(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 553 152**

(21) Número de solicitud: 201430854

(51) Int. CI.:

G06F 11/08 (2006.01)

(12) SOLICITUD DE PATENTE A1

(22) Fecha de presentación:

03.06.2014

(43) Fecha de publicación de la solicitud:

04.12.2015

(71) Solicitantes:

UNIVERSIDAD CARLOS III DE MADRID (100.0%)
Av. Gregorio Peces Barba, 1
28919 Leganés (Madrid) ES

(72) Inventor/es:

PLEITE GUERRA, Jorge y
JIMÉNEZ OLAZÁBAL, Andrés

(74) Agente/Representante:

GONZÁLEZ AHIJADO, Ángel

(54) Título: **Método para la detección y corrección de errores en memorias volátiles**

(57) Resumen:

La invención describe un método para la detección y corrección de múltiples errores en los datos almacenados en la memoria volátil de un sistema electrónico, como los causados por radiación cósmica en sistemas aeroespaciales, basado en una codificación/decodificación piramidal a través de los llamados bits semilla HSB que son soldados o fijados físicamente en la placa del circuito impreso.

La fase de codificación agrupa en bloques los bits de la memoria seleccionados de forma equiespaciada, calcula las palabras de chequeo de detección y corrección para cada bloque, y calcula y fija físicamente los bits semilla HSB.

La fase de decodificación calcula periódicamente en dos funciones duplicadas las palabras de chequeo de detección y corrección de cada bloque, detecta y corrige errores en estas palabras de chequeo a partir de los bits semilla HSB fijados en codificación, y detecta y corrige errores en los datos almacenados en la memoria.

ES 2 553 152 A1

DESCRIPCIÓN

Método para la detección y corrección de errores en memorias volátiles

OBJETO DE LA INVENCIÓN

El objeto de la invención se engloba dentro de las técnicas para la detección y corrección de múltiples errores que puedan producirse en los datos de configuración a proteger almacenados en la memoria volátil de un sistema electrónico, y más específicamente de aquellos errores causados por el impacto de la radiación cósmica basada en iones pesados, neutrones y protones, en sistemas electrónicos aeroespaciales.

ANTECEDENTES DE LA INVENCIÓN

Históricamente, la industria aeroespacial continúa la mejora de su competitividad basada en dos puntos básicos, la reducción de costes y la introducción de tecnologías seguras y fiables. El rápido crecimiento en el uso de sistemas electrónicos complejos para realizar actividades o funcionalidades para la mayoría de aplicaciones críticas de vuelo ha creado nuevas necesidades para la certificación aeronáutica de estos nuevos sistemas críticos embarcables, mediante las regulaciones de EASA (del inglés "European Aviation Safety Agency") y FAA (del inglés "Federal Aviation Administration of United States"), como RTCA-DO254 / EUROCAE-ED80 para hardware complejo como FPGAs (del inglés "Field Programmable Gate Array") y RTCA-DO178 / EUROCAE-ED12 para sistemas basados en módulos Software embebidos en microprocesadores.

Actualmente diferentes técnicas son usadas para mitigar errores del tipo modificación por fenómeno único o SEU en memorias (del inglés "Single Event Upset"). Estas técnicas están basadas en procesos específicos de la tecnología de memorias, células de memoria protegidas contra radiación, códigos de detección y corrección de errores, redundancias Software, redundancia modular triple (TMR) o incluso soluciones a nivel de sistema.

Las siguientes referencias como estado del arte son aquí citadas como de interés en el campo al que pertenece la invención:

- Carl Carmichael en Xilinx, Inc según "SEU Mitigation Techniques for Virtex FPGAs in Space Applications" donde se presentan diferentes soluciones basadas en el particionado de la lógica implementada en FPGAs y la redundancia de estos particionados para la mitigación de los fallos producidos por radiación. Aunque proporciona una gran capacidad de detección y corrección de errores, la mayor desventaja de este sistema de replicación de datos es el tamaño de la memoria necesaria ya que se duplica o triplica el tamaño requerido.

- Sandi Habinc en Gaisler Research según "Functional Triple Modular Redundancy (FTMR)" donde se presentan soluciones como la técnica de Triple Redundancia Modular (TMR) que implica tener tres circuitos lógicos funcionales para determinar el fallo de uno de ellos mediante una comparación de los resultados de los tres circuitos. La mayor desventaja de estos sistemas es que sólo protege la parte lógica programada por el usuario de una FPGA pero no la parte interna de enrutado de los circuitos de la FPGA.

- Ghazanfar-Hossein Asadi and Mehdi Baradaran Tahoori en Computer Engineering Northeastern University según "Soft Error Mitigation for SRAM-based FPGAs" donde se presenta una técnica redundante donde la corrección de integridad del estado de la memoria se realiza mediante una FPGA auxiliar para el chequeo continuo de la palabra de corrección de redundancia cíclica CRC o "checkword" de la FPGA principal.

- Michael Nicolaidis de IROC Technologies según "Radiation Induced Single-word Multiple-bit Upsets Correction in SRAM" donde se presenta la técnica basada en la detección de errores mediante la monitorización del consumo de corriente de cada una de los bloques o "frames" de la memoria. Una vez se detecta el fallo, se utiliza la codificación Hamming para la corrección de errores.

- Fernanda de Lima en Universidade Federal do Rio Grande do Sul según "A Multiple Bit Upset Tolerant SRAM memory" donde se presenta la técnica de corrección y detección de errores en memorias basadas en codificaciones Reed-Solomon y Hamming. Estos sistemas implican introducir una latencia alta en la lectura y escritura de los datos de la memoria, ya que es necesaria la elección de un sistema Reed-Solomon basado en la corrección de hasta 7 errores consecutivos.

Dado el estado del arte descrito, se plantean unos objetivos necesarios para evaluar las mejoras del sistema con respecto a sistemas ya existentes, como son:

1. Capacidad de corrección de múltiples errores simultáneos o MBU (del inglés "Multi-Bit Upset") como mejora al sistema TMR.

2. Ocupación del sistema o Latencia en el proceso como mejora a los sistemas actuales de detección y corrección de errores, debido a la necesidad de codificar y decodificar en cada proceso.

3. Eficiencia en cuanto a la necesidad de memoria extra requerida, como mejora de los sistemas de partición y replicación de datos.

DESCRIPCIÓN DE LA INVENCIÓN

Muchos estudios y campañas de ensayos se han desarrollado para caracterizar el tipo de fallos SEU y MBU en las memorias volátiles sometidas a radiación solar basada en iones pesados, neutrones y protones. Además, las nuevas técnicas de fabricación de los transistores utilizados en los bits de las memorias han sido capaces de reducir su tamaño incrementando su densidad. Este incremento de la densidad conlleva que la capacidad o carga crítica de los bits de memoria sea inferior, haciendo que los bits de memoria sean más susceptibles a las pequeñas descargas de corriente producidas por iones pesados, neutrones y protones cuando pasan a través del substrato de silicio.

Estos estudios han desvelado el incremento de la capacidad de estas partículas para producir MBU dentro de una memoria. Ensayos se han realizado tanto para incidencia normal a la superficie como para incidencia con 60º donde se aprecia aún más el fallo por MBU, hasta un máximo de 6 o 7 bits afectados por partícula (figura 1). La figura 2 muestra la arquitectura utilizada basada en la disposición característica geográfica de errores debido a MBU.

La innovación de la solución propuesta es un método basado en una codificación/decodificación piramidal multi-nivel gracias a los llamados bits semilla que, soldados o fijados físicamente en la placa del circuito impreso o HSB (del inglés "Hardwired Seed bits"), son calculados en la fase de codificación y contienen suficiente

información como para restaurar una porción del código de programación original en la fase de decodificación en cada ciclo de refresco de las memorias volátiles.

Acorde a las figuras 3 y 4, existen dos procesos definidos en diferentes fases:

- Fase de codificación (figura 3) para la obtención de los bits semilla HSB, realizada durante el diseño del equipo electrónico una vez que el código de configuración a proteger está definido, constituida por:

    - Capa inferior de la codificación, basada en la agrupación en bloques de los bits de la memoria escogidos por ejemplo de manera equidistante geográfica (figura 3). Esta separación geográfica entre los bits pertenecientes a cada bloque es adecuada para minimizar el efecto de los errores esperados según la caracterización de los patrones de errores múltiples en ráfaga (figura 1), que provocan tanto SEU como MBU sobre varios bits contiguos almacenados originalmente en la memoria volátil del receptor. Se crea la unidad mínima analizada llamada bloque o "frame" donde no aparecerán errores múltiples. Esto es debido a que la separación geográfica descrita está caracterizada por tener una distancia mínima superior a la máxima longitud registrada y esperada de bits afectados por ráfaga en MBU. (figura 1), por lo que los bits afectados son distribuidos en bloques o "frames" diferentes de tal forma que no más de un bit erróneo por bloque o "frame" es esperado

    - De cada bloque o "frame" se obtienen dos palabras de chequeo CRC o "checkwords" con los siguientes objetivos:

        - Detección de error o cambio de bit en cada bloque o "frame" mediante una palabra de detección de errores o chequeo de tipo bit de paridad que es un proceso con baja latencia de procesado. Se determina de forma que el número total de bits en cada bloque o “frame” sea par. Esta detección es posible gracias a la disposición equidistante de los bits de los bloques o "frames" adecuada según la caracterización de la fuente de fallos debidos a SEU o MBU, y que permite que siempre exista un solo fallo máximo por bloque o "frame" incluso en el peor caso esperado de fallos debidos a MBU.

- Corrección de un solo bit erróneo en los "frames" detectados anteriormente como irregulares mediante el CRC de paridad descrito en el apartado anterior. La codificación utilizada para la corrección del bit afectado en cada bloque o "frame" irregular se basa en una codificación de los datos para obtener una palabra de chequeo CRC de corrección, por ejemplo basada en la suma en complemento a dos basada en funciones XOR de todos los bits del bloque o "frame" (figura 2).

o Todos los bits del CRC de paridad junto con los bits de CRC de corrección de todos los bloques o "frames" se codifican de nuevo para obtener los bits semilla HSB mediante una codificación que permita detectar posibles errores, por ejemplo mediante la suma en complemento a dos basada en funciones XOR, que se fijan físicamente a una referencia de alimentación en la placa del circuito impreso del sistema mediante resistencias pull-up, resistencias pull-down, puentes tipo "jumper", directamente soldados, o cualquier otro método equivalente que garantice su disponibilidad sin errores.

• Fase de decodificación (figura 4) durante la operación normal de funcionamiento del equipo electrónico:

o La función de decodificación se realiza periódicamente en busca de errores en cada uno de los bloques o "frames".

o La funcionalidad lógica desarrollada para el proceso de decodificación es duplicada e implementada físicamente en lugares distintos del receptor (figura 3). Esta lógica duplicada recibe el nombre de “Func1” (15) para la función primaria y “Func2” (16) para la función redundada. Además es imprescindible que cada una de las áreas de implementación de estas dos funciones estén separadas del área donde se implementen los bits de los CRCs de paridad y corrección.

o Lectura mediante puertos dedicados de los HSB externos fijados físicamente (22).

o Con los HSB se comprueba el posible error en los CRC (13) de paridad y de corrección guardados en memoria mediante las funciones “Func1” (15) y “Func2” (16). Ambas funciones contienen la misma funcionalidad lógica y

están implementadas en lugares segregados del receptor. En ambas funciones se utiliza de nuevo la codificación empleada para el proceso de detección/corrección. De esta manera se hace la comparación bit a bit con los HSB leídos (proceso inverso al anteriormente descrito más la comparación de resultados).

- Si el resultado de ambas funciones “Func1” y “Func2” es igual y sin error una vez comprobadas con respecto a la referencia HSB, esto significa que no hay error en los CRC de paridad y corrección. De la misma manera, se procede a comprobar secuencialmente los siguientes bloques o "frames".

- Si el resultado de ambas funciones “Func1” y “Func2” es igual con el mismo valor una vez comprobadas respecto a la referencia HSB, esto significa que hay error en los CRC de paridad y corrección indicando el resultado de ambas funciones el bit erróneo a corregir.

  - Con los CRC (13) de paridad y de corrección ya verificados, corregidos y sin error mediante el proceso anteriormente descrito, se comprueba ya el estado de los datos de cada uno de los bloques o "frames" que componen la memoria. Primero se comprueba la paridad de los datos de cada bloque o "frame" y se compara con el CRC de paridad.

    - Si no existe error entre el cálculo de paridad del bloque o "frame" y el CRC de paridad correspondiente, se procede al cálculo de la paridad del siguiente bloque o "frame".

    - Si se detecta error entre el cálculo de la paridad del bloque o "frame" y el CRC de paridad correspondiente, se utiliza de nuevo la codificación empleada en el proceso de detección y corrección del bloque o "frame" para hacer la comparación bit a bit con el CRC de corrección. De esta manera se obtiene el bit afectado modificado y se corrige a su estado anterior.

- Si el resultado de ambas funciones “Func1” y “Func2” es distinto, se interpreta que se ha producido un error en la lógica de una de las funciones y no en los datos de los CRC de paridad y corrección. Al ser un proceso periódico, no se contemplan los errores acumulados, por lo que no pueden existir errores

simultáneos en la lógica de algunas de las funciones “Func1” y “Func2” y en los CRC de paridad y corrección. Esto es debido a que tanto las funciones como los CRC de paridad y corrección están separados físicamente en áreas distintas del receptor y por tanto un MBU no puede llegar a afectar simultáneamente. Al existir el fallo en una de las funciones, el resultado de la comprobación de la otra función será que no existe error entre la comparación de los CRC de paridad y corrección con respecto a los HSB. De esta manera se puede concluir que la función cuyo resultado es un error indeterminado es la función que ha fallado, pudiéndose discriminar qué función “Func1” o “Func2” es la que está realmente afectada por un error, sin necesidad de triplicar la lógica de decodificación e implementación de un votador (sistema comparador de las tres salidas de los sistemas triplicados capaz de detectar cuál de ellas es diferente a las otras dos). Se utiliza la función sin fallo para la corrección del error existente en la lógica de la función que falla comprobando que el error se encuentra en el área destinada a la lógica de la función que falla.

- Con ambas funciones “Func1” y “Func2” ya corregidas y con los CRC de paridad y de corrección ya verificados, se comprueba ya el estado de los datos de cada uno de los bloques o "frames" que componen la memoria. Primero se comprueba la paridad de los datos de cada bloque o "frame" calculando su CRC de paridad y se compara con el CRC de paridad almacenado ya verificado y corregido.

    - Si no existe error entre el cálculo de paridad del bloque o "frame" y el CRC de paridad almacenado correspondiente, se procede al cálculo de la paridad del siguiente bloque o "frame".

    - Si se detecta error entre el cálculo de la paridad del bloque o "frame" y el CRC de paridad almacenado correspondiente, se utiliza de nuevo la codificación empleada en el proceso de detección y corrección del bloque o "frame" para hacer la comparación bit a bit con el CRC de corrección. De esta manera se obtiene el bit afectado modificado y se corrige a su estado anterior.

Las ventajas de este sistema son la reducción del tamaño de memoria necesaria

tanto volátil como no volátil del sistema redundante, ya que en el sistema habitualmente utilizado TMR (Redundancia Modular Triple), la memoria debe triplicarse, siendo necesaria además la implementación de un sistema votador para discriminar qué sistema ha fallado para aislar y/o corregir el fallo. Gracias a la reducción del sistema de la arquitectura piramidal, no es necesario triplicar las funcionalidades críticas por lo que la energía consumida y disipada es también menor.

Comparado con otros sistemas redundantes basados en sistemas de detección/corrección de errores, las características de este sistema son:

- Capacidad de corrección de múltiples errores producidos simultáneamente gracias a la disposición geográfica de los bits que impide el fallo múltiple en cada bloque o "frame"(figura 1).

- Baja latencia de procesado gracias a la separación de detección y corrección. Se utiliza el sistema de paridad para comprobar la existencia de errores, y el uso de la función de detección/corrección de errores sólo es aplicado una vez que se ha detectado un error. Además, la disposición geográfica permite el uso de los sistemas más sencillos y rápidos, ya que sólo es necesaria la corrección de un solo error por bloque o "frame".

- Eficiencia en el porcentaje entre el número de bits extra necesarios (HSB y CRCs por utilizar sistemas de corrección de un solo bit) y los bits de datos totales almacenados.

- La referencia externa fija (HSB) es reducida y no afectada por radiación, ya que está basada en la implementación física en la placa del circuito impreso.

BREVE DESCRIPCIÓN DE LAS FIGURAS

Para complementar la descripción de esta invención, se acompaña como parte integrante de la misma un juego de esquemas con carácter ilustrativo y no limitativo, que se resume a continuación.

La figura 1 muestra diferentes ejemplos de errores producidos por los efectos de la radiación que causan el cambio de estado en varios bits de la memoria volátil: error simple (1), errores múltiples (2 a 11). La representación de los errores son marcados en

negro.

La figura 2 muestra un ejemplo de la matriz utilizada en la estructura basada en la disposición característica geográfica de cada uno de los 64 bloques o "frames" (para una capacidad de corrección de hasta 8x8 bits) de bits cuando se produce un fallo debido a SEU o MBU. De esta manera se consigue que sólo exista como máximo un fallo en un bit en todo el bloque o "frame" ya que la distancia mínima entre los bits que componen un bloque o "frame" es mayor que la distancia detectada en el peor caso esperado de errores debidos a MBU. Se define cada una de las celdas o bit de la matriz según el número de bloque o "frame" al que pertenecen y el orden del bit dentro del bloque o "frame". Por ejemplo, f45.8 corresponde al bit 8 del bloque o "frame" 45.

Los CRC de paridad y corrección son codificados secuencialmente basados en máscaras espaciales para obtener los bits implicados en cada bloque o "frame", similares a los bits marcados en la figura 2, equidistantes una cantidad determinada de bits (en el ejemplo: 8x8). La primera detección de paridad de baja latencia de procesado de datos desvela posibles bloque o "frames" con algún bit alterado, para posteriormente sólo corregir secuencialmente los bloques o "frames" erróneos.

La figura 3 muestra el resultado del proceso de codificación en los tres niveles de la arquitectura de bits, mapa de bits a proteger (12), palabras de chequeo CRC de paridad y corrección (13), HSB (14), lógica implementada de la “Func1” (15) y lógica implementada de la “Func2” redundada (16), de la arquitectura piramidal obtenidos secuencialmente durante la fase de codificación.

En la decodificación, mostrada la figura 4, la referencia del proceso de protección son los bits semilla HSB (19), de la arquitectura. Se utilizan las palabras de chequeo CRC (18) ya calculadas y corregidas, para la detección de errores con baja latencia de procesado y corrección de bits sólo en bloques o "frames" irregulares (17) del mapa de bits. En esta arquitectura también se encuentran separados espacialmente las funciones lógicas para la decodificación (15) y (16).

La figura 5 muestra un ejemplo de implementación que utiliza circuitos tipo FPGA (23). El mapa de bits de datos (20) y los CRC de paridad y corrección (21) son almacenados en la memoria volátil del sistema, mientras que los bits semilla HSB (22), son fijados físicamente y externamente a la FPGA. La lógica desarrollada para el proceso

de decodificación es duplicada e implementada físicamente en lugares distintos del receptor. Esta lógica duplicada recibe el nombre de “Func1” (24) para la función primaria y “Func2” (25) para la función redundada

REALIZACION PREFERENTE DE LA INVENCION

A continuación se expone un caso de uso de esta invención. La técnica propuesta se centra en equipos electrónicos que utilizan FPGAs con tecnología interna basada en memoria volátil tipo RAM, donde la secuencia de bits de configuración es la encargada de definir la funcionalidad del sistema, según el contenido interno de las tablas de consulta o LUTs (del inglés "lookup tables"), matrices de conexión interna, entradas, salidas, etc.

Más concretamente, se trata de un sistema aeroespacial basado en una FPGA de tecnología RAM donde se ha programado una serie de funcionalidades críticas de vuelo. Para ello, después de la fase de diseño del equipo basado en el lenguaje VHDL, éste se compila para ser descargado en la memoria ROM del sistema.

La codificación se realiza durante la fase de diseño del equipo, una vez que todo el código de configuración de la FPGA está definido. Mediante el estudio de los patrones de errores creados por el SEU cuando se manifiesta afectando a varios bits simultáneamente, MBU, se define entonces la unidad mínima de análisis, llamada bloque o "frame". Esto es, la distancia mínima de separación necesaria entre los bits que componen cada uno de los bloques o "frames" debe ser superior a la distancia detectada en el peor caso de errores propagados que afectan a varios bits simultáneamente. Estos bloques o "frames" se codifican mediante el diseño de la máscara para conseguir una capacidad de corrección de múltiples errores provocados por MBU.

Utilizando el sistema de codificación, para una FPGA que utiliza una memoria de 64 Kbits para definir sus funcionalidades, este código de 64 Kbits se define la máscara del bloque o "frame" para la detección de baja latencia de procesado (64 bits totales, uno por cada bloque o "frame") y corrección de errores (640 bits totales, 10 bits por bloque o "frame"). En la figura 2, se muestra cómo para un mismo código y gracias a la disposición geográfica de los bits de los 64 bloques o "frames", todos los errores esperados causados por MBU pueden ser detectados/corregidos en un pocos ciclos de refresco (primeras simulaciones estiman un tiempo de respuesta de análisis de toda la memoria inferior a décimas de milisegundos).

El diseño de este sistema está también basado en un chequeo de paridad para la detección de aquellos bloques o "frames" que estén afectados por bits erróneos debido a la propagación múltiple del SEU. Este proceso de detección se ha escogido por la baja latencia de procesado utilizada en este sistema para cada bloque o "frame" en caso de múltiples errores. De esta manera sólo se utiliza el CRC de corrección en aquellos bloques o "frames" afectados por errores.

En el caso práctico expuesto para una memoria de 64 Kbits, se utilizan 64 bloques o "frames" de 1 Kbit, equiespaciados geográficamente en la memoria cada 8 bits en las dos dimensiones. La primera codificación (13) contiene 1 bit para el CRC de paridad y 10 bits para el CRC de corrección para protección de cada bloque o "frame" mediante la codificación para detección/protección, por ejemplo mediante la suma en complemento a dos basada en funciones XOR. Para proteger la memoria total (12) son necesarios 704 bits (11 bits en cada uno de los 64 "frames") que corresponden sólo con el 1,08% de la memoria total.

El CRC de corrección se basa en la suma complemento a dos de los bits de datos. Por ejemplo, el cálculo del CRC de corrección de la palabra de datos "1010011" es "110". La palabra de suma de control tiene la propiedad de que los bits de datos de información originales también aparecen en la palabra de datos de corrección y sin cambios junto con los bits de CRC de corrección añadidos. Este procedimiento se utiliza porque los bits de datos de información deben mantenerse en su forma original debido a que es el código desarrollado que define la funcionalidad del sistema. Por lo tanto, se dice que el código es sistemático.

Para un conjunto de datos de 7 bits, los 3 bits del CRC de corrección se obtienen de la siguiente manera:

C1 = D1 xor D3 xor D5 xor D7,

C2 = D2 xor D3 xor D6 xor D7 y

C3 = D4 xor D5 xor D6 xor D7

En el ejemplo de datos "1010011", la palabra completa con los bits del CRC de corrección es "1010011110". Los bits del CRC de corrección C1 a C3, se obtienen mediante la suma complemento a dos de los bits de datos, D1 a D7, incluidos en un conjunto único y determinadas por el formato binario de su posición de bit. Todas los

CRC de corrección se obtienen con los bits de datos. La matriz generadora requerida para esta codificación es G = (I_K | A). Si existen k bits de datos, la matriz debe contener al menos k palabras de código linealmente independientes para producir combinaciones lineales de dos o más códigos en el conjunto. La manera más fácil de obtener k códigos linealmente independientes es elegir la matriz identidad como I_K en la matriz generadora, mientras que A es la representación del módulo lineal en complemento a 2 de los bits de datos. El proceso de codificación se representa en forma de matriz como v = u G, donde u es el bloque de bits de datos de información, v la palabra de suma de corrección completa y G de la matriz generadora.

Es importante destacar que la matriz generadora es de dimensión kxn donde k es la dimensión de los bits de datos y n es la longitud total de la palabra suma que incluye los bits de datos y el CRC de corrección. Esta forma especial corresponde a un código sistemático que contiene la matriz de identidad kxk y kx(n-k) de matriz de bits de suma de corrección. Por lo tanto, el mismo ejemplo se obtiene mediante V = [1 0 1 0 0 1 1] G = [1 0 1 0 0 1 1 1 1 0].

La detección de errores consiste en la decisión de qué bit está en fallo mediante el análisis de la información del síndrome. El síndrome se obtiene de la matriz de corrección compuesta por el (n-k)x n de la matriz generadora, G:

$H = (A^T \mid I_{N-K})$

Síndrome: s = v H

Los bits del CRC de corrección general se cubren la siguiente manera:

- C1 cubre todas las posiciones de bits que tienen el bit menos significativo establecido en 1: bit 1, 3, 5, 7, etc. siendo siempre el bit menos significativo para el cálculo del síndrome se valora como 2^0, al detectar un error.
- C2 cubre todas las posiciones de bit que tienen el segundo bit menos significativo puesto a 1: bit 2, 3, 6, 7, 10, 11, etc. se valora como 2^1 para el cálculo del síndrome.
- C3 cubre todas las posiciones de bits que tienen el tercer bit menos significativo establecido en 1: Bits 4-7, 12-15, 20-23, etc., en este ejemplo es el bit más significativo y para el cálculo del síndrome se valora como 2^2, cuando se detecta un error.

- C4 cubre todas las posiciones de bits que tienen el cuarto bit menos significativo establecido en 1: Bits 8-15, 24-31, 40-47, etc., se valora como 2^3 para el cálculo del síndrome.

Al tener m bits en el CRC de corrección, este cálculo puede cubrir un numero de bits de datos hasta 2^m - 1. La longitud total (suma de CRC de corrección más los bits de datos) se obtiene mediante la suma de los bits de datos, además del CRC de corrección, 2^m + m - 1. Como m varía, todos los posibles CRC de corrección con la máxima capacidad se pueden conseguir. En los ejemplos anteriores, 3 bits de CRC de corrección cubren hasta 7 bits de datos, mientras que 5 bits de CRC cubrirían hasta 31 bits de datos. Cualquier combinación no optimizada intermedia, podría ser también construida. Una trama de 22 bits de datos se podría construir aunque no optimizada y también ser cubiertos por 5 bits CRC de corrección.

Sólo si se detecta un error en un bloque o "frame" mediante el CRC de paridad, se utilizarán los bits redundantes del CRC de corrección (13) para detectar la posición del fallo y corregir el bit erróneo de ese bloque o "frame". La totalidad de la memoria es chequeada secuencialmente mediante los 64 bloques o "frames" creados en esta aplicación. Si se encuentra un bloque o "frame" con fallo en el CRC de paridad, se procede al proceso de corrección del bit afectado mediante el CRC de corrección.

Una vez el bit erróneo es detectado y corregido, se procede al análisis del CRC de paridad del siguiente bloque o "frame". En caso de que no se encuentre un error en el bloque o "frame", secuencialmente y manteniendo la disposición geográfica de la máscara que define el bloque o "frame", se procederá al chequeo de cada uno de los bloques o "frames" de la memoria. En el caso mostrado en la figura 2, implicaría 64 pasos (uno por bloque o "frame") para completar el estudio de fallos en toda la memoria.

Una vez obtenida la secuencia de bits (13) de la capa inferior (704 bits) de CRC de paridad y corrección, ésta se utiliza como datos de entrada a proteger para la capa superior de la codificación (14). Se ha utilizado la misma codificación de detección/protección de la etapa anterior basada en la suma en complemento a dos que permite detectar/corregir un error en cada ciclo de operación para obtener la semilla HSB (14).

Una vez que ambas capas se han codificado (13 y 14), los bits semilla HSB se fijan

físicamente a una referencia de alimentación (22) según la figura 5 en la placa electrónica del equipo electrónico mediante 10 resistencias pull-up, resistencias pull-down, puentes tipo "jumpers", directamente soldados, o cualquier otro método equivalente que garantice su disponibilidad sin errores. De esta manera, se asegura que no existe la posibilidad de que estos datos semilla sean alterados por la radiación existente o cualquier otro agente externo.

La lógica desarrollada para el proceso de decodificación es duplicada e implementada físicamente en lugares distintos del receptor (figura 3). Esta lógica duplicada recibe el nombre de “Func1” (15 y 24) para la función primaria y “Func2” (16 y 25) para la función redundada. Además es imprescindible que cada una de las áreas de implementación de estas dos funciones esté separada del área donde se implementen los bits de los CRCs (18 y 21) de paridad y corrección. Si existe fallo en una de las funciones, el resultado de la comprobación de la otra función será que no existe error entre la comparación de los CRC (18) de paridad y corrección con respecto a los HSB (19). De esta manera se puede concluir que la función cuyo resultado es un error indeterminado es la función que ha fallado, pudiéndose discriminar que función “Func1” (24) o “Func2” (25) es la que está realmente afectada por un error, sin necesidad de triplicar la lógica de decodificación e implementación de un votador. Se utiliza la función sin fallo para la corrección del error existente en la lógica de la función fallada comprobando que el error se encuentra en el área destinada a la lógica de la función que falla.

REIVINDICACIONES

1. Método para la detección y corrección de errores en memorias volátiles caracterizado por contener al menos:

   -Fase de codificación (fig.3) realizada durante el diseño del equipo electrónico para la obtención de los bits semilla HSB.

   -Fase de decodificación (fig.4) realizada durante la operación normal de funcionamiento del equipo electrónico utilizando los bits semilla HSB calculados en la fase de codificación para la detección/corrección de errores.

2. Método para la detección y corrección de errores según la reivindicación 1 caracterizado porque la fase de codificación incluye al menos las etapas de:
   - Agrupación en bloques de los bits de la memoria seleccionados de forma equiespaciada.
   - Obtención de una palabra de chequeo para detección de errores para cada bloque o CRC de paridad.
   - Obtención de una palabra de chequeo para corrección de errores para cada bloque o CRC de corrección.
   - Obtención de los bits semilla HSB a partir de todas las palabras calculadas de chequeo para detección errores o CRCs de paridad y de chequeo para corrección de errores o CRCs de corrección.
   - Fijación física de los bits semilla HSB calculados en el circuito impreso mediante resistencias pull-up, resistencias pull-down, puentes tipo "jumper", directamente soldados, o cualquier otro método equivalente que garantice su disponibilidad sin errores.

3. Método para la detección y corrección de errores según las reivindicaciones anteriores caracterizado porque la fase de decodificación incluye al menos las etapas de:
   - Cálculo periódico mediante dos funciones duplicadas Func1 (15) y Func2 (16) de las palabras de chequeo para detección o CRCs de paridad y de chequeo para corrección o CRCs de corrección, a partir de los bits de datos almacenados en memoria mediante el mismo procedimiento de agrupación de bits y algoritmos de detección y corrección de errores utilizados en la fase de codificación.
   - Lectura de los bits semilla HSB fijados físicamente en el circuito impreso en la fase de codificación.

- Comprobación de errores en las palabras de chequeo de detección y chequeo de corrección mediante la comprobación con los bits semilla HSB leídos mediante las funciones duplicadas Func1 (15) y Func2 (16).
- Cuando el resultado de las funciones duplicadas Func1 y Func2 es idéntico y la comprobación de errores no indica errores para un bloque determinado, se concluye que ese bloque está libre de errores y ambas funciones Func1 (15) y Func2 (16) son correctas.
- Cuando el resultado de las funciones duplicadas Func1 y Func2 es idéntico y la comprobación de errores en las palabras de chequeo de detección y chequeo de corrección indica error, ambas funciones Func1 (15) y Func2 (16) son correctas y el resultado indica dónde está el bit erróneo a corregir en las palabras de chequeo de detección y chequeo de corrección.
- Cuando el resultado de las funciones duplicadas es distinto, se interpreta que se ha producido un error en la lógica de una de las funciones y no en los datos de las palabras de chequeo de detección y chequeo de corrección. No se toma ninguna acción y se continúa el proceso periódico a la espera del resultado del siguiente bloque o "frame" para confirmar el error en una de las lógicas de las funciones “Func1” o “Func2”. En este caso, se comprueba qué función está en fallo.
- Con ambas funciones “Func1” y “Func2” ya corregidas y con los CRC de paridad y de corrección ya verificados, se comprueba entonces el estado de los datos de cada uno de los bloques o "frames" que componen la memoria.

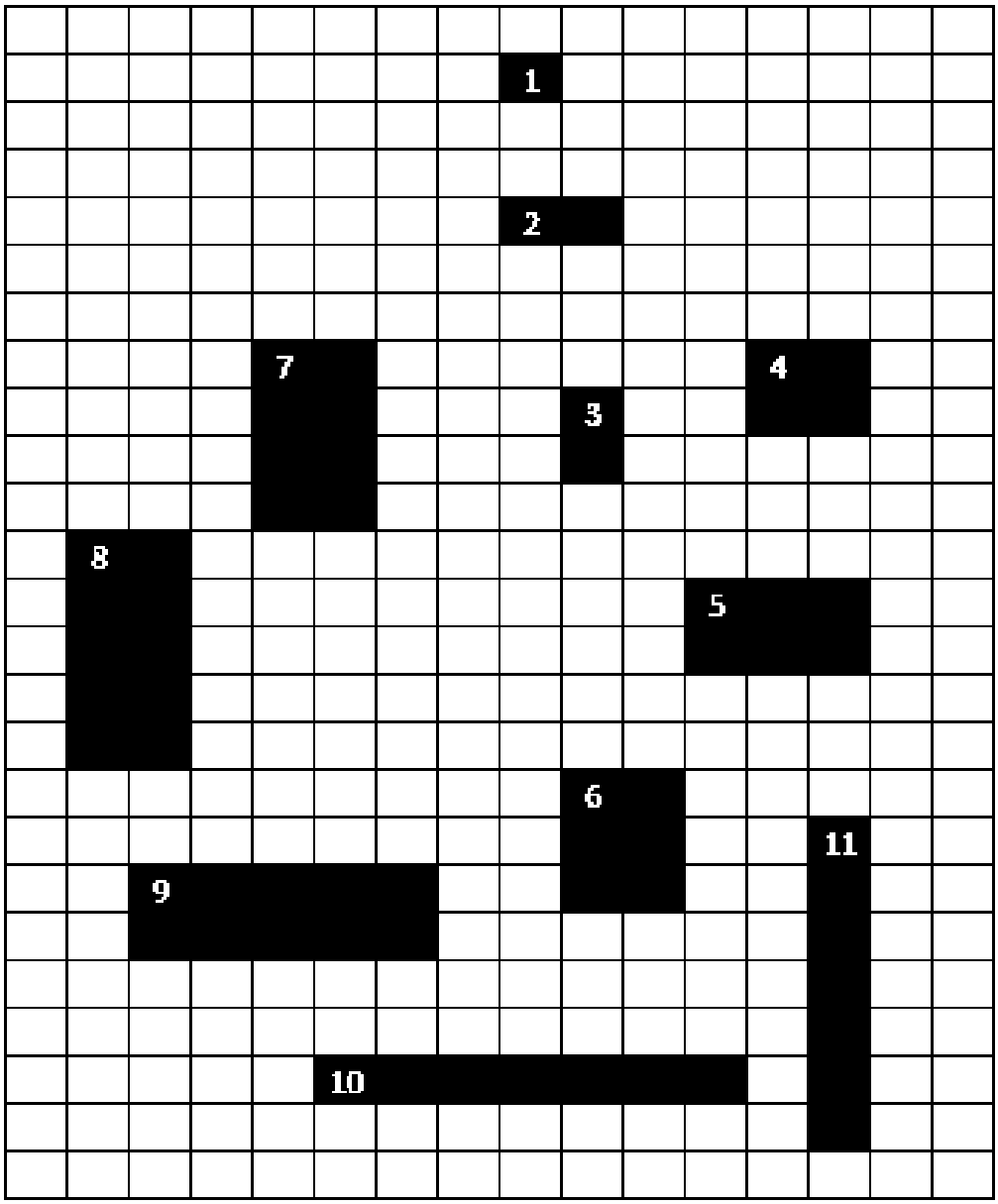

FIG. 1

f1.1	f2.1	f3.1	f4.1	f5.1	f6.1	f7.1	f8.1	f1.2	f2.2	f3.2	f4.2	f5.2	f6.2	f7.2	f8.2
f9.1	f10.1	f11.1	f12.1	f13.1	f14.1	15.1	f16.1	f9.2	f10.2	f11.2	f12.2	f13.2	f14.2	15.2	f16.2
f17.1	f18.1	f19.1	f20.1	f21.1	f22.1	f23.1	f24.1	f17.2	f18.2	f19.2	f20.2	f21.2	f22.2	f23.2	f24.2
f25.1	f26.1	f27.1	f28.1	f29.1	f30.1	f31.1	f32.1	f25.2	f26.2	f27.2	f28.2	f29.2	f30.2	f31.2	f32.2
f33.1	f34.1	f35.1	f36.1	f37.1	f38.1	f39.1	f40.1	f33.2	f34.2	f35.2	f36.2	f37.2	f38.2	f39.2	f40.2
f41.1	f42.1	f43.1	f44.1	f45.1	f46.1	f47.1	f48.1	f41.2	f42.2	f43.2	f44.2	f45.2	f46.2	f47.2	f48.2
f49.1	f50.1	f51.1	f52.1	f53.1	f54.1	f55.1	f56.1	f49.2	f50.2	f51.2	f52.2	f53.2	f54.2	f55.2	f56.2
f57.1	f58.1	f59.1	f60.1	f61.1	f62.1	f63.1	f64.1	f57.2	f58.2	f59.2	f60.2	f61.2	f62.2	f63.2	f64.2
f1.3	f2.3	f3.3	f4.3	f5.3	f6.3	f7.3	f8.3	f1.4	f2.4	f3.4	f4.4	f5.4	f6.4	f7.4	f8.4
f9.3	f10.3	f11.3	f12.3	f13.3	f14.3	15.3	f16.3	f9.4	f10.4	f11.4	f12.4	f13.4	f14.4	15.4	f16.4
f17.3	f18.3	f19.3	f20.3	f21.3	f22.3	f23.3	f24.3	f17.4	f18.4	f19.4	f20.4	f21.4	f22.4	f23.4	f24.4
f25.3	f26.3	f27.3	f28.3	f29.3	f30.3	f31.3	f32.3	f25.4	f26.4	f27.4	f28.4	f29.4	f30.4	f31.4	f32.4
f33.3	f34.3	f35.3	f36.3	f37.3	f38.3	f39.3	f40.3	f33.4	f34.4	f35.4	f36.4	f37.4	f38.4	f39.4	f40.4
f41.3	f42.3	f43.3	f44.3	f45.3	f46.3	f47.3	f48.3	f41.4	f42.4	f43.4	f44.4	f45.4	f46.4	f47.4	f48.4
f49.3	f50.3	f51.3	f52.3	f53.3	f54.3	f55.3	f56.3	f49.4	f50.4	f51.4	f52.4	f53.4	f54.4	f55.4	f56.4
f57.3	f58.3	f59.3	f60.3	f61.3	f62.3	f63.3	f64.3	f57.4	f58.4	f59.4	f60.4	f61.4	f62.4	f63.4	f64.4
f1.5	f2.5	f3.5	f4.5	f5.5	f6.5	f7.5	f8.5	f1.6	f2.6	f3.6	f4.6	f5.6	f6.6	f7.6	f8.6
f9.5	f10.5	f11.5	f12.5	f13.5	f14.5	15.5	f16.5	f9.6	f10.6	f11.6	f12.6	f13.6	f14.6	15.6	f16.6
f17.5	f18.5	f19.5	f20.5	f21.5	f22.5	f23.5	f24.5	f17.6	f18.6	f19.6	f20.6	f21.6	f22.6	f23.6	f24.6
f25.5	f26.5	f27.5	f28.5	f29.5	f30.5	f31.5	f32.5	f25.6	f26.6	f27.6	f28.6	f29.6	f30.6	f31.6	f32.6
f33.5	f34.5	f35.5	f36.5	f37.5	f38.5	f39.5	f40.5	f33.6	f34.6	f35.6	f36.6	f37.6	f38.6	f39.6	f40.6
f41.5	f42.5	f43.5	f44.5	f45.5	f46.5	f47.5	f48.5	f41.6	f42.6	f43.6	f44.6	f45.6	f46.6	f47.6	f48.6
f49.5	f50.5	f51.5	f52.5	f53.5	f54.5	f55.5	f56.5	f49.6	f50.6	f51.6	f52.6	f53.6	f54.6	f55.6	f56.6
f57.5	f58.5	f59.5	f60.5	f61.5	f62.5	f63.5	f64.5	f57.6	f58.6	f59.6	f60.6	f61.6	f62.6	f63.6	f64.6

FIG. 2

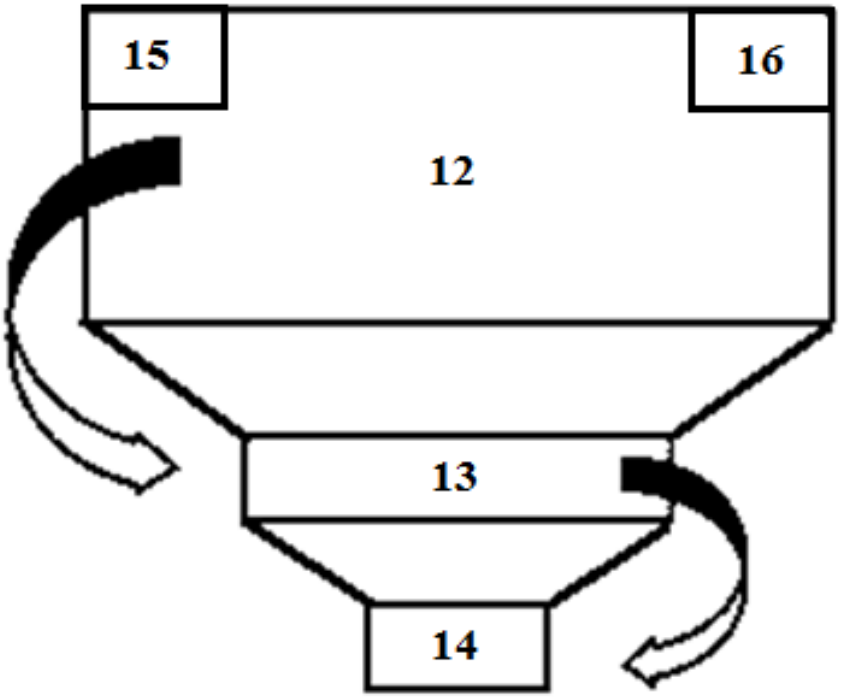

FIG. 3

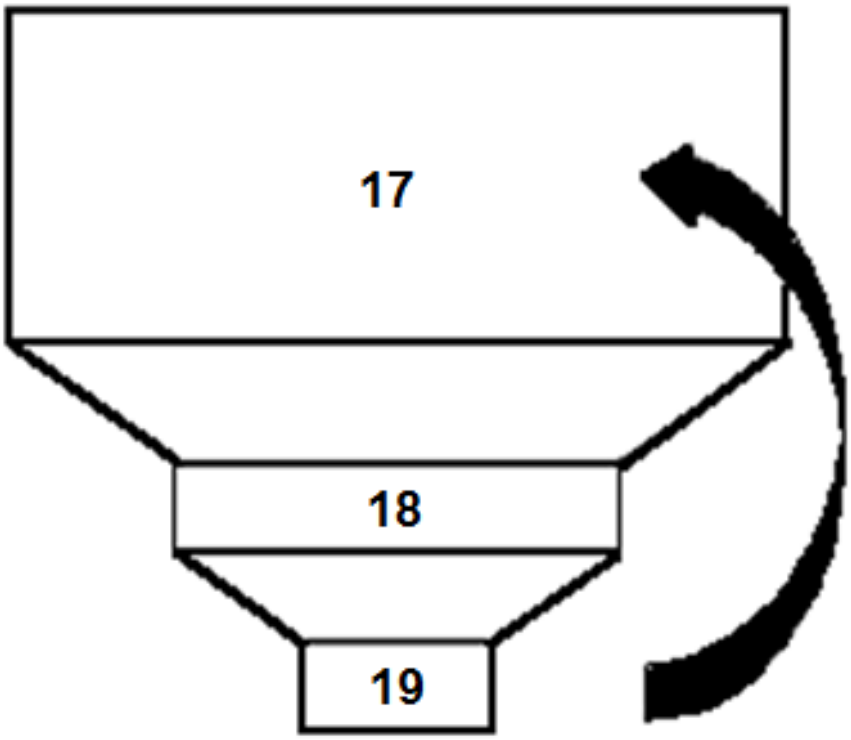

FIG. 4

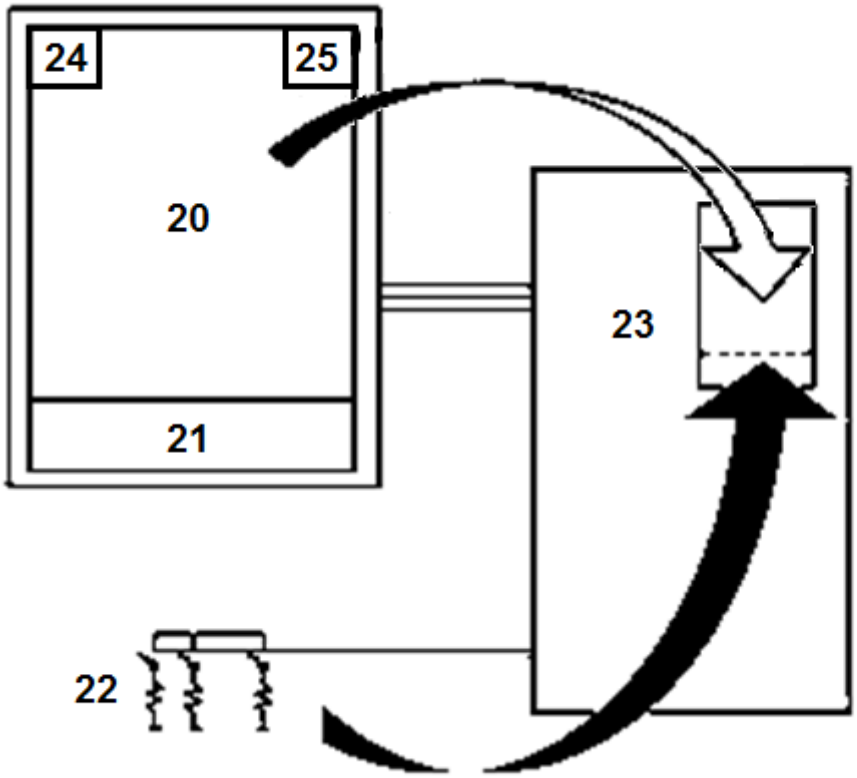

FIG. 5

OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

(21) N.º solicitud: 201430854

(22) Fecha de presentación de la solicitud: 03.06.2014

(32) Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TECNICA

(51) **Int. Cl. :** ***G06F11/08*** (2006.01)

DOCUMENTOS RELEVANTES

Categoría	(56) **Documentos citados**	**Reivindicaciones afectadas**
X	EP 2343705 A1 (SONY CORP) 13.07.2011, Párrafos 6-8; reivindicaciones; figura 1.	1-2
A	GHERMAN V et al. System-level hardware-based protection of memories against soft-errors. Design, Automation&Test in Europe Conference&Exhibition, 2009. DATE '09, 20090420 IEEE, Piscataway, NJ, USA 20.04.2009 VOL: Págs: 1222-1225 ISBN 978-1-4244-3781-8; ISBN 1-4244-3781-4 Doi:10.1109/DATE.2009.5090849.	1-2
A	SAEED SHAMSHIRI et al. End-to-end error correction and online diagnosis for on-chip networks. Test Conference (ITC), 2011 IEEE International, 20110920 IEEE 20.09.2011 VOL: Págs: 1-10 ISBN 978-1-4577-0153-5; ISBN 1-4577-0153-7 Doi:10.1109/TEST.2011.6139156.	1-2
A	US 2004158796 A1 (KOBAYASHI SHOEI et al.) 12.08.2004	1
A	US 6085348 A (SHIMIZU TETSUYA) 04.07.2000	1
A	US 7203890 B1 (NORMOYLE KEVIN B) 10.04.2007	1
A	US 6604222 B1 (JENSEN DAVID W) 05.08.2003	1
A	SHENG YANG et al. Reliable State Retention-Based Embedded Processors Through Monitoring and Recovery. IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, 20111201 IEEE SERVICE CENTER, PISCATAWAY, NJ, US 01.12.2011 VOL: 30 No: 12 Págs: 1773-1785 ISSN 0278-0070 Doi: 0.1109/TCAD.2011.2166590.	1
A	US 2009193314 A1 (MELLIAR-SMITH PETER MICHAEL et al.) 30.07.2009	1

Categoría de los documentos citados
X: de particular relevancia
Y: de particular relevancia combinado con otro/s de la misma categoría
A: refleja el estado de la técnica
O: referido a divulgación no escrita
P: publicado entre la fecha de prioridad y la de presentación de la solicitud
E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

El presente informe ha sido realizado

☒ para todas las reivindicaciones ☐ para las reivindicaciones nº:

Fecha de realización del informe	**Examinador**	**Página**
25.09.2015	M. Muñoz Sánchez	1/5

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

G06F

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

INVENES, EPODOC, WPI, NPL, XPI3E, XPIEE, XPIETF

Fecha de Realización de la Opinión Escrita: 25.09.2015

Declaración

Novedad (Art. 6.1 LP 11/1986)	Reivindicaciones	1-3	**SI**
	Reivindicaciones		**NO**
Actividad inventiva (Art. 8.1 LP11/1986)	Reivindicaciones	3	**SI**
	Reivindicaciones	1-2	**NO**

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

Base de la Opinión.-

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

1. Documentos considerados.-

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

Documento	Número Publicación o Identificación	Fecha Publicación
D01	EP 2343705 A1 (SONY CORP)	13.07.2011
D02	GHERMAN V et al. System-level hardware-based protection of memories against soft-errors. Design, Automation&Test in Europe Conference&Exhibition, 2009. DATE '09, 20090420 IEEE, Piscataway, NJ, USA 20.04.2009 VOL: Págs: 1222-1225 ISBN 978-1-4244-3781-8; ISBN 1-4244-3781-4 Doi: 10.1109/DATE.2009.5090849.	20.04.2009
D03	SAEED SHAMSHIRI et al. End-to-end error correction and online diagnosis for on-chip networks. Test Conference (ITC), 2011 IEEE International, 20110920 IEEE 20.09.2011 VOL: Págs: 1-10 ISBN 978-1-4577-0153-5; ISBN 1-4577-0153-7 Doi: 10.1109/TEST.2011.6139156.	20.09.2011
D04	US 2004158796 A1 (KOBAYASHI SHOEI et al.)	12.08.2004
D05	US 6085348 A (SHIMIZU TETSUYA)	04.07.2000
D06	US 7203890 B1 (NORMOYLE KEVIN B)	10.04.2007
D07	US 6604222 B1 (JENSEN DAVID W)	05.08.2003
D08	SHENG YANG et al. Reliable State Retention-Based Embedded Processors Through Monitoring and Recovery. IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, 20111201 IEEE SERVICE CENTER, PISCATAWAY, NJ, US 01.12.2011 VOL: 30 No: 12 Págs: 1773-1785 ISSN 0278-0070 Doi: 10.1109/TCAD.2011.2166590.	01.12.2011
D09	US 2009193314 A1 (MELLIAR-SMITH PETER MICHAEL et al.)	30.07.2009

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

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

Reivindicaciones independientes

Reivindicación 1: el documento D01 divulga un método de codificación con doble comprobación (códigos externo e interno, el segundo obtenido en una etapa posterior al primero) de errores basada en la paridad empleando también entrelazado por bloques. La redundancia es de dos niveles calculándose la paridad de la paridad del código externo (pár. 6, 7, 8). Asimismo el documento D01 divulga un método de decodificación con funcionamiento inverso al de la codificación. Con la estructura planteada se consiguen dispersar los errores de ráfaga de modo que resulten aleatorios. Dentro de la generalidad de la redacción de la reivindicación 1 se considerarían como bits semilla HSB cualesquiera que permitieran reconstruir la información de usuario y fijados en hardware, siendo sólo esta fijación entonces la diferencia con el documento D01 y teniendo ella como efecto técnico la mayor estabilidad de la información, en el sentido de su manipulación/ persistencia. Esta característica se considera opcional y comúnmente conocida en los términos generales de la reivindicación 1 y, en consecuencia, resulta evidente para el experto en la materia. Para ilustrar este hecho se cita el documento D02. Por tanto, el documento D01 afecta a la actividad inventiva de la reivindicación 1 según el art. 8.1 de la Ley 11/86 de patentes.

Reivindicaciones dependientes

Reivindicación 2: la diferencia aportada al objeto de la invención por el contenido de la reivindicación 2 se refiere a la forma de la fijación física de los bits semilla HSB. Esta fijación no se encuentra divulgada en D01 pero per se es una opción alternativa a otras con igual resultado de carencia de errores, considerándose, en consecuencia, evidente para el experto en la materia. Por tanto, el documento D01 también afecta a la actividad inventiva de la reivindicación 2 según el art. 8.1 de la Ley 11/86 de patentes.

Reivindicación 3: la diferencia aportada al objeto de la invención por el contenido de la reivindicación 3 se refiere a la fase de decodificación con dos funciones duplicadas cuyos resultados se comparan para detectar errores en su lógica combinándose este resultado con los resultados de las comprobaciones de paridad para detectar errores en los datos almacenados con el efecto técnico de aumentar la capacidad de detección/ corrección de errores del método.
El documento D02 por su parte divulga un método para mejorar capacidad de tolerancia a errores en memoria principal mediante códigos de detección y corrección de errores (EDAC). El almacenamiento de cada tipo de código se hace en posiciones de memoria fijas. No se mencionan las dos funciones duplicadas ni la lógica que las relacionaría con la detección y corrección de errores.

El documento D03 tampoco menciona las dos funciones duplicadas ni la lógica que las relacionaría con la detección y corrección de errores limitándose a presentar una solución de corrección de errores para errores aleatorios y de ráfaga basada en el entrelazado.
Así, la reivindicación 3 tiene actividad inventiva según el art. 8.1 de la Ley 11/86 de patentes.