

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 764 990**

51 Int. Cl.:

G06F 11/16 (2006.01)

H03K 19/007 (2006.01)

H03K 19/177 (2006.01)

G01R 31/317 (2006.01)

G01R 31/3177 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **16.04.2018** **E 18167418 (5)**

97 Fecha y número de publicación de la concesión europea: **09.10.2019** **EP 3392773**

54 Título: **Circuito integrado digital protegido contra errores transitorios**

30 Prioridad:

20.04.2017 FR 1700443

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

05.06.2020

73 Titular/es:

**THALES (100.0%)
Tour Carpe Diem, Place des Corolles,
Esplanade Nord
92400 Courbevoie, FR**

72 Inventor/es:

**OSTER, YANN NICOLAS PIERRE y
BARRES, LOÏC PIERRE**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 764 990 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Circuito integrado digital protegido contra errores transitorios

La invención se refiere al campo de los circuitos integrados digitales, por ejemplo, circuitos de lógica programable tales como los circuitos FPGA (*Field Programmable Gate Array*) o circuitos integrados específicos para una aplicación o ASIC (*Application Specific Integrated Circuit*).

De forma más precisa, la invención se refiere a un circuito integrado digital que integra un mecanismo de protección contra los errores transitorios.

Los circuitos integrados digitales pueden estar sometidos a radiaciones o partículas ionizantes que producen errores transitorios en el circuito y perturban el funcionamiento nominal.

En el caso de aplicaciones aerotransportadas o aeroespaciales, los circuitos integrados digitales están expuestos a efectos de radiaciones ionizantes que provienen del entorno exterior. En una menor medida, incluso para aplicaciones embarcadas en sistemas en tierra, las cajas en las cuales los circuitos están encapsulados generan radiaciones alfa que producen errores con una ocurrencia tanto más mayor cuando los elementos son complejos y comprenden un número muy grande de puertas lógicas.

Los efectos producidos por radiaciones en un circuito pueden ser progresivos o singulares. En el primer caso, los defectos generados por las radiaciones son acumulados progresivamente hasta generar errores a partir de un cierto umbral de radiaciones acumuladas. En el segundo caso, una sola partícula ionizante o un solo evento pueden generar un defecto inmediato que puede afectar a las memorias, los biestables o las puertas lógicas que contienen el circuito. Si el efecto genera una inversión de un bit, se habla de "*single event upset*" (SEU), si por el contrario impacta varios bits, se habla de "*multiple-bit upset*" (MBU). Los efectos de las radiaciones pueden tomar la forma de parásitos o impulsos que se propagan en el circuito y afectan a la lógica combinatoria al nivel de un cable eléctrico o de una puerta lógica (se habla de "*single event transient*" (SET) en este caso. Estos errores (SEU, MBU, SET) son reversibles y no afectan a la interpretación de las señales eléctricas en la lógica secuencial y combinatoria, con la consecuencia de, errores lógicos que afectan a circuito.

El documento US 2004/219739A1 describe un circuito de respaldo y de recuperación de estados pasados de un biestable cuando se han detectado errores por medios de detección.

El problema que plantea la invención consiste en proteger un circuito digital contra errores reversibles inducidos por radiaciones, sin penalizar demasiado la complejidad lógica del circuito y la disipación, y de manera transparente para el servicio o la misión realizada por el circuito, en particular concerniente al ritmo de ejecución de tratamientos, y en particular sin interrupción del servicio.

Son conocidas varias soluciones que permiten luchar contra los efectos de radiaciones en un circuito integrado digital.

Una primera solución consiste en utilizar un blindaje metálico para limitar la interacción de las partículas o radiaciones ionizantes con el sustrato del chip del circuito integrado. Por razones de volumen y de masa, el espesor del blindaje debe sin embargo permanecer limitado lo que no permite proteger totalmente el circuito de errores transitorios. Por tanto, esta primera solución resulta insuficiente.

Una segunda solución consiste en utilizar una tecnología de silicio sobre aislante (SOI) como alternativa al silicio bruto, para realizar el circuito integrado. Esta tecnología permite reducir los errores transitorios gracias a la utilización de sustratos fuertemente resistentes, sin embargo, produce un coste de fabricación más elevado sin suprimir totalmente los efectos de las radiaciones.

Una tercera solución conocida está basada en el empleo de una biblioteca específica de puertas lógicas que integran de forma nativa un cierto nivel de redundancia gracias a un diseño específico de las puertas lógicas. Esta solución depende del fabricante de circuitos integrados y corresponde a una tecnología particular. El diseño de puertas lógicas integra márgenes superiores con respecto a la concepción de transistores, con capacidades más fuertes, tiempos de ascenso más lentos, células para filtrar los impulsos parásitos y redundancias en las estructuras con bucle con el fin de hacer la lógica tolerante a los impulsos parásitos. Un inconveniente de esta solución es que es específica para un tipo de tecnología dada presentando por tanto una oferta limitada y rendimientos reducidos. De hecho, el hecho de integrar la redundancia de las estructuras lógicas induce un lugar ocupado más importante, un consumo más elevado y una velocidad de ejecución más lenta.

Otra solución consiste en implementar la redundancia a nivel funcional del circuito, de su arquitectura. Por ejemplo, se conoce proteger el contenido de las memorias para la utilización de códigos detectores y correctores de errores, tal como el código de Hamming extendido.

Para proteger las puertas lógicas de un circuito, se conoce del mismo modo la técnica denominada de triplicación modular y de voto mayoritario (TMR) que permite corregir un error en un proceso de una función entre un grupo de

tres procesos de la misma función que operan en paralelo. Este principio puede ser aplicado al nivel de biestable o de un conjunto lógico combinatorio o incluso de una función. Permite proteger la lógica secuencial y la lógica combinatoria contra un error irrelevante en uno de los tres procesos. Por el contrario, si dos errores intervienen simultáneamente en dos procesos, no pueden ser corregidos. Por otro lado, un inconveniente de esta solución es que resulta muy costosa en complejidad lógica y en disipación.

Se conocen del mismo modo las técnicas de detección y corrección de errores transitorios descritos en el artículo "*Power consumption improvement with residue code for fault tolerance on SRAM FPGA, Frédéric Amiel et. Al, ISEP*". Éste artículo presenta métodos de detección y de corrección de errores transitorios por replicación de la función y comparación de los resultados, así como métodos de detección de errores por proyección modulada de la función y comparación de resultados.

Se conoce también el artículo "*Designing fault-tolerant techniques for SRAM-based FPGAs, F. Gusmao de Lima Kastensmidt, IEEE design & test of computers*" que presenta el efecto de las radiaciones en circuitos integrados FPGA SRAM así como una implantación clásica de protección por triplicación modular y voto mayoritario TMR. Este artículo propone del mismo modo una técnica de protección de la lógica combinatoria menos costosa que el método TMR con:

- una replicación de circuitos combinatorio para detectar un error,
- una modificación de estos circuitos para poder volver a realizar en caso de error el cálculo en un ciclo suplementario a partir de operandos después de la codificación, siendo el resultado de codificado y después comparado con el primer resultado con el fin de identificar cuál proceso de entre los dos es un error.

La lógica secuencial permanece protegida por el método TMR.

Esta técnica de protección presenta el inconveniente de ser demasiado costosa en recursos lógicos debido a la modificación en profundidad de toda la lógica combinatoria y del método TMR para proteger los biestables, con una reducción significativa de la velocidad de los circuitos.

La invención contempla resolver las limitaciones de las soluciones de la técnica anterior citadas anteriormente proponiendo una solución de protección de un circuito integrado digital que sea ligeramente complejo y que permita proteger todos los recursos lógicos de una función implementada por un circuito sin interrumpir el servicio y sin producir impacto en el ritmo de ejecución de la función tal y como se observa desde las interfaces de entrada y de salida.

La invención tiene por objeto un circuito integrado digital que comprende un conjunto lógico que comprende un bloque lógico funcional, una unidad lógica de detección de errores transitorios que afecta al bloque lógico funcional, una memoria FIFO de entrada para alimentar de muestras el bloque lógico funcional, una memoria FIFO de salida para recibir muestras a la salida del bloque lógico funcional, una memoria de búfer alimentada de muestras por la memoria FIFO de entrada, y una unidad lógica de control apta para controlar, el acceso de lectura en la memoria FIFO de entrada y el acceso de escritura en la memoria FIFO de salida y configurada para, cuando se detecta un error por la unidad lógica de detección de errores transitorios, reinicializar la unidad lógica de detección de errores transitorios y el bloque lógico funcional, suspender el acceso de escritura en la memoria FIFO de salida y conmutar la entrada del bloque lógico funcional hacia la salida de la memoria de búfer.

Según un aspecto particular, el circuito integrado digital según la invención comprende un multiplexor para conectar la entrada del bloque lógico funcional a la memoria FIFO de entrada o a la memoria de búfer.

Según un aspecto particular de la invención, la unidad lógica de control está configurada para controlar el multiplexor en función del resultado de la unidad lógica de detección de errores.

Según un aspecto particular de la invención, la frecuencia del reloj en el seno del conjunto lógico es superior a la frecuencia media de escritura en la memoria FIFO de entrada.

Según un aspecto particular de la invención, la actividad en el seno del conjunto lógico se hace irregular.

Según un aspecto particular de la invención, el ritmo Ra de los tratamientos en el seno del conjunto lógico es controlado por la unidad lógica de control por medio de una señal de activación.

Según un aspecto particular de la invención, el ritmo Ra de los tratamientos en el seno del conjunto lógico es controlado por la unidad lógica de control en función del nivel de llenado de la memoria FIFO de salida o de una detección de error señalada por la unidad lógica de detección de errores transitorios.

Según un aspecto particular de la invención, la unidad lógica de control está configurada para modificar el ritmo RA de los tratamientos en el seno del conjunto lógico en función de una comparación del nivel de llenado de la memoria FIFO de salida en al menos tres umbrales.

5 Según un aspecto particular de la invención, la memoria FIFO de entrada y la memoria FIFO de salida están dimensionadas para almacenar al menos $p \cdot L_p$ datos, con p el número máximo de errores consecutivos tolerables que afectan en ráfaga al bloque lógico funcional y L_p una estimación de la latencia del bloque lógico funcional.

Según un aspecto particular de la invención, el tamaño de la memoria de búfer depende de la latencia del bloque lógico funcional.

10 Según un aspecto particular de la invención, la unidad lógica de detección de errores transitorios comprende un bloque lógico funcional redundante y una unidad de comparación de resultados producidos respectivamente por el bloque lógico funcional y el bloque lógico funcional redundante.

Según un aspecto particular de la invención, el bloque lógico funcional es de latencia constante.

15 Según un aspecto particular de la invención, el bloque lógico funcional está provisto de al menos una segunda entrada y de al menos una segunda salida para, cuando están conectadas entre sí, realizar al menos un bucle de reacción externa, dicho conjunto lógico comprende al menos una segunda memoria de búfer alimentada de muestras por dicha al menos una segunda salida del bloque lógico funcional, estando configurada la unidad lógica de control para, cuando un error es detectado por la unidad lógica de detección de errores transitorios, conmutar dicha al menos una segunda entrada del bloque lógico funcional de dicha al menos una segunda salida de bloque lógico funcional hacia la salida de dicha al menos una segunda memoria de búfer.

20 Según un aspecto particular de la invención, dicho conjunto lógico comprende al menos un segundo multiplexor para conectar dicha al menos una segunda entrada del bloque lógico funcional a dicha al menos una segunda salida del bloque lógico funcional o a dicha al menos una segunda memoria de búfer.

Según un aspecto particular de la invención, la unidad lógica de control está configurada para controlar el segundo multiplexor de cada conjunto en función del resultado de la unidad lógica de detección de errores.

25 Según una variante particular de la invención, el circuito integrado digital según la invención comprende una pluralidad de dichos conjuntos lógicos y en los cuales una memoria FIFO de entrada de un primer conjunto lógico es una memoria FIFO de salida para un segundo conjunto lógico situado aguas arriba.

Otras características y ventajas de la presente invención aparecerán mejor de la lectura de la descripción siguiente en relación a los dibujos adjuntos que representan:

- 30
- la figura 1, un diagrama sinóptico de un circuito integrado según la invención, adaptado a las funciones de latencia constante
 - las figuras 2a y 2b, dos ejemplos de aplicación de la invención en uno o varios bloques de tratamiento funcional,
 - la figura 2c, una ilustración, para el ejemplo de la figura 2b, de una secuencia de inicialización del circuito protegido por la invención,
- 35
- las figuras 3a y 3b, dos diagramas que ilustran respectivamente el nivel de llenado de una memoria FIFO de salida y de una memoria FIFO de entrada en el transcurso de la ejecución de la invención,
 - la figura 4a, un esquema de un circuito que comprende varios bloques lógicos funcionales de latencia constante,
- 40
- la figura 4b, un esquema de un circuito que comprende varios bloques lógicos funcionales con bucle de reacción,
 - la figura 5, un diagrama sinóptico de una variante de realización del circuito integrado digital según la invención adaptado a las funciones con bucle de reacción,
- 45
- las figuras 6a, 6b, 6c, tres esquemas que ilustran la transformación de un bloque lógico funcional que comprende un bucle de reacción interna en un bloque lógico funcional para el cual el bucle de reacción se hace externo y se puede abrir.

La invención consiste en implementar en el circuito integrado digital, un mecanismo de protección contra los errores transitorios para cada función del circuito a proteger.

La figura 1 representa una parte 100 de un circuito integrado que comprende un bloque 101 lógico funcional que ejecuta una función sensible a los errores transitorios, que la invención contempla proteger. El bloque 101 lógico funcional puede corresponderse a una función lógica elemental o a un conjunto de funciones lógicas interconectadas para realizar un tratamiento particular. Desde este punto de vista, el bloque 101 lógico funcional es modular y corresponde a un reagrupamiento de una o varias puertas lógicas del circuito a proteger. El mecanismo de protección a los errores proporcionado por la invención es construido alrededor del bloque 101 lógico funcional a proteger. Por tanto, en un mismo circuito integrado, varios bloques lógicos funcionales pueden protegerse mediante la adición de un mecanismo de protección asociado a cada uno de los bloques. Una limitación del bloque 101 lógico funcional es que debe tener una latencia constante, dicho de otra manera, el número de muestras de entrada necesarias para generar un resultado debe ser constante sean cual sean los datos de entrada. El bloque 101 lógico funcional está integrado en la unidad 102 lógica de detección de errores transitorios.

El mecanismo de protección según la invención comprende principalmente una unidad 102 lógica de detección de errores transitorios, una memoria 103 FIFO de entrada para recibir datos en forma de muestras y alimentar la entrada de la unidad 102 lógica de detección de errores transitorios que incluye el bloque 101 lógico funcional, una memoria 104 FIFO de salida para recibir las muestras producidas por el bloque 101 lógico funcional de salida, una memoria 105 de búfer y una unidad 106 lógica de control. La memoria 103 FIFO de entrada y la memoria 104 FIFO de salida son llenadas sobre la marcha según un principio conocido denominado de primero en entrar, primero en salir o *First In First Out* en inglés. La memoria 103 FIFO de entrada es alimentada por tratamientos funcionales situados aguas arriba. La memoria 104 FIFO de salida proporciona datos a los tratamientos funcionales situados aguas abajo. La memoria 105 de búfer es llenada sobre la marcha con las L últimas muestras leídas en la memoria 103 FIFO de entrada, siendo L el tamaño de la memoria 105 de búfer.

La unidad 106 lógica de control es apta para controlar los diferentes elementos del mecanismo de protección. En particular, controla la actividad de la unidad 102 lógica de detección de errores, la actividad del bloque 101 lógico funcional, el acceso a la lectura en la memoria 103 FIFO de entrada y el acceso a la escritura en la memoria 104 FIFO de salida.

El principio de protección del bloque 101 de lógica funcional es el siguiente. La unidad 102 lógica de detección de errores permite, cada vez que se produce un resultado por el bloque 101 de lógica funcional, detectar que un error transitorio ha perturbado la ejecución del tratamiento. Son posibles varias soluciones conocidas para implementar un mecanismo de detección de errores.

Una primera solución consiste en añadir un segundo proceso 110 del bloque 101 de lógica funcional que es ejecutado en paralelo con los mismos datos de entrada y por tanto se espera que se produzca el mismo resultado. Un comparador 111 permite detectar una diferencia entre los resultados producidos por los dos procesos 101, 110 del bloque lógico funcional, esta diferencia que se traduce en un error transitorio en uno de los dos procesos.

Esta primera solución puede ser menos costosa en el número de operaciones a realizar y en puertas lógicas implementando una proyección modulada de la función a redundar. Dicho de otra manera, el segundo proceso 110 del bloque lógico funcional reproduce los tratamientos del primer proceso 101 pero proyectando los datos los operandos y los resultados intermedios, en cualquier punto del bloque, a un valor N modulado, donde N es un número entero estrictamente positivo. El comparador 111 realiza a continuación una comparación entre el resultado del primer proceso 101 del bloque lógico funcional, proyectado a su valor N modulado y el resultado producido por el segundo proceso 110. Esta variante de implementación no es sin embargo compatible con funciones lineales, y no permite detectar todos los errores.

Son posibles otras soluciones de detección de errores, tales como la utilización de una codificación de detector de errores aplicada a los datos antes de ser tratados por el bloque 101 lógico funcional. La unidad 102 lógica de detección de errores comprende por tanto un módulo de detección de errores basado en la decodificación del código detector de errores.

Sin salir del ámbito de la invención, se puede contemplar cualquier mecanismo que permita detectar un error en el resultado producido por el bloque 101 lógico funcional, por redundancia espacial, por adición de redundancia a los datos a tratar o por otros medios.

La unidad 102 lógica de detección de errores está conectada a la unidad 106 lógica de control con el fin de informar de la detección de un error. Cuando se produce este evento, la unidad 106 lógica de control prohíbe provisionalmente la escritura en la memoria 104 FIFO de salida, prohíbe provisionalmente la lectura en la memoria 103 FIFO de entrada, reinicializa la unidad 102 lógica de detección de errores y el bloque 101 lógico funcional, y después controla el acceso de lectura de la memoria 105 de búfer para proporcionar una secuencia de muestras, anteriores al error, en la entrada de todos los procesos 101, 110 del bloque lógico funcional con el fin de restaurar el estado de este bloque. La conmutación de la entrada del bloque 101 lógico funcional de la memoria 103 FIFO de entrada hacia la memoria 105 de búfer puede realizarse por medio de un multiplexor 107 o cualquier otro medio equivalente. El bloque 101 lógico funcional ejecuta a continuación su tratamiento en los datos leídos en la memoria de búfer que son idénticos a los datos ya tratados anteriormente y que han conducido a un resultado erróneo debido

a un error transitorio. Bajo el control de la unidad 106 lógica de control, la memoria 105 de búfer cesa de grabar las muestras leídas en la memoria 103 FIFO de entrada siempre que el estado interno del bloque 101 funcional no haya sido completamente restaurado.

5 En funcionamiento nominal, es decir cuando no se ha detectado ningún error en la salida del bloque 101 lógico funcional, las muestras son leídas directamente en la memoria 103 FIFO de entrada. En paralelo, las L últimas muestras leídas en la memoria 103 FIFO de entrada son en cada instante almacenadas en la memoria 105 de búfer que está conectada a la memoria 103 FIFO de entrada. El tamaño L de la memoria 105 de búfer está directamente ligado a la latencia del bloque 101 lógico funcional. Es al menos igual a esta latencia, expresada en número de muestras. De hecho, la función de la memoria 105 de búfer es conservar, en cada instante, las L últimas muestras
10 leídas en la memoria 103 FIFO de entrada para permitir una restauración del estado interno del bloque 101 lógico funcional y una repetición de la función ejecutada por el bloque 101 lógico funcional sobre estas L últimas muestras si se produce un error transitorio. El término “repetición” se emplea para designar una nueva ejecución del tratamiento operado por un bloque lógico funcional en las mismas muestras que durante la secuencia anterior.

15 La invención se refiere al caso general de circuitos integrados digitales, para los cuales se puede difícilmente respaldar y restaurar el estado interno, al contrario que en el caso de circuitos digitales de tipo microprocesador, microcontrolador, procesador multi-núcleo, procesador de núcleos múltiples, procesador gráfico GPU o procesador de tratamiento de señal DSP cuyo estado interno es completamente definido por un grupo limitado de registros accesibles en escritura y lectura a través de un bus.

20 En lugar de respaldar el estado interno, se respalda la secuencia anterior de las muestras de entrada. En lugar de restaurar un estado interno archivado anteriormente, se restaura indirectamente el estado interno repitiendo el tratamiento de una secuencia anterior, hasta el punto de la latencia de la función.

25 Como se explicó al principio de este documento, un error transitorio es reversible y es suficiente generalmente reinicializar el bloque 101 lógico funcional para suprimir el efecto de error. Si a pesar de esto se detecta un nuevo error por la unidad 102 lógica de detección de errores, el mecanismo de repetición es iterado tantas veces como sea necesario hasta que el resultado producido por el bloque 101 lógico funcional sea validado. Por tanto, la memoria de búfer permite remontar en el tiempo respaldando los datos necesarios para restaurar el estado interno del bloque 101 lógico funcional y repetir o ejecutar de nuevo la función afectada por el error después de su reinicialización. El tamaño de la memoria de búfer es por tanto al menos igual a la latencia del bloque 101 lógico funcional.

30 Un objetivo de la invención es proteger el bloque 101 lógico funcional contra errores transitorios garantizando en todo momento una continuidad del servicio del circuito integrado. Dicho de otra manera, el mecanismo de repetición activado tras la detección de un error no debe perturbar el funcionamiento global del circuito y en particular no debe modificar su velocidad de ejecución tal y como se puede observar desde las interfaces de entrada y de salida.

35 Por ello, el diseño del circuito está constituido de dos dominios de reloj distintos. La parte 120 protegida del circuito integrado, que comprende la unidad 102 lógica de detección de errores, el bloque 101 lógico funcional, la unidad 106 lógica de control, la memoria 105 de tampón y el multiplexor 107, tiene una cadencia a un ritmo más rápido que el ritmo medio nominal de funcionamiento de las interfaces de entrada y de salida del bloque 101 lógico funcional. La memoria 103 FIFO de entrada y la memoria 104 FIFO de salida son memorias sincronizadas de doble puerta y de doble reloj también designadas por el acrónimo inglés “*dual-clock FIFO*” que están interconectadas entre los dos dominios de reloj. Dicho de otra manera, cada memoria FIFO tiene un puerto de entrada asociado a un primer reloj y un puerto de salida asociado un segundo reloj. Esto permite desacoplar el ritmo de datos entre el dominio de reloj
40 externo y el dominio de reloj interno acelerado.

45 La aceleración del ritmo de funcionamiento del sistema de protección permite compensar el retardo inducido por los tratamientos suplementarios necesarios para la corrección del error con el fin de conservar un funcionamiento sin error al ritmo medio nominal tal y como se puede observar en las interfaces del bloque 120 lógico protegido por la invención. Además, el ritmo acelerado del sistema de protección se hace irregular, con el fin de poder disponer de un margen temporal para efectuar la reinicialización de la unidad 102 lógica de detección de errores que incluye el bloque 101 lógico funcional, para repetir el tratamiento implementado por este bloque, y para recuperar el retardo en el tratamiento de datos en caso de detección de error.

La secuenciación del sistema de protección se asegura por la unidad 106 lógica de control que controla:

- 50
- la fuente de datos de entrada de la unidad 102 lógica de detección de errores que incluye el bloque 101 lógico funcional, es decir la memoria 103 FIFO de entrada o la memoria 105 de búfer,
 - la reinicialización de la unidad 102 lógica de detección de errores que incluye el bloque 101 lógico funcional,
 - la actividad de la unidad 102 lógica de detección de errores que incluye el bloque 101 lógico funcional,
 - la transmisión de datos de salida del bloque 101 lógico funcional hacia la memoria 104 FIFO de salida.

La secuenciación es irregular y está condicionada por la disponibilidad de datos en la memoria 103 FIFO de entrada, por el sitio disponible en la memoria 104 FIFO de salida, por la detección de un error por la unidad 102 lógica de detección de errores y por la latencia del bloque 101 lógico funcional.

5 Por ejemplo, cuando la memoria 104 FIFO de salida está llena, la unidad 106 lógica de control bloquea la actividad de la unidad 102 lógica de detección de errores que incluye el bloque 101 lógico funcional lo que impide nuevos accesos de escritura en la memoria 104 FIFO de salida hasta que se libera el espacio. Del mismo modo, si la memoria 103 FIFO de entrada está vacía, la unidad 106 lógica de control bloquea la actividad de la unidad 102
10 lógica de detección de errores que incluye el bloque 101 lógico funcional lo que impide nuevos accesos de lectura en la memoria 103 FIFO de entrada siempre que las nuevas muestras no estén disponibles en esta memoria. Por tanto, la memoria 104 FIFO de salida es apta para comunicar a la unidad 106 lógica de control su estado de llenado. Del mismo modo, para la memoria 103 FIFO de entrada.

15 En lo que se refiere a la continuidad del servicio y de la frecuencia media de ejecución de los tratamientos tales y como se pueden observar en el exterior del sistema 120 de protección depende directamente del dimensionamiento de la memoria 103 FIFO de entrada y de la memoria 104 FIFO de salida. Por tanto, este dimensionamiento depende en particular de la latencia del bloque 101 lógico funcional, del número máximo de errores a tolerar en ráfaga, y de la aceleración de la frecuencia de tratamiento en el seno del sistema 120 de protección.

20 La aceleración irregular del ritmo de ejecución en el seno del sistema 120 de protección se puede realizar por medio, por un lado, de un reloj regular a una frecuencia acelerada, y por otro lado, con la ayuda de una señal de activación (en inglés: *clock enable*) que condiciona la toma en cuenta del cero por los biestables de la lógica secuencial, estando controlada esta señal de activación por la unidad 106 lógica de control. Esto permite implantaciones sincronas en el reloj continuo, con la posibilidad de regular dinámicamente el ritmo de secuenciación por la señal de activación, según la necesidad. La aceleración irregular puede comprender ciclos eventualmente inactivos. Por ello, la señal de activación tiene la capacidad de congelar la lógica secuencial para forzar los ciclos inactivos.

25 La invención se puede aplicar globalmente al conjunto de una cadena 200 de tratamientos o a varios bloques 201, 202, 203 de una cadena de tratamiento como la que se ilustra en las figuras 2a y 2b. Cada bloque de la cadena de tratamiento se realiza con una latencia fija y constante que depende de la complejidad de los tratamientos en cada bloque.

30 En el caso de la figura 2a, el circuito protegido por la invención comprende una sola memoria FIFO de entrada $FIFO_e$ y una sola memoria FIFO de salida $FIFO_s$ que sirven para interconectar el dominio de reloj interno a la cadena 200 de tratamiento protegida por la invención y el dominio de reloj externo correspondiente a las entradas y salidas de la cadena 200 de tratamiento.

35 En el caso de la figura 2b, cada bloque protegido por la invención es interconectado con un bloque aguas arriba a través de una memoria FIFO de entrada y con un bloque aguas abajo a través de una memoria FIFO de salida. Por tanto, en el ejemplo de la figura 2b, que comprende tres bloques 201, 202, 203, el circuito comprende una memoria FIFO de entrada $FIFO_e$ conectada a la entrada del primer bloque 201, una memoria FIFO de salida $FIFO_s$ conectada a la salida del último bloque 203 y dos memorias FIFO intermedias $FIFO_{i1}$, $FIFO_{i2}$ que sirven a la vez de memoria FIFO de entrada y de memoria FIFO de salida. La memoria $FIFO_{i1}$ sirve de memoria de entrada al bloque 202 y de memoria de salida al bloque 201. La memoria $FIFO_{i2}$ sirve de memoria de entrada al bloque 203 y de memoria de salida al bloque 202.

40 Como se explicó anteriormente, el dominio de reloj interno en un bloque 200, 201, 202, 203 de tratamiento protegido por la invención se acelera con respecto a la necesidad nominal. Para asegurar una continuidad de servicio en el conjunto del circuito, el nivel de llenado de las memorias FIFO de entrada y de salida se controla por la unidad lógica de control de cada bloque.

45 Se describe a continuación la lógica de control del nivel de llenado de una memoria FIFO de interfaz de entrada $FIFO_e$ o de salida $FIFO_s$.

50 De forma ideal, para poder efectuar una repetición de un tratamiento funcional tras la detección de un error en este tratamiento, la memoria FIFO de salida en la interfaz externa debe estar casi llena, con el fin de optimizar la capacidad para proporcionar los datos de salida válidos durante la repetición, durante la cual se suspende la escritura en la memoria FIFO de salida. Del mismo modo, la memoria FIFO de entrada en la interfaz externa debe estar casi vacía, con el fin de optimizar la capacidad de almacenar datos de entrada durante una repetición, durante la cual se suspende la lectura en la memoria FIFO de entrada. Se puede por tanto optimizar el dimensionamiento de las dos memorias FIFO regulando el funcionamiento nominal en estos niveles de llenado deseados. Dicho de otra manera, fuera de las fases de recuperación de error, el nivel de llenado de una memoria FIFO de entrada es controlado para estar casi vacío y el nivel de llenado de una memoria FIFO de salida es controlado para estar casi
55 lleno. La expresión "casi vacío" debe comprenderse como que significa que la memoria casi vacía contiene un número mínimo de datos o muestras, típicamente del orden de la decena de muestras. Del mismo modo, la expresión "casi lleno" debe comprenderse como que significa que la memoria casi llena está casi completamente

llena con un espacio de memoria libre correspondiente a un orden de magnitud de una decena de datos o muestras. Estas consideraciones no se aplican por supuesto a las memorias FIFO intermedias que presentan la doble función de memoria de entrada y de salida $FIFO_{i1}$, $FIFO_{i2}$.

5 Una memoria FIFO que se llena provoca la suspensión del tratamiento aguas arriba. Una memoria FIFO que se vacía provoca la suspensión del tratamiento aguas abajo.

10 En el caso de las memorias $FIFO_{i1}$, $FIFO_{i2}$ intermedias que interconectan dos bloques de tratamiento protegidos por la invención, hace falta conciliar los objetivos de llenado para las necesidades a la vez en la entrada y la salida. En este caso, el nivel de llenado deseado óptimo es por tanto un nivel medio de un 50%. En la salida de una cadena de tratamiento, se podrá observar una ruptura de secuencia de resultados (lo que se corresponde también a resultados erróneos) en particular si:

- la memoria FIFO de entrada de un bloque está llena mientras se ordenan nuevas escrituras
- la memoria FIFO de entrada de un bloque está vacía mientras se ordenan nuevas lecturas,
- la memoria FIFO de salida de un bloque está vacía mientras se ordenan nuevas lecturas,
- la memoria FIFO de salida de un bloque está llena mientras se ordenan nuevas escrituras.

15 Por tanto, hace falta gestionar dinámicamente el nivel de llenado de las diferentes memorias FIFO de interfaz (entre bloques y con el exterior de los bloques) para evitar estas rupturas de secuencia en la salida. Esta gestión dinámica es efectuada por la unidad 106 lógica de control de cada bloque protegido por la invención. En la práctica, la actividad de cada bloque está condicionada por el nivel de llenado de su memoria FIFO de entrada y de su memoria FIFO de salida. Esta actividad influye en la actividad de otros bloques por intervalos, sirviendo las memorias FIFO de interfaz. A ambos lados de la cadena de tratamiento, los datos de entrada llegan a un ritmo R dado y los datos de salida deben ser generados a este mismo ritmo R en el caso de un tratamiento de cadencia única.

20 El ritmo R es un ritmo medio de funcionamiento de una cadena de tratamiento de cadencia única. Se corresponde al flujo de muestras de entrada o de salida. Este ritmo medio corresponde también a una frecuencia F de reloj media.

25 La lógica de un bloque 120 funcional protegido por la invención tiene una cadencia a un ritmo R_a en un reloj de frecuencia F_a acelerada relativamente a la frecuencia F de reloj media externa al bloque 120: $F_a = \gamma * F$ y $R_a = \gamma * R$ (con $\gamma > 1$). La frecuencia F de reloj media externa al bloque 120 se corresponde con la frecuencia media de escritura en la memoria FIFO de entrada del bloque 120. La tasa γ de aceleración de la frecuencia puede elegirse muy próxima a 1, típicamente $\gamma=1,05$ o $\gamma=1,1$ para limitar la diferencia entre el ritmo R medio funcional y el ritmo R_a acelerado de secuenciación de la lógica del bloque 120 funcional. Esto permite limitar el impacto del dispositivo de protección en la velocidad de tratamiento. El bloque 120 funcional funciona en un reloj a la frecuencia F_a y de manera condicional con una señal de activación que controla la actividad de bloque según un funcionamiento irregular como se explicó anteriormente.

35 Para un bloque 120 de tratamiento dado, se evita el llenado excesivo de la memoria FIFO de entrada aumentando la actividad media del bloque, lo que induce un aumento del ritmo medio de lectura de esta memoria FIFO de entrada. La actividad media modulada (disminuida o aumentada) dinámicamente a través del pilotaje de la señal de activación específica en el bloque de tratamiento. La señal de activación es controlada por la unidad 106 lógica de control. Esta señal de activación es típicamente generada por un secuenciador, en el reloj a la frecuencia F_a de funcionamiento del bloque 120 protegido. Si se denomina γ_i a la tasa de actividad (variable con $0 \leq \gamma_i \leq 1$) de la señal de activación por tanto el ritmo de tratamiento de bloque 120 es igual a $\gamma_i * R_a = \gamma_i * \gamma * R$. La tasa γ_i de actividad varía dinámicamente y de forma independiente para cada bloque de tratamiento protegido.

40 Siempre que la aceleración γ y el ritmo R medios son fijos, la tasa γ_i de actividad de la señal de activación se puede ajustar dinámicamente por una lógica de control secuenciada en el reloj acelerado (a la frecuencia F_a). La generación de la señal de activación permite por tanto ajustar dinámicamente el ritmo de tratamiento del bloque 120. Cuando la memoria 103 FIFO de entrada del bloque 120 presenta un nivel de llenado demasiado elevado, el ritmo de tratamiento del bloque 120 se aumenta con el fin de leer la memoria 103 FIFO de entrada más rápidamente que se llena. Tan pronto como su nivel de llenado desciende por debajo de un umbral dado, el ritmo de tratamiento del bloque 120 se ralentiza, siempre a través de la orden de la señal de activación. El control del nivel de llenado de las memorias 103, 104 FIFO es también efectuado por la unidad 106 lógica de control.

50 El aumento puntual del ritmo de tratamiento del bloque 120 conduce a un aumento del nivel de llenado de su memoria 104 FIFO de salida (si un bloque situado aguas abajo al bloque 120 no es en sí mismo acelerado) y a una disminución del nivel de llenado de su memoria 103 FIFO de entrada (si un bloque situado aguas arriba al bloque 120 no es en sí mismo acelerado). Esto permite transformar un excedente de datos de la memoria 103 FIFO de entrada en un excedente de datos en la memoria 104 FIFO de salida.

5 La disminución puntual del ritmo de tratamiento del bloque 120 conduce a una disminución del nivel de llenado de su memoria 104 FIFO de salida (si un bloque situado aguas abajo al bloque 120 no es en sí mismo ralentizado) y a un aumento del nivel de su memoria 103 FIFO de entrada (si un bloque situado aguas arriba al bloque 120 no es en sí mismo ralentizado). Esto permite transformar indirectamente un excedente de datos de la memoria 104 FIFO de salida en un excedente de datos en la memoria 103 FIFO de entrada. Por tanto, la unidad 106 lógica de control controla dinámicamente la actividad de cada bloque de la cadena de tratamiento para ajustar los niveles de llenado de las memorias FIFO, de aguas arriba hacia aguas abajo, o de aguas abajo hacia aguas arriba.

10 Con el fin de ser operativo, el mecanismo de protección aportado por la invención debe ser inicializado, en particular para llenar las memorias FIFO intermedias $FIFO_{i1}$ y $FIFO_{i2}$ y la memoria FIFO de salida $FIFO_s$ con los niveles deseados de llenado, respectivamente a medio lleno y casi lleno como se describió anteriormente. Esta operación de inicialización es, por ejemplo, asegurada por una secuencia de puesta en marcha tal como se ilustra en la figura 2c.

15 La figura 2c esquematiza, para el ejemplo de dispositivo descrito en la figura 2b, el llenado progresivo de las memorias FIFO durante cinco etapas consecutivas ET1, ET2, ET3, ET4, ET5. Al final de la primera etapa ET1, la memoria FIFO de entrada $FIFO_e$ es llenada a su nivel óptimo (casi vacío). Al final de la segunda etapa ET2, la primera memoria FIFO intermedia $FIFO_{i1}$ es llenada a su nivel óptimo (50%). Al final de la tercera etapa ET3, la segunda memoria FIFO intermedia $FIFO_{i2}$ es llenada a su nivel óptimo (50%). Durante la cuarta etapa ET4, la memoria FIFO de salida $FIFO_s$ está en proceso de llenado. Al final de la cuarta etapa ETA, la memoria FIFO de salida $FIFO_s$ está llenada a su nivel óptimo (casi llena). ET5 representa el estado final de inicialización donde todas las memorias FIFO están llenas a sus niveles deseados.

20 Al final del procedimiento de inicialización que comprende las etapas de llenado sucesivas descritas en la figura 2c, todas las memorias FIFO del circuito (protegido por la invención) están llenas a su nivel óptimo permitiendo por tanto implementar el mecanismo de protección según la invención, conservando una continuidad de servicio.

25 De forma más precisa, el mecanismo de protección según la invención puede ser activado por cada uno de los bloques 201, 202, 203 lógicos funcionales, ya que sus memorias FIFO de salida respectivas son llenadas a su nivel de llenado óptimo.

30 Dicho de otra manera, para el bloque 201 lógico funcional alimentado por una memoria FIFO de entrada $FIFO_e$, la secuencia de inicialización se termina y el mecanismo de protección puede ser activado cuando su memoria FIFO de salida $FIFO_{i1}$ ha alcanzado su nivel de llenado óptimo. La actividad en la puesta en marcha del bloque 201 está condicionada por el llenado de la memoria FIFO de entrada $FIFO_e$ a su nivel de llenado óptimo.

Para el bloque 202 lógico funcional, alimentado por una memoria FIFO intermedia $FIFO_{i1}$, la secuencia de inicialización se termina y el mecanismo de protección se puede activar cuando su memoria FIFO de salida $FIFO_{i2}$ ha alcanzado su nivel de llenado óptimo. La actividad en la puesta en marcha del bloque de 202 está condicionada por el llenado de su memoria FIFO de entrada $FIFO_{i1}$ a su nivel de llenado óptimo.

35 En el caso de un único bloque protegido según la invención (ejemplo de la figura 2a), su actividad en la puesta en marcha está condicionada por el llenado de la memoria FIFO de entrada $FIFO_e$ a su nivel de llenado óptimo.

Durante esta fase de inicialización o de puesta en marcha, para cada bloque 201, 202, 203 lógico funcional protegido, la unidad 106 lógica de control controla la actividad del bloque por la señal de activación del bloque, teniendo en cuenta los niveles de llenado de la memoria FIFO aguas arriba y de la memoria FIFO aguas abajo.

40 En el régimen establecido, después de la secuencia de inicialización descrita en la figura 2c y fuera de las secuencias de corrección de error, cada bloque lógico funcional protegido según la invención gestiona el nivel de llenado de su memoria FIFO aguas arriba (a la salida del bloque) para mantener el nivel de llenado deseado. De forma más precisa, la unidad 106 lógica de control de cada bloque vigila y controla este nivel de llenado con la ayuda de la señal de activación que permite modificar el ritmo de actividad en el seno del bloque.

45 Para el caso de un bloque 201, 202 conectado a una memoria FIFO intermedia $FIFO_{i1}$, $FIFO_{i2}$ aguas abajo, el nivel de llenado deseado es de un 50%. Cuando el nivel de llenado de esta memoria sobrepasa un umbral alto $S_h = 50\% + \Delta$, donde Δ es un parámetro de la invención, la unidad 106 lógica de control suspende la actividad del bloque a través de la señal de activación. Después, cuando el nivel de llenado de esta memoria desciende por debajo de un umbral medio $S_m = 50\%$, la unidad 106 lógica de control vuelve a lanzar la actividad del bloque al ritmo R nominal.

50 Cuando el nivel de llenado de la memoria extiende por debajo de un umbral bajo $S_b = 50\% - \Delta$, la unidad 106 lógica de control acelera la actividad del bloque al ritmo R con la señal de activación. Después, cuando el nivel de llenado sobrepasa de nuevo el umbral medio $S_m = 50\%$, la unidad 106 lógica de control ralentiza la actividad del bloque para volver al ritmo R nominal.

5 En resumen, para gestionar dinámicamente el nivel de llenado de una memoria FIFO intermedia situada aguas abajo de un bloque protegido por la invención, la unidad 106 lógica de control del bloque controla la actividad de bloque por la señal de activación, teniendo en cuenta el nivel de llenado de la memoria FIFO aguas abajo con respecto a tres umbrales: $S_b = 50\% - \Delta$, $S_m = 50\%$ y $S_h = 50\% + \Delta$. Δ es un parámetro de la invención típicamente del orden del porcentaje de ocupación de la memoria correspondiente a una decena de muestras.

En el caso de un bloque 203 conectado a una memoria FIFO de salida $FIFO_s$ aguas abajo, el principio descrito anteriormente es similar pero los tres umbrales son definidos de la manera siguiente: $S_b = 100\% - 2\Delta$, $S_m = 100\% - \Delta$ y $S_h = 100\% - \delta$. δ es otro parámetro de la invención típicamente del orden de un porcentaje de ocupación de la memoria correspondiente a una o dos muestras.

10 Por tanto, el control del nivel de llenado de la memoria FIFO de salida permite regular este nivel de llenado a un nivel casi lleno, evitando la saturación de la memoria FIFO.

15 Para gestionar dinámicamente el nivel de llenado de la memoria $FIFO_s$ de salida, la unidad 106 lógica de control controla la actividad del último bloque 203 protegido, por la señal de activación del bloque, teniendo en cuenta el nivel de llenado de la memoria $FIFO_s$ aguas abajo con respecto a tres umbrales: $S_b = 100\% - 2\Delta$, $S_m = 100\% - \Delta$ y $S_h = 100\% - \delta$.

Se ilustran a continuación diferentes casos de ajuste dinámico de los niveles de llenado de las memorias FIFO para el ejemplo de la figura 2b.

20 Un primer ejemplo de aplicación se refiere a un escenario en el que se detecta un error en el bloque 202 funcional. Según la lógica implementada por la invención, tal como se describe en la figura 1, se efectúa una repetición del tratamiento del bloque 202 funcional lo que conlleva un retardo en el tratamiento de datos de entrada por el bloque 202. En consecuencia, la memoria FIFO de salida $FIFO_{i2}$ del bloque 202 está con un déficit de datos y presenta un nivel de llenado más reducido que el requerido por la misma. Por el contrario, la memoria FIFO de entrada $FIFO_{i1}$ del bloque 202 está en excedencia de datos y presenta un nivel de llenado más elevado que el requerido por la misma. Para regular dinámicamente estos niveles de llenado, la unidad 106 lógica de control del bloque 202 activa una señal de activación para acelerar el ritmo de tratamiento del bloque 202 con el fin de regularizar el retardo de tratamiento de datos de entrada. Esta aceleración tiene por efecto una disminución del nivel de llenado de la memoria FIFO de entrada $FIFO_{i1}$ y un aumento del nivel de llenado de la memoria FIFO de salida $FIFO_{i2}$. Esta ilustración de aplicación de la invención en este primer ejemplo muestra que el mecanismo de corrección por repetición con aceleración dinámica propuesto por la invención permite restaurar el estado de buen funcionamiento de la cadena de tratamiento, con, en particular, la restauración de los niveles de llenado esperados por las memorias FIFO.

35 Otro ejemplo de aplicación se refiere a un caso de déficit de datos en la memoria FIFO de salida $FIFO_s$ del conjunto de la cadena de tratamiento. Cuando la unidad lógica de control de bloque 203 detecta que el nivel de llenado de la memoria $FIFO_s$ es demasiado reducido, activa aceleración de la frecuencia de tratamiento del bloque 203. Esta aceleración tiene por consecuencia una disminución del nivel de llenado de la memoria FIFO de entrada $FIFO_{i2}$ que juega por tanto la función de memoria de salida para bloque 202 situado aguas arriba. Por tanto, por efecto de cascada, el bloque 202 después del bloque 201 acelera su frecuencia de tratamiento con el fin de regular el nivel de llenado de las memorias de salida. Estas aceleraciones en cadena se traducen finalmente en un déficit de datos en la memoria FIFO de entrada $FIFO_e$ de la cadena de tratamiento que es equivalente al déficit inicial de la memoria FIFO de salida $FIFO_s$.

Controlando dinámicamente la frecuencia de tratamiento de cada bloque 201, 202, 203, se puede de la misma manera reportar un excedente de datos en la memoria FIFO de salida $FIFO_s$ hacia un excedente de datos en la memoria FIFO de entrada $FIFO_e$ ralentizando el funcionamiento de cada bloque.

45 De forma similar, se puede reportar un excedente de datos en la memoria FIFO de entrada $FIFO_e$ hacia un excedente de datos en la memoria FIFO de salida $FIFO_s$ acelerando el funcionamiento de cada bloque.

De forma similar, se puede reportar un déficit de datos en la memoria FIFO de entrada $FIFO_e$ hacia un déficit de datos en la memoria FIFO de salida $FIFO_s$ ralentizando el funcionamiento de cada bloque.

En la práctica, las memorias FIFO de entrada y de salida de la cadena de tratamiento no se supone que están en déficit o en excedencia de datos, fuera del régimen transitorio correspondiente a la puesta en marcha de la cadena.

50 En general, la función aguas abajo se interconecta con la memoria FIFO, cuya lectura está condicionada a la disponibilidad de datos. Del mismo modo, la cadena no se pone en marcha más que cuando el nivel de llenado de la memoria FIFO de entrada es satisfactorio. De hecho, durante el funcionamiento nominal de la cadena, todas las memorias FIFO tienen un nivel de llenado normal. Por tanto, los únicos casos de memorias FIFO en déficit o en excedencia de datos se corresponden a las fases de corrección por reducción que siguen a una detección de error.

Las figuras 3a y 3b ilustran respectivamente en dos diagramas la evolución del nivel de llenado de una memoria 104 FIFO de salida (figura 3a) y de una memoria 103 FIFO de entrada (figura 3b) en el transcurso del tiempo que sigue a una detección de error.

5 En el instante t_0 se detecta un error transitorio en la salida del bloque 101 lógico funcional. A continuación de la detección de este error, durante una primera fase ϕ_1 , el tratamiento es interrumpido en este bloque, el acceso de lectura de la memoria 103 FIFO de entrada es suspendido, y el acceso de escritura de la memoria 104 FIFO de salida es suspendido. La memoria 103 FIFO de entrada continúa por tanto de llenarse mientras que la memoria 104 FIFO de salida continúa de vaciarse a un ritmo R , como se indica en las figuras 3a y 3b. La unidad 102 lógica de detección de errores que incluye el bloque 101 lógico funcional se reinicializa, el estado anterior al error detectado se restaura indirectamente a partir del tratamiento acelerado al ritmo R_a de los datos L almacenados en la memoria 105 de buffer.

15 La primera fase ϕ_1 se termina en un instante t_1 cuando el tratamiento del bloque 101 funcional se retoma a partir del resultado que había sido detectado en el error, y con un ritmo $R_a > R$ acelerado. En una segunda fase ϕ_2 , el bloque 101 funcional es por tanto autorizado de nuevo a leer los datos en la memoria 103 FIFO de entrada. Los resultados del tratamiento ejecutado por el bloque 101 funcional son escritos al ritmo R_a en la memoria 104 FIFO de salida más rápidamente que son leídos por el bloque situado aguas abajo, por lo que la memoria 104 FIFO de salida se llena de nuevo como se ilustra en la figura 3a.

20 La segunda fase ϕ_2 se termina en un instante t_2 cuando el nivel de llenado de la memoria 104 FIFO de salida ha alcanzado un umbral predeterminado considerado óptimo. En una tercera fase ϕ_3 , el bloque 101 lógico funcional reanuda ahora un ritmo R de tratamiento nominal. Las capacidades de protección son por tanto restauradas.

Si se denomina N al número de datos presentados en la memoria 104 FIFO de salida en el instante t_0 y que se corresponden al nivel de llenado óptimo, el tiempo necesario para vaciar la memoria 104 FIFO de salida de sus N datos durante la primera fase ϕ_1 es igual a $T_{\text{drenage}} = N/F$.

25 El tiempo necesario para ralentizar la unidad 102 lógica de detección de errores que incluye el bloque 101 funcional lógico, para restaurar su estado anterior tratando L muestras anteriores al error, para reanudar el tratamiento en el resultado que estaba en error, y para reanudar la escritura de los resultados en la memoria 104 FIFO de salida durante la segunda fase ϕ_2 es igual a $T_{\text{reanudación}} = (L+\epsilon)/F_a$, con L la latencia de bloque 101 expresada en ciclos, ϵ un margen igual a algunos ciclos, típicamente inferior a 10 ciclos, para tomar en cuenta los asincronismos eventuales y la sincronización $F_a = \gamma * F$ y $\gamma > 1$, siendo F expresada en hercios.

30 La condición que garantiza una reanudación transparente para los bloques aguas abajo del bloque 120, dicho de otra manera, que garantiza que la detección de errores es imperceptible a la salida de la cadena de tratamiento es: $T_{\text{reanudación}} < T_{\text{drenage}}$. Si esta condición es respetada, la memoria 104 FIFO de salida no está jamás completamente vacía. Esta condición equivale por tanto a la condición siguiente:

$$N > (L + \epsilon) / \gamma.$$

35 El razonamiento anterior para la memoria 104 FIFO de salida es válido para todas las memorias FIFO_i, que sirven a la vez en la entrada y en la salida. Para el bloque 101 considerado, la memoria 103 FIFO de entrada es normalmente utilizada con N datos disponibles. Durante la corrección por repetición en el bloque 101 considerado, la memoria 103 FIFO de entrada es llenada con N nuevos datos, sin ser leída durante la fase ϕ_1 . Por tanto, cada memoria FIFO debe por consiguiente poder almacenar $2*(N + \epsilon)$ datos. En la práctica, se elige un tamaño de memoria FIFO superior a $2*(L+\epsilon)$ para cada memoria FIFO.

En el caso de una memoria 103 FIFO de entrada que no sirva de memoria FIFO de salida para un bloque protegido según la invención, la necesidad de la capacidad de la memoria FIFO se puede reducir a $(L+\epsilon)$ datos, ya que el nuevo llenado óptimo para la memoria FIFO de entrada está casi vacío.

45 El tiempo $T_{\text{restauración}}$ corresponde a la duración durante la cual se trata un error y se restauran las capacidades de corrección, es decir los niveles de llenado de las memorias FIFO. Es igual a $T_{\text{restauración}} = T_{\text{reanudación}} + T_{\text{llenado FIFO}}$.

$T_{\text{llenado FIFO}}$ designa el tiempo necesario para llenar la memoria FIFO de entrada (o la memoria FIFO de salida) a su nivel de llenado óptimo.

Durante la fase ϕ_2 , la memoria 104 FIFO de salida es leída a un ritmo R por un bloque aguas abajo, y se llena a un ritmo R_a acelerado por el bloque 101 actual. Globalmente, su nivel de llenado aumenta al ritmo $R(\gamma-1)$.

50 Por tanto, se deduce que $T_{\text{llenado FIFO}} = N/(F(\gamma-1))$ y por tanto

$$T_{\text{restauración}} = (L + \epsilon) / F_a + N / (F * (\gamma - 1)) = (L + \epsilon) / (\gamma * F) + N / (F * (\gamma - 1)).$$

El razonamiento desarrollado anteriormente para el dimensionamiento de las memorias FIFO y el tiempo de restauración de las capacidades de corrección se corresponde al tratamiento para un error en un bloque de la cadena de tratamiento. Se puede entender este razonamiento en el caso de una sucesión de errores en el seno de la función 101 protegida (sea a nivel de la cadena global, sea a nivel del bloque de tratamiento) durante la fase ϕ_1 en el transcurso de la repetición. La necesidad del dimensionamiento de las memorias FIFO es diferente en este caso.

Durante la fase ϕ_1 de una repetición, los datos de entrada para el bloque 101, anteriores a la manifestación del error están siempre disponibles e integrados en la memoria 105 de buffer, aguas abajo de la memoria 103 FIFO de entrada. Durante esta fase ϕ_1 , la memoria 103 FIFO de entrada no es ya leída, sólo los datos archivados en la memoria 105 de buffer son utilizados para restaurar el estado anterior de la función del bloque 101 de tratamiento. La ocurrencia de un segundo error durante esta fase conduce a volver a lanzar esta fase ϕ_1 (de inicialización/restauración del estado anterior) por tanto, la duración de esta fase ϕ_1 se duplicará como mucho. En el caso general, la duración de la fase ϕ_1 es inferior o igual a $p \cdot T_{\text{reanudación}}$ en el caso de p errores separados consecutivamente de menos de $T_{\text{reanudación}}$. Hay igualdad cuando el nuevo error se produce al final de la fase ϕ_1 . En este caso, cualquier nueva reanudación es realizada a partir de los mismos datos de entrada, archivados en la memoria 105 de buffer.

Pero los nuevos datos de entrada continúan llegando al ritmo R y llenan la memoria 103 FIFO de entrada cuya lectura por el bloque 101 ha sido suspendida durante una duración al menos igual a $p \cdot T_{\text{reanudación}}$. La memoria 103 FIFO de entrada debe por tanto disponer de $p \cdot (L + \epsilon)$ sitios libres en el régimen establecido. Además, la fase aguas abajo espera nuevos datos de salida. La memoria 104 FIFO de salida por tanto dispone de $p \cdot (L + \epsilon)$ datos en el régimen establecido. Por consiguiente, para tolerar p errores consecutivos separados cada uno del anterior por un intervalo de tiempo de duración inferior a $T_{\text{reanudación}}$ (nuevos errores en fase ϕ_1), las memorias FIFO_e de entrada y de salida FIFO_s deben dimensionarse para almacenar $p \cdot (L + \epsilon)$ datos. Las memorias FIFO_i intermedias que toman a la vez la función de memoria de entrada y de memoria de salida entre dos bloques protegidos deben estar dimensionadas para almacenar $2p \cdot (L + \epsilon)$ datos.

Si otro error aparece durante la fase ϕ_2 de repetición de bloque 101, los niveles de llenado de las memorias FIFO de entrada y de salida no son todavía completamente restaurados. Un nuevo error al principio de la fase ϕ_2 conduce a volver a lanzar la secuencia completa de corrección (fase ϕ_1 y después fase ϕ_2), pero con los márgenes reducidos de N datos en los niveles de llenado de las memorias FIFO de entrada y de salida.

Por tanto, para tolerar p errores consecutivos separados cada uno del anterior de un intervalo de tiempo de duración superior a $T_{\text{reanudación}}$ e inferiores a $T_{\text{restauración}}$ (nuevos errores en fase ϕ_2), las memorias FIFO deben también ser dimensionadas para almacenar $2p \cdot (L + \epsilon)$ datos.

Integrando el margen ϵ en una estimada L_p de la latencia de la función 101, el tamaño mínimo de las memorias FIFO es igual a $p \cdot L_p$ o $2p \cdot L_p$ dependiendo de si la memoria FIFO tiene sólo una función de memoria de entrada o de memoria de salida o dependiendo de si la memoria FIFO acumula las dos funciones.

La descripción de la invención que precede se refiere a bloques 101 lógicos funcionales que tengan una latencia constante. Esto corresponde a circuitos organizados en una cascada de fases, cuyas señales intermedias y de salida son siempre propagadas a una fase aguas abajo, sin retornar en bucle hacia aguas arriba. Estos circuitos tienen una latencia constante. Los resultados de la salida sólo dependen de los N últimos datos de entrada. Un ejemplo típico de circuitos de latencia constante es dado por la familia de filtros RIF (FIR en inglés). En la figura 4a, se ha representado esquemáticamente dicho circuito en el cual los rectángulos 401, 402, 403 representan registros y las nubes 404, 405 representan la lógica combinatoria.

En los circuitos que realizan funciones con bucles de reacción (o *feedback loops* en inglés) que corresponden a funciones recursivas, ciertas señales intermedias o de salida son reutilizadas aguas arriba, formando por tanto los bucles denominados de reacción. En este caso de figura, el estado del circuito depende generalmente del histórico de las señales de entrada en concurrencia con una profundidad infinita. Un ejemplo típico de circuito numérico recursivo es dado por la familia de filtros RII (IIR en inglés). En la figura 4b, se representa esquemáticamente un circuito que comprende tratamientos 410, 411 con bucles de reacción.

La figura 5 representa un esquema de una variante de realización de la invención que se refiere a un circuito integrado digital que comprende un bloque 501 lógico funcional que realiza una función recursiva con un bucle de reacción interno. Los elementos que portan las mismas referencias numéricas en las figuras 1 a 5 son idénticos y son gestionados de la misma manera que en el caso correspondiente a la protección de funciones de latencia constante.

Para adaptar la invención a la protección de funciones de bucles de reacción, el dispositivo de corrección por repetición puede ser adaptado con el fin de poder restaurar el estado interno de la función, a partir de los datos de entrada anteriores y los valores archivados de diferentes señales que hacen bucle hacia aguas arriba en el seno del bloque 501 funcional a proteger.

Para poder proteger el bloque 501 lógico funcional que comprende uno o varios bucle(s) de reacción interno(s), este bloque 501 es transformado con el fin de implementar el o los bucle(s) de reacción en externo, es decir en el exterior del bloque 501, y de hacer que estos bucle(s) externo(s) se puedan abrir. La apertura de un bucle externo se realiza por medio de un multiplexor 503 que selecciona los datos de la segunda memoria 502 de buffer.

5 Por tanto, el circuito 500 según esta valiente de realización en la invención comprende, además de los elementos ya descritos en la figura 1, para cada señal de bucle de reacción, una segunda memoria 502 de buffer asociada a un segundo multiplexor 503 controlado por la unidad 106 lógica de control. El segundo multiplexor 503 está controlado para seleccionar alternativamente o bien los datos aguas arriba del bucle de reacción interno del bloque 501 lógico funcional, o bien los datos archivados en la segunda memoria 502 de buffer que se corresponden a los datos aguas
10 abajo del bucle de reacción interno del bloque 501 lógico funcional en un instante anterior a la aparición de un error. El multiplexor 503 está conectado a una segunda entrada de bloque 501 para introducir los datos para retornar en bucle aguas arriba en el bloque 501 lógico funcional. Con referencia la figura 5, el multiplexor 503 selecciona la salida de la segunda memoria 502 de buffer durante la primera fase ϕ_1 y la salida específica del bloque 501 durante la fase ϕ_2 o la fase ϕ_3 . El multiplexor 503 puede ser reemplazado por cualquier otro medio equivalente que permita a
15 la unidad 106 lógica de control seleccionar los datos a retornar en bucle, resultantes o bien de la segunda memoria 502 de buffer, o bien de la fuente aguas abajo del bucle de reacción interno del bloque 501 lógico funcional.

El bloque 501 lógico funcional transformado dispone de una salida y de una entrada específica para, respectivamente, extraer los datos de la función que se va a retornar en bucle y para introducir aguas arriba los datos extraídos a retornar en bucle. Esto permite implantar el bucle de reacción en el interior del circuito 501, con el
20 fin de archivar la secuencia de la señal digital del bucle de reacción y para poder introducir una secuencia anterior de la señal del bucle de reacción.

Este principio se ilustra mejor en las figuras 6a, 6b, 6c que esquematizan, en tres esquemas, la transformación de un bloque 601 lógico funcional que comprende un bucle de reacción interno en un bloque 603 lógico funcional para el cual el bucle de reacción se hace externo y se puede abrir.

25 La figura 6a representa un bloque 601 lógico funcional que comprende un bucle 610 de reacción interno que tiene por función volver a introducir los datos en la salida del registro 611 hacia la entrada de la lógica 612 combinatoria. El bloque 601 lógico funcional comprende una entrada E_1 y una salida S_1 .

La figura 6b representa una primera transformación del bloque 601 hacia un bloque 602 transformado en el cual el bucle 610 de reacción se hace externo 620. Para esto, el bloque 602 está provisto de una segunda entrada E_2 y de una segunda salida S_2 . La segunda salida S_2 recibe los datos en la salida del registro 611. La segunda entrada E_2 permite introducir los datos del bucle 620 de reacción externo hacia la entrada de la lógica 612 combinatoria.
30

La figura 6c representa una segunda transformación del bloque 602 hacia un bloque 603 transformado conforme a la invención. Este bloque 603 es modificado ya que comprende un multiplexor 630 para conectar la segunda entrada E_2 o bien al bucle 620 de reacción, en cuyo caso el funcionamiento del bloque 603 es en bucle cerrado, o bien a otra fuente 631, en cuyo caso el funcionamiento del bloque 603 es en bucle abierto.
35

El principio descrito en la figura 6c se utiliza para proteger el bloque funcional lógico según la invención.

Volviendo a la figura 5, que se refiere a los recursos de protección específicos para las funciones con el bucle de reacción, la segunda memoria 502 de buffer archiva la secuencia de N valores anteriores de datos extraídos del bucle de reacción. El acceso de escritura y de lectura a esta segunda memoria 502 de buffer es gestionado de la
40 misma manera que la primera memoria 105 de buffer en la entrada del bloque 501 funcional protegido. La profundidad N se corresponde aquí al número de tapas de registros en cascada en la función realizada por el bloque 501 haciendo la abstracción de los bucles (considerados abiertos). El tamaño de esta memoria 502 de buffer puede ser reducido según la posición del punto de introducción aguas arriba de la señal del bucle de reacción con respecto a la salida de la función. El tamaño de la memoria 502 de buffer es como mucho igual al número de fases de
45 registros, cuando la salida se retorna en bucle en la fase de entrada. Aunque en la figura 5, se ha representado una segunda memoria 502 de buffer y un segundo multiplexor 503, hace falta en general tantos conjuntos de memorias 502 de buffer y de multiplexores 503 como de bucles de reacción. El dimensionamiento de estas memorias de buffer para archivar los datos resultantes del bucle de reacción soporta el caso de errores simples y en ráfaga. Por tanto, la adaptación del circuito propuesto a la figura 5 permite tener en cuenta las funciones del bucle de reacción.

50 El dimensionamiento de las memorias FIFO en la entrada, en la salida e intermedias se realiza de la misma manera que para las funciones de latencia constante, considerando que los bucles de reacción externos de la función están abiertos. Conceptualmente, la apertura de los bucles permite volver al caso de una función de latencia constante. En la práctica, el hecho de implantar los bucles de reacción en el exterior del bloque 501 de lógica funcional, a través de una entrada y una salida específica para la fuente y el punto de introducción de la señal del bucle de reacción,
55 vuelven a transformar la función con bucle de reacción en una función sin bucle con latencia constante, pero con varias salidas y varias entradas.

La invención presenta numerosas ventajas en comparación a las soluciones de la técnica anterior. Permite realizar una corrección de los errores transitorios que afectan a una o varias funciones implementadas en un circuito integrado digital, de manera transparente para las otras funciones situadas aguas arriba o aguas abajo. Dicho de otra manera, se asegura una continuidad de servicio, sin afectar al ritmo de ejecución de los tratamientos y la secuenciación de las muestras en el seno del circuito integrado digital. El mecanismo de protección a los errores aportado por la invención no perturba significativamente el funcionamiento nominal de un circuito integrado, en particular, no ralentiza significativamente su frecuencia de ejecución ni degrada sus rendimientos de tratamiento. La invención permite proteger todos los recursos lógicos de un circuito integrado digital, así como tanto la lógica secuencial como la lógica combinatoria. No está sometido a limitaciones en comparación con la tecnología utilizada para realizar el circuito integrado. Presenta una complejidad lógica y una disipación reducida en comparación a soluciones usuales, en particular las basadas en una triplicación modular y un voto mayoritario (TMR). El mecanismo propuesto permite corregir los errores lo antes posible y no se contenta con enmascararlos. Incluso si un error transitorio afecta globalmente a la unidad 102 lógica de detección de errores, la unidad 102 lógica de detección de errores señala eventualmente un falso error (falso positivo), que conlleva una repetición inútil del bloque 101 lógico funcional pero que no tiene consecuencia. El dimensionamiento de las memorias FIFO de entrada y de salida del dispositivo según la invención permite adaptar la capacidad de corrección del dispositivo en función de un escenario en el peor caso contemplado. Por ejemplo, si se anticipa una ráfaga de p errores repartidos en un corto periodo de tiempo, el dimensionamiento de las memorias FIFO se puede efectuar a partir del parámetro p y de la latencia del bloque lógico funcional a proteger. La invención permite proteger un bloque 101 lógico funcional contra errores transitorios. Por otro lado, las memorias FIFO 103 de entrada y 104 de salida, así como las memorias 105 de buffer y la unidad 106 lógica de control pueden, por otro lado, ser igualmente protegidas por medios de protección conocidos, por ejemplo utilizando códigos correctores de errores o un método de triplicación modular y voto mayoritario TMR.

Aunque la invención ha sido descrita en el ámbito de los tratamientos de carencia única, es decir para los cuales el ritmo en la interfaz de entrada del bloque 120 y el ritmo en la interfaz de salida son idénticos, se aplica de manera idéntica a tratamientos de cadencia múltiple, es decir para los cuales los ritmos de interconexión de entrada y de salida son diferentes.

REIVINDICACIONES

1. Circuito integrado digital que comprende un conjunto (120, 520) lógico que comprende un bloque (101, 501) lógico funcional, una unidad (102, 512) lógica de detección de errores transitorios que afectan al bloque (101, 501) lógico funcional, una memoria (103) FIFO de entrada para alimentar de muestras el bloque (101, 501) lógico funcional, una memoria (104) FIFO de salida para recibir muestras en la salida del bloque (101, 501) lógico funcional, una memoria (105) de buffer alimentada de muestras por la memoria (103) FIFO de entrada, y una unidad (106) lógica de control apta para controlar, el acceso de lectura en la memoria (103) FIFO de entrada y el acceso de escritura en la memoria (104) FIFO de salida y configurada para, cuando se detecta un error por la unidad (102, 512) lógica de detección de errores transitorios, reinicializar la unidad (102, 512) lógica de detección de errores transitorios y el bloque (101, 501) lógico funcional, suspender el acceso de escritura en la memoria (104) FIFO de salida y conmutar la entrada del bloque (101, 501) lógico funcional hacia la salida de la memoria (105) de buffer.
2. Circuito integrado digital según la reivindicación 1 que comprende un multiplexor (107) para conectar la entrada del bloque (101, 501) lógico funcional a la memoria (103) FIFO de entrada o a la memoria (105) de buffer.
3. Circuito integrado digital según la reivindicación 2, en el cual la unidad (106) lógica de control está configurada para controlar el multiplexor (107) en función del resultado de la unidad (102, 512) lógica de detección de errores.
4. Circuito integrado digital según una de las reivindicaciones anteriores en el cual la frecuencia de reloj en el seno del conjunto (120, 520) lógico es superior a la frecuencia media de escritura en la memoria (103) FIFO de entrada.
5. Circuito integrado digital según la reivindicación 4 en el cual la actividad en el seno del conjunto (120, 520) lógico se hace irregular.
6. Circuito integrado digital según una de las reivindicaciones 4 o 5, en el cual el ritmo Ra de los tratamientos en el seno del conjunto (120, 520) lógico es controlado por la unidad (106) lógica de control por medio de una señal de activación.
7. Circuito integrado digital según una de las reivindicaciones 4 a 6, en el cual el ritmo Ra de los tratamientos en el seno del conjunto (120, 520) lógico es controlado por la unidad (106) de control en función del nivel de llenado de la memoria (104) FIFO de salida o de una detección de error señalada por la unidad (102, 512) lógica de detección de errores transitorios.
8. Circuito integrado digital según la reivindicación 7 en el cual la unidad (106) lógica de control está configurada para modificar el ritmo Ra de los tratamientos en el seno del conjunto (120, 520) lógico en función de una comparación del nivel de llenado de la memoria (104) FIFO de salida con al menos tres umbrales.
9. Circuito integrado digital según una de las reivindicaciones anteriores en el cual la memoria (103) FIFO de entrada y la memoria (104) FIFO de salida están dimensionadas para almacenar al menos $p \cdot L_p$ datos, con p el número máximo de errores consecutivos tolerables que afectan en ráfaga al bloque (101, 501) lógico funcional y L_p una estimación de la latencia del bloque lógico funcional.
10. Circuito integrado digital según una de las reivindicaciones anteriores, en el cual el tamaño de la memoria (105) de buffer depende de la latencia del bloque (101, 501) lógico funcional.
11. Circuito integrado digital según una de las reivindicaciones anteriores, en el cual la unidad (102) lógica de detección de errores transitorios comprende un bloque (110) lógico funcional redundante y una unidad (111) de comparación de resultados producidos respectivamente por el bloque (101, 501) lógico funcional y el bloque (110) lógico funcional redundante.
12. Circuito integrado digital según una de las reivindicaciones anteriores en el cual el bloque (101, 501) lógico funcional tiene una latencia constante.
13. Circuito integrado digital según una de las reivindicaciones 1 a 11, en el cual el bloque (501, 603) lógico funcional está provisto de al menos una segunda entrada (E_2) y de al menos una segunda salida (S_2) para, cuando están conectadas entre ellas, realizar al menos un bucle (620) de reacción externo, dicho conjunto (520) lógico comprende al menos una segunda memoria (502) de buffer alimentada de muestras por dicha al menos una segunda salida (S_2) del bloque (501, 603) lógico funcional, estando configurada la unidad (106) lógica de control para, cuando se detecta un error por la unidad (512) lógica de detección de errores transitorios, conmutar dicha al menos una segunda entrada (E_2) del bloque (501, 603) lógico funcional de dicha al menos una segunda salida (S_2) de dicho bloque (501, 603) lógico funcional hacia la salida de dicha al menos una segunda memoria (502) de buffer.
14. Circuito integrado digital según la reivindicación 13, en el cual dicho conjunto (520) lógico comprende al menos un segundo multiplexor (503) para conectar dicha al menos una segunda entrada (E_2) del bloque (501, 603) lógico

funcional a dicha al menos una segunda salida (S_2) del bloque (501, 603) lógico funcional o a dicha al menos una segunda memoria (502) de buffer.

5 15. Circuito integrado digital según la reivindicación 14 en el cual la unidad (106) lógica de control está configurada para controlar el segundo multiplexor (503) de cada conjunto en función del resultado de la unidad (512) lógica de detección de errores.

16. Circuito integrado digital según una de las reivindicaciones anteriores que comprende una pluralidad de dichos conjuntos (201, 202, 203) lógicos y en el cual una memoria FIFO de entrada ($FIFO_{i1}$) de un primer conjunto (202) lógico es una memoria FIFO de salida para un segundo conjunto (201) lógico situado aguas arriba.

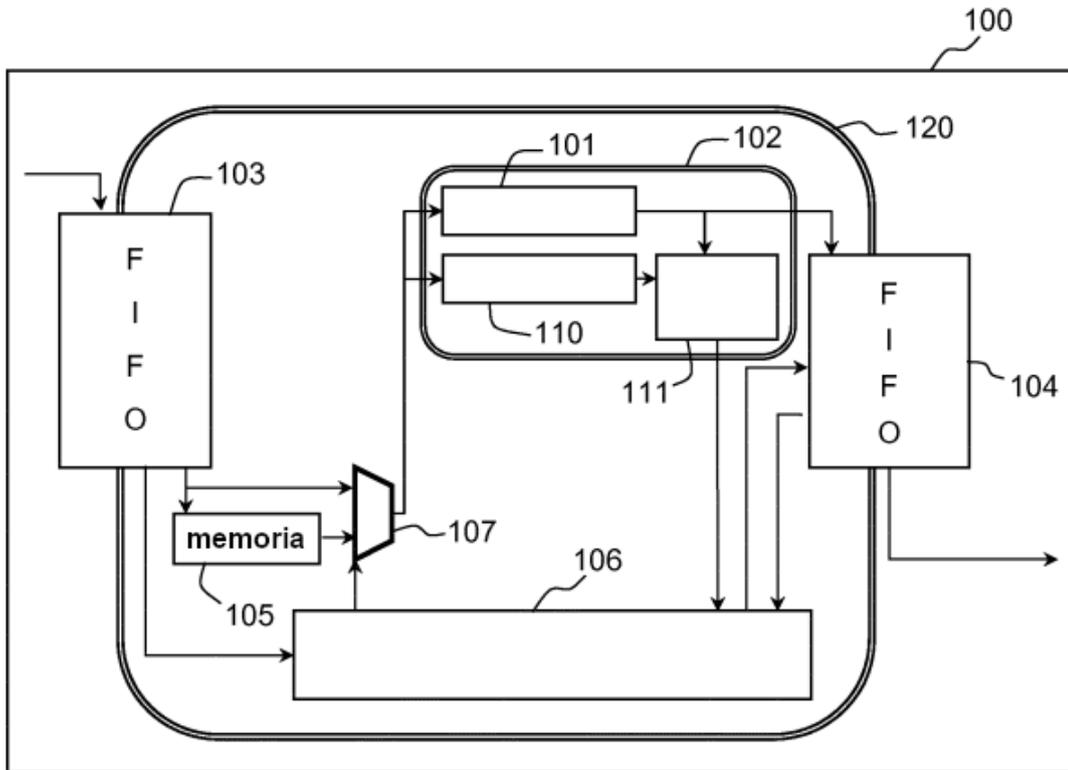


FIG.1

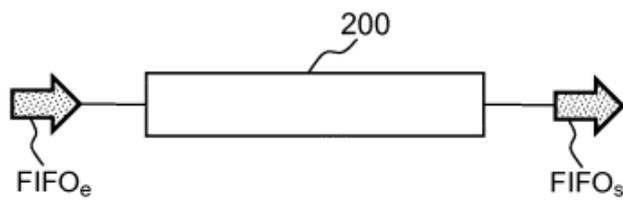


FIG.2a

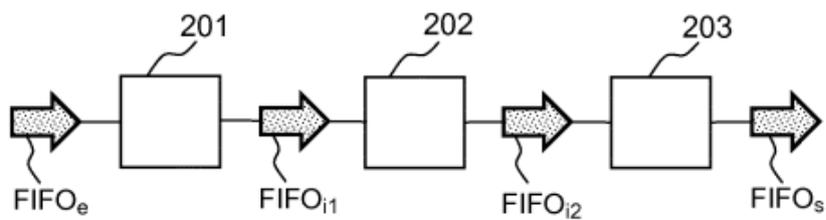


FIG.2b

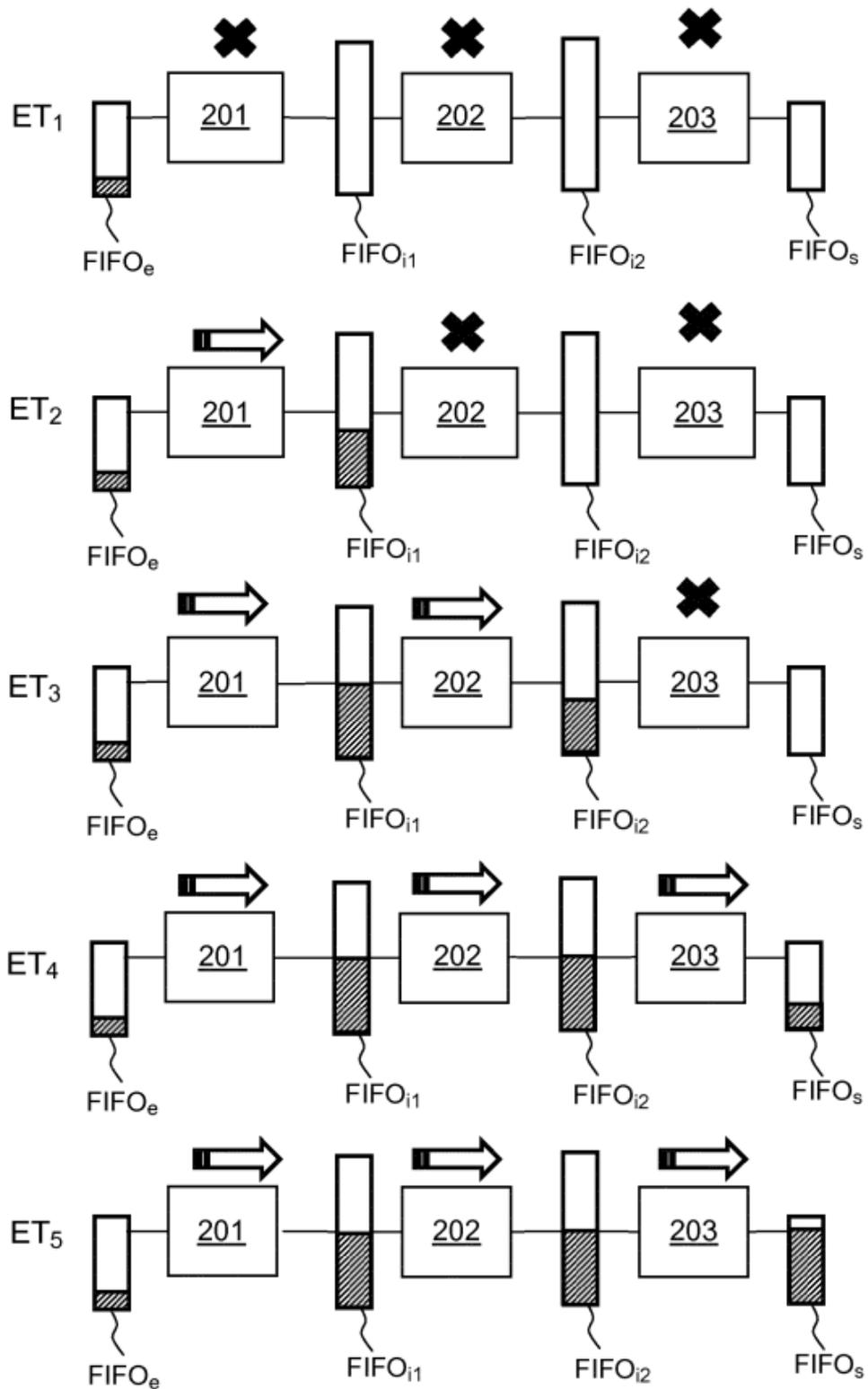


FIG.2c

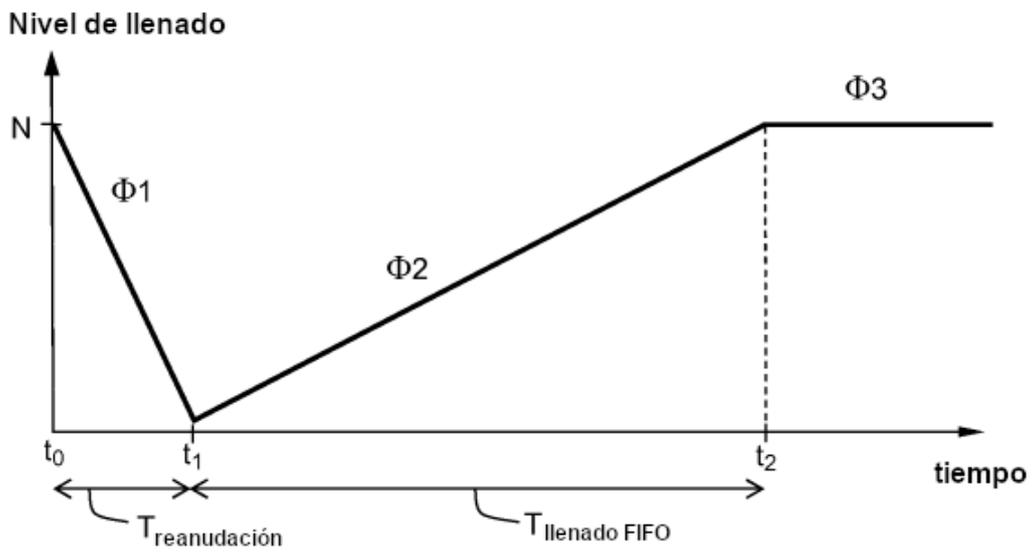


FIG.3a

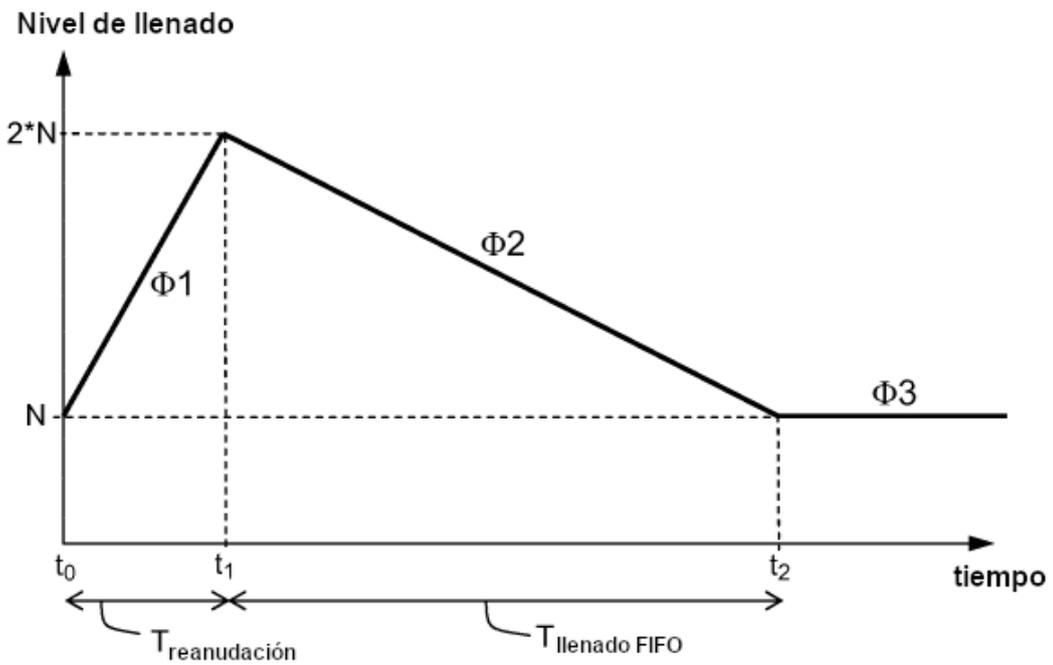


FIG.3b

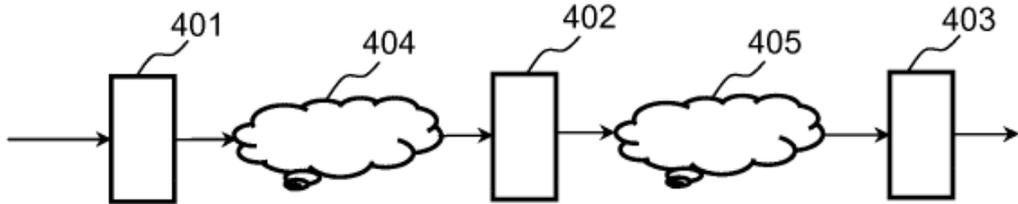


FIG.4a

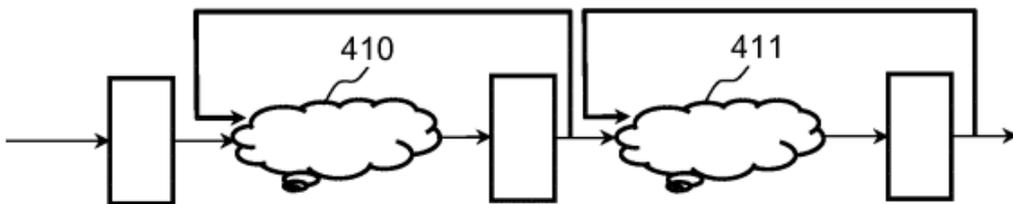


FIG.4b

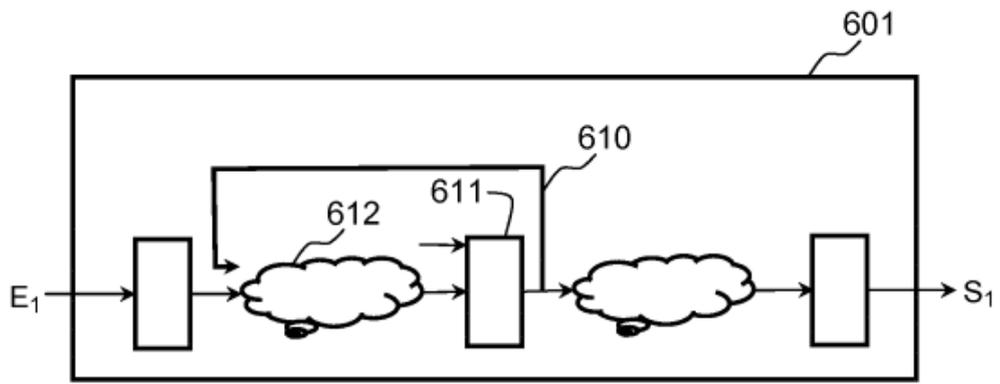


FIG. 6a

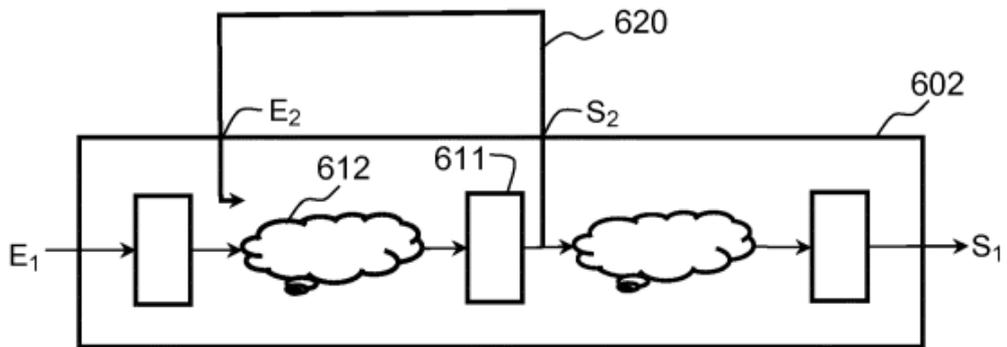


FIG. 6b

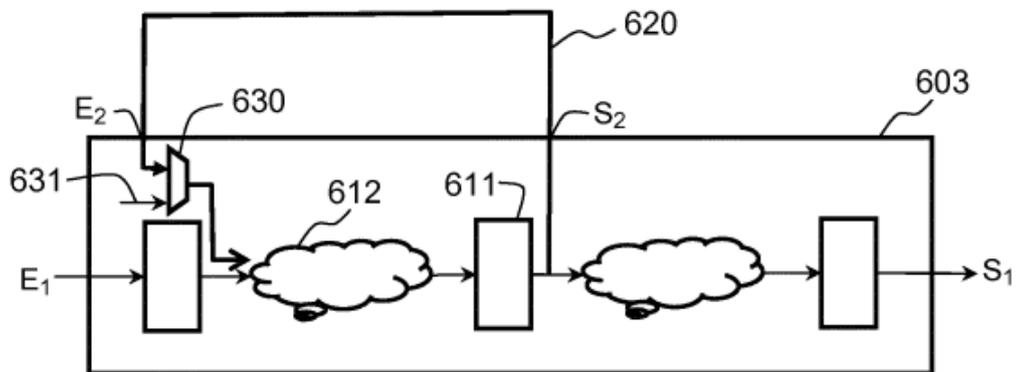


FIG. 6c