(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 309 687**

(51) Int. Cl.:
***G06F 11/16*** (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **05291923 .0**
(96) Fecha de presentación : **16.09.2005**
(97) Número de publicación de la solicitud: **1764694**
(97) Fecha de publicación de la solicitud: **21.03.2007**

(54) Título: **Procedimiento y sistema de control redundante para ordenadores seguros.**

(45) Fecha de publicación de la mención BOPI:
**16.12.2008**

(45) Fecha de la publicación del folleto de la patente:
**16.12.2008**

(73) Titular/es: **Siemens Transportation Systems S.A.S.**
**48-56, rue Barbès**
**92542 Montrouge, FR**

(72) Inventor/es: **Fumery, Benoit;**
**Essame, Didier y**
**Forin, Philippe**

(74) Agente: **Elzaburu Márquez, Alberto**

## DESCRIPCIÓN

Procedimiento y sistema de control redundante para ordenadores seguros.

La invención se refiere a los sistemas de tratamiento que necesitan una gran seguridad de funcionamiento, en especial los sistemas de tratamiento utilizados en el transporte de personas. Más particularmente, la invención se refiere a un sistema de tratamiento que dispone de dos ordenadores, *a priori* intrínsecamente seguros, para garantizar un nivel de seguridad suficiente asegurando al mismo tiempo, en particular mediante una puesta en redundancia, una buena disponibilidad del sistema en caso de fallo de uno de los ordenadores.

Dos ejemplos de tal sistema, fuera del dominio de los transportes, se describen en US4823256 y en US2005/0149795 donde un procedimiento de control del sistema que consta de dos ordenadores redundantes que disponen cada uno de salidas seguras que pueden adoptar un estado restrictivo o un estado permisivo. D1= US-A-4 823 256 divulga un procedimiento de control de un sistema que consta al menos de dos ordenadores dispuestos cada uno de salidas seguras que pueden adoptar un estado restrictivo o un estado permisivo. Dichos ordenadores pueden igualmente intercambiar información entre ellos y cada ordenador dispone de un mismo programa de cálculo para definir los estados de salida, un ordenador es el master y el otro el slave y funcionan según un modo redundante, denominado "modo divergente", este modo comprende también las etapas de recepción, comparación, detección y prohibición de la transición de un estado de mando restrictivo hacia un estado permisivo en caso de anomalía de uno de los ordenadores. En contraste con la invención, al menos uno de los ordenadores o uno de sus programas idénticos de cálculo puede ser defectuoso, es decir no intrínsecamente seguro. Los ordenadores pueden intercambiar información entre ellos, cada ordenador dispone del mismo programa de cálculo para definir los estados de salida, un ordenador es el master y el otro el slave y funcionan según un modo redundante que corresponde a un modo divergente que comprende las etapas de recepción, comparación, detección y prohibición de la transición de un estado de mando restrictivo hacia un estado permisivo en caso de anomalía de uno de los ordenadores.

Las aplicaciones relativas al tratamiento de control/mando en condiciones de seguridad son comúnmente utilizadas en el transporte automatizado, como por ejemplo, los trenes urbanos automáticos. Estas aplicaciones utilizan ordenadores intrínsecamente seguros que son capaces de garantizar un funcionamiento seguro y capaces de detectar un error de funcionamiento en el tratamiento efectuado. Un ejemplo de ordenador se describe en EP-A-1 089 175. El ordenador aplica una técnica que consiste en detectar errores por medio de un sistema de codificación de la información y, si llega el caso, pone los mandos en una posición segura para los pasajeros, pudiendo ir justo hasta la parada del tren.

La parada de emergencia de un tren no es buena ni para los pasajeros ni para la rentabilidad económica de los trenes. Por esta razón, el sistema de mano debe duplicarse para poder tolerar las faltas: los ordenadores se organizan en pares, para que uno pueda reemplazar al otro en caso de fallo. Sin embargo, esta duplicación puede ser fuente de numerosos problemas de seguridad. Se pueden imaginar, en efecto, numerosas situaciones en las que un ordenador de socorro que no tenga una visión coherente del entorno podría comprometer la seguridad del mando si viniera a tomar el control.

Para evitar la incompatibilidad de dos ordenadores, es conocido el asegurarse de que reciben y procesan exactamente las mismas informaciones, y en el mismo orden. Sin embargo, a pesar de la seguridad de que tienen las mismas entradas, no se excluye que los dos ordenadores puedan producir resultados diferentes, que pueden ser seguros tomados individualmente, pero que puede implicar un fallo más importante si persisten. Igualmente, una divergencia de resultados puede muy bien ser sólo temporal sin que esta pausa produzca un problema serio y sin que sea necesario tomar medidas de seguridad que pueden entrañar un perjuicio al tráfico.

Un sistema de tratamiento redundante se revela en el artículo de D. Essamé, J. Arlat and D. Powell, "PADRE: A Protocol for Asymmetric Duplex Redundancy", publicado por Dependable Computing for Critical Applications (DCCA-7), C. B. Welnstock and J. Rushby, Eds., y en Dependable Computing and Fault-Tolerant Systems 12, pp. 229-248, IEEE Computer Society Press, 1999 (Proc. IFIP 7th Working Conf. held in San Jose, CA, USA, January 1999). Este artículo describe, desde un punto de vista teórico, un sistema asincrónico de gestión de redundancias dobles a base de unidad de tratamientos intrínsecamente seguros. En particular, este artículo introduce el principio de la detección de posibles incoherencias de contexto entre las unidades de tratamiento redundantes. Sin embargo, el asincronismo de este mecanismo de gestión y la falta de información precisa sobre la manera de detectar las incoherencias de contexto hacen que el mecanismo de este artículo no sea totalmente aplicable a un sistema de control/mando que suministra salidas todo o nada seguras que necesitan una fuerte reactividad.

La invención tal como lo reivindica la reivindicación 1, propone por lo tanto realizar un sistema redundante seguro que utiliza dos ordenadores intrínsecamente seguros dispuestos cada uno de salidas todo o nada seguras y que aplican un proceso de gestión de la redundancia sincrónica. En particular, la invención proporciona un mecanismo adaptado para detectar incoherencias de contexto sobre salidas todo o nada seguras y en particular, directamente aplicable en los sistemas ferroviarios. El procedimiento permitirá así gestionar las incoherencias sobre las salidas de los dos ordenadores cuando estos funcionan en modo redundante.

Otros perfeccionamientos se proponen por las reivindicaciones del procedimiento dependientes 2 a 6. La reivindicación independiente 7 propone un sistema que corresponde a cualquiera de las reivindicaciones del procedimiento.

Este nuevo protocolo de gestión de redundancia no pretende asegurar y garantizar en seguridad de manera permanente la igualdad de los contextos de los dos ordenadores. Su aplicación sobre el plano de seguridad es más simple, parte de la definición de un nuevo principio, que es el siguiente: toda divergencia de contextos entre los ordenadores sigue siendo no peligrosa, mientras ésta no aparece de manera contraria a la seguridad en las "variables de seguridad" salientes del ordenador (a las salidas "todo o nada" seguras o a las salidas de mensajes de seguridad que deben transmitirse). Al contrario, una divergencia de contextos corre el peligro de convertirse en peligrosa, si para uno de los dos ordenadores hay al menos una de las salidas seguras emitidas por un sistema al exterior más permisivo que aquella que lo calculó. En este caso, la seguridad puede estar comprometida y el ordenador en cuestión no debería ser autorizado a reanudar el control En efecto si esto se produce, nos encentraríamos ante una configuración que sería normalmente imposible con un sólo ordenador, la seguridad por lo tanto no se garantizaría ya forzosamente. Hay que recordar, que cada ordenador es intrínsecamente seguro, puede en caso de avería emitir salidas más restrictivas pero nunca a la inversa, es decir, más permisivas.

El protocolo propuesto aquí, realiza los controles que garantizan la seguridad esencialmente bajo los tratamientos de los ordenadores (a nivel de un programa informático, también denominado "Applicatif"); estos controles están relacionados al principio básico definido anteriormente e intervienen principalmente sobre todas las salidas de seguridad del ordenador.

El control en seguridad de la igualdad de contextos entre los dos ordenadores se vuelve así solamente necesario en el arranque o en el inicio de la reintegración de un ordenador que antes habría sido aislado a consecuencia, por ejemplo, de un defecto de funcionamiento.

Este nuevo protocolo no necesita realmente más un control en seguridad de la igualdad de las "Entradas" o de los "Mensajes recibidos" al nivel de los ordenadores; pues no habrá exigencia debido al protocolo sobre la naturaleza de las entradas, en particular, las aplicaciones de seguridad podrán también ocuparse de las entradas funcionales.

Del mismo modo, como este protocolo opera de control de la seguridad principalmente bajo la aplicación, no necesitará garantizar que los programas de aplicación de los dos ordenadores ejecuten bien el mismo programa en el mismo momento.

Por lo tanto, este protocolo no requiere que el conjunto de programas tratados por los ordenadores estén todos vinculados a las metas de seguridad: las variables o los procedimientos de carácter no seguro pueden coexistir dentro de los programas.

Este protocolo permite igualmente la puesta en "cable OU" de las salidas de "todo o nada" de seguridad entre los dos ordenadores.

Los controles asociados al principio de base definido precedentemente y ejercido entre las salidas de seguridad de los dos ordenadores, permiten facilitar la puesta en ejecución sobre el plano de seguridad de una función de reintegración, por ejemplo de uno de los dos ordenadores.

Así, la invención es un procedimiento de control de un sistema que consta por lo menos de dos ordenadores que disponen cada uno de salidas seguras que pueden tomar un estado restrictivo o un estado permisivo. Cabe aclarar que el estado restrictivo es un estado intrínsecamente seguro y que todo paso incluso intempestivo a este estado no es jamás contrario a la seguridad. Dichos ordenadores pueden intercambiar información entre ellos. Cada ordenador dispone de un mismo programa de cálculo para definir los estados de salida. Un ordenador es el master y el otro el slave. Dicho procedimiento consta de un funcionamiento en modo redundante en el cual cada uno de los ordenadores calcula los estados de salida, y consta de las etapas de recepción de los estados de recepción de los estado de salida determinados por el otro ordenador; de comparación, para cada salida, de los estados determinados por los dos ordenadores, de detección de un estado de divergencia de funcionamiento si los dos ordenadores tiene determinado dos estados diferentes para la misma salida segura; si se detectada una divergencia para por lo menos una salida segura, de determinación del estado de dicha salida divergente al prohibir una transición de un estado restrictivo hacia un estado permisivo.

Para una salida temporalizable, el estado de la salida divergente guarda el nivel determinado durante el ciclo de cálculo precedente. Para una salida no temporalizable, el estado de la salida divergente toma el estado restrictivo.

Según otro aspecto, la invención es igualmente un sistema de tratamiento que dispone de dos ordenadores seguros, cada ordenador consta de entradas para recibir los datos del exterior; de salidas para abastecer de estados de salida que pueden tomar un estado restrictivo o un estado permisivo; de salidas de mensajes, también denominadas salidas distantes; una memoria para memorizar programas y datos; al menos un procesador para ejecutar los programas contenidos en la memoria, particularmente un programa para determinar el estado de las salidas a partir de las datos de entrada y de variables internas memorizadas en la memoria, de medios de sincronización y de comunicación para sincronizarse con el otro ordenador e intercambiar información con éste. Cada ordenador consta por otro lado de partes de programa de ordenador memorizados en la memoria, dichas partes de programa constan de instrucciones ejecutables por el procesador para realizar el proceso de control del sistema tal como se definió anteriormente.

La invención se comprenderá mejor y otras particularidades y ventajas aparecerán a la lectura de la descripción que sigue, la descripción hace referencia a las figuras anexas entre las cuales:

La figura 1 representa un sistema de tratamiento de acuerdo con la invención.

La figura 2 representa un diagrama de los estados de funcionamiento de un ordenador del sistema de tratamiento.

Las figuras 3 a 5 representan los organigramas puestos en funcionamiento por el sistema de tratamiento para funcionar según la invención.

La figura 1 representa un sistema de tratamiento de acuerdo con la invención. El sistema de tratamiento se describe como está incorporado en un vehículo ferroviario, ya que se trata de una aplicación que corresponde al nivel de seguridad requerido. Sin embargo, tal sistema de tratamiento puede utilizarse en un equipamiento fijo de gestión ferroviaria o también en otro ámbito distinto del transporte ferroviario, si dicho ámbito necesita la utilización de sistemas de tratamiento seguros que tienen un nivel de seguridad comparable con los sistemas de transporte ferroviario. El sistema de tratamiento consta de primeros y segundos ordenadores 10 y 20, por ejemplo del tipo descrito en EP-A-1 089 175. Cada uno de los ordenadores 10 o 20 es un ordenador intrínsicamente seguro.

Cada ordenador 10 o 20 dispone de una pluralidad de entradas locales, de una pluralidad de entradas distantes, de una pluralidad de salidas distantes, de una pluralidad de salidas seguras, de entradas/salidas de sincronización, de entradas/salidas de conexión bidireccional así como de entradas de mando. Cada ordenador 10 o 20 dispone además de un procesador, de una memoria y de medios de seguridad. Por regla general, sólo una unidad (master), por ejemplo el ordenador 10 emite las salidas distantes.

Las entradas distantes de un ordenador corresponden a los mensajes enviados por un sistema, como una unidad de gestión del tráfico que informa al ordenador embarcado, proporcionando por ejemplo, la distancia libre delante de él o que lo separa del próximo punto de parada. Las entradas locales están conectadas a los sensores y se toman muestras cíclicamente por el ordenador con el fin de tener en cuenta por ejemplo un estado físico del vehículo.

Las salidas seguras se conectan a los órganos que actúan directamente sobre el vehículo. Estas salidas seguras son las salidas que pueden tomar dos estados que clásicamente son un estado restrictivo y un estado permisivo. Se habla de estado restrictivo, cuando éste implica una acción de tipo seguro, lo que significa que activa o que detiene una acción con el fin de traer al órgano de mando a un estado de seguridad con respecto a los pasajeros. A título de ejemplo, el estado restrictivo de un freno corresponde a la presión del freno, y el estado restrictivo de la tracción corresponde a la parada de la tracción. Además, para paliar el fallo de alimentación, es corriente tener un estado restrictivo que tiene un nivel lógico cero que corresponde a una ausencia de energía.

Las salidas distantes son unas salidas que permiten, por ejemplo, enviar mensajes a una unidad de gestión del tráfico. Los mensajes enviados son, por ejemplo, los mensajes de estado del vehículo y los mensajes de posicionamiento.

La memoria del ordenador sirve para memorizar los datos y los programas. Entre los programas, estos incluyen un programa de cálculo para determinar en seguridad el estado de las salidas en función de los datos de entrada y de variables internas. Otro programa se refiere al funcionamiento en redundancia objeto de la invención. Los programas contenidos en la memoria se constituyen de instrucción comprensible por el procesador para que éste pueda ejecutarlos.

Cada uno de los ordenadores 10 y 20 disponen de los mimos programas para poder efectuar los mismos cálculos. El programa de cálculo se ejecuta cíclicamente con el fin de obtener de manera regular un cálculo del estado de salida en función de los estados de entrada. El tiempo de cálculo es muy rápido con el fin de obtener un cálculo llamado “tiempo real” que permite reaccionar cuanto antes.

A título de ejemplo, un ciclo de cálculo se realiza, por ejemplo, en un centenar de milésimas de segundo.

El cálculo de los estados de salida efectuados par cada ordenador 10 y 20 consiste, en el cada ciclo, en sacar muestras de todas las entradas, efectuar los cálculos necesarios para determinar los estados de salida y proporcionar estos estados sobre las diferentes salidas con el fin, por un lado, de comandar, por ejemplo, los órganos del vehículo y, de otra parte, de informar por lo menos a una unidad de gestión del tráfico. El cálculo de los estados de salida se efectúa de manera segura por cada ordenador, lo que significa que incluso en caso de fallo, los estados de salida proporcionados por cada ordenador de manera independiente deben corresponder a un estado de seguridad para los pasajeros del vehículo.

Cada ordenador está equipado de medios de seguridad que permiten detectar su propio fallo. Los medios de seguridad participan activamente en la seguridad intrínseca del ordenador. De acuerdo con una técnica conocida, los medios de seguridad cooperan con el procesador con el fin de efectuar los cálculos codificados en paralelo de los programas que deben de asegurarse con el fin de detectar una posible incoherencia en el desarrollo de los programas. Otras técnicas conocidas podrían utilizarse con este fin, pero la invención no se retrasará allí. En caso de detección de un fallo, los medios de seguridad neutralizan con bastante rapidez el ordenador colocando todas sus salidas en un estado restrictivo. Una vez neutralizada, el ordenador puede funcionar de nuevo a condición de ser reinicializado.

Este sistema de tratamiento, por ejemplo, se coloca en un vehículo ferroviario automático. Las entradas locales corresponden a entradas de información que proceden de elementos del vehículo. Las entradas distantes son las entradas procedentes del exterior del vehículo, por ejemplo, a través de un enlace de radio que permite comunicar con la unidad de gestión del tráfico. Las salidas distantes son unas salidas que sirven para enviar mensajes a la unidad de gestión del tráfico, por ejemplo con la ayuda del enlace de radio. Las salidas seguras son las salidas de comando de órganos del vehículo. Las conexiones bidireccionales de sincronización de los ordenadores 10 y 20 se conectan en conjunto con el fin de asegurar la sincronización de los dos ordenadores, Las entradas/salidas bidireccionales de comunicación de los dos ordenadores se conectan en conjunto con el fin de que los dos ordenadores puedan intercambiar datos entre ellos.

Un circuito de control 30 funciona en paralelo a los ordenadores 10 y 20 y recibe sin interrupción o de manera muy regular una información del funcionamiento de cada ordenador 10 y 20. La información del funcionamiento se proporciona por los medios de seguridad de cada ordenador 10 y 20 e indica si el ordenador está defectuoso o si funciona de manera normal. El circuito de control 30 determina cual de los ordenadores 10 o 20 debe de ser master y cual debe de ser slave. La determinación del estado de master o slave de un ordenador se hace con la ayuda de un programa que determina a un master sobre la base de datos arbitrarios (por ejemplo la hora o el día), de estadística de fiabilidad o de datos de mantenimiento. Pero el circuito de control 30 toma igualmente en consideración los fallos de los ordenadores para cambiar de master si esto es necesario. Cuando en un ordenador 10 o 20 se detecta un fallo, el circuito de control 30 reacciona para asegurar que el ordenador funciona en slave o, si era antes master, hacerlo pasar a slave. El circuito de control 30 permite por otro lado garantizar, de manera segura, que uno sólo de los dos ordenadores es designado como master.

Las salidas seguras se conectan a un circuito de adición 40, que en realidad puede ser, por ejemplo, realizado con la ayuda de cables OU, que vinculan en conjunto la salida de cada uno de los ordenadores. Con este fin, es necesario que cada ordenador disponga de salidas que autoricen este tipo de cableado. La ventaja de utilizar los cables OU y que baste con colocar todas las salidas de un ordenador a cero para que éstos no sean tomados en consideración (se recuerda que el estado cero o la ausencia de energía de una salida corresponden al estado restrictivo de ésta). Esto permite igualmente tener un sistema de agujas relativamente simple y por lo tanto seguro.

Como se conoce particularmente por el artículo de Essamé y al., la puesta en redundancia de dos ordenadores del tipo seguro implica una reacción que puede ser no tan segura. Con el fin particular de paliar algunos problemas vinculados a la puesta en redundancia, el enlace de sincronización entre los dos ordenadores 10 y 20 permite asegurar la buena sincronización temporal de los dos ordenadores 10 y 20. Igualmente, el enlace de comunicación bidireccional entre los dos ordenadores 10 y 20 se utiliza para el intercambio de datos entre los dos ordenadores. Los datos intercambiados son por ejemplo los datos de entrada y de salida de los dos ordenadores 10 y 20, o también los datos internos para poner al día el contexto de un ordenador con el contexto del otro ordenador. Los datos intercambiados por este enlace de comunicación permiten hacer funcionar los dos ordenadores con un mismo contexto, en particular en el momento de una reintegración (de un ordenador). Sin embargo, como ya se ha citado anteriormente: una vez que están en modo redundante, los dos ordenadores se aseguran únicamente que toman bien las mismas entradas y que tienen bien las mismas salidas. De ahí, los datos intercambiados por este enlace de comunicación permiten hacer funcionar los dos ordenadores en redundancia al tiempo que se garantiza la seguridad del sistema.

Para realizar la puesta en redundancia de dos ordenadores 10 y 20, se define para cada ordenador la posibilidad de funcionar en un estado master o en un estado de slave, el estado master o el estado slave se determinan por el circuito de control 30. Por otra parte, se definen igualmente dos modos de funcionamiento, un modo llamado redundante, donde los dos ordenadores trabajan simultáneamente y un modo denominado aislado, donde un sólo ordenador está activo. Además, cuando un ordenador falla, se vuelve a arrancar y reinicializar. La figura 2 representa un diagrama de estados de funcionamiento de un ordenador que puede tomar un estado entre los cinco estados representados, a saber un estado 100 dónde el ordenador funciona como master y en modo aislado, un estado 200 donde el ordenador funciona como master y de modo redundante, un estado 300 donde el ordenador funciona como slave y en modo redundante, en estado 400 donde el ordenador funciona como slave y en modo aislado, y un estado 500 donde el ordenador funciona según un modo de fallo o de arranque del sistema. Las flechas entre los diferentes estados representan los cambios de estado autorizados según la invención.

Así, cuando un ordenador master funciona en modo aislado 100, éste puede evolucionar hacia el modo redundante siguiendo siendo master 200 o hacia el modo de fallo 500. Cuando un ordenador amo funciona en modo redundante 200, éste puede evolucionar hacia el modo aislado siguiendo siendo master 100 o hacia un estado de slave siguiendo siendo redundante 300. Cuando un ordenador slave funciona en modo redundante 300, el ordenador puede evolucionar hacia el estado de master permaneciendo en modo redundante 200, hacia el modo aislado siguiendo siendo slave 400 o hacia el modo de fallo 500. Cuando un ordenador slave funciona en modo aislado 400, éste puede evolucionar hacia el modo redundante permaneciendo slave 300, a condición de que su contexto sea idéntico al contexto del master, o hacia el modo de fallo 500. Cuando un ordenador funciona en modo de fallo 500, éste puede evolucionar hacia el estado de slave en modo aislado 400 o hacia el estado de master en modo aislado 100. La evolución de un ordenador del estado de fallo hacia el estado de master en modo aislado 100 se produce sólo cuando los dos ordenadores están en el modo de fallo 500, es decir en el arranque o en la reactivación de todo el sistema. Estas diferentes evoluciones de un modo de funcionamiento hacia el otro se limitan con el fin de garantizar que un ordenador cuyo contexto no es coherente con el estado del sistema no pueda tomar el control.

Los ordenadores funcionan siempre en el mismo modo, o redundante o aislados. El circuito de control 30 permite garantizar que un solo ordenador es master, pero cada ordenador 10 o 20 determinan en seguridad el modo de funcionamiento que debe de aplicarse.

El modo redundante es el modo de funcionamiento nominal de protocolo de funcionamiento según la invención. Cuando se está en modo redundante, los dos ordenadores funcionan en paralelo y siempre es posible que uno de los dos ordenadores tome el relevo del otro en caso de fallo.

En el modo aislado, solamente el ordenador master se considera como el que funciona de manera segura. El ordenador slave se considera como el que no tiene un funcionamiento correcto y se debe de arrancar de nuevo, se encuentra entonces en un estado de espera donde éste va a intentar recuperar el modo redundante.

El arranque consiste en reinicializar todos los datos internos del ordenador después de colocarse en modo aislado tan pronto como el arranque se lleve a cabo.

*Determinación del master y del slave*

Esta etapa sirve para determinar en función de los datos arbitrarios o estadísticos o aún de mantenimiento, a cual de los ordenadores 10 o 20 (según la figura 1) debe de designarse como master. A título de ejemplo, un dato arbitrario puede determinar que los días pares se designe al ordenador 10, en lo sucesivo U1, como master y los días impares al ordenador 20, en lo sucesivo U2, como master. Un dato estadístico puede por ejemplo determinar la elección en función de un índice de avería de U1 y U2. Un dato de mantenimiento puede orientar la elección de U1 o U2 en función de la realización de las tareas de mantenimiento realizadas por los ordenadores durante su funcionamiento favoreciendo como master al ordenador más disponible.

*Funcionamiento en modo redundante*

El funcionamiento en modo redundante corresponde a un funcionamiento nominal del sistema de tratamiento. Uno de los ordenadores funciona de master y el otro de slave. Sin embargo, los dos ordenadores funcionan en paralelo usando los mismos datos para obtener normalmente los mismos resultados. La figura 3 representa el organigrama de funcionamiento de un ordenador que funciona en modo redundante. La única diferencia de funcionamiento entre un master y un slave se refiere a las salidas distantes que son neutralizadas por el ordenador slave.

De acuerdo con el protocolo aplicado en la invención, los dos ordenadores se aseguran mutuamente el sincronizarse entre sí e intercambiar sus entradas, etapa 201, una con la otra con el fin de mantener el mismo contexto. En efecto, cada ordenador recibe las entradas locales a partir de sensores que le son propios o de sensores comunes a los dos ordenadores 10 y 20. Cada ordenador saca muestras de las entradas locales. Aunque los ordenadores 10 y 20 se sincronizan entre sí, una entrada analógica puede evolucionar entre los dos momentos de muestreo consustancial a cada ordenador. Así, es posible que una misma entrada tenida en cuenta por cada ordenador tenga un valor diferente y corra el riesgo de arrastrar las etapas diferentes de salida. Con este fin, los ordenadores intercambian entre ellos sus datos de entrada con el fin de devolver sus entradas idénticas.

A continuación, cada ordenador pone en marcha su programa de cálculo, etapa 202. El programa de cálculo de cada ordenador determina entonces los estados de salida $S_A$ con arreglo a las entradas. Los programas de cálculo son idénticos y los resultados deben de ser normalmente los mismos.

Al final del cálculo, cada ordenador recibe los estados de salida $S_B$ calculados por el otro ordenador con el fin de efectuar una comprobación de redundancia, etapa 203.

Como se ha indicado anteriormente, y sin que haya un fallo en uno de los ordenadores, éstos pueden a pesar de todo tener un funcionamiento no idéntico que es fuente de fallo global del sistema. La comparación de los estados de salida de los dos ordenadores sirve para detectar una divergencia de las salidas, test 204.

Si todos los estados de salida son idénticos, lo que corresponde a un funcionamiento normal, entonces el ordenador master envía los mensajes calculados sobre las salidas distantes con destinación a una unidad distante, y los dos ordenadores envían simultáneamente sus estados de salida calculados $S_A$ sobre sus salidas S, etapa 205. Los estados de salida de los dos ordenadores se recombinan con la ayuda del circuito de adicción 40. A continuación se efectúa un nuevo ciclo de cálculo cambiando de nuevo las entradas, regreso a la etapa 201. Este es el funcionamiento normal del sistema.

Si durante el funcionamiento normal del sistema, uno de los ordenadores 10 o 20 esta defectuoso y si éste es el ordenador master, el circuito de control 30 intercambia los ordenadores master y slave. La toma en consideración de la calidad de master y de slave se hace por lo tanto antes de efectuar de nuevo la etapa 201.

Al comparar las salidas, es posible detectar una divergencia sobre una o varias salidas. La divergencia se caracteriza por lo menos por un estado de salida diferente para los dos ordenadores. Esta divergencia indica que la seguridad del sistema corre el riesgo de no ser ya asegurada puesto que los dos ordenadores tienen comportamientos antagónicos. El cambio sobre uno de los dos ordenadores no es fácil de determinar de manera instantánea. En efecto, puede muy

bien tener una divergencia sobre una salida sin que ninguno de los dos ordenadores tenga un fallo. Esta divergencia puede ser una divergencia puntual sin ninguna incidencia, o puede igualmente ser el anuncio de un fallo más grave del sistema. La parada arbitraria de uno de los dos ordenadores correría el riesgo de hacerlo indisponible mientras que el otro ordenador se comprueba es un ordenador defectuoso.

En caso de divergencia, varios casos tiene que considerarse. Un primer caso concierne a las salidas distantes. Estas salidas distantes corresponden a mensajes sobre el estado del vehículo y su posición que son enviados a la unidad distante. En caso de divergencia sobre las salidas distantes, conviene no emitir mensajes. Es preferible no enviar un mensaje a enviar un mensaje falso. Si la divergencia persiste sobre las salidas distantes, convendrá entonces cambiar al modo aislado. Tal operación se puede realizar en algunos ciclos de cálculo. Este primer caso no es crítico y es diferente de las salidas seguras, no se trata en los organigramas descritos.

Un segundo caso, más preocupante, se refiere a las salidas seguras. Se distinguen las salidas seguras de tipo temporizables y las salidas seguras del tipo no temporizables. Una salida segura está considerada como temporizable si su mantenimiento en un estado permisivo durante un cierto tiempo es posible sin que esto perjudique la seguridad de los pasajeros, a pesar de que debe pasar a un estado restrictivo. Una salida segura está considerada como no temporizable si esta corre el riesgo, por su mantenimiento en un estado permisivo, de perjudicar la seguridad de los pasajeros.

En caso de divergencia de los dos ordenadores, se pasa a un modo de funcionamiento restrictivo. Los ordenadores determinan entonces los estados de salida S a aplicar de manera restrictiva, etapa 206. Cualquiera que sea el tipo de salida segura en cuestión dónde una divergencia se produce, conviene, según la invención, prohibir toda transición de un estado restrictivo hacia un estado permisivo de la salida segura.

Si se trata de una divergencia sobre una salida segura no temporizable, conviene forzar su estado a un estado restrictivo.

Si se trata de una divergencia sobre una salida segura temporizable, es posible mantener su estado en el estado precedente. La determinación del estado debe necesariamente evitar realizar una transición de un estado restrictivo hacia un estado permisivo. Además, un ordenador no determina la salida permisiva si calculó una salida restrictiva. La determinación así hecha por los dos ordenadores se puede encontrar que es la misma o siempre divergente, de todas formas la combinación de las salidas efectuadas con la ayuda del circuito de adición 40 garantiza el mantenimiento de la salida divergente en el estado precedente.

Los estados de salida S están determinados y aplicados, conviene entonces determinar si la divergencia era sólo una divergencia puntual o si la divergencia está vinculada a un fallo de uno de los dos ordenadores.

Si la divergencia persiste durante una duración de divergencia $\Delta$ t que va más allá de una duración predeterminada k, test 207, entonces los ordenadores cambian al modo aislado, etapa 208. Desde que un ordenador 10 o 20 cambia al modo aislado, fuerza al otro ordenador 20 o 10 a cambiar. La duración de la divergencia se mide en número de ciclos de cálculo. Esta duración de divergencia $\Delta$ t se pone a cero tan pronto como se constata una divergencia, etapa 209, y ella es, por ejemplo, incrementada después de cada ciclo de cálculo divergente, etapa 210.

Mientras que la duración predeterminada k no se alcanza, los ciclos de cálculo se enlazan intercambiando las entradas, etapa 211, realizando el cálculo de los estado de salida $S_A$, etapa 212, y comparando los estado calculados $S_A$ con los estados $S_B$ calculados por el otro ordenador, etapa 213 y test 214.

Si durante un ciclo de cálculo, las salidas no son más divergentes, entonces el sistema va a reanudar su modo de funcionamiento normal al aplicar sobe sus salida S los estados de salida calculados $S_A$, etapa 215. Los ordenadores ejecutan a continuación la etapa 201.

Hasta que un ciclo de cálculo pone de manifiesto una divergencia, el funcionamiento sigue siendo restrictivo. Se incrementa la duración de la divergencia $\Delta$ t y se determinan los estados de salida S de manera restrictiva en la etapa 208.

En caso de divergencia, el estado restrictivo de una salida corre el riesgo de ser mantenido, mientras hay divergencias de los dos ordenadores. Si el funcionamiento restrictivo dura demasiado tiempo, puede haber indisponibilidad. Para evitar esto, es conveniente utilizar una duración predeterminada k relativamente breve, siendo suficientemente largo para permitir una conmutación master/slave. A titulo de ejemplo, conviene una duración predeterminada k igual a dos o tres ciclos de cálculo.

Si durante el tiempo predeterminado un ordenador es defectuoso, los cambios de estado, master/slave se pueden determinar por el circuito de control 30. Sin embargo la consideración del estado master o slave se hace sólo después de la etapa 210.

Hay que tener en cuenta que en el momento de la determinación de la salida, etapa 206, los dos ordenadores pueden determinar un mismo estado si la salida precedente era restrictiva, pero pueden determinar dos estados diferentes si el estado precedente era permisivo.

Con el fin de determinar el origen del fallo, cada ordenador dispone de un circuito de detección de la disfunción (equivalente a un perro guardián) que permite indicar al circuito de comando 30 si es o no defectuoso. Si es el ordenador master el que es defectuoso, entonces hay un cambio de master y de slave que se realiza por el circuito de comando 30. Este cambio se produce durante la duración predeterminada k y sigue siendo completamente transparente al nivel del funcionamiento global del sistema de tratamiento.

El ordenador defectuoso será automáticamente posicionado en slave por el circuito de comando 30, y el paso al estado restrictivo de las salidas del ordenador defectuoso provoca la incoherencia de las salidas que harán cambiar al ordenador master en el modo aislado, mientras que el ordenador defectuoso cambia al modo defectuoso.

Así, en caso de detección de un fallo en el ordenador master, el ordenador master se vuelve slave y viceversa. El cambio en el modo aislado se hace sin parar el funcionamiento del sistema conservando al mismo tiempo el ordenador más apto para dirigir el sistema.

*Funcionamiento en el modo aislado*

En el modo aislado, solamente el ordenador master funciona de manera activa, dado que el ordenador slave ha sido neutralizado. La figura 4 representa el organigrama de funcionamiento de un ordenador cuando éste está en un estado de master. La figura 5 representa el organigrama de funcionamiento de un ordenador cuando éste está en un estado de slave. Estos dos organigramas se ejecutan en paralelo. Los dos ordenadores quedan sincronizados, es decir que los ciclos de cálculo se desarrollan al mismo tiempo.

El modo aislado es un modo temporal de funcionamiento. Toma parte durante el arranque del sistema donde los dos ordenadores arrancan desde un estado de inicialización, en caso de detección de una incoherencia entre los dos ordenadores o de reactivación de uno sólo de los ordenadores. El paso del modo de arranque al modo aislado en un estado master sólo puede producirse cuando los dos ordenadores arrancan, o reinicializan, juntos y para eso, los ordenadores deben de estar en el estado de inicialización (el conjunto de las variables y salidas del ordenador al estado restrictivo).

El ordenador master, según la figura 4, sólo tiene en cuanta sus entradas para calcular los estados de salida S, etapa 101, que se aplican, etapa 102, sobre las salidas S del sistema. Las entradas y salidas del ordenador slave no son tenidas en cuenta por el ordenador master. De este modo, el sistema funciona de manera segura como un único ordenador. Sin embargo, el ordenador master comunica sus entradas, las variables internas y sus estados de salida al ordenador slave. En funcionamiento normal, el ordenador master encadena así los ciclos de cálculo. Aquí, las salidas del ordenador slave son totalmente forzadas a ser restrictivas hacia el exterior en el modo aislado.

Con el fin de poder funcionar de nuevo en modo redundante, el ordenador master debe primero transmitir su contexto al ordenador slave. A continuación, verifica si es posible un cambio a un modo redundante, test 104. Un cambio de modo es posible sólo si el ordenador slave hace la petición después de haber comprobado que es convergente con los estados de salida del ordenador master. El master verifica a continuación que hay una identidad de contexto, es decir una identidad de las variables internas entre el slave y el master. Para reducir el tiempo de control, puede utilizarse una firma de las variables internas. Pero la comprobación de la coherencia de contexto puede llevarse a cabo sobre varios ciclos de cálculo. Si existe cambio al modo redundante, etapa 105, este vaivén implica al mismo tiempo el cambio del ordenador slave al modo redundante.

En caso de un fallo del ordenador master antes del paso al modo redundante, esto podría causar una indisponibilidad del sistema hasta una reactivación del ordenador.

En el modo aislado el sistema se comporta en el peor de los casos como un único ordenador. Los estados de salida siguen, por lo tanto, siendo seguros ya que el ordenador es intrínsecamente seguro.

Para prevenir una indisponibilidad del material, es por tanto necesario, volver a pasar al modo redundante lo más rápido posible. Para el ordenador slave, el modo aislado es un modo de puesta en coherencia con el ordenador master. Por ejemplo, un ordenador slave que se encuentra en el modo aislado acaba de volver a arrancar con las variables internas que han sido reinicializadas, o cuando estaba previamente en el modo redundante pero divergente con el ordenador master. En todos los casos, el contexto de funcionamiento del ordenador slave se supone diferente del contexto de funcionamiento del ordenador master.

El ordenador slave efectúa ciclos de cálculo dónde recibe las entradas del ordenador master, etapa 401, acompañado de datos contextuales del ordenador master para poner al día sus variables internas. Al ser muy importante el número de variable internas, por ejemplo algunos millares, la transferencia de valores internos se puede repartir sobre varios ciclos de cálculo. Ciertas variables pueden evolucionar de un ciclo a otro. Éstas deberán ser, en este caso, transferidas algunas veces. Luego, el ordenador slave efectúa un cálculo de los estados de salida $S_A$, etapa 402. Recibe también los estados de salida $S_B$ del ordenador master, etapa 403, para verificar sus estados de salida calculados $S_A$. Las salidas del ordenador slave con destino al exterior se colocan en cualquier caso en un estado cero, etapa 404, a fin de que el circuito de adicción 40 proporcione los estados de salida únicamente del ordenador master.

Una etapa 405 esta prevista, con el fin de que pida al ordenador slave pasar a master, pero volveremos de nuevo sobre este punto más adelante.

De lo contrario (caso N), el ordenador slave verifica si sus salidas son divergentes o no con el ordenador master, test 406.

Si el ordenador slave es divergente test 406, caso N, entonces se arranca de nuevo, etapa 407.

Si el ordenador slave es convergente, test 406, caso O, entonces pide al ordenador master cambiar al modo redundante, etapa 408. Esta petición activa el proceso de verificación 409 del contexto en un ciclo determinado, esto corresponde al conjunto de sus datos internos memorizabes y de seguridad por el ordenador master. Si la verificación de los datos internos por el master acaba de manera positiva, el master cambia al modo redundante arrastrando al ordenador slave al modo redundante, etapa 410. La verificación del contexto correspondiente a un ciclo dado y efectuado por el amo puede tener lugar sobre varios ciclos de cálculo. En cambio, si en el curso de uno de estos ciclos de cálculo, el slave no tiene más sus salidas coherentes con el ordenador master según la etapa 406, caso N, entonces se arranca de nuevo en la fase 407 y deja de pedir el cambio en el modo redundante, lo que anula la verificación hecha por el master.

Mientras el ordenador slave es divergente, no es posible volver a cambiar al modo redundante. Si el ordenador slave no llega a converger con el master, es porque el intercambio de datos internos no es suficiente y que es preferible reinicializar al ordenador slave para volver a salir sobre una mejor base.

En los casos en los que el ordenador amo se vuelva a su vez defectuoso mientras que el ordenador esclavo esta asilado pero en funcionamiento, el circuito de control 30 va a indicar al esclavo que cambie a amo. Ahora bien, tal toma de enlace no es factible debido a que el contexto de slave no se considera coherente con el estado general del sistema. Conviene volver a arrancar al slave. El test 405 verifica si se le pide al slave de pasar a master con el fin de efectuar, después de la etapa 405, si es el caso (O), la etapa de reinicialización 407 y de volver a arrancar todo el sistema.

El procedimiento que acaba de ser descrito es un programa de supervisión de la puesta en redundancia de dos ordenadores. La persona experta apreciará, para poner en ejecución la invención, que tendrá que hacer una simple actualización del programa en un sistema.

## REIVINDICACIONES

1. Procedimiento de control de un sistema de tratamiento que requiere una gran seguridad para un transporte, que consta por lo menos de dos ordenadores (10, 20) dispuestos cada uno de salidas seguras que pueden estar en un estado restrictivo o en un estado permisivo, el estado restrictivo es un estado de carácter intrínsecamente seguro, dichos ordenadores pueden intercambiar información entre ellos, cada ordenador intrínsecamente seguro está en disposición de un mismo programa de cálculo para definir los estados de salida ($S_A$), un ordenador es master y el otro es slave, dónde los dos ordenadores están provistos de medios de seguridad para detectar sus propios fallos de funcionamiento, y en el cual, en caso de detección de fallos en el ordenador master, el ordenador master se convierte en slave, este procedimiento consta de:

a) un funcionamiento en modo redundante en el que cada uno de los ordenadores calcula (202, 212) los estados de salida, **caracterizado** por las etapas siguientes:

- recepción (203, 213) de los estados de salida ($S_a$) determinados por el otro ordenador,
- comparación (204, 214), para cada salida, de los estados determinados por los dos ordenadores,
- detección de un estado de funcionamiento divergente, si los dos ordenadores han determinado dos estados diferentes para una misma salida segura,
- si una divergencia se detecta para al menos una salida segura, la determinación (206) del estado de dicha salida divergente, impide una transición de un estado restrictivo hacia un estado permisivo,

b) un funcionamiento temporal en modo aislado, en el que:

- si un estado de divergencia persiste mas allá de una primera duración predeterminada (k), el funcionamiento del ordenador cambia a un modo aislado (208) en el cual el ordenador master no toma en cuenta los estados de salida del ordenador slave y dónde el ordenador slave no proporciona más salidas al exterior,
  - para el ordenador slave, el modo aislado es un modo de puesta en coherencia con el ordenador master,
- con el fin de poder funcionar de nuevo en modo redundante, el ordenador slave pide al ordenador master cambiar al modo redundante, en el que este cambio de modo es posible sólo si el ordenador slave hizo la petición después de haber constatado que es convergente con las etapas de salida del ordenador master.

2. Procedimiento según la reivindicación 1, en el que

el estado de la salida divergente guarda un estado determinado durante un ciclo de control precedente.

3. Procedimiento según una de las reivindicaciones 1 o 2, en el que

el estado de salida divergente cambia al estado restrictivo.

4. Procedimiento según una de las reivindicaciones precedentes, en el que

en el modo aislado, el ordenador slave recibe (401) informaciones de entrada y de contexto del ordenador master, calcula (402) los estados de salida ($S_A$) tomando en cuenta las informaciones del ordenador master, compara (403) los estados de salida calculados ($S_A$) de los estados de salida ($S_B$) del ordenador master y detecta (405) si existe un estado convergente en el que todos los estados de salida determinados por el ordenador slave corresponden a los estados de salida del ordenador master.

5. Procedimiento según una de las reivindicaciones precedentes, en el que

en el modo aislado, el ordenador master verifica (104) el contexto del ordenador slave y cambia (105) al modo redundante si hay identidad de contextos entre los dos ordenadores.

6. Procedimiento según una de las reivindicaciones precedentes, en el que

si un estado de divergencia persiste, entonces el ordenador slave se reinicializa (407).

7. Sistema de tratamiento dispuesto de dos ordenadores seguros (10, 20), en el que cada ordenador consta de:

- las entradas para recibir los datos de exterior,
- las salidas para proporcionar los estados de salida que pueden tomar un estado restrictivo o un estado permisivo,
- una memoria para memorizar los programas y los datos,
- al menos un procesador para ejecutar los programas contenidos en la memoria, en particular un programa para determinar el estado de salida a partir de los datos de entrada y de variables internas memorizadas en la memoria,
- los medios de sincronización y de comunicación para sincronizarse con el otro ordenador e intercambiar información con él,

**caracterizado** porque,

cada ordenador consta de partes de programas de ordenador memorizados en la memoria, dichas partes de programa comportan las instrucciones que pueden ser ejecutadas por el procesador para realizar el procedimiento de cualquiera de las reivindicaciones 1 a 6.

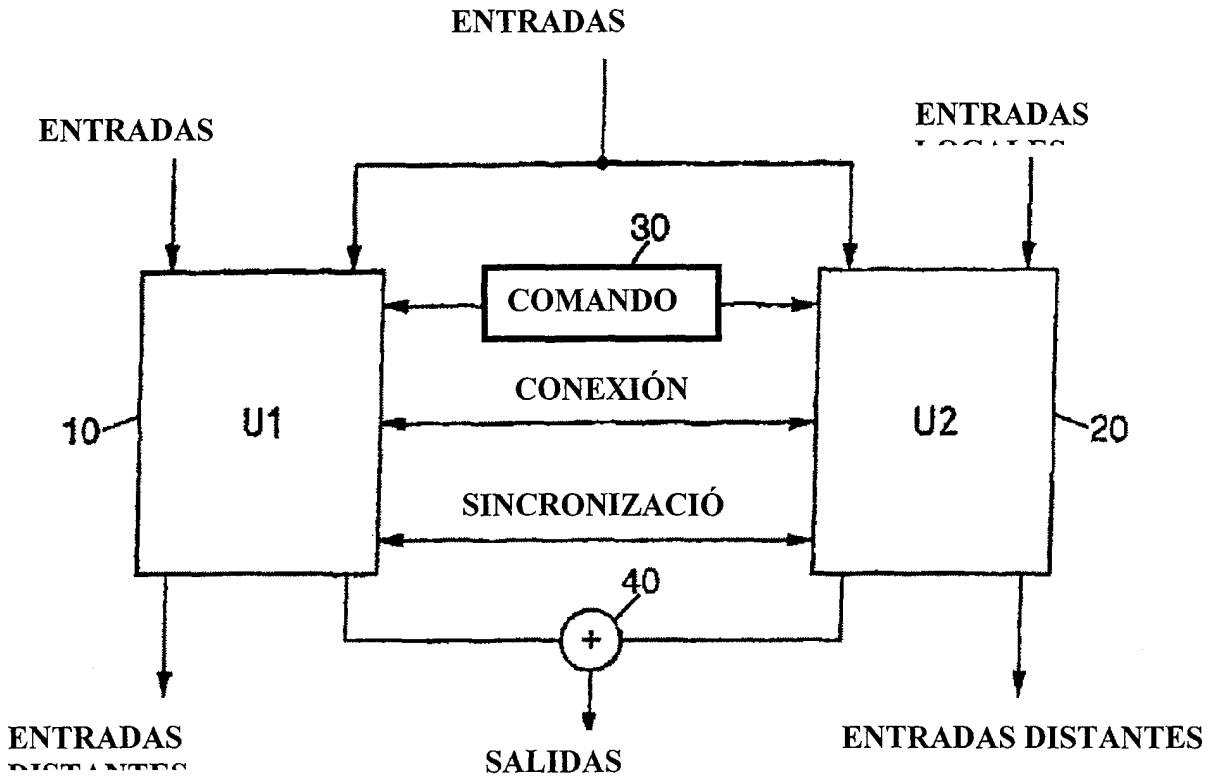
ENTRADAS
ENTRADAS
ENTRADAS
30
COMANDO
CONEXIÓN
10
U1
U2
20
SINCRONIZACIÓ
40
+
ENTRADAS
SALIDAS
ENTRADAS DISTANTES

FIG. 1

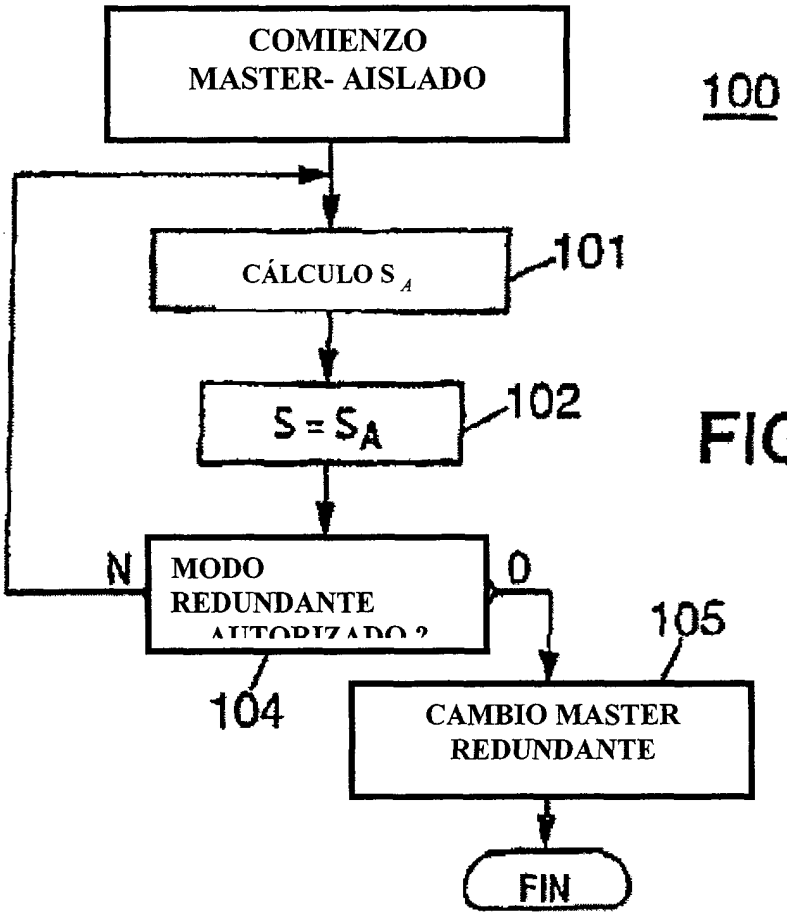
COMIENZO
MASTER- AISLADO
100
CÁLCULO S A
101
S = SA
102
N
MODO
REDUNDANTE
0
105
104
CAMBIO MASTER
REDUNDANTE
FIN

FIG. 4

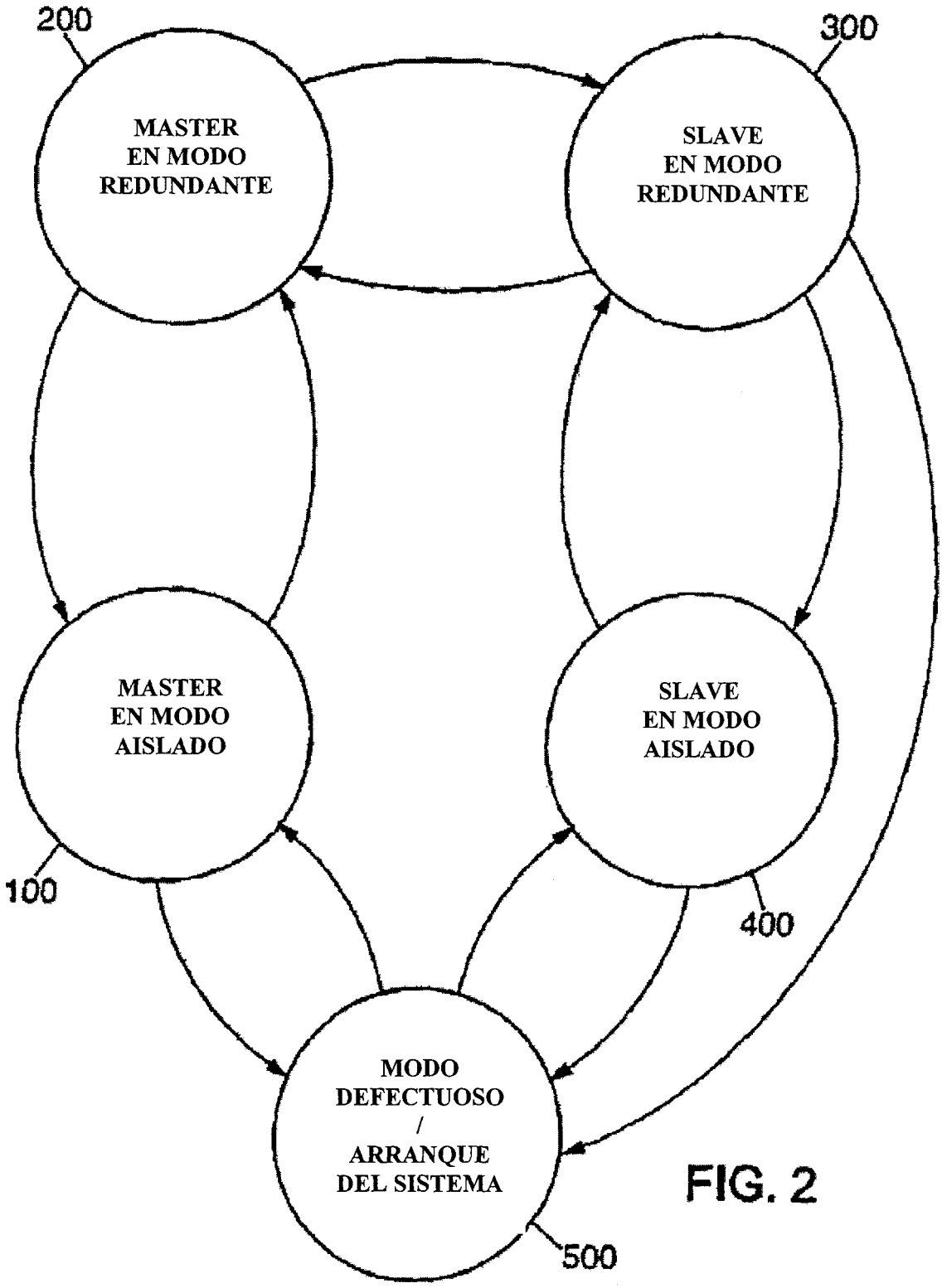
200
MASTER
EN MODO
REDUNDANTE
300
SLAVE
EN MODO
REDUNDANTE
MASTER
EN MODO
AISLADO
100
SLAVE
EN MODO
AISLADO
400
MODO
DEFECTUOSO
/
ARRANQUE
DEL SISTEMA
500

FIG. 2

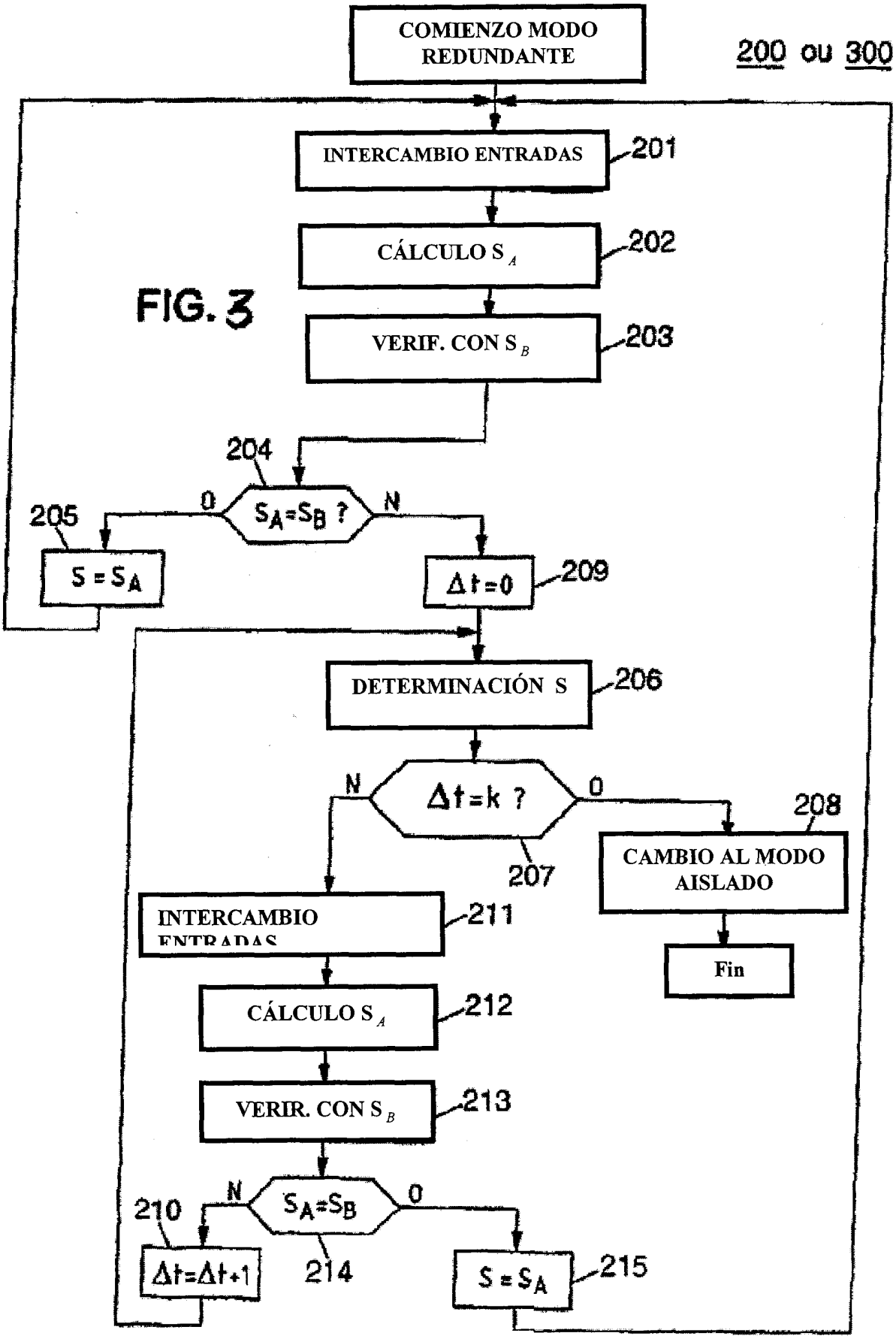
COMIENZO MODO REDUNDANTE
200 ou 300
INTERCAMBIO ENTRADAS
201
CÁLCULO $S_A$
202
VERIF. CON $S_B$
203
204
$S_A=S_B$ ?
O
N
205
$S = S_A$
$\Delta t=0$
209
DETERMINACIÓN S
206
$\Delta t=k$ ?
N
O
207
208
CAMBIO AL MODO AISLADO
Fin
INTERCAMBIO ENTRADAS
211
CÁLCULO $S_A$
212
VERIR. CON $S_B$
213
$S_A=S_B$
N
O
210
$\Delta t=\Delta t+1$
214
$S = S_A$
215

FIG. 3

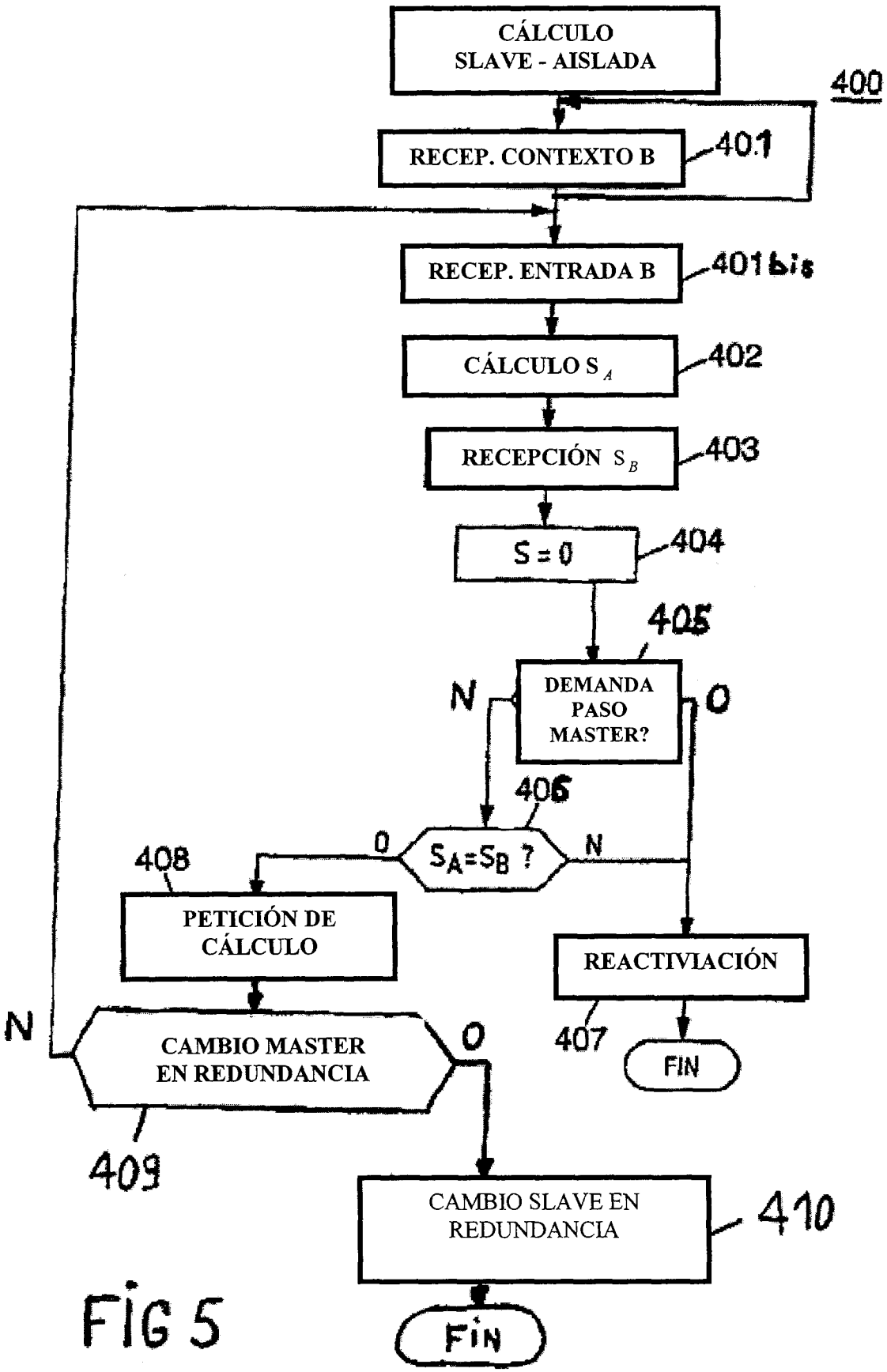
CÁLCULO
SLAVE - AISLADA
400
RECEP. CONTEXTO B
401
RECEP. ENTRADA B
401 bis
CÁLCULO $S_A$
402
RECEPCIÓN $S_B$
403
S = 0
404
405
DEMANDA
PASO
MASTER?
N
O
406
$S_A$=$S_B$ ?
O
N
408
PETICIÓN DE
CÁLCULO
REACTIVIACIÓN
N
CAMBIO MASTER
EN REDUNDANCIA
O
407
FIN
409
CAMBIO SLAVE EN
REDUNDANCIA
410
FIN

FIG 5