(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 724 117**

(51) Int. CI.:

G06F 7/00 (2006.01)
G06F 21/75 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **18.03.2009 PCT/EP2009/053212**

(87) Fecha y número de publicación internacional: **01.10.2009 WO09118264**

(96) Fecha de presentación y número de la solicitud europea: **18.03.2009 E 09725641 (6)**

(97) Fecha y número de publicación de la concesión europea: **13.02.2019 EP 2257904**

(54) Título: **Procedimiento de protección de circuitos de criptografía programable y circuito protegido por dicho procedimiento**

(30) Prioridad:

25.03.2008 FR 0851904

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
06.09.2019

(73) Titular/es:

INSTITUT MINES TELECOM (50.0%)
37-39, rue Dareau
75014 Paris, FR y
CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (50.0%)

(72) Inventor/es:

GUILLEY, SYLVAIN;
DANGER, JEAN-LUC y
HOOGVORST, PHILIPPE

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Procedimiento de protección de circuitos de criptografía programable y circuito protegido por dicho procedimiento

La presente invención se refiere a un procedimiento de protección de un circuito de criptografía programable. Se refiere igualmente a un circuito protegido por un procedimiento de ese tipo.
Se aplica principalmente para proteger este tipo de circuitos contra los ataques de análisis diferencial del consumo.

La criptografía tiene principalmente por objeto proteger:

- ya sea al secreto de la información por medio del cifrado y de su operación dual: el descifrado;
- o ya sea únicamente su integridad, mediante las operaciones de firma y de verificación de firma.

La criptografía utiliza unos métodos matemáticos seguros, en el sentido de que no existen en el estado actual de los conocimientos publicados unos métodos de ataque más rápidos que el ataque exhaustivo que corresponde al ensayo de todas las claves posibles.
En general, los métodos de cifrado implican unos cálculos complejos necesarios para la seguridad de los sistemas. Esta complejidad no plantea problemas particulares a los ordenadores pero constituye un inconveniente en el caso de dispositivos de gran público que no incluyen una gran potencia de cálculo, en general controlados por unos microprocesadores de reducido coste. Las consecuencias pueden ser entonces de varios órdenes, así por ejemplo una tarjeta bancaria requeriría varios minutos para firmar una transición o un decodificador digital de televisión de pago no podría seguir la velocidad de las informaciones en reproducción.
Para paliar este tipo de problemas sin aumentar el precio de los sistemas, es habitual añadir una ayuda a la unidad central que controla el dispositivo, en general en la forma de un coprocesador dedicado a la criptografía.

Sin embargo, tanto si se implementa mediante la unidad central como mediante un coprocesador especializado, el algoritmo de criptografía se implementa en todos los casos mediante un dispositivo físico, electrónico. Los dispositivos electrónicos presentan imperfecciones inevitables vinculadas a las propiedades inherentes de las leyes de la electricidad.
Es así que unos sistemas criptográficos seguros desde el punto de vista matemático pueden ser atacados aprovechando las imperfecciones de los sistemas físicos que implementan el algoritmo:

- la duración de los cálculos puede depender de los valores de los datos, en particular en unos sistemas programables optimizados en el tiempo, lo que puede dar lugar a los ataques de tipo “timing attack” que permiten en ciertos casos encontrar la totalidad de las claves secretas a partir de simples medidas de tiempos de ejecución;
- el consumo eléctrico instantáneo puede depender igualmente de los datos, lo que puede dar lugar a series de ataques tales como:
    - el SPA que intenta diferenciar las operaciones ejecutadas por una unidad central a partir de una medida de su consumo eléctrico medido durante una operación criptográfica;
    - el análisis diferencial del consumo DPA que utiliza operaciones estadísticas sobre numerosas medidas de consumo eléctrico, efectuadas durante operaciones de criptografía sobre unos mensajes aleatorios y con una clave constante para validar o invalidar una hipótesis realizada sobre la parte limitada de la clave;
    - los ataques de tipo “template” que:
        - en una primera fase utilizan un dispositivo idéntico al dispositivo atacado, con la diferencia de que este dispositivo idéntico no contiene ningún secreto, para construir unos modelos de consumo indexados por el valor de una parte limitada de la clave y;
        - en una segunda fase utilizan algunas medidas de consumo del dispositivo atacado para determinar el modelo cuyos consumos medidos están más próximos y así determinar el valor de esta sub-clave;
- cualquier corriente eléctrica que circule en un conductor genera un campo electromagnético cuya medida puede dar lugar a ataques idénticos en su principio a los ataques que tratan sobre el consumo eléctrico, principalmente por DPA;
- finalmente, unos ataques que perturban el funcionamiento de los sistemas con el fin de aprovechar los resultados falsos para encontrar los secretos del sistema.

Se llama “canal oculto” a toda imperfección de un dispositivo físico que implemente un algoritmo de criptografía y susceptible de dejar fluir información vinculada a los secretos conservados en la memoria del dispositivo.

Los circuitos de configurables de tipo FPGA (Field Programmable Gate Arrays) son muy utilizados en las aplicaciones que precisan de la criptografía. Al menos dos razones justifican esta circunstancia. En primer lugar, las normas de criptografía evolucionan rápidamente, ciertos algoritmos que presentan vulnerabilidades son sustituidas por otros que corrigen los fallos. Además los parámetros de criptografía, como el tamaño de las claves, son igualmente variables. Es por tanto necesaria una flexibilidad, pero sin comprometer el rendimiento. En efecto, los algoritmos de criptografía protegen porque sus cálculos son complejos. Las FPGA responden perfectamente a esta necesidad de flexibilidad y de potencia. En segundo lugar, ciertas aplicaciones de criptografía se difunden en

volúmenes reducidos. Este es por ejemplo el caso de los sistemas integrados principalmente en satélites. La solución FPGA es entonces más rentable con una implementación dedicada del tipo ASIC por ejemplo. Como todos los circuitos de criptografía las FPGA equipadas con funciones de criptografía, son vulnerables a los ataques, principalmente por los canales ocultos.

Una solución de contramedida conocida para contrarrestar los ataques, principalmente por medición de consumo eléctrico utiliza la lógica diferencial, más particularmente la duplicación de las redes lógicas. De ese modo cada puerta lógica es duplicada en dos puertas físicas duales que funcionan en lógica complementaria de tal manera que en todo momento una puerta dual consume, haciendo el consumo eléctrico independiente de los datos y por tanto inutilizable principalmente para una DPA. Para que se asegure un número de transiciones constantes en cada cálculo, por tanto un consumo constante, la lógica diferencial necesita dos fases de trabajo:

- una fase de precarga para poner en un estado conocido las variables y;
- una fase de evaluación en la que el cálculo se efectúa con un número constante de transiciones.

La complejidad de un circuito de criptografía está así más que duplicada debido a la utilización de la lógica diferencial y de las conexiones de doble carril necesarias para su implementación.

Aunque ciertas FPGA integran unas protecciones contra la piratería de su configuración, ninguna se ha concebido para resistir ataques sobre su implementación. Las protecciones provienen por tanto de soluciones a nivel de RTL (Register Transfer Level) principalmente como la lógica WDDL propuesta en el documento de K.Tiri y I. Verbauwhede “A logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation” en Proceedings of DATE’04, páginas 246-251, febrero de 2004, o también la lógica MDPL propuesta en el documento de T.Popp y S.Mangard “Masked Dual Rail Pre-Charge Logic : DPA Resistance without routing Constraints” en LNCS, editor, Proceedings of CHES’05, volumen 3659 de LNCS, páginas 172-186, Springer, septiembre de 2005. Estas soluciones son insuficientes porque presentan unos sesgos lógicos y tecnológicos que pueden ser aprovechados por un atacante.

En todos los tipos de lógica diferencial propuestos, a pesar del aparente equilibrio del consumo, fenómenos de segundo orden dejan aparecer desequilibrios y de ese modo fugas de información. Los fenómenos más importantes son principalmente la evaluación anticipada y las diferentes tecnologías de las redes diferenciales.

El artículo de A. Razafindraibe, M. Roberts y P. Maurine titulado “Analysis Improvement of Dual Rail Logic as a Countermeasure Against DPA” publicado en 2007 divulga un mecanismo de aseguramiento de los circuitos contra ataques de tipo DPA, es decir contra los ataques por análisis diferencial del consumo. Este planteamiento conocido bajo la denominación técnica “STTL” sigue siendo sin embargo complejo (número de hilos requerido), impone el empleo de retardos y es relativamente lento.

Un objeto de la invención es principalmente permitir superar estos fenómenos y aumentar significativamente la dificultad de los ataques por medidas del consumo eléctrico, en particular en los circuitos de criptografía de tecnología FPGA. Con este fin, la invención tiene por objeto un procedimiento de protección de un circuito de criptografía programable, utilizando dicho procedimiento unas puertas compuestas a su vez por células basadas en memoria que definen la función lógica de cada célula, estando configurado el circuito de manera que integre una red diferencial adecuada para efectuar unos cálculos sobre unas variables binarias compuestas de pares de señales, incluyendo la red diferencial una primera red de células que realizan unas funciones lógicas sobre la primera componente de los pares y una segunda red de células duales que funcionan en lógica complementaria sobre la segunda componente de los pares. El cálculo incluye dos fases: una fase de precarga que pone las dos señales de todas las variables en un estado idéntico conocido (por ejemplo 0), una fase de evaluación en la que se efectúa el cálculo propiamente dicho por las células, en la que está activa una única señal de las dos que representan cada variable, y una fase de sincronización antes de cada fase de precarga y de evaluación.

La fase de sincronización de las variables se efectúa por ejemplo a nivel de un grupo de variables y se intercala antes de la fase de evaluación en cada célula adecuada para recibir varias señales que transmiten unas variables de entrada, efectuándose la sincronización sobre la señal más retardada.

La fase de sincronización de las variables puede efectuarse también a nivel de un grupo de variables e intercalarse antes de la fase de precarga en cada célula de cálculo adecuada para recibir varias señales que transmiten unas variables de entrada, efectuándose la sincronización sobre la señal más retardada.

La fase de sincronización se realiza por ejemplo, para cada célula de la red diferencial, mediante un mecanismo de reunión que utiliza unas células de unanimidad cuyas entradas son comunes a las entradas de dicha célula de la red diferencial y cuya salida controla el funcionamiento de dicha célula, teniendo lugar la reunión cuando hay unanimidad de valores sobre las entradas de las células de unanimidad, no cargándose las salidas de la puerta más que cuando la reunión se alcanza como resultado de la sincronización.

Para facilitar la comprensión se considerará que todos los pares de señales (correspondientes a las variables no complementadas y complementadas) están en el estado (0, 0) durante la fase de precarga. El razonamiento se aplicará también con el estado (1, 1).

Las células de unanimidad son de dos tipos: unanimidad a uno U_1, y unanimidad a cero U_0, que tienen las entradas comunes y permiten respectivamente la fase de evaluación y la fase de precarga:

- la célula U_1 de unanimidad a 1 genera una señal que permite la evaluación a partir de que todas las variables de entrada hayan dejado el estado de precarga. Esta condición se convierte en decir que todos los pares de señales de entrada se pasan del estado de precarga (0, 0) a uno de los estados (0, 1) o (1,0).

- la célula U_0 de unanimidad a 0 genera una señal que permite la precarga a partir de que todas las variables de entrada hayan dejado el estado de evaluación. Esta condición se convierte en decir que todos los pares de señales de entrada han pasado del estado de evaluación (0, 1) o (1, 0) al estado de precarga (0, 0).

Las células de cálculo se congelan (o memorizan) en tanto que las unanimidades a uno o cero estén inactivas. Esto se asegura por las funciones de memoria de reunión.
En un modo de realización particular, una puerta recibe la señal global PRE de puesta a cero de las variables de entrada antes de la fase de precarga. Esta señal es común a todas las puertas y se adelanta con relación a las otras señales. La precarga asegurada por PRE permite eliminar a la vez la célula de unanimidad U_0 y la memorización necesaria para la reunión.
La fase de sincronización utiliza por ejemplo:

- una célula U_1 de unanimidad que generan una señal que permite la evaluación a partir de que todas las variables de entrada hayan dejado el estado de precarga;
- una puerta que combina la salida de la célula U_1 y de la señal PRE de puesta a cero;

La señal $U / \overline{PRE}$ combinada de sincronización que permite según su valor binario la fase de precarga ($U / \overline{PRE} = 0$) o la fase de evaluación ($U / \overline{PRE} = 1$).

La invención tiene igualmente por objeto un circuito protegido según el procedimiento anteriormente descrito. El circuito programable, por ejemplo de criptografía, incluye unas puertas a su vez compuestas por células basadas en memoria que definen la función lógica de cada célula, integrando dicho circuito una red diferencial adecuada para efectuar cálculos sobre variables binarias compuestas de pares de señales, incluyendo la red diferencial una primera red de células que realizan las funciones lógicas sobre la primera componente de los pares y una segunda red de células duales que funcionan en lógica complementaria sobre la segunda componente de los pares. Una etapa de cálculo incluye una fase de precarga que pone a las variables en un estado conocido a la entrada de las células, una fase de evaluación en la que se efectúa un cálculo por las células y una fase de sincronización antes de cada fase de precarga y de evaluación.

En un modo de realización particular, una célula de la red diferencial incluye la memoria que define su función lógica asociada a un árbol de multiplexores, recibiendo las entradas de los multiplexores de la primera columna del árbol los valores de la memoria, formando la salida del último multiplexor la salida de la célula, la señal $U / \overline{PRE}$ combinada de sincronización controla los multiplexores de la primera columna, siendo controlados los multiplexores de las otras columnas por las señales de entrada de la célula.
Los pares de señales de las variables de entrada se asocian por ejemplo a la misma columna de multiplexores en su célula respectiva.
El circuito incluye por ejemplo al menos una puerta protegida, utilizándose cuatro células de 2n entradas para generar una puerta protegida de 2n-1 entradas y utilizándose dos células para realizar la unanimidad U_1 y 2 células para la red diferencial.
Incluye por ejemplo al menos una puerta protegida, utilizándose ocho células de 2n entradas para generar una puerta protegida de 2n entradas, utilizándose cuatro células para realizar las unanimidades U_1 y U_0, utilizándose dos células para la red diferencial y utilizándose dos células para la reunión lo que permite congelar las salidas de la puerta, debiendo respetar las funciones lógicas utilizadas la propiedad de crecimiento.
El circuito puede en un modo de realización incluir al menos una célula protegida, utilizándose dos células de 2n entradas para generar una puerta protegida de n entradas, utilizándose las dos células para realizar la red diferencial que integra la unanimidad, debiendo respetar las funciones lógicas utilizadas la propiedad de crecimiento.

Surgirán otras características y ventajas de la invención con ayuda de la descripción que sigue realizada con relación a los dibujos adjuntos que representan:

- la figura 1, una puerta “Y” en lógica diferencial;
- la figura 2, una presentación de las fases de una etapa de cálculo en lógica diferencial en un circuito programable, de tipo FPGA por ejemplo;
- la figura 3, un ejemplo de evaluación anticipada en lógica diferencial;
- la figura 4, una ilustración de fases de sincronización intercaladas entre las fases de precarga y de evaluación en una etapa de cálculo en el interior de un circuito en lógica BCDL (Balanced Cell-Based Differential Logic) según la invención;
- la figura 5, una ilustración de las diferencias de caminos en el interior de una red diferencial para llegar a una célula de cálculo;
- la figura 6, el principio de funcionamiento de la puerta BCDL que realiza la sincronización de las variables a la entrada seguida por el cálculo en la red diferencial;
- la figura 7, una ilustración de las fases de cálculo simplificadas en una puerta BCDL según la invención;
- la figura 8, el principio de funcionamiento de una puerta BCDL para la implementación de las fases de cálculo simplificadas antes citadas con ayuda de una señal global;
- la figura 9, una presentación de los cronogramas de las señales que participan en las fases de cálculo antes mencionadas;

- la figura 10, una representación de la arquitectura de una célula de cálculo en una red programable, principalmente del tipo FPGA;
- las figuras 11a, 11b, 11c y 11d, una ilustración de los equilibrios de consumo en el interior del circuito según la invención;
- la figura 12, una presentación de la estructura de una puerta BCDL;
- la figura 13, un ejemplo de realización de la función de unanimidad en una célula FPGA;
- la figura 14, un ejemplo de realización de una puerta BCDL que no tiene una entrada de precarga global, permitiendo aceptar un gran número de entradas y necesitando células de reunión;
- la figura 15, un ejemplo de realización de una puerta BCDL que tiene pocas entradas y en la que las funciones de unanimidad y de cálculo se integran en la misma célula.

La figura 1 presenta una puerta “Y” 1, 2 en lógica WDDL como ejemplo de ilustración del principio de la lógica diferencial. Esta se compone de dos redes lógicas duales 1, 2, que funcionan en lógicas complementarias. Los datos se representan en doble carril, estando formada cada variable lógica *a* por un par de señales (a_t, a_f) codificadas de la siguiente manera:

- (0, 0) para el estado de reposo: el valor de *a* no está definido, se indica por Ω en lo que sigue:
- (1, 0) es un estado activo en el que *a* = 1;
- (0, 1) es un estado activo en el que *a* = 0.

Una puerta lógica H de dos entradas *a* y *b* y una salida *s* se representa físicamente mediante dos puertas 1, 2 que tienen respectivamente las funciones lógicas $T(a_t, b_t)$ y $F(a_f, b_f)$ tales que:

- $s_t = T(a_t, b_t)$
- $s_f = F(a_f, b_f)$

La red lógica “verdadero” corresponde a la función *T* que proporciona la señal s_t. La red lógica dual “falso” corresponde a la función *F* que proporciona la señal dual s_f. La figura 1 ilustra la puerta “Y” en la que la red “verdadero” 1 que realiza la función *T* recibe las dos entradas a_t y b_t no complementadas. La función dual “O” realiza la función *F*. Para una señal *x* se verifican las relaciones siguientes:

- $T(x) = H(x)$
- $F(x) = \overline{H}(\overline{x})$

La figura 2 presenta las fases de una etapa de cálculo en lógica diferencial, por ejemplo del tipo WDDL (Wave Dynamic Differential Logic). Esta etapa incluye unas fases sucesivas de precarga 21 y de evaluación 22. Unos ejemplos de estados de las variables a_t, b_t, a_f, b_f de entrada y de las variables s_t, s_f de salida correspondientes se presentan con relación a unas fases de precarga y de evaluación. Los cronogramas de la figura 2 muestran que el número de transiciones es el mismo, tres en este caso, durante el paso de la fase de precarga a la fase de evaluación y viceversa. Como el consumo está directamente vinculado al número de transiciones en las tecnologías electrónicas, de tipo CMOS principalmente, de ese modo se equilibra el consumo.
Sin embargo, a pesar de un aparente equilibrio del consumo, unos fenómenos de segundo orden pueden dejar lugar a fugas de información. Por ejemplo si a_t se adelanta o se retrasa con relación a a_f, el desfase temporal puede ser percibido por un atacante que deduce por tanto el valor de la variable *a*. Este fenómeno puede desbaratarse utilizando una interconexión de doble carril equilibrada, es decir con dos líneas perfectamente equilibradas desde un punto de vista eléctrico, principalmente en términos de longitud y de capacidad. Suponiendo que las líneas de doble carril estén equilibradas, existen otros fenómenos que dejan posibilidades de ataques en las lógicas actualmente propuestas. Como se ha indicado anteriormente, los más importantes son principalmente la evaluación anticipada y las diferencias tecnológicas de las redes diferenciales.

La figura 3 ilustra unos programas en lógica WDDL con una evaluación anticipada. Si la señal *a* se adelanta o se retrasa con relación a la señal *b*, puede producirse una evaluación anticipada como lo ilustra esta figura 3. El retardo entre los componentes de las señales *a* y *b* se refleja por tanto a la salida de la puerta 1, 2 por la diferencia lógica entre la función “Y” y la función “O”. Los desfases Δt_1 y Δt_2 permiten saber si la señal *a* vale 0 o 1. Más precisamente en la figura 3, *b* es siempre más rápida que *a*. Según que *b* valga 0, como en la primera mitad de los cronogramas o 1, como en la segunda mitad, la salida se evalúa menos rápido, en Δt_1, o más rápido, en Δt_2, lo que delata su valor. Del mismo modo si hay una evaluación 22 anticipada, existe una precarga 21 anticipada que hace que unas señales sean más rápidas para pasar a 0 que otras.
Dos razones hacen principalmente que *b* sea más rápida que *a*:

- los datos b_t y b_f se conectan directamente a la puerta mientras que a_t y a_f pasan por una multitud de interconexiones y por tanto de elementos de conmutación;
- los datos a_t y a_f pasan por unas puertas intermedias mientras que b_t y b_f llegan directamente.

Otro fenómeno, que implica una vulnerabilidad a los ataques por análisis del consumo, vinculado a la tecnología es la diferencia de energía gastada entre una red lógica y su complementaria. Por ejemplo en lógica WDDL, para la puerta “Y” si *a* vale 1, la puerta “Y” conmuta si *b* vale 1 si no es la puerta “O”. Es posible por tanto deducir el valor de *b* si las transiciones de las puertas “Y” y “O” no gastan la misma energía. La lógica MDPL se aprovecha de este

problema pero con un sobrecoste. Es necesario inicialmente disponer de un generador verdaderamente aleatorio que produzca un bit de máscara por ciclo de reloj. Además debe dedicarse una entrada de cada puerta para la máscara.
Debido a su naturaleza diferencial, la lógica WDDL es necesariamente dos veces más compleja que una lógica normal. Además una limitación importante se añade en la elección de las funciones *T* y *F* que deben ser crecientes. Esta condición permite a la vez evitar las conmutaciones parásitas durante fases de cálculo y garantizar la propagación del valor de precarga a lo largo del cono de lógica. Esta limitación de crecimiento limita el tipo de células en una FPGA. La lógica MDPL es todavía más compleja de realizar en FPGA.

La invención realiza principalmente un nuevo tipo de lógica que elimina los defectos de la evaluación anticipada y unas diferencias tecnológicas que se llamará en lo que sigue lógica BCDL, según la expresión anglosajona “Balanced Cell-based Differential Logic”. En un circuito que funcione en lógica BCDL:

- se añade una etapa de sincronización de los datos a cada puerta antes del paso efectivo a la fase de precarga o de evaluación;
- la sincronización se efectúa a nivel de un grupo de datos;
- el cálculo de sincronización se realiza en paralelo con el cálculo efectuado por el circuito.

La invención aplica este funcionamiento:

- a nivel global, es decir entre las células de cálculo *T*, *F*;
- a nivel local, es decir en el interior de las células de cálculo.

La figura 4 ilustra la etapa de sincronización. Esta etapa de sincronización se efectúa antes de la fase de evaluación, en cada puerta, a través de las redes duales *T*, *F*. Permite principalmente evitar los problemas de evaluación y de precarga anticipadas.
La sincronización consiste en esperar a la señal más retardada. Las causas de la evaluación anticipada provocada por la diferencia de tiempos de cálculo entre dos señales se aniquilan por tanto por la espera a la señal más retardada.

La figura 4 ilustra las cuatro fases 41, 42, 43, 44 sucesivas de sincronización y de cálculo que componen un ciclo de precarga y de evaluación. Una primera fase 41 ejecuta el cálculo de precarga, una segunda fase 42 efectúa una primera sincronización por la espera por ejemplo a la última señal en el estado 1. Esta fase es seguida por la fase 43 de evaluación a su vez seguida por una fase 44 de sincronización antes de la etapa 41 de precarga del ciclo siguiente. Esta etapa 44 de sincronización impone la espera por ejemplo a la última señal a 0.

La figura 5 ilustra una parte de las redes lógicas duales que incluyen dos registros 51, 52 y unas puertas duales *T*, *F* 53, 54, 55. Un primer camino 501 transporta una señal a procedente de un primer registro 51 y un segundo camino 502 transporta una señal b procedente de un segundo registro 52. Los dos caminos se reagrupan a la entrada de una puerta *T*, *F* 55, proporcionando esta última una señal de salida s. La señal a se retarda en el primer camino por una sucesión de puertas 53, 54. Estas puertas reciben por otra parte en la entrada unas señales procedentes de otros caminos 503, 504. A la entrada de cada puerta un bloque 56, 57, 58, efectúa una sincronización. En el ejemplo de la figura 5, la puerta 55 final que proporciona la señal s no efectúa su cálculo más que en un tiempo t_2 correspondiente a la llegada de la señal más retardada, en concreto la señal a en este ejemplo.
La sincronización en lógica asíncrona se realiza entre dos señales con una célula de reunión “RV”. Las células RV no pasan a un valor lógico L más que si las dos entradas tienen el mismo valor lógico L, si no estas no cambian de estado. Una célula RV es por tanto una memoria que no cambia de estado más que si hay unanimidad, a 0 o a 1, de las señales de entrada. En la lógica BCDL la reunión se efectúa sobre un grupo de datos en una misma célula del circuito FPGA. Se utilizan por ejemplo unas células U_0, U_1 específicas.
Una célula U_1 genera una señal que permite la evaluación, esta señal pasa a 1 a partir de que todos los datos hayan dejado el estado Ω anteriormente definido, más particularmente la señal indicada $U_1(x, y, \ldots)$ se define por la relación siguiente:

$$\mathbf{U_1(x, y, \ldots) = 1 \text{ si } x \neq (0, 0) \text{ e } y \neq (0, 0) \ldots, \text{ si no } U_1(x, y, \ldots) = 0} \qquad \mathbf{(1)}$$

Una célula U_0 genera una señal que permite el retorno al estado Ω de las salidas. Esta señal pasa a 1 a partir de que cada una de las entradas esté en el estado Ω de acuerdo con la relación siguiente:

$$\mathbf{U_0(x, y, \ldots) = 1 \text{ si } x = y = (0, 0) \ldots, \text{ si no } U_0(x, y, \ldots) = 0} \qquad \mathbf{(2)}$$

El arranque del cálculo no se realiza por tanto más que si hay unanimidad, es decir si U_0 o U_1 están activos, y el cálculo se congela mientas que no haya unanimidad, es decir si U_0 y U_1 están inactivos.

La figura 6 ilustra el principio de la sincronización en grupo de datos. Los datos a, b, c entran en paralelo hacia la puerta dual *T*, *F* 60, que efectúa el cálculo de precarga o de evaluación, y hacia las células U_1 61 y U_0 62. La célula U_1 envía una señal 63 de autorización de evaluación a la puerta 60 y la célula U_0 envía una señal 64 de autorización de precarga a la puerta.

El cálculo de la precarga es más simple que el cálculo de evaluación porque todas las señales deben pasar al estado 0 mientras que la evaluación corresponde a un verdadero cálculo sobre unas señales que llevan información. Esta propiedad puede aprovecharse en la lógica BCDL suprimiendo por ejemplo la fase 44 de sincronización antes de la precarga utilizando una señal global de puesta al estado 0 por ejemplo, más rápida que las otras señales.
La figura 7 ilustra esta señal PRE que pone a 0 el conjunto de las señales justamente antes de la fase 41 de precarga. El ciclo de precarga y de evaluación no incluye en este caso más que tres fases entre ellas solo una 42 de sincronización antes de la fase 43 de evaluación. La estructura de la puerta se encuentra simplificada por dos razones principalmente:

- no es necesario ya efectuar la unanimidad a 0 porque es sustituida por la señal PRE;
- estando adelantada la señal PRE, no es ya necesario memorizar la salida de la puerta para que esta no cambie más que durante la unanimidad a 0 o a 1 (la reunión). La salida cambia a 0 a continuación tras la llegada de PRE y no puede pasar a 1 más que si hay unanimidad a 1.

La figura 8 ilustra una estructura de puerta en lógica BCDL simplificada porque la unanimidad a 0 necesaria en el caso de evaluación, tal como se ilustra por la figura 7, es sustituida por una puerta “Y” 71 que recibe la orden de paso a precarga por la señal de precarga global PRE. La señal $U / \overline{PRE}$ en la salida 72 de la puerta “Y” permite de ese modo sincronizar el cálculo:

- cuando $U / \overline{PRE}$ pasa a 0, justamente después de la señal PRE. La precarga es forzada independientemente de las entradas y;
- cuando $U / \overline{PRE}$ pasa a 1, indicando que una parte de la señal PRE está a 1 y que por otra parte se efectúa la reunión de los valores de entrada, se inicia la fase de evaluación.

La figura 9 presenta los cronogramas de las señales *a*, *b*, *s* y $U / \overline{PRE}$ en el curso de las fases 21 de precarga y 22 de evaluación. Estos cronogramas muestran que la fase 21 de precarga está forzada independientemente de las entradas durante la transición 91 de la señal $U / \overline{PRE}$.

En las soluciones anteriormente expuestas, aplicadas a nivel global, la lógica BCDL permite combatir los problemas vinculados a la evaluación anticipada a nivel del conjunto del circuito. La robustez con respecto a ataques debe verificarse también localmente a nivel de una única puerta BCDL, en particular para evitar las diferencias tecnológicas y la evaluación anticipada local. Por otro lado, la adición de la sincronización no debe hacerse en detrimento de un gran aumento de la complejidad.
La figura 10 ilustra la estructura de una célula de un circuito FPGA de tres entradas a, b, c. En un circuito FPGA la lógica se realiza en unas células basadas en memoria que incluyen unas tablas 101, llamadas LUT según la expresión anglosajona “Look Up Table”, que definen la función lógica de la célula. La figura 10 muestra que la estructura de la célula se basa en una tabla 101 (LUT) asociada a un árbol de multiplexores 102 que forman tres columnas 103, 104, 105. La tabla 101 memoriza los valores binarios de la función para cada uno de los tripletes (x, y, z) en las que *x*, *y* y *z* toman el valor 0 o 1. La primera columna, o columna de entrada, está formada por multiplexores 102 del primer nivel del árbol, estando formada la tercera columna 105 por el multiplexor del último nivel, es decir el multiplexor de salida. Las entradas a, b, c controlan los multiplexores. El número de columnas corresponde así al número de entradas. Cada uno de los valores s(x, y, z) de la función está presente en la entrada del multiplexor de la columna de entrada 103. Las diferentes combinaciones de los valores binarios a, b, c que controlan los niveles de multiplexores permiten seleccionar las entradas de la columna de entrada 103.
Según la invención, la robustez local se mejora a partir de los dos modos de conexión siguientes:

- la señal $U / \overline{PRE}$ entra en la primera columna del árbol de multiplexores;
- los pares de entradas “verdadero” e_t y “falso” e_f se asocian a la misma patilla respectiva sobre la puerta T y en la puerta F.

Estos modos de conexión permiten obtener unos grandes resultados en lo que se refiere a la seguridad local, son por otro lado muy poco costosos de implementar.
En primer lugar, no hay conmutaciones parásitas. Conmutando la señal $U / \overline{PRE}$ primero durante la fase de precarga, los equipos potenciales internos se fuerzan todos a 0 sin el riesgo de conmutaciones parásitas cuando las entradas de datos conmutan. Igualmente, esta señal conmuta en último lugar antes de la fase de evaluación. De ese modo, los multiplexores controlados por los datos seleccionan ceros. Posteriormente, la señal $U / \overline{PRE}$ deja que el valor de la función transite a través de los multiplexores 102 pre-situados. En segundo lugar, se obtiene una complejidad reducida. En efecto, por la misma razón, no es necesario tener funciones crecientes para evitar las conmutaciones parásitas porque los multiplexores en las entradas ya están situados correctamente. Esto permite principalmente utilizar todas las funciones posibles, en número de 2^n, para una tabla LUT de n entradas, ofreciendo un potencial de optimización mucho mayor que con un subconjunto de funciones crecientes. Por ejemplo para una LUT de 4 entradas, no hay más que 166 funciones crecientes entre las 65536 funciones posibles.
En tercer lugar, se obtiene una fuerte reducción del sesgo tecnológico. El número total de conmutaciones de los equipotenciales de T y de F no cambia en función de las combinaciones de las entradas. Este número es constante,

igual a $2^n - 1$ cuando n es el número de entradas de la LUT. Esto hace así difícil discriminar la actividad de T de la de F porque el perfil de consumo es idéntico a nivel del par T y F. Además, la sucesión de las conmutaciones de los multiplexores en el tiempo es independiente de los datos.
Finalmente hay ausencia de evaluación y de precarga anticipadas en el interior de la tabla LUT. En efecto, es la señal $U / \overline{PRE}$ la que retarda la evaluación llegando siempre después de las señales y que, sin esperar a los datos, fuerza la precarga. En otros términos, la evaluación se retarda siempre y la precarga siempre se anticipa, y esto independientemente de los datos.

Las figuras 11a, 11b, 11c y 11d ilustran el equilibrio de los consumos durante las conmutaciones en lógicas BCDL en el ejemplo de una tabla LUT de tres entradas. Más particularmente, estas cuatro figuras presentan todas las combinaciones en una puerta XOR de dos entradas cuando conmuta la señal $U / \overline{PRE}$. Para todas estas figuras, la célula de arriba es la puerta “verdadero” T y la célula de abajo la puerta “falso” F. Los picos de consumo de energía global en función del tiempo que corresponden a las conmutaciones de los circuitos T y F se representan por una curva 111, estando representados los picos con relación a unas columnas y correspondiendo a la energía consumida durante la conmutación de sus columnas correspondientes. A título de ejemplo, en la figura 11a corresponde al caso en que la señal de entrada, conectada a las segundas 104 y tercera 105 columnas es igual a (0, 0). Los multiplexores 112, 113, 114, 115, 116, 117, 118 representados en un grueso verán conmutar su salida. La curva 111 muestra los picos de consumos de energía que corresponden a esta primera combinación (0, 0). Las curvas 111 de consumos asociados a las combinaciones de las figuras siguientes son idénticas. Hay por tanto un equilibrio global a nivel del par T, F, tanto si es en términos de tiempos de conmutación como de consumo energético. En otros términos, como lo muestran las figuras 11a, 11b, 11c y 11d, hay un número simultáneo de conmutaciones para cada combinación (0, 0), (0, 1), (1, 0), (1, 1).

A partir de un programa de realización de una aplicación en una FPGA cualquiera, el paso a lógica BCDL puede hacerse de una manera automática. Una herramienta de análisis, obtenida a partir de las herramientas estándar de las FPGA permite operar una transformación hacia las lógicas presentadas anteriormente. El análisis se limita a una sustitución de los elementos lógicos hacia las variantes de las puertas BCDL. El enrutado de los pares de hilos de interconexión debe realizarse de manera equilibrada.
La figura 12 ilustra la estructura de una puerta BCDL que contiene cuatro células 121, 122, 123, 124, a saber dos para la unanimidad a 1 121, 122, una para la función T 123 y una para su función dual F 124. Es posible la colocación en cascada de las células U_1 descritas anteriormente.
Una puerta BCDL se compone entonces por ejemplo de:

- dos células duales 123, 124, que funcionan en lógicas complementarias, recibiendo cada célula n entradas, Ei_t para la célula T y Ei_f para la célula F, variando i de 1 a n, las células T y F proporcionan respectivamente las componentes s_t y s_f de la señal de salida;
- dos células U_1 para hacer la unanimidad a 1 y generar la señal $U / \overline{PRE}$, recibiendo cada célula n entradas que corresponden a una mitad de los pares (Ei_t, Ei_f)

La figura 13 ilustra un ejemplo de realización de la célula U'_1 122 unida en cascada a la célula U_1 para formar la unanimidad. Las entradas (Ei_t, Ei_f)... (Ej_t, Ej_f) se unen a unas puertas “O” 121, las salidas de estas puertas “O” se combinan en la entrada de una puerta “Y” 122, combinándose la señal invertida PRE en la entrada de esta puerta 122 con las salidas de las puertas “O”. La salida de la puerta “Y” 122 proporciona la señal $U / \overline{PRE}$.

En el caso de la implementación en la puerta BCDL sin precarga global y con un gran número de entradas, puede no utilizarse la señal de precarga global PRE. La evolución de los cálculos se hace entonces en cuatro fases como se ha indicado anteriormente. Esto permite principalmente ganar una entrada en las tablas LUT de las puertas T y F y aumentar de ese modo el número de entradas. Por el contrario es necesario sustituir la precarga global por el cálculo de unanimidad a 0. Por otro lado, es necesario hacer la reunión (congelar el cálculo) cuando no hay unanimidad. Se utilizan con este fin unas células específicas de reunión.
Una puerta BCDL sin precarga global, ilustrada por la figura 14, se compone por tanto de:

- dos células 123, 124 que funcionan en lógicas complementarias para las funciones T y F, recibiendo cada célula n entradas;
- al menos dos células 143, 144 de reunión RV, que proporcionan las componentes s_t y s_f de la señal de salida, asociada respectivamente a las células de las funciones T y F, dependiendo el número exacto del número de entradas de la LUT.
- dos células U_0 141, 142 para hacer la unanimidad a 0, cuyas salidas se unen a las entradas de las células de reunión;
- dos células U_1 121, 122 para hacer la unanimidad a 1, cuyas salidas se unen a las entradas de las células de reunión;

La puerta ilustrada por la figura 14 está adaptada principalmente para la implementación de una caja de sustitución del algoritmo de cifrado DES.
La puerta BCDL sin precarga global debe respetar siempre la condición de crecimiento sobre las funciones, como la

lógica WDDL por ejemplo.

En el caso de la implementación de una puerta BCDL sin precarga global con pocas entradas, la función de unanimidad a 1 puede integrarse en las células T y F como lo ilustra por ejemplo la figura 15. No es ya necesario entonces utilizar la señal PRE porque la función de unanimidad a 1 repone a cero las funciones T y F con la llegada de la señal más rápida. Este principio no es aplicable sin embargo cuando las células U_1 están separadas porque existe entonces una carrera de las señales, pudiendo pasar las células T y F a la fase de precarga más rápido que lo previsto si la célula U_1 es más lenta. En este caso, hay de nuevo propagación del valor 0 de la precarga porque la señal PRE no existe y es necesario recurrir a funciones crecientes para evitar las conmutaciones parásitas y propagar la precarga a lo largo del cono lógico. En el ejemplo de la figura 15, las componentes duales (Ei_t, Ei_f) de las señales de entrada atacan una puerta “O” 153 integrada en las células T 151 y F 152, estando unidas estas entradas por otro lado a las entradas de las puertas T y F propiamente dichas de las células 151, 152. Las salidas de estas células así como las salidas de las puertas “O” 153 se combinan a la entrada de la puerta “Y” 154 cuyas salidas proporcionan las componentes s_t y s_f de la señal de salida. La figura 15 ilustra una puerta en el caso particular de una función de reducido número de entradas. La lógica para la unanimidad a 1 está integrada en las células T y F, permitiendo así pasar a la fase de precarga con la señal más rápida.

REIVINDICACIONES

1. Procedimiento de protección de un circuito de criptografía programable, utilizando dicho procedimiento unas puertas a su vez compuestas por células basadas en memoria (101) que definen la función lógica de cada célula, estando configurado el circuito de manera que integra una red diferencial adecuada para efectuar cálculos sobre variables binarias compuestas de pares de señales, incluyendo la red diferencial una primera red (1) de células que realizan unas funciones lógicas (T) sobre la primera componente de los pares y una segunda red (2) de células duales que funcionan en lógica complementaria (F) sobre la segunda componente de los pares, incluyendo una etapa de cálculo una fase (41) de precarga que pone las variables en un estado conocido a la entrada de las células, una fase (43) de evaluación en la que se efectúa un cálculo por las células, **caracterizado porque**:

- las puertas están en lógica BCDL, siendo la lógica BCDL una lógica diferencial que comprende además una etapa de sincronización global de los datos previamente a la fase de precarga y una etapa de sincronización local previa a la fase de evaluación.

2. Procedimiento según la reivindicación 1, **caracterizado porque** la fase (42) de sincronización de las variables se efectúa a nivel de un grupo de variables y se intercala antes de la fase de evaluación en cada célula adecuada para recibir varias señales que transmiten unas variables de entrada, efectuándose la sincronización sobre la señal más retardada.

3. Procedimiento según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** la fase (44) de sincronización de las variables se efectúa a nivel de un grupo de variables y se intercala antes de la fase de precarga en cada célula de cálculo adecuada para recibir varias señales que transmiten unas variables de entrada, efectuándose la sincronización sobre la señal más retardada.

4. Procedimiento según una cualquiera de las reivindicaciones 2 o 3, **caracterizado porque** la fase (42, 44) de sincronización se realiza, para cada célula (53, 54, 55) de la red diferencial, mediante un mecanismo (56, 57, 58) de reunión que utiliza unas células de unanimidad cuyas entradas son comunes a las entradas de dicha célula de la red diferencial y cuya salida controla el funcionamiento de dicha célula, teniendo lugar la reunión cuando hay unanimidad de valores en las entradas de las células de unanimidad, no cambiando las salidas de la puerta más que cuando la reunión se alcanza como resultado de la sincronización.

5. Procedimiento según la reivindicación 4, **caracterizado porque** la fase de sincronización utiliza dos células de unanimidad U_1 (61), U_0 (62) que tienen las entradas comunes a la red diferencial de cálculo y permiten respectivamente la fase de evaluación y la fase de precarga:

- generando la célula U_1 una señal (63) que permite la evaluación a partir de que todas sus entradas, pares de señales asociadas a cada variable, hayan dejado el estado de precarga;
- generando la célula U_0 una señal (64) que permite la precarga a partir de que todas sus entradas, pares de señales asociadas a cada variable, hayan pasado al estado de precarga.

6. Procedimiento según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** utiliza para la fase de sincronización una señal (PRE) global de puesta a cero de las variables de entrada antes de la fase de precarga, en lugar de la célula U_0, adelantándose la señal (PRE) con relación a las otras señales.

7. Procedimiento según la reivindicación 6, **caracterizado porque** la fase de sincronización utiliza:

- una célula U_1 (61) que tiene las entradas comunes a la red diferencial de cálculo y que genera una señal que permite la evaluación a partir de que todas sus entradas hayan dejado el estado de precarga;
- una puerta (71) que combina la salida de la célula U_1 y la señal (PRE) de puesta a cero;

permitiendo la señal ($U / \overline{PRE}$) combinada de sincronización según su valor binario la fase de precarga o la fase de evaluación.

8. Circuito programable, que incluye unas puertas a su vez compuestas por células basadas en memoria (101) que definen la función lógica de cada célula, integrando dicho circuito una red diferencial adecuada para efectuar cálculos sobre variables binarias compuestas de pares de señales, incluyendo la red diferencial una primera red (1) de células que realizan unas funciones lógicas (T) sobre la primera componente de los pares y una segunda red (2) de células duales que funcionan en lógica complementaria (F) sobre la segunda componente de los pares, incluyendo una etapa de cálculo una fase (41) de precarga que pone las variables en un estado conocido a la entrada de las células, una fase (43) de evaluación en la que se efectúa un cálculo por las células, **caracterizado porque**:

- las puertas están en lógica BCDL, siendo la lógica BCDL una lógica diferencial que comprende además una etapa de sincronización global de los datos previamente a la fase de precarga y una etapa de sincronización local previa a la fase de evaluación.

9. Circuito según la reivindicación 8, **caracterizado porque** la fase (42) de sincronización de las variables se efectúa

a nivel de un grupo de variables y se intercala antes de la fase de evaluación en cada célula adecuada para recibir varias señales que transmiten unas variables de entrada, efectuándose la sincronización sobre la señal más retardada.

10. Circuito según una cualquiera de las reivindicaciones 8 o 9, **caracterizado porque** la fase (44) de sincronización de las variables se efectúa a nivel de un grupo de variables y se intercala antes de la fase de precarga en cada célula de cálculo adecuada para recibir varias señales que transmiten unas variables de entrada, efectuándose la sincronización sobre la señal más retardada.

11. Circuito según una cualquiera de las reivindicaciones 9 o 10, **caracterizado porque** la fase (42, 44) de sincronización se realiza, para cada célula (53, 54, 55) de la red diferencial, mediante un mecanismo (56, 57, 58) de reunión que utiliza unas células de unanimidad cuyas entradas son comunes a las entradas de dicha célula de la red diferencial y cuya salida controla el funcionamiento de dicha célula, teniendo lugar la reunión cuando hay unanimidad de valores en las entradas de las células de unanimidad, no cambiando las salidas de la puerta más que cuando la reunión se alcanza como resultado de la sincronización.

12. Circuito según la reivindicación 11, **caracterizado porque** la fase de sincronización utiliza dos células de unanimidad U_1 (61), U_0 (62) que tienen las entradas comunes a la red diferencial de cálculo y permiten respectivamente la fase de evaluación y la fase de precarga:

- generando la célula U_1 una señal (63) que permite la evaluación a partir de que todas sus entradas, pares de señales asociadas a cada variable, hayan dejado el estado de precarga;
- generando la célula U_0 una señal (64) que permite la precarga a partir de que todas sus entradas, pares de señales asociadas a cada variable, hayan pasado al estado de precarga.

13. Circuito según una cualquiera de las reivindicaciones 8 a 12, **caracterizado porque** utiliza para la fase de sincronización una señal (PRE) global de puesta a cero de las variables de entrada antes de la fase de precarga, en lugar de la célula U_0, adelantándose la señal (PRE) con relación a las otras señales.

14. Circuito según la reivindicación 13, **caracterizado porque** la fase de sincronización utiliza:

- una célula U_1 (61) que tiene las entradas comunes a la red diferencial de cálculo y que genera una señal que permite la evaluación a partir de que todas sus entradas hayan dejado el estado de precarga;
- una puerta (71) que combina la salida de la célula U_1 y la señal (PRE) de puesta a cero;

permitiendo la señal ($U / \overline{PRE}$) combinada de sincronización según su valor binario la fase de precarga o la fase de evaluación.

15. Circuito según la reivindicación 14, **caracterizado porque** una célula de la red diferencial incluyendo la memoria (101) que define su función lógica asociada a un árbol de multiplexores (103, 104, 105), recibiendo las entradas de los multiplexores de la primera columna (103) del árbol los valores de la memoria, formando la salida del último multiplexor (102) la salida de la célula, la señal ($U / \overline{PRE}$) combinada de sincronización controla los multiplexores de la primera columna (103), estando controlados los otros multiplexores de las otras columnas por las señales de entrada de la célula.

16. Circuito según la reivindicación 15, **caracterizado porque** los pares de señales de las variables de entrada se asocian a la misma columna de multiplexores en su célula respectiva.

17. Circuito según una cualquiera de las reivindicaciones 8 a 16, **caracterizado porque** incluye al menos una puerta protegida, utilizándose cuatro células de 2n entradas para generar una puerta protegida de 2n-1 entradas (12) y utilizándose dos células para realizar la unanimidad U_1 y 2 células para la red diferencial.

18. Circuito según una cualquiera de las reivindicaciones 8 a 17, **caracterizado porque** incluye al menos una puerta protegida, utilizándose ocho células de 2n entradas para generar una puerta (14) protegida de 2n entradas, utilizándose cuatro células para realizar las unanimidades U_1 y U_0, utilizándose dos células para la red diferencial y utilizándose dos células para la reunión que permite congelar las salidas de la puerta, debiendo respetar las funciones lógicas utilizadas la propiedad de crecimiento.

19. Circuito según una cualquiera de las reivindicaciones 8 a 18, **caracterizado porque** incluye al menos una célula protegida, utilizándose dos células de 2n entradas para generar una puerta (15) protegida de n entradas, utilizándose las dos células para realizar la red diferencial que integra la unanimidad, debiendo respetar las funciones lógicas utilizadas la propiedad de crecimiento.

20. Circuito según una cualquiera de las reivindicaciones 8 a 19, **caracterizado porque** efectúa una función de criptografía.

21. Circuito según una cualquiera de las reivindicaciones 8 a 20, **caracterizado porque** dicho circuito es del tipo FPGA.

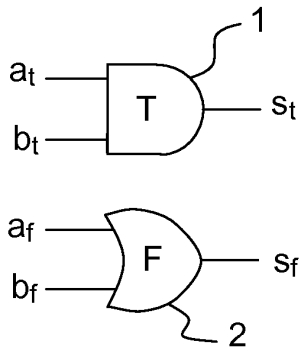

FIG.1

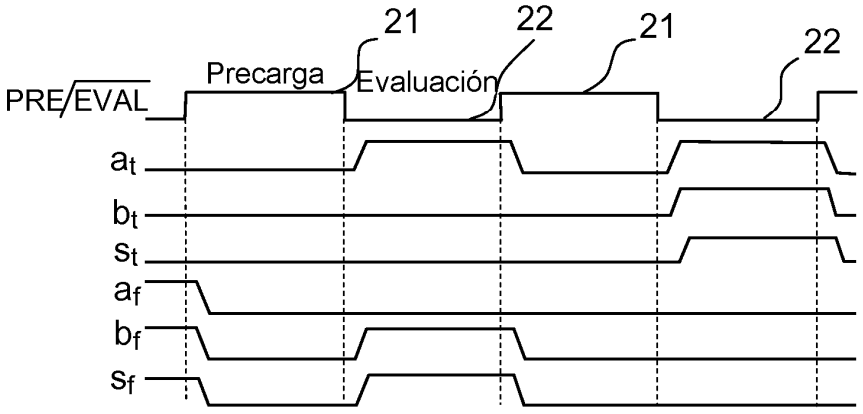

FIG.2

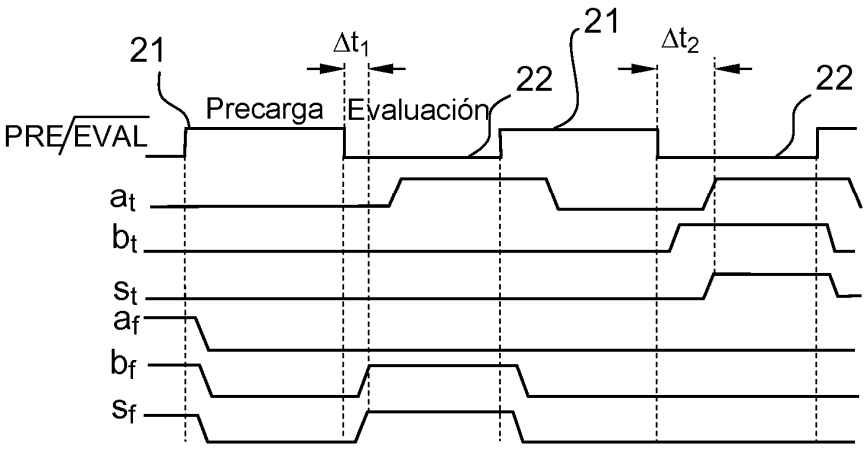

FIG.3

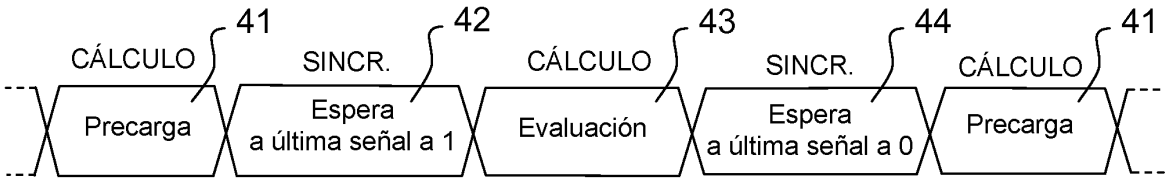

FIG.4

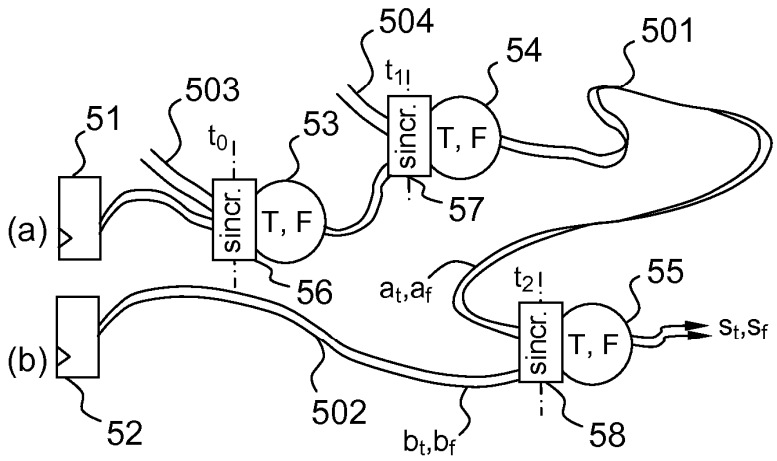

FIG.5

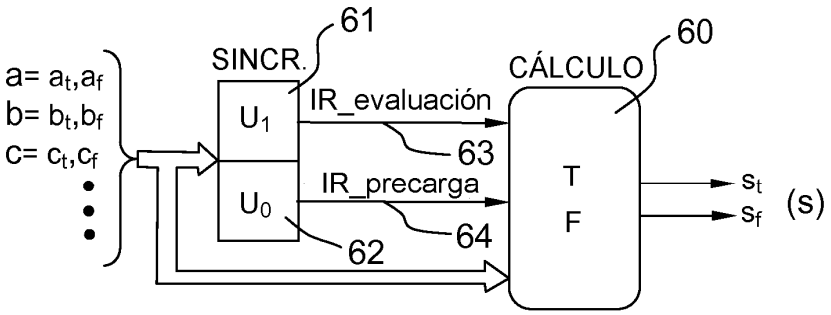

FIG.6

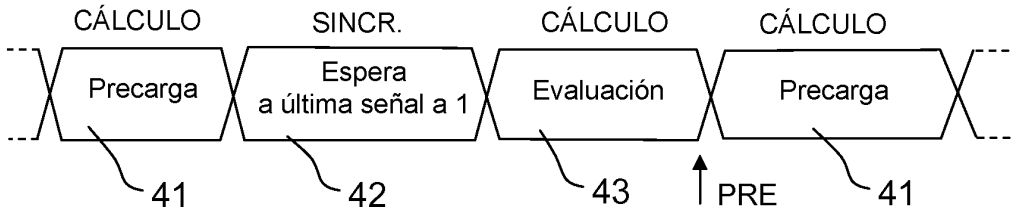

FIG.7

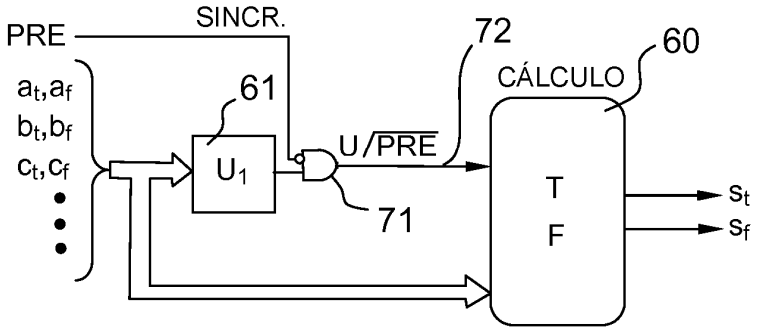

FIG.8

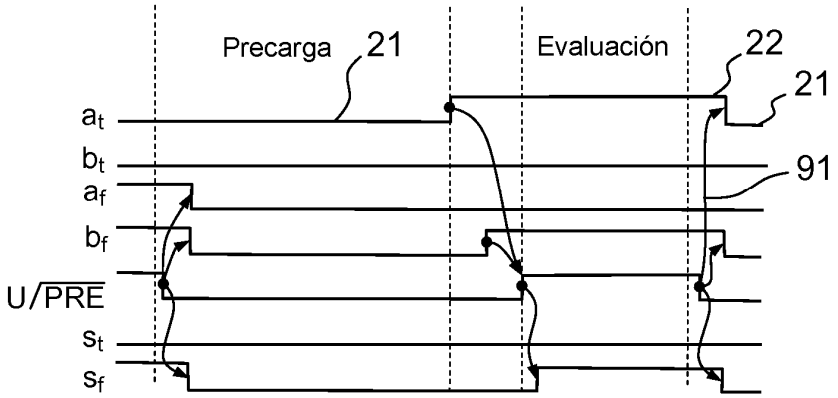

FIG.9

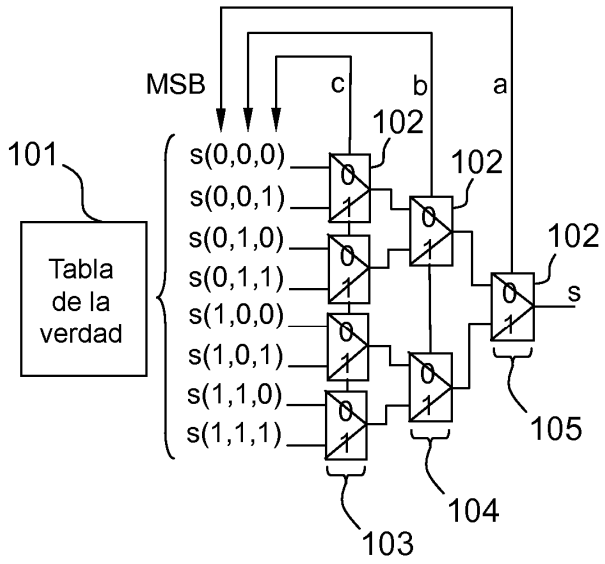

FIG.10

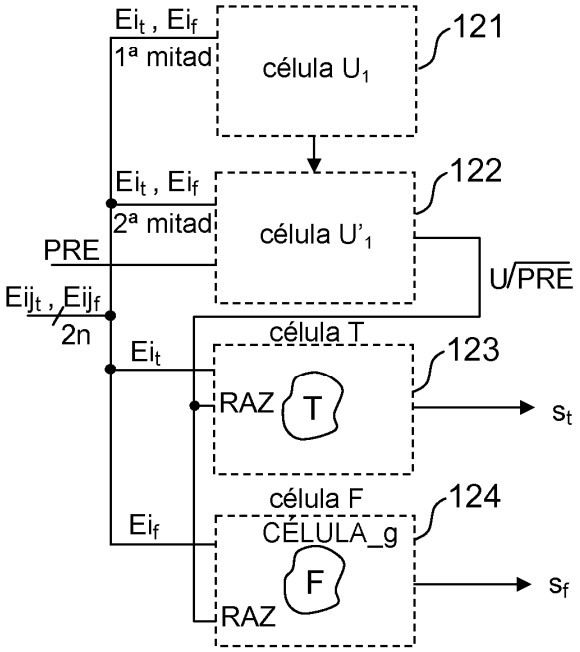

FIG.12

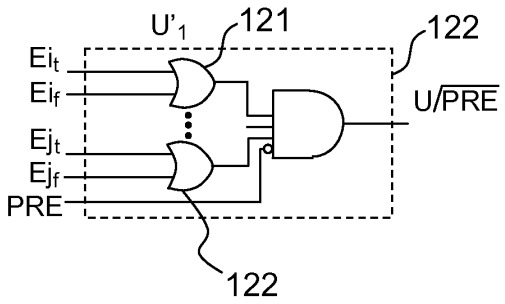

FIG.13

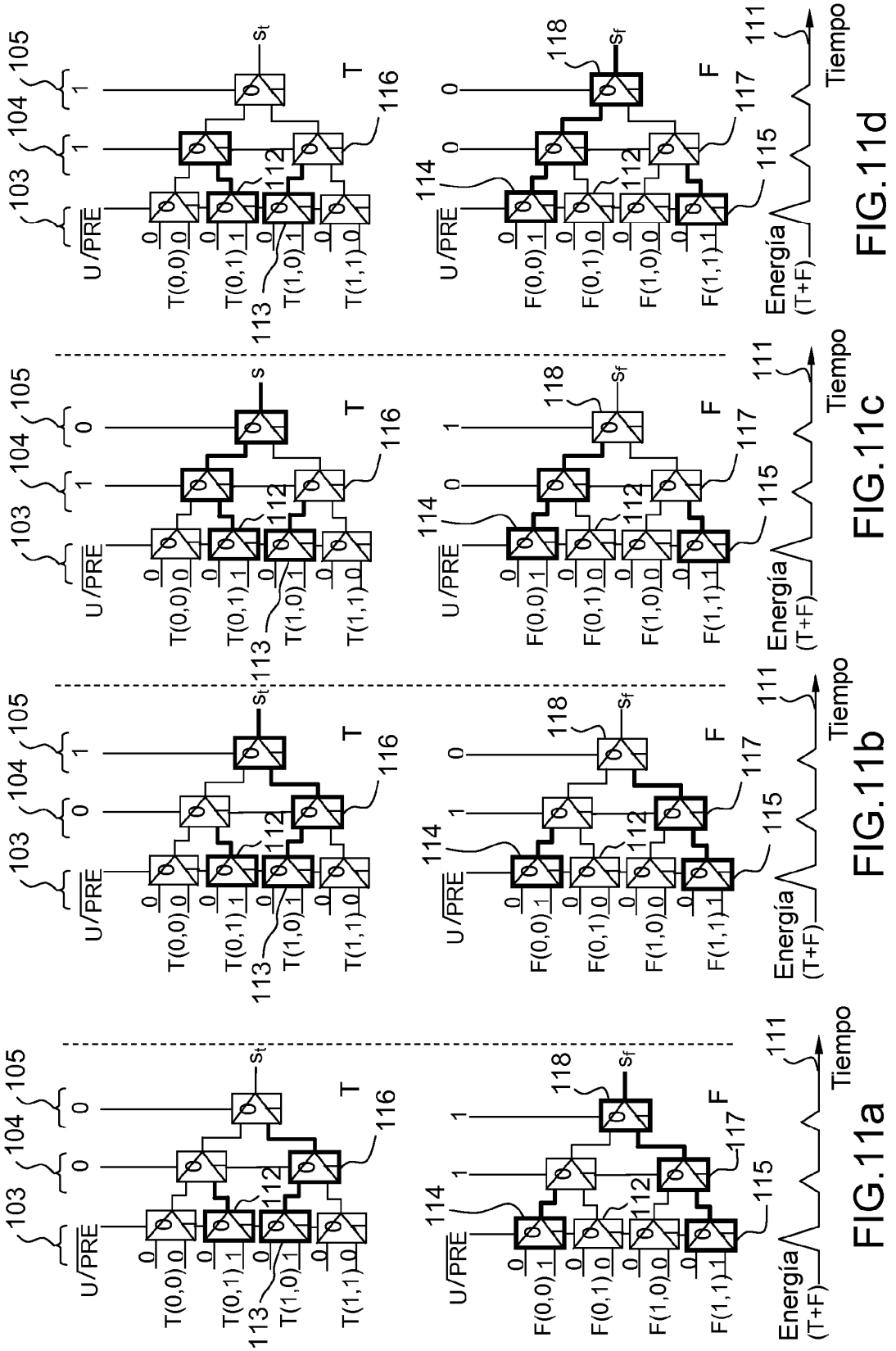

FIG.11a

FIG.11b

FIG.11c

FIG.11d

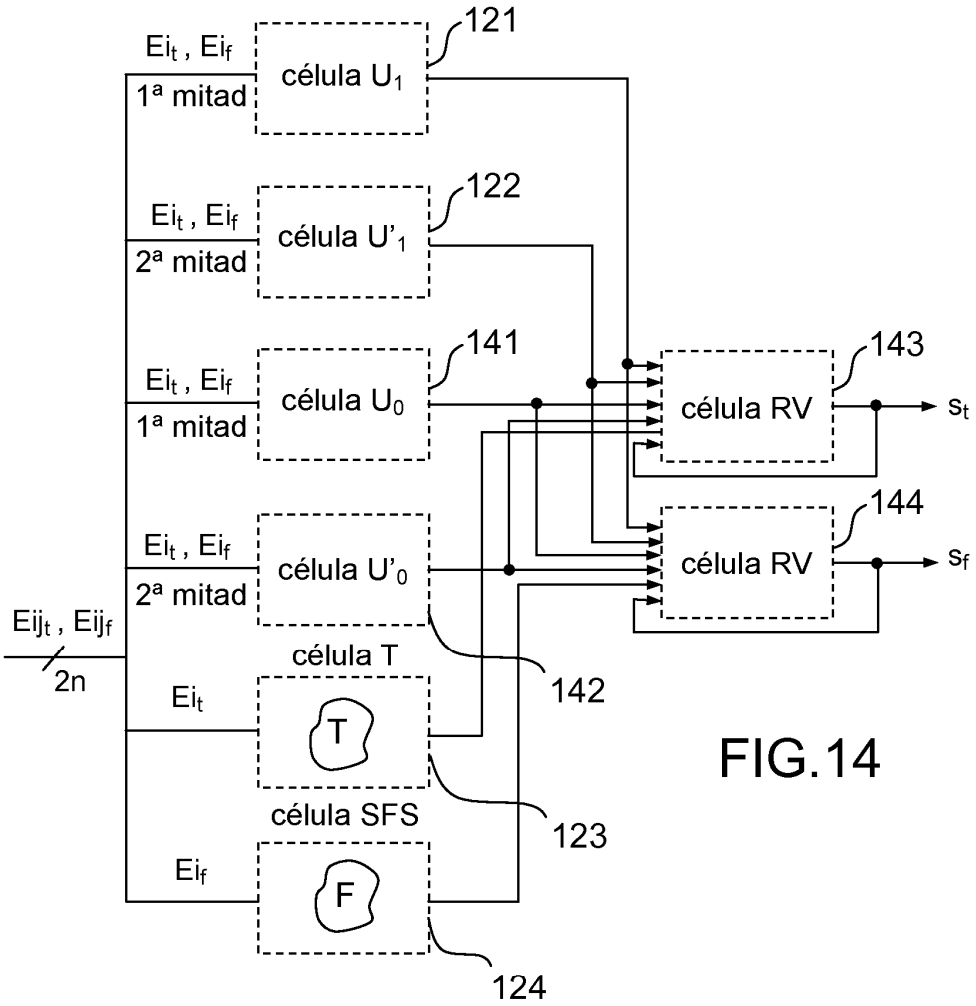
Eit , Eif
1ª mitad
célula U1
121
Eit , Eif
2ª mitad
célula U'1
122
Eit , Eif
1ª mitad
célula U0
141
Eit , Eif
2ª mitad
célula U'0
Eijt , Eijf
2n
célula T
Eit
T
142
célula SFS
Eif
F
123
124
célula RV
143
st
célula RV
144
sf

FIG.14

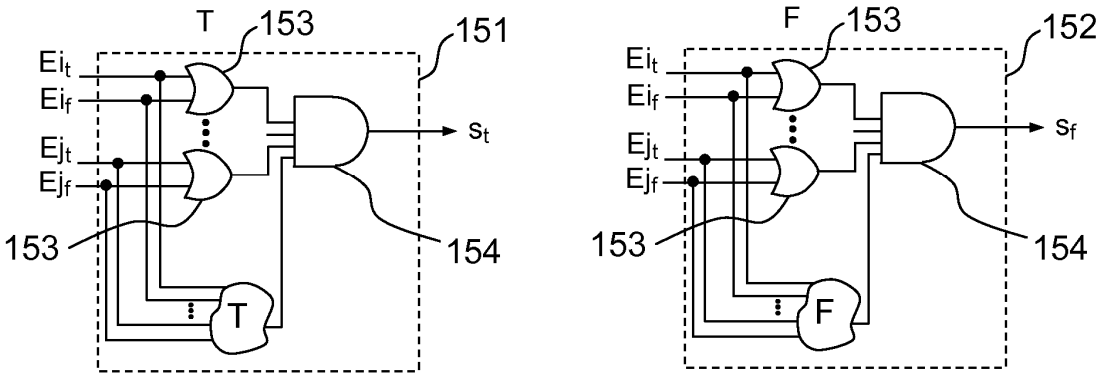
T
153
151
Eit
Eif
Ejt
Ejf
153
st
154
T
F
153
152
Eit
Eif
Ejt
Ejf
153
sf
154
F

FIG.15