(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 523 129**

(51) Int. CI.:

G06F 11/10 (2006.01)
G06F 11/20 (2006.01)
G05B 9/03 (2006.01)
G06F 11/16 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **15.02.2012 E 12155519 (7)**

(97) Fecha y número de publicación de la concesión europea: **03.09.2014 EP 2490124**

(54) Título: **Controlador de procesos doblemente redundante**

(30) Prioridad:

16.02.2011 US 201113029102

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.11.2014

(73) Titular/es:

INVENSYS SYSTEMS INC. (100.0%)
Intellectual Property Department Building 51
Bristol Park 33 Commercial Street
Foxboro, MA 02035, US

(72) Inventor/es:

GALE, ALAN A.;
KLING, ANDREW L.;
TIMPERLEY, MARK E.;
BASS, LAWRENCE T.;
LAVALLEE, JOHN J.;
CRANSHAW, GEORGE W. y
FOSKETT, ALAN M.

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Controlador de procesos doblemente redundante

Antecedentes

Los sistemas de control de procesos pueden ser implementados para que controlen automáticamente procesos industriales en función de una lógica y/o de reglas predefinidas. Los procesos industriales pueden ser realizados por motores, válvulas, calentadores, bombas y similares, a los que se puede denominar dispositivos de proceso o dispositivos en servicio, en plantas industriales, refinerías, plantas de procesamiento de alimentos y otras plantas. Los sistemas de control de procesos pueden monitorizar parámetros y/o propiedades de procesos en curso recibiendo salidas de sensores acoplados a los procesos, por ejemplo sensores de temperatura, sensores de presión, sensores de movimiento, sensores de peso, sensores de densidad, sensores de caudal y otros sensores. Dispositivos automatizados de control, por ejemplo controladores, pueden regular y controlar los dispositivos de proceso en función de los parámetros y las propiedades detectados en función de una lógica predefinida y/o de entradas de instrucciones procedentes, por ejemplo, de una interfaz hombre-máquina.

Las metodologías actuales para su uso en la redundancia de sistemas de control incluyen comúnmente triple redundancia modular (TMR) con votación en todas las interfaces y/o diversos procedimientos de cambio en caliente en los que un módulo activo controla todos los procesos y un módulo en espera está listo para tomar el releve en caso de cualquier fallo. Normalmente, los enfoques de TMR permiten que los tres tramos del controlador participen de un cambio votado a las estaciones extremas, teniendo todos los miembros un voto igual y una votación de 2 de 3 para resolver ligeras diferencias de sincronización. La TMR puede verificar votos en todos los puntos finales del sistema. Esto puede conllevar triplicar algunos componentes de soporte físico en el controlador, aumentando el coste y una votación individualizada en los puntos finales, lo que aumenta la complejidad de los puntos finales. Los procedimientos de cambio en caliente dependen de que el dispositivo activo realice una autodiagnosis de un fallo y se apague para permitir que el módulo en espera sea promovido al papel activo. Tales procedimientos de cambio en caliente pueden conllevar que se apliquen valores erróneos al proceso hasta que los diagnósticos detecten un módulo activo defectuoso, que el módulo activo defectuoso se apague y que el módulo en espera sea promovido al estado activo.

Otros sistemas actuales pueden utilizar un enfoque activo dual que no requiere la votación de los puntos finales. Por ejemplo, dos módulos de control pueden ejecutar la misma secuencia de operaciones y producir las mismas configuraciones y/o las mismas instrucciones de un proceso. Las configuraciones y/o las instrucciones producidas por los dos módulos de control pueden ser verificadas por soporte físico de comunicaciones para que estén de acuerdo antes de remitirlas a otros dispositivos, por ejemplo a dispositivos de procesos controlados. En una realización, este enfoque puede conllevar un soporte físico dedicado para la sincronización del estado, la sincronización de relojes y la comparación del contenido de los mensajes.

Los sistemas actuales, así como los descritos en el presente documento, pueden ser implementados en un procesador que forme parte de un ordenador. Como apreciará un experto en la técnica, los ordenadores y los sistemas de ordenadores pueden darse de formas, tamaños e instrumentos diferentes. A la vez, puede decirse que los ordenadores y/o los sistemas de ordenadores comprenden algunos elementos comunes y que tienen algunos elementos opcionales comunes. Por ejemplo, un sistema de ordenador puede comprender un procesador (al que se puede denominar unidad central de procesador o CPU) que está en comunicación con dispositivos de memoria que incluyen un almacenamiento secundario, memoria de solo lectura (ROM), memoria de acceso aleatorio (RAM), dispositivos de entrada/salida (E/S) y dispositivos de conectividad de red. El procesador puede ser implementado como uno o más chips de CPU.

Se entiende que programando y/o cargando instrucciones ejecutables en el sistema de ordenadores, cambian al menos una de la CPU, la RAM y la ROM, transformando el sistema de ordenadores en parte en una máquina o un aparato particular que tiene la funcionalidad novedosa enseñada por la presente divulgación. Es fundamental para las técnicas de ingeniería eléctrica y de ingeniería de soporte lógica que la funcionalidad que pueda ser implementada por la carga de soporte lógico ejecutable en un ordenador pueda convertirse en una implementación de soporte físico mediante reglas de diseño bien conocidas. Las decisiones de si implementar un concepto en soporte lógico o en soporte físico normalmente dependen de consideraciones de estabilidad del diseño y del número de unidades que hayan de producirse, no en ninguna cuestión implicada en traducir del dominio del soporte lógico al dominio del soporte físico. En general, puede preferirse que un diseño que esté aún sujeto a un cambio frecuente se implemente en soporte lógico, porque volver a poner en circulación una implementación de soporte físico es más caro que volver a poner en circulación un diseño de soporte lógico. En general, puede preferirse que un diseño que sea estable que se produzca en gran volumen se implemente en soporte físico, por ejemplo en un circuito integrado para aplicaciones específicas (ASIC), porque para grandes tiradas de producción la implementación de soporte físico puede ser menos cara que la implementación de soporte lógico. A menudo, un diseño puede ser desarrollado y probado en forma de soporte lógico y después ser transformado, mediante reglas de diseño bien conocidas, en una implementación equivalente de soporte físico en un circuito integrado para aplicaciones específicas que cablea físicamente las instrucciones del soporte lógico. De la misma manera que una máquina controlada por un nuevo

ASIC es una máquina o aparato particular, también un ordenador que haya sido programado y/o cargado con instrucciones ejecutables puede ser visto como una máquina o aparato particular.

El almacenamiento secundario comprende normalmente una o más unidades de disco o unidades de cinta y se usa para el almacenamiento no volátil de datos y como un dispositivo de almacenamiento de datos de desbordamiento si la RAM no es lo suficientemente grande para contener todos los datos de trabajo. El almacenamiento secundario puede usarse para almacenar programas que se cargan en la RAM cuando tales programas son seleccionados para su ejecución. La ROM se usa para almacenar instrucciones y quizás datos que se leen durante la ejecución de programas. La ROM es un dispositivo de memoria no volátil que normalmente tiene una pequeña capacidad de memoria con respecto a la mayor capacidad de memoria del almacenamiento secundario. La RAM se usa para almacenar datos volátiles y quizás para almacenar instrucciones. El acceso tanto a la ROM como a la RAM es normalmente más rápido que al almacenamiento secundario. En algunos contextos, el almacenamiento secundario, la RAM y/o la ROM pueden ser denominados medios de almacenamiento legibles por ordenador y/o medios no transitorios legibles por ordenador.

Los dispositivos de E/S pueden incluir impresoras, monitores de vídeo, pantallas de cristal líquido (LCD), pantallas táctiles, teclados, teclados numéricos, conmutadores, selectores, ratones, ratones de bola de mando, dispositivos de reconocimiento de voz, lectores de tarjetas, lectores de cinta de papel u otros dispositivos de entrada bien conocidos.

Los dispositivos de conectividad de red pueden adoptar la forma de módems, baterías de módems, tarjetas Ethernet, tarjetas de interfaz de bus serie universal (USB), interfaces de serie, tarjetas Token Ring, tarjetas de interfaz de datos distribuida por fibra (FDDI), tarjetas de red inalámbrica de área local (WLAN), tarjetas transceptoras de radio, tales como las de acceso múltiple por división de código (CDMA), el sistema global para comunicaciones móviles (GSM), la evolución a largo plazo (LTE), la interoperabilidad mundial para el acceso por microondas (WiMAX) y/u otras tarjetas transceptoras de radio de protocolos con interfaz aérea y dispositivos de red bien conocidos. Estos dispositivos de conectividad de red pueden permitir que el procesador se comunique con Internet o una o más intranets. Con una conexión de red, se contempla que el procesador pueda recibir información de la red o que pueda enviar información a la red en el curso de la realización de las etapas de los procedimientos descritos más arriba. tal información, que se representa a menudo como una secuencia de instrucciones que ha de ser ejecutada usando un procesador, puede ser recibida de la red y enviada a la misma, por ejemplo en forma de una señal de datos de ordenador implementada en una onda portadora.

Tal información, que puede incluir datos o instrucciones que hayan de ser ejecutadas usando, por ejemplo, un procesador, puede ser recibida de la red y enviada a la misma, por ejemplo en forma de una señal de datos de ordenador de banda base o una señal implementada en una onda portadora. La señal de banda base o la señal implementada en la onda portadora generada por los dispositivos de conectividad de red pueden propagarse en o por la superficie de conductores eléctricos, en cables coaxiales, en guiaondas, en un conducto óptico, por ejemplo una fibra óptica, o por el aire o el espacio libre. La información contenida en la señal de banda base o la señal embebida en la onda portadora puede ser ordenada según diferentes secuencias, según pueda resultar deseable ya sea para el procesamiento o la generación de información o la transmisión o la recepción de la información. La señal de banda base o la señal embebida en la onda portadora, u otros tipos de señales usadas actualmente o desarrolladas en el futuro, pueden ser generadas según varios procedimientos bien conocidos para un experto en la técnica. La señal de banda base y/o la señal embebida en la onda portadora pueden ser denominadas en algunos contextos señal transitoria.

El procesador ejecuta instrucciones, códigos, programas de ordenador, secuencias de órdenes a los que accede desde un disco duro, un disco flexible, un disco óptico (todos estos diversos sistemas a base de discos pueden ser considerados almacenamiento secundario), ROM, RAM o los dispositivos con conectividad de red. Aunque solo se muestra un procesador, puede haber presentes múltiples procesadores. Así, aunque puede decirse que las instrucciones son ejecutadas por un procesador, las instrucciones pueden ser ejecutadas simultáneamente, en serie o ejecutadas de otra forma, por uno o múltiples procesadores. Las instrucciones, los códigos, los programas de ordenador, las secuencias de órdenes y/o los datos a los que se pueda acceder desde el almacenamiento secundario, por ejemplo discos duros, discos flexibles, discos ópticos y/u otro dispositivo, la ROM, y/o la RAM pueden denominarse en algunos contextos instrucciones no transitorias y/o información no transitoria.

En una realización, el sistema de ordenador puede comprender dos o más ordenadores en comunicación mutua que colaboran para llevar a cabo una tarea. Por ejemplo, pero no a título de limitación, una aplicación puede estar dividida de tal modo que permita el procesamiento concurrente y/o paralelo de las instrucciones de la aplicación. Alternativamente, los datos procesador por la aplicación pueden ser divididos de tal modo que permitan el procesamiento concurrente y/o paralelo de diferentes porciones de un conjunto de datos por los dos o más ordenadores. En una realización, el sistema de ordenadores puede emplear un soporte lógico de virtualización para proporcionar la funcionalidad de varios servidores que no está directamente limitado al número de ordenadores del sistema de ordenadores. Por ejemplo, el soporte lógico de virtualización puede proporcionar veinte servidores virtuales en cuatro ordenadores físicos. En una realización, la funcionalidad dada a conocer más arriba puede ser proporcionada ejecutando la aplicación y/o las aplicaciones en un entorno informático en nube. La informática en

nube puede comprender proporcionar servicios informáticos mediante una conexión de red que use recursos informáticos con cambio dinámico de escala. La informática en nube puede estar soportada, al menos en parte, por un soporte lógico de virtualización. Puede establecerse un entorno informático en nube por parte de la empresa y/o puede ser objeto de contratación a un tercer proveedor cuando sea necesario. Algunos entornos informáticos en nube pueden comprender recursos informáticos en nube propiedad de la empresa, y operados por la misma, así como recursos informáticos en nube contratados y/o arrendados a un tercer proveedor.

En una realización, parte o la totalidad de la funcionalidad dada a conocer más arriba puede ser proporcionada como un producto de programa de ordenador. El producto de programa de ordenador puede comprender uno o más medios de almacenamiento legibles por ordenador que tienen implementados en los mismos código de programa utilizable por ordenadores para implementar la funcionalidad dada a conocer más arriba. El producto de programa de ordenador puede comprender estructuras de datos, instrucciones ejecutables y otro código de programa utilizable por ordenadores. El producto de programa de ordenador puede ser implementado en medios de almacenamiento de ordenador extraíbles y/o medios de almacenamiento de ordenador no extraíbles. El medio de almacenamiento extraíble legible por ordenador puede comprender, sin limitación, una cinta de papel, una cinta magnética, un disco magnético, un disco óptico, un chip de memoria de estado sólido, por ejemplo cinta magnética analógica, discos compactos de memoria de solo lectura (CD-ROM), discos flexibles, memorias USB, tarjetas digitales, tarjetas multimedia y otros. El producto de programa de ordenador puede ser adecuado para que el sistema de ordenadores cargue al menos porciones del contenido del producto de programa de ordenador del almacenamiento secundario, de la ROM, de la RAM y/o de otra memoria no volátil y memoria volátil del sistema de ordenadores. El procesador puede procesar las instrucciones ejecutables y/o las estructuras de datos en parte accediendo al producto de programa de ordenador, por ejemplo leyendo de un disco CD-ROM insertado en una unidad periférica de disco del sistema de ordenadores. Alternativamente, el procesador puede procesar las instrucciones ejecutables y/o las estructuras de datos accediendo remotamente al producto de programa de ordenador, por ejemplo descargando las instrucciones ejecutables y/o las estructuras de datos desde un servidor remoto a través de los dispositivos de conectividad de la red. El producto de programa de ordenador puede comprender instrucciones que promuevan la carga y/o la copia de datos, estructuras de datos, ficheros y/o instrucciones ejecutables del almacenamiento secundario, de la ROM, de la RAM y/o de otra memoria no volátil y memoria volátil del sistema de ordenadores.

En algunos contextos, la señal de banda base y/o la señal implementada en una onda portadora pueden ser denominadas señal transitoria. En algunos contextos, el almacenamiento secundario, la ROM y/o la RAM pueden ser denominados medios no transitorios legibles por ordenador y/o medios de almacenamiento legibles por ordenador. Asimismo, una realización de RAM dinámica de la RAM puede ser denominada medio no transitorio legible por ordenador, porque aunque la RAM dinámica recibe energía eléctrica y es operada según su diseño, por ejemplo durante un periodo de tiempo durante el cual el ordenador está encendido y es operativo, la RAM dinámica almacena la información que se escribe en ella. De manera similar, el procesador puede comprender una RAM interna, una ROM interna, una memoria temporal y/o otros bloques, secciones o componentes internos de almacenamiento no transitorio a los que se puede denominar en algunos contextos medios no transitorios legibles por ordenador o medios de almacenamiento legibles por ordenador.

El documento EP 1 426 862 A2 versa sobre la sincronización del procesamiento de datos dentro de elementos redundantes de procesamiento de un sistema de procesamiento de datos.

El documento EP 1 283 468 A2 da a conocer un aparato según el preámbulo de la reivindicación 1.

Es un objeto de la presente invención mejorar la sincronización entre procesadores doblemente redundantes que ejecutan una aplicación común de control de procesos.

Se logra el objeto de la invención por medio de un controlador de procesos doblemente redundante según las características de las reivindicaciones independientes. en las reivindicaciones dependientes se dan a conocer realizaciones preferentes.

Sumario

En una realización se da a conocer un controlador de procesos doblemente redundante. El controlador de procesos comprende un primer procesador, una primera memoria y una primera instancia de una aplicación de control de procesos almacenada en la primera memoria. El controlador de procesos comprende, además, un segundo procesador, una segunda memoria, y una segunda instancia de la aplicación de control de procesos almacenada en la segunda memoria. Cuando es ejecutada por el primer procesador, la primera instancia de la aplicación de control de procesos escribe en la segunda memoria una primera información de sincronización, lee de la primera memoria una segunda información de sincronización y, cuando la segunda información de sincronización no coincide con la primera información de sincronización después del transcurso de un intervalo predeterminado de tiempo de espera, realiza una función de resincronización. Cuando es ejecutada por el segundo procesador, la segunda instancia de la aplicación de control de procesos escribe en la primera memoria la segunda información de sincronización, lee de la segunda memoria la primera información de sincronización y, cuando la primera información de sincronización no coincide con la segunda información de sincronización después del transcurso de un intervalo predeterminado de tiempo de espera, realiza una función de resincronización.

En una realización no reivindicada se da a conocer un procedimiento de transmisión de un mensaje de datos. El procedimiento comprende la formación de una primera carga útil y de una primera comprobación de redundancia cíclica (CRC) por parte de un primer procesador de un controlador de procesos doblemente redundante y la formación de una segunda carga útil y de una segunda comprobación de redundancia cíclica por parte de un segundo procesador de un controlador de procesos doblemente redundante. El procedimiento comprende, además, la comparación de la primera comprobación de redundancia cíclica con la segunda comprobación de redundancia cíclica por parte del primer procesador y, cuando la primera comprobación de redundancia cíclica y la segunda comprobación de redundancia cíclica coinciden, la transmisión del mensaje de datos que comprende la primera carga útil y la primera comprobación de redundancia cíclica.

En una realización se da a conocer un controlador de procesos. El controlador de procesos comprende un primer módulo. El primer módulo comprende un primer procesador que ejecuta un sistema operativo multitarea en tiempo real y un controlador de comunicaciones de control de enlace de datos de alto nivel (HDLC) acoplado al primer procesador. El primer procesador forma un primer mensaje que comprende una primera carga útil de datos y una primera comprobación de redundancia cíclica (CRC) y transmite el primer mensaje al controlador de comunicaciones de control de enlace de datos de alto nivel. El controlador de comunicaciones de control de enlace de datos de alto nivel recibe el primer mensaje, transmite el primer mensaje a un dispositivo en servicio, calcula una segunda comprobación de redundancia cíclica en función del mensaje y, cuando la segunda comprobación de redundancia cíclica es diferente de la primera comprobación de redundancia cíclica, transmite un mensaje de error al primer procesador.

En una realización se da a conocer un controlador de procesos doblemente redundante. El controlador comprende un primer procesador, una primera memoria, una primera instancia de un sistema operativo multitarea en tiempo real (RTOS) y una primera instancia de una aplicación de control de procesos almacenada en la primera memoria. El controlador comprende, además, un segundo procesador, una segunda memoria, una segunda instancia del sistema operativo multitarea en tiempo real y una segunda instancia de la aplicación de control de procesos almacenada en la segunda memoria. Cuando es ejecutada por el primer procesador en un contexto proporcionado por la primera instancia del sistema operativo multitarea en tiempo real, la primera instancia de la aplicación de control de procesos escribe en la segunda memoria una primera información de sincronización usando una función de sincronización proporcionada por la primera instancia del sistema operativo multitarea en tiempo real, lee de la primera memoria una segunda información de sincronización, realiza una función de resincronización cuando la segunda información de sincronización no coincide con la primera información de sincronización después del transcurso de un intervalo predeterminado de tiempo de espera, y llama a la función de sincronización proporcionada por la primera instancia del sistema operativo multitarea en tiempo real antes de invocar una función de eventos establecidos proporcionada por la primera instancia del sistema operativo multitarea en tiempo real. Cuando es ejecutada por el segundo procesador, la segunda instancia de la aplicación de control de procesos escribe en la primera memoria la segunda información de sincronización usando la función de sincronización proporcionada por la segunda instancia del sistema operativo multitarea en tiempo real, lee de la segunda memoria la primera información de sincronización, y realiza la función de resincronización cuando la primera información de sincronización no coincide con la segunda información de sincronización después del transcurso de un intervalo predeterminado de tiempo de espera, caracterizándose porque la primera información de sincronización comprende uno de varios valores de estado que proporciona una indicación de qué instrucción de la aplicación de control de procesos ha ejecutado recientemente el primer procesador (102a), y la segunda información de sincronización comprende uno de varios valores de estado que proporciona una indicación de qué instrucción de la aplicación de control de procesos ha ejecutado recientemente el segundo procesador (102b).

En una realización no reivindicada se da a conocer un controlador de procesos doblemente redundante. El controlador comprende un primer procesador, una primera memoria y una primera instancia de una aplicación de control de procesos almacenada en la primera memoria. El controlador comprende, además, un segundo procesador, una segunda memoria y una segunda instancia de la aplicación de control de procesos almacenada en la segunda memoria. Cuando es ejecutada por el primer procesador, la primera instancia de la aplicación de control de procesos forma una primera carga útil y una primera comprobación de redundancia cíclica (CRC), compara la primera comprobación de redundancia cíclica con una segunda comprobación de redundancia cíclica determinada por la segunda instancia de la aplicación de control de procesos en función de una segunda carga útil formada por la segunda instancia de la aplicación de control de procesos que se ejecuta en el segundo procesador, y transmite la primera carga útil y la primera comprobación de redundancia cíclica cuando la primera comprobación de redundancia cíclica coincide con la segunda comprobación de redundancia.

En una realización adicional no reivindicada se da a conocer un controlador de procesos doblemente redundante. El controlador comprende un primer procesador, una primera memoria, una primera instancia de un sistema operativo multitarea en tiempo real (RTOS) y una primera instancia de una aplicación de control de procesos almacenada en la primera memoria. El controlador comprende, además, un segundo procesador, una segunda memoria, un reloj, una segunda instancia del sistema operativo multitarea en tiempo real y una segunda instancia de la aplicación de control de procesos almacenada en la segunda memoria. Cuando es ejecutada por el primer procesador en un contexto proporcionado por la primera instancia del sistema operativo multitarea en tiempo real, la primera instancia de la aplicación de control de procesos escribe en la segunda memoria una primera información de sincronización

que indica una sincronización de ciclo de reloj usando una función de sincronización proporcionada por la primera instancia del sistema operativo multitarea en tiempo real. Cuando es ejecutada por el segundo procesador, la segunda instancia de la aplicación de control de procesos ajusta el reloj en función de la primera información de sincronización almacenada en la segunda memoria.

Estas y otras características serán entendidas con mayor claridad a partir de la siguiente descripción detallada tomada junto con las reivindicaciones y los dibujos adjuntos.

Breve descripción de los dibujos

Para una comprensión más completa de la presente divulgación, se hace referencia ahora a la siguiente descripción, tomada en conexión con la descripción detallada y los dibujos adjuntos, en los que números de referencia semejantes representan partes semejantes.

La FIG. 1 es un diagrama de bloques de un sistema de control de procesos según una realización de la divulgación.

La FIG. 2 es un diagrama de bloques de una porción de un módulo procesador según una realización de la divulgación.

La FIG. 3 es un diagrama de flujo de un procedimiento según una realización de la divulgación.

La FIG. 4 es un diagrama de flujo de un procedimiento según una realización de la divulgación.

Descripción detallada

Debe entenderse desde el principio que aunque en lo que sigue se ilustran implementaciones ilustrativas de una o más realizaciones, los sistemas y los procedimientos dados a conocer pueden ser implementados usando cualquier número de técnicas, con independencia de que se conozcan en la actualidad o no estén aún en existencia. La divulgación no debe ser limitada en modo alguno a las implementaciones, las técnicas y los dibujos ilustrativos ilustrados más adelante, sino que pueden ser modificados dentro del alcance de las reivindicaciones adjuntas junto con su alcance total de equivalentes.

En el presente documento se enseña un controlador de procesos doblemente redundante. En una realización, el controlador de procesos es adecuado para su uso en un entorno de control de procesos en tiempo real de alta fiabilidad. Puede usarse el controlador de procesos para monitorizar y controlar varios dispositivos de procesos o dispositivo en servicio, tales como válvulas, bombas, motores, calentadores y otros dispositivos. Puede usarse el controlador de procesos en fábricas, refinerías, fábricas de productos químicos, fábricas de alimentos y otras plantas. Los controladores de procesos pueden causar un daño considerable en el supuesto caso de que fallen. Los controladores fallidos de procesos pueden causar lesiones al personal de la planta. Los controladores fallidos de procesos pueden causar daños en la maquinaria o el material. Se apreciará, además, que es deseable que los controladores de procesos reciban datos de entrada resultado de instrucciones, reciban valores detectados de parámetros y/o propiedades, determinen salidas de control apropiadas y transmitan estas salidas de control puntualmente y en momentos debidamente planificados.

El controlador de procesos comprende dos módulos, siendo cada uno adecuado para proporcionar la función de control de procesos. Durante la operación, un primer módulo opera como módulo primario que recibe entradas de sensores procedentes de los dispositivos de proceso, transmite las salidas de control a los dispositivos de proceso y transmite mensajes a interfaces hombre-máquina (HMI), estaciones de trabajo y/o dispositivos automatizados de control de nivel superior según una aplicación de control y/o un programa de ordenador de control. El segundo módulo opera como módulo de copia que recibe las mismas entradas de sensor procedentes de los dispositivos de proceso, determina pero no transmite salidas de control a los dispositivos de proceso y determina pero no transmite mensajes a las HMI, a las estaciones de trabajo y/o a los dispositivos automatizados de control de nivel superior según la misma aplicación de control. El primer módulo y el segundo módulo pueden ejecutar instancias separadas de la misma aplicación de control. En el supuesto caso de que el primer módulo experimentara un fallo o un error, es deseable que este fallo sea detectado y que el segundo módulo asuma el papel del módulo primario de inmediato.

Para soportar un intercambio sin interrupciones del papel en caso de fallo, es deseable, además, que el primer módulo y el segundo módulo ejecuten las mismas instrucciones de la aplicación de control sustancialmente al mismo tiempo y que sigan el mismo recorrido de ejecución por la aplicación de control, con la excepción de instrucciones selectivas que solo ejecuta el módulo primario o que solo ejecuta el módulo de copia. Los dos módulos ejecutan instrucciones de sincronización en puntos designados en la secuencia de instrucciones de la aplicación de control. Cuando lleva a cabo una instrucción de sincronización, el primer módulo escribe información en una ubicación predefinida de memoria asociada con el segundo módulo que identifica el estado de sincronización del primer módulo, y el segundo módulo escribe información en una ubicación predefinida de memoria asociada con el primer módulo que identifica el estado de sincronización del segundo módulo. Si el estado de sincronización del módulo hermano no coincide con el estado de sincronización del módulo titular dentro de un intervalo predefinido de tiempo de espera, el módulo titular declara un error de sincronización y ejecuta una rutina de recuperación. Si coinciden los

estados de sincronización, cada módulo sigue ejecutando instrucciones de control. Al insertar instrucciones de sincronización en puntos apropiados en el código, puede mantenerse sincronizada la ejecución separada de la misma aplicación de control por parte de los dos módulos, dentro de los límites de diseño.

Cada módulo comprende un reloj que gobierna el ritmo de la ejecución de instrucciones por parte del módulo. En una realización, la aplicación de control ejecutada por el módulo en el papel de copia determina una diferencia horaria entre el módulo de copia y el módulo primario, basada en la operación de sincronización, y ajusta el reloj del módulo de copia para que esté alineado con el reloj del módulo primario. Este ajuste de reloj permite definir el intervalo de tiempo de espera de la sincronización en una duración más breve y contribuye a reducir los tiempos de espera de sincronización, aumentado con ello la eficiencia de procesamiento de al menos uno de los módulos.

Cuando se da salida a información destinada a las HMI, las estaciones de trabajo y/o dispositivos automatizados de control de nivel superior, el módulo primario y el módulo de copia crean cada uno una carga útil de datos y calculan una comprobación de redundancia cíclica (CRC) de la carga útil de datos. En algunos contextos, puede denominarse a la carga útil de datos cuerpo de mensaje. El módulo de copia envía al módulo primario el valor de CRC que ha sido calculado. Si el valor de CRC calculado por el módulo de copia coincide con el valor de CRC calculado por el módulo primario, el módulo primario transmite un mensaje que comprende tanto la carga útil de datos como el valor de CRC a las HMI, las estaciones de trabajo y/o dispositivos automatizados de control de nivel superior. Cuando los valores de CRC no coinciden, el módulo primario ejecuta una rutina de recuperación.

En una realización, el módulo primario y el módulo de copia comprenden cada uno un dispositivo lógico programable complejo (CPLD) que ejecuta la aplicación de control y un controlador de control de enlace de datos de alto nivel (HDLC) que puede transmitir mensajes a dispositivos de control y recibirlos de los mismos, por ejemplo cuando el módulo titular está ejecutándose en el papel de módulo primario. El CPLD determina una carga útil de datos y una primera CRC de la carga útil de datos y envía al controlador de HDLC un mensaje que comprende la carga útil de datos y la CRC. El controlador de HDLC transmite el mensaje al dispositivo de control apropiado y recibe concurrentemente el mismo mensaje. El controlador de HDLC calcula una segunda CRC de la carga útil de datos que transmitió, y si la primera CRC y la segunda CRC no coinciden, el controlador de HDLC envía un mensaje de error al CPLD. Este procedimiento puede promover que el CPLD identifique un error en la salida del mensaje al dispositivo de control y retransmita el mensaje más rápidamente de lo que lo haría si el controlador de HDLC simplemente se desconectara por tiempo cuando el dispositivo de control no devolviera puntualmente un acuse de recibo al controlador de HDLC.

La divulgación enseña una funcionalidad de sincronización que no depende de máquinas de estado basadas en un soporte físico. En vez de ello se emplea un procedimiento de sincronización a base de mensajes para mantener iguales todas las operaciones en dos entornos separados de un sistema operativo, por ejemplo en dos módulos procesadores separados. Los dos módulos procesadores intercambian una etapa de sincronización basada en mensajes que puede ser ejecutada en los momentos críticos en el código del sistema operativo y la aplicación en un enlace estándar Gigabit Ethernet entre los dos módulos.

La divulgación enseña, además, la implementación de la funcionalidad de sincronización usando un soporte lógico que permite el intercambio de varias CRC del ámbito de la pila antes de la transmisión de mensajes, por ejemplo la transmisión de un mensaje a un componente de control de mayor nivel y/o a un dispositivo en servicio. Esto permite que el soporte lógico detecte cualquier diferencia en las configuraciones del proceso sin comparación del soporte físico. Estas CRC pueden ser ya parte integral de la metodología de pila común, pero en el presente documento se usan únicamente para la verificación de la capa de aplicación. La divulgación enseña el uso de estas CRC para verificar la equivalencia de contenido en los dobles controladores activos. Los dos miembros de un par de módulos verifican un contexto correcto al examinar ambos la CRC (que expresa el contenido) de la transmisión del otro, lo que permite que un soporte físico genérico logre la misma funcionalidad que un soporte físico construido con un fin dedicado con lógica de comparación. La capacidad de usar un soporte físico genérico puede proporcionar varias ventajas en algunas circunstancias, por ejemplo promover diseños de soporte físico más flexibles y mantenibles, promover el uso de una variedad más amplia de componentes disponibles comercialmente y otras ventajas.

La divulgación enseña, además, el ajuste dinámico de relojes para reducir el sesgo acumulado del reloj. En general, puede resultar difícil mantener la sincronización de dos módulos con los relojes de los sistemas operativos produciendo un tiempo de espera en intervalos diferentes definidos por temporizadores internos. Los sistemas anteriores creaban circuitos especiales de sincronización para derivar el reloj de los sistemas operativos de ambos módulos e interrumpir la señal de un único origen para eliminar cualquier sesgo. La metodología enseñada en el presente documento ajusta dinámicamente la interrupción de tiempo de espera del reloj en uno de los módulos procesadores para que coincida con el periodo del otro módulo procesador. Esto se realiza a través de una modificación determinista del tiempo de espera por medio de los mensajes de sincronización.

Pasando ahora a la FIG. 1, se describe un sistema 100. El sistema 100 comprende un primer módulo procesador 102a, un segundo módulo procesador 102b, un dispositivo 104 en servicio, un bus A 106a de entrada/salida (ES) de proceso, un bus B 106b de ES de proceso y una placa base 108. En algunos contextos, el primer módulo procesador 102a y el segundo módulo procesador 102b y la placa base 108 pueden ser denominados controlador

109 de procesos doblemente redundante y/o procesador doblemente redundante. Alternativamente, los módulos procesadores 102 y la placa base 108 pueden ser denominados procesador de control, controlador de unidades o controlador. En una realización, los módulos procesadores 102 se comunican con un sistema 112 de control distribuido (DCS) a través de una red 110. El DCS 112 puede comprender una o más estaciones 114 de trabajo y un sistema 116 de ordenadores. Aunque en la FIG. 1 el controlador 109 de procesos doblemente redundante y el DCS 112 son mostrados por separado para mover la facilidad de comprensión y centrar la atención en el controlador 109 de procesos doblemente redundante, se entiende que el DCS 112 podría alternativamente ser abstraído para que comprenda el controlador 109 de procesos doblemente redundante.

En una realización, la red 110 proporciona vías duales de comunicación desde el primer módulo procesador 102a al DCS 112 para que si una de las vías de comunicación no está disponible por alguna razón, por ejemplo debido a un fallo, el primer módulo procesador 102a pueda seguir comunicándose con el DCS 112 por la otra vía de comunicación. Asimismo, la red 110 puede proporcionar vías duales de comunicación desde el segundo módulo procesador 102b al DCS 112. En una realización, la red 110 pueden proporcionar las vías duales de comunicación al menos en parte usando varios conmutadores para proporcionar una malla de conmutadores y/o una malla de vías de comunicación. La red 110 puede comprender, además, uno o más divisores de señal para garantizar que un mensaje transmitido al módulo procesador 102 operando en el modo primario también sea transmitido al módulo procesador 102 operando en el modo de copia. La red 110 puede enlazar los módulos procesadores 102 con el DCS 112 a través de cualquiera de enlaces inalámbricos, enlaces alámbricos y/o enlaces de fibra. En una realización, la red 110 puede ser cualquiera de una red pública, una red privada y/o una combinación de las mismas.

Los módulos procesadores 102 controlan el dispositivo 104 en servicio y monitorizan uno o más parámetros del dispositivo 104 en servicio a través de buses 106 de ES de proceso. Los buses 106 de ES de proceso proporcionan vías duales de comunicación desde los módulos procesadores 102 al dispositivo 104 en servicio para que si una de las vías de comunicaciones no está disponible por alguna razón, por ejemplo debido a una conexión suelta, un hilo o cable cortado o un enlace inalámbrico interrumpido, los módulos procesadores 102 pueden seguir comunicándose con el dispositivo 104 en servicio por la otra vía de comunicación. El módulo procesador 102 que opera en el modo primario monitoriza uno o más parámetros del dispositivo 104 en servicio y, a la vez, envía instrucciones de control al dispositivo 104 en servicio a través de los buses 106 de ES de proceso. El módulo procesador 102 que opera en el modo de copia monitoriza uno o más parámetros del dispositivo 104 en servicio y también monitoriza las instrucciones de control enviadas por el módulo procesador 102 que esté operando en el modo primario. En una realización, el módulo procesador 102 que opera en el modo de copia no envía instrucciones de control al dispositivo 104 en servicio.

Aunque en la FIG. 1 aparece marcado un solo dispositivo 104 en servicio, se entiende que los módulos procesadores 102 pueden controlar y monitorizar varios dispositivos 104 en servicio. Los dispositivos 104 en servicio comprenden cualquiera de varios equipos de plantas, equipos de procesos, equipos de fábricas y otros equipos. Los dispositivos 104 en servicio pueden ser denominados dispositivos y/o dispositivos de proceso en algunos contextos. Los dispositivos 104 en servicio pueden comprender un componente lógico acoplado a uno o más dispositivos electromecánicos, por ejemplo una válvula, una bomba, un motor, un calentador, un dispositivo de transporte y otros dispositivos. El componente lógico de los dispositivos 104 en servicio puede, además, estar acoplado a uno o más sensores para detectar un parámetro operativo del dispositivo electromecánico o de un parámetro físico con el que interactúa el dispositivo electromecánico, por ejemplo una presión, una temperatura, una densidad u otra característica u otra propiedad. En algunos casos, el dispositivos 104 en servicio puede comprender un componente lógico acoplado a uno o más sensores, pero no a ningún dispositivo electromecánico.

En combinación con los módulos procesadores 102 y el DCS 112, los dispositivos 104 en servicio pueden proporcionar de forma conjunta un proceso automatizado, tal como un proceso de producción química, un proceso de refinado de petróleo, un proceso de fabricación de vidrio, un proceso de producción de alimentos, un proceso de generación de energía eléctrica y/u otros procesos. En una realización, los módulos procesadores 102 controlan a los dispositivos 104 en servicio según una instrucción proporcionada por el sistema 116 de ordenadores y transmiten valores de parámetros al ordenador 116 y opcionalmente a las estaciones 114 de trabajo. El sistema 116 de ordenadores puede ejecutar una aplicación de alto nivel de control de procesos o monitorizar y controlar varios procesadores doblemente redundantes que, a su vez, controlan y monitorizan varios dispositivos 104 en servicio. En algunos contextos, el sistema 116 de ordenadores puede ser denominado dispositivo automatizado de control de nivel superior.

Cada uno de los procesadores 102 ejecuta una copia del mismo programa de ordenador y/o de la aplicación de control. Dicho en otras palabras, cada uno de los procesadores 102 ejecuta una instancia del mismo programa de ordenador. En un modo operativo, uno de los procesadores 102 se ejecuta en un modo primario, y el otro procesador 102 se ejecuta en un modo de copia. La descripción siguiente puede dar por sentado que el primer módulo procesador 102a se ejecuta en el modo primario y que el segundo módulo procesador 102b se ejecuta en el modo de copia, pero se entiende que los papeles de los módulos procesadores 102 pueden invertirse. En particular, en una condición de fallo, el programa ejecutado por los módulos procesadores 102 puede identificar el fallo y coordinar la permuta de los papeles primario/de copia entre los módulos procesadores 102, según se expondrá adicionalmente más adelante en el presente documento.

El módulo procesador 102 que se ejecuta en el modo primario transmite instrucciones al dispositivo 104 en servicio y transmite valores de parámetros al DCS 112. El módulo procesador 102 que se ejecuta en el modo de copia recibe las instrucciones transmitidas por el módulo procesador 102 que ejecuta en el modo primario y recibe los valores de parámetros transmitidos por el módulo procesador 102 que ejecuta en el modo primario. Ambos módulos procesadores 102 reciben las instrucciones transmitidas por el DCS 112 y reciben los valores de parámetros transmitidos por el dispositivo 104 en servicio.

El sistema 116 de ordenadores puede comprender uno o más ordenadores que se ejecutan una aplicación de alto nivel de control de procesos que interactúa con el controlador 109 de procesos doblemente redundante. En lo que sigue se describen con detalle sistemas de ordenadores. El sistema 116 de ordenadores puede proporcionar entradas de alto nivel de control al controlador 109 de procesos doblemente redundante, por ejemplo una entrada de control del punto de referencia de la temperatura de un horno. El controlador 109 de procesos doblemente redundante puede controlar el dispositivo 104 en servicio, por ejemplo varios termistores que modulan la energía eléctrica consumida por elementos calentadores resistivos y, por ende, el calor emitido por los elementos calentadores resistivos, en función de la entrada de control de alto nivel del punto de referencia de la temperatura de un horno y en función de los valores de temperatura del sensor del horno recibidos del dispositivo 104 en servicio. En una realización, el sistema 116 de ordenadores puede ser un sistema de ordenadores de alta fiabilidad, y el enlace de comunicación entre el sistema 116 de ordenadores y la red 110 puede ser proporcionado por vías duales de comunicación.

Las estaciones 114 de trabajo también pueden ser implementadas como ordenadores. Una de las estaciones 114 de trabajo puede proporcionar una funcionalidad de interfaz hombre-máquina (HMI). Las estaciones 114 de trabajo promueven la monitorización del proceso controlado y/o de procesos por parte de los usuarios y/o de operarios de fábricas. Las estaciones 114 de trabajo pueden promover, además, que los usuario y/o los operarios de fábricas transmitan datos de entrada al sistema 116 de ordenadores para seleccionar modos operativos del proceso o los procesos controlados y/o para introducir valores indicados de algunos parámetros de procesos. Una o más de las estaciones 114 de trabajo pueden comunicarse con el controlador 109 de procesos doblemente redundante independientemente del sistema 116 de ordenadores, por ejemplo en un modo de mantenimiento de operación y/o en un modo de prueba de operación.

De forma deseable, los módulos procesadores 102 ejecutan las mismas instrucciones del programa de control sustancialmente al mismo tiempo, por ejemplo dentro de una diferencia predefinida de tiempos de ejecución. Esto puede ser denominado ejecución síncrona de instrucciones por parte de los módulos procesadores 102 y/u operación síncrona de los módulos procesadores 102. Se entiende que aunque en algunos contextos pueda usarse el término “síncrono” de modo que signifique incidencia exactamente simultánea de eventos, “síncrono”, según es usado en el presente documento, significa sustancialmente simultáneo dentro de un umbral predefinido de diferencia horaria, por ejemplo dentro de aproximada 2 ms de diferencia horaria o de desfase temporal. Al ejecutar las mismas instrucciones de forma síncrona, es decir, dentro de un umbral predefinido de diferencia horaria, puede reducirse la dificultad de que el procesador primario se recupere de un error y/o de la permuta de papeles entre los módulos procesadores 102.

Para promover la operación síncrona, el programa de control incluye varias instrucciones de sincronización distribuidas entre las instrucciones del programa de control. Cuando uno de los módulos procesadores 102 ejecuta una instrucción de sincronización del programa de control —recordando que ambos módulos procesadores 102 ejecutan una instancia del mismo programa de control—, escribe un mensaje de sincronización en una memoria del otro módulo procesador 102 y aguarda a leer un mensaje correspondiente de sincronización escrito en su propia memoria por el otro módulo procesador 102 antes de seguir el procesamiento de instrucciones subsiguientes. Si cualquiera de los dos módulos procesadores 102 no lee el mensaje de sincronización previsto en su memoria dentro de un periodo de tiempo predeterminado, por ejemplo 2 ms, el módulo procesador 102 titular lleva a cabo una acción de recuperación de un fallo. En algunos contextos, el periodo predeterminado de tiempo puede ser denominado intervalo predeterminado de tiempo de espera.

En una realización, el primer módulo procesador 102a comprende una primera unidad central 118a de procesador (CPU), una primera interconexión 119a, un primer reloj 120a, una primera memoria 122a, y una primera matriz 126a de puertas programables *in situ* (FPGA). La primera matriz 126a de puertas programables *in situ* comprende y/o implementa un primer controlador 128a de enlace de datos de alto nivel (HDLC). Se entiende que una matriz de puertas programables *in situ* es una especie de dispositivo lógico programable complejo (CPLD). En otra realización puede usarse otra especie de CPLD que no sea una FPGA en lugar de la primera FPGA 126a. Alternativamente, puede usarse un dispositivo lógico distinto de un CPLD en lugar de la FPGA 126a, por ejemplo un circuito integrado para aplicaciones específicas (ASIC), un microcontrolador, un microprocesador u otro componente lógico electrónico. En una realización, la funcionalidad de la FPGA 126a y del controlador de HDLC 128a puede ser implementada en componentes separados en vez de la forma integrada descrita en el presente documento. La primera memoria 122a puede comprender una primera ubicación 124a de la memoria del estado de sincronización. El primer módulo procesador 102a puede ser implementado en una placa de circuitos, en dos placas de circuitos o en un número mayor de placas de circuitos, y estas una o más placas de circuitos pueden estar encerradas en un paquete tal como una caja de equipo electrónico.

En una realización, el segundo módulo procesador 102b comprende una segunda unidad central 118b de procesador, una segunda interconexión 119b, un segundo reloj 120b, una segunda memoria 122b, una segunda FPGA 126b y un segundo controlador de HDLC 128b. La segunda memoria 122b puede comprender una segunda ubicación 124b de la memoria del estado de sincronización. El segundo módulo procesador 102b puede ser implementado en un número cualquiera de placas de circuitos que pueden estar encerradas en un paquete tal como una caja de equipo electrónico. Los comentarios sobre la implementación de realizaciones alternativas del primer módulo procesador 102a se aplican igualmente al segundo módulo procesador 102b, pero no se repiten aquí en aras de la brevedad. En una realización, la placa base 108 puede proporcionar una estructura mecánica para sujetar y montar los módulos procesadores 102, para sujetar los conectores y para sostener componentes de soporte, por ejemplo una fuente de alimentación u otros componentes.

En una realización, ambos módulos procesadores 102 ejecutan un sistema operativo multitarea en tiempo real (RTOS), y la aplicación de control que ejecutan ambos módulos procesadores 102 se ejecuta en un contexto proporcionado por el RTOS. Por ejemplo, las unidades centrales 118 de procesamiento ejecutan instanciaciones de la aplicación de control en un sistema operativo multitarea en tiempo real que también se ejecuta en las unidades centrales 118 de procesamiento. Algunos RTOS disponibles comercialmente incluyen el Nucleus RTOS, vendido por la División de Sistemas Embebidos de Mentor Graphics, de Wilsonville, Oregón; VxWorks, vendido por Wind River Systems, de Alameda, California; uno o más RTOS vendidos por Green Hills Software, de Santa Bárbara, California; y otros. Los RTOS también pueden ser desarrollados a medida por una organización cuando desarrolla la aplicación de control y el controlador 109 de procesos doblemente redundante. Sin limitación, generalmente cabe esperar de un RTOS multitarea que proporcione una planificación determinista priorizada de tareas, de modo que una tarea de mayor prioridad que esté lista para su procesamiento no aguardará a que complete su procesamiento una tarea de menor prioridad. En una realización, un RTOS disponible comercialmente puede ser ampliado para que proporcione una llamada de instrucción de sincronización para su uso por parte del programa de control. Alternativamente, puede desarrollarse una rutina de soporte lógico que promueva la funcionalidad de generación y transmisión de mensajes de sincronización, posiblemente usando una o más llamadas al sistema del RTOS para completar la transmisión del mensaje de sincronización. Esta rutina de soporte lógico puede ser escrita de tal modo que pueda ser invocada en cualquiera de varios módulos, tareas, subrutinas y/u otros componentes de la aplicación de control.

Cuando el primer módulo procesador 102a ejecuta una instrucción de sincronización, escribe un primer mensaje de sincronización en la segunda ubicación 124b de la memoria del estado de sincronización de la segunda memoria 122b asociada con el segundo módulo procesador 102b. Este primer mensaje de sincronización identifica un estado de sincronización del primer módulo procesador 102a, un valor enumerado correspondiente a uno de varios valores diferentes de estados de sincronización. En algunos contextos, se puede denominar valor de estado a un valor de estado de sincronización, y se puede denominar estado a un estado de sincronización. En algunos contextos, el valor de estado de sincronización y posiblemente otros datos pueden ser denominados información de sincronización.

Otros datos pueden comprender un número de secuencia o un número de identidad de instrucciones de sincronización. Dado que la aplicación de control puede comprender muchas instrucciones de sincronización, por ejemplo cientos de instrucciones de sincronización o miles de instrucciones de sincronización, identificar sin más un valor de estado puede no ubicar suficientemente el punto de procesamiento de la aplicación de control. La información que combina tanto un valor de estado como un número de secuencia u otra información identificativa puede ser útil para identificar unívocamente un punto de ejecución en la aplicación de control.

El valor de estado que el primer módulo procesador 102a escribe en la segunda ubicación 124b de la memoria del estado de sincronización proporciona una indicación de qué instrucción de la aplicación común de control ha ejecutado recientemente el primer módulo procesador 102a, y el segundo módulo procesador 102b puede analizar esa indicación para determinar si los módulos procesadores 102 están en sincronización. En una realización, puede enviarse un mensaje de sincronización cuando ocurre un evento de ciclo de reloj del RTOS, y, en este caso, l información de sincronización puede identificar el evento de ciclo de reloj del RTOS en vez de un estado de sincronización. En otra realización, el evento de ciclo de reloj del RTOS puede ser gestionado y/o tratado como uno entre varios valores del estado de sincronización.

En una realización, hay ocho valores de estado o valores de sincronización diferentes, pero en otra realización puede haber menos o más valores de estado diferentes. Un primer valor de estado puede corresponder a una interrupción del reloj, por ejemplo una interrupción de reloj generada por el reloj 120 y/o un evento de ciclo de reloj del RTOS. Esto también puede ser denominado valor de estado de ciclo de sistema operativo o ciclo de SO. Un segundo valor de estado puede corresponder a una interrupción del bus de entrada/salida de proceso, por ejemplo asociado con una interrupción o con que se reciban datos de entrada del dispositivo 104 en servicio. Un tercer valor de estado puede corresponder a una conmutación de tareas del sistema operativo en tiempo real. Un cuarto valor de estado puede corresponder al envío de un mensaje desde el módulo procesador primario 102 al DCS 112 y/o al sistema 116 de ordenadores. Un quinto valor de estado puede corresponder a una sincronización horaria externa. Un sexto valor puede corresponder a un evento de recepción de un mensaje procedente del DCS 112. Un séptimo valor de estado puede corresponder a un intercambio de mensajes. Un octavo valor de estado puede corresponder a

una solicitud de resincronización de los módulos procesadores 102, a lo que se puede denominar solicitud de casamiento.

Asimismo, cuando el segundo módulo procesador 102b ejecuta una instrucción de sincronización, escribe en la primera ubicación 124a de la memoria del estado de sincronización de la primera memoria 122a asociada con el primer módulo procesador 102a un segundo mensaje de sincronización que identifica un estado de sincronización del segundo módulo procesador 102b. El valor de estado que el segundo módulo procesador 102b escribe en la primera ubicación 124a de la memoria del estado de sincronización proporciona una indicación de qué instrucción de la aplicación común de control ha ejecutado recientemente el segundo módulo procesador 102b, y el primer módulo procesador 102a puede analizar esa indicación para determinar si los módulos procesadores 102 están en sincronización.

Después de que cualquiera de los dos módulos procesadores 102 escribe el mensaje de sincronización en la ubicación 124 de la memoria del estado de sincronización de su módulo procesador 102 correlacionado, aguarda un periodo predefinido de tiempo que el valor de estado en su propia ubicación 124 de la memoria del estado de sincronización coincida con lo que escribió. Si no se determina una coincidencia del valor de estado antes de que expire el periodo predefinido de tiempo o el intervalo predefinido de tiempo de espera, el módulo procesador titular 102 puede declarar que el módulo procesador 102 correlacionado está desincronizado y puede iniciar una rutina de recuperación para resincronizar los dos módulos procesadores 102. La expiración del periodo predefinido de tiempo o el intervalo predefinido de tiempo de espera pueden ser denominados en algunos contextos expiración por tiempo o expiración de la sincronización por tiempo.

El siguiente ejemplo de código muestra una llamada de sincronización, a la que se puede denominar (ftsync_i()), antes de que el sistema operativo efectúe una llamada de eventos establecidos del SO. En una realización, la función permite la operación de sincronización descrita en el presente documento. Un experto en la técnica apreciará que el código en cuestión puede ser escrito en cualquier lenguaje de programación adecuado. Además, un experto en la técnica apreciará inmediatamente que el código puede ser escrito de varias maneras diferentes, que tienen distintas variaciones diferentes del ejemplo explícito de código proporcionado a continuación.

```
STATUS EVC_Set_Events(NU_EVENT_GROUP *event_group_ptr, UNSIGNED events,
OPTION operation)
{
   R1 EV_GCB *event_group; /* Puntero al bloque de control de eventos */
   R2 EV_SUSPEND *suspend_ptr; /* Puntero al bloque de suspensión */
   R3 EV_SUSPEND *next_ptr; /* Puntero a la suspensión siguiente */
   R4 EV_SUSPEND *last_ptr; /* Puntero al último bloque de suspensión */
   UNSIGNED consume; /* Banderas de eventos a consumir */
   UNSIGNED compare; /* Variable de comparación de eventos */
   INT preempt; /* Bandera requerida de prioridad */
   NU_SUPERV_USER_VARIABLES /* Conmutar al modo supervisor */
   NU_SUPERVISOR_MODE();
   /* Mover el puntero al grupo de eventos de entrada al puntero interno. */
   event_group = (EV_GCB *) event_group_ptr;
   /* Sincronizar con el compañero de FT si están casados e inhabilitar las
   interrupciones. */
   ftsync_i(OS_SID);
   /* Proteger contra el acceso simultáneo al grupo de eventos. */
   TCT_System_Protect();
   /* Realizar la operación especificada en las benderas de eventos actuales del
   grupo. */
   if (operation & EV_AND)
      /* Y de los eventos especificados con los eventos actuales. */
      event_group -> ev_current_events = event_group -> ev_current_events & events;
   else
      /* O de los eventos especificados con los eventos actuales. */
      event_group -> ev_current_events = event_group -> ev_current events | events;
```

Pasando ahora a la FIG. 4, se describe un procedimiento 250. El procedimiento 250 representa de manera abstracta el código ejemplar descrito arriba. En una realización se invoca o se ejecuta el procedimiento 250 cuando se invoca el procedimiento EVC_Set_Events( ) anteriormente descrito. En una realización puede invocarse el procedimiento 250 antes que el procedimiento EVC_Set_Events( ) y/o antes de realizar una llamada al procedimiento EVC_Set_Events( ). En el bloque 252, se realiza una conmutación el modo supervisor. En el bloque 254, se mueve el punto al grupo de eventos al puntero interno, por ejemplo al puntero event_group. En el bloque 256, se lleva a cabo la sincronización con el otro módulo procesador llamando una subrutina o función de sincronización. En una realización, la subrutina de sincronización puede llamarse ftsync_i( ). En el bloque 258, se protege contra el acceso simultáneo al grupo de eventos, por ejemplo usando un mecanismo de acceso exclusivo, tal como un semáforo u

otro mecanismo. En el bloque 260, se actualizan las banderas de eventos actuales del grupo de eventos. A continuación, termina el procedimiento 250. En la FIG. 4 y en el procedimiento 250 se muestra que una sincronización entre los módulos procesadores puede efectuarse antes de que se realicen llamadas señaladas del sistema operativo, por ejemplo antes de la actualización de las banderas de eventos actuales u otras operaciones tales como el envío de una instrucción de control al dispositivo 104 en servicio.

Con referencia nuevamente a la FIG. 1, si el primer módulo procesador 102a alcanza una instrucción dada de sincronización en la aplicación común de control antes que el segundo módulo procesador 102b, el primer módulo procesador 102a escribe en la segunda ubicación 124b de la memoria del estado de sincronización un mensaje de sincronización que comprende su valor de estado, lee el valor de estado almacenado en la primera ubicación 124a de la memoria del estado de sincronización, determina que los valores de estado en las ubicaciones 124 de la memoria del estado de sincronización no coinciden y aguarda hasta que el valor de estado almacenado en la primera ubicación 124a de la memoria del estado de sincronización sea revisado para corresponder con el valor de estado que escribió en la segunda ubicación 124b de la memoria del estado de sincronización. El primer módulo procesador 102a puede leer reiteradamente en la primera ubicación 124a de la memoria del estado de sincronización y efectuar la comparación. Alternativamente, el primer módulo procesador 102a puede leer periódicamente en la primera ubicación de la memoria del estado de sincronización, por ejemplo cada 100 µs, cada 500 µs, cada 1 ms, o algún otro intervalo periódico, y llevar a cabo la comparación.

Si el valor de estado escrito por el primer módulo procesador 102a y el valor de estado leído por el primer módulo procesador 102a de la primera ubicación 124a de la memoria del estado de sincronización coinciden antes de la expiración del periodo predefinido de tiempo, el primer módulo procesador 102a sigue ejecutando las instrucciones subsiguientes de la aplicación de control. Sin embargo, si el primer módulo procesador 102a experimenta una tiempo de espera de sincronización, el primer módulo procesador 102a puede iniciar una rutina de recuperación para resincronizar con el segundo módulo procesador 102b. La rutina de recuperación puede ser denominada en algunos contextos procedimiento de resincronización o función de resincronización. En otra circunstancia, el segundo módulo procesador 102b puede alcanzar una instrucción dada de sincronización en la aplicación común de control antes que el primer módulo procesador 102a, y entonces el comportamiento del primer módulo procesador 102a descrito más arriba sería realizado en su lugar por el segundo módulo procesador 102b.

En una realización, un procedimiento de resincronización o una función de resincronización puede comprender la interrupción breve del procesamiento de control y la copia de todo el contexto del primer módulo procesador 102a al segundo módulo procesador 102b, lo que puede ser denominado en algunos contextos recasamiento en caliente. El contexto puede incluir valores de registros y/o valores de pila mantenidos por el primer módulo procesador 102a. Alternativamente, o además, el procedimiento de resincronización puede comprender la permuta de papeles entre el primer módulo procesador 102a y el segundo módulo procesador 102b, de modo que el módulo procesador 102 que operaba anteriormente en modo primario realice una transisicón a una operación en el modo de copia, y que el módulo procesador 102 que anteriormente operaba en el modo de copia realice una transición a la operación en el modo primario.

Puede decirse que la compartición de valores de estado descrita más arriba implementa un secuenciador de estados del procesador doblemente redundante. En algunos contextos, puede decirse que la aplicación de control comprende o implementa un secuenciador de estados. La función secuenciadora de estados de la aplicación de control rastrea el estado del módulo procesador titular 102 y promueve el mantenimiento de la sincronización con el correspondiente módulo procesador 102.

En una realización, el primer módulo procesador 102a escribe el primer mensaje de sincronización en la segunda ubicación 124b de la memoria del estado de sincronización en la segunda memoria 122b y el segundo módulo procesador 102b escribe el segundo mensaje de sincronización en la primera ubicación 124a de la memoria del estado de sincronización en la primera memoria 122a a través de un enlace de comunicaciones Ethernet de un gigabit (1 G) que proporciona entre los módulos procesadores 102 el sistema 100. Por ejemplo, la primera interconexión 119a en el primer módulo procesador 102a proporciona un primer puerto estándar de comunicaciones que se acopla con un segundo puerto estándar de comunicaciones en la segunda interconexión 119b en el segundo módulo procesador 102b para proporcionar un enlace de comunicaciones entre los módulos procesadores 102 para promover la función de sincronización. En otra realización, se proporciona la transmisión de mensajes de sincronización usando un enlace de comunicaciones diferente. En una realización, los mensajes de sincronización pueden ser formateados como una trama Ethernet que tiene aproximadamente 13 octetos de datos. El enlace de comunicaciones puede implementarse con un acoplamiento de transformador para promover el aislamiento eléctrico entre los dos módulos procesadores 102.

En una realización, el periodo predeterminado de tiempo o el intervalo predefinido de tiempo de espera, que también puede ser denominado periodo de tiempo de espera de sincronización, puede estar en el intervalo de 50 µs a 50 ms. Alternativamente, en una realización, el periodo de tiempo de espera de sincronización puede estar en el intervalo de 500 µs a 10 ms. En una realización, el periodo de tiempo de espera de sincronización puede ser de aproximadamente 2 ms. Alternativamente puede emplearse otro periodo de tiempo de espera de sincronización. En combinación con la presente divulgación, un experto en la técnica seleccionará fácilmente un periodo

predeterminado de tiempo efectivo para sincronizar la ejecución de instrucciones del programa de control entre los módulos procesadores 102. Una consideración en la determinación del periodo de tiempo de espera de sincronización puede ser la frecuencia o la granularidad de los relojes 120 y/o la deriva entre los relojes 120.

Se entiende que cabe esperar que los relojes 120 tengan una deriva mutua: cabe esperar que un reloj 120 opere más rápido que el otro reloj 120, aunque sea solamente ligeramente más rápido. Dado que los módulos procesadores 102 ejecutan instrucciones a un ritmo fijados por sus respectivos relojes 120, la ejecución de instrucciones del módulo procesador 102 que tiene el reloj 120 más lento irá cada vez más a la zaga en la ejecución de instrucciones del otro módulo procesador 102 hasta que ocurre la tiempo de espera de sincronización. Uno de los resultados de la resincronización de los módulos procesadores 102 puede ser la puesta a cero del desfase entre las ejecuciones de instrucciones de los dos módulos procesadores 102. Sin embargo, después del procedimiento de resincronización, el módulo procesador 102 que tiene el reloj 120 más lento irá cada vez más a la zaga en la ejecución de instrucciones del otro módulo procesador 102 hasta que se repite la tiempo de espera de sincronización, y este ciclo se repetirá. En general, es poco deseable que la resincronización se repita periódicamente en ausencia de verdaderas condiciones de error, porque durante la resincronización el procesador de control doblemente redundante no está ejerciendo control sobre los dispositivos 104 en servicio. Esto es análogo a un coche que ruede calle abajo mientras el conductor quita las manos del volante durante un intervalo de tiempo.

En una realización, si la resincronización sucede demasiado a menudo, los dos módulos procesadores 102 ejecutan un procedimiento de recuperación, estableciendo, por ejemplo, que el módulo procesador 102 que operaba anteriormente en el modo de copia opere en el modo primario y estableciendo que el módulo procesador 102 que operaba anteriormente en el modo primario opere en el modo de copia. El procedimiento de recuperación puede comprender, además, la realización de diagnósticos en el reloj del módulo procesador 102 que se ha determinado que es inexacto. El programa de control puede invocar el procedimiento de recuperación cuando ocurre un número predefinido de resincronizaciones dentro de una ventana temporal predefinida. Por ejemplo, el programa de control puede invocar el procedimiento de recuperación cuando ocurren más de 5 resincronizaciones en un intervalo temporal de un minuto.

En una realización, el procesador 109 de control doblemente redundante compensa automáticamente la deriva del reloj haciendo que el módulo procesador 102 que está operando en el papel de copia ajuste periódicamente su reloj 120 para que se alinee con el reloj 120 del módulo procesador 102 que opera en el papel primario, por ejemplo ajustando su reloj 120 compensar un desfase entre el reloj 120 de copia y el reloj primario 120 o ajustando su reloj 120 para compensar un adelanto temporal entre el reloj 120 de copia y el reloj primario 120.

La compensación automática de la deriva del reloj puede reducir la frecuencia de las resincronizaciones. Además, la compensación de la deriva del reloj puede permitir la reducción del periodo de tiempo de espera de sincronización. Por ejemplo, en presencia de la deriva del reloj, el periodo de tiempo de espera de sincronización puede configurarse muy largo para reducir la frecuencia de las resincronizaciones. En consecuencia, el módulo procesador 102 que tiene el reloj 120 más rápido malgasta cada vez más tiempo. Al reducir el periodo de tiempo de espera de sincronización, el módulo procesador 102 que tiene el reloj 120 más rápido puede malgastar menos tiempo. Además, cuando realmente ocurre un problema que hace que los módulos procesadores 102 estén desincronizados, la condición de desincronización puede ser detectada y abordada más inmediatamente.

En una realización, el módulo procesador 102 que opera en el modo de copia determina una tasa media de deriva de reloj entre los dos módulos procesadores 102 y profilácticamente corrige su propio reloj 120 para minimizar la deriva de reloj experimentada. Esto puede ser denominado en algunos contextos ajuste de una tasa de adelanto temporal por ciclo de reloj o una tasa de demora temporal por ciclo de reloj. Se entiende que las instrucciones de la aplicación de control pueden incorporar instrucciones de compensación de la deriva del reloj que son ejecutadas selectivamente por el módulo procesador 102 que se ejecuta en el modo de copia y que no son ejecutadas por el módulo procesador 102 que se ejecuta en el modo primario. En una realización, la aplicación de control puede ejecutar una instrucción de sincronización que promueva la determinación de la deriva del reloj. En algunos contextos, esta instrucción de sincronización puede ser denominada sincronización del ciclo del reloj. Dicho en otras palabras, la sincronización con el fin de determinar la deriva de reloj entre los dos módulos procesadores 102 puede ser denominado en algunos contextos sincronización del ciclo del reloj.

En una realización, cuando ha de transmitirse un mensaje desde el controlador 109 de procesos doblemente redundante al DCS 112, ambos módulos procesadores 102, que se supone que ejecutan las instrucciones de la aplicación de control en sincronización mutua, según se ha descrito más arriba, generan un cuerpo de mensaje y un valor de comprobación de redundancia cíclica (CRC) calculado en el cuerpo del mensaje. Se intercambia un mensaje de sincronización entre los dos módulos procesadores 102 que indican que una transmisión está pendiente, y el módulo procesador 102 que opera en el modo de copia incluye el valor de la CRC que calculó en el mensaje de sincronización que transmite a la ubicación 124 de la memoria del estado de sincronización del módulo procesador 102 que opera en el modo primario.

El módulo procesador 102 que opera en el modo primario compara la CRC que calculó con la CRC calculada por el módulo procesador 102 que opera en el modo de copia. Si las CRC coinciden, el módulo procesador 102 que opera

en el modo primario transmite el cuerpo del mensaje y la CRC al DCS 112 y/o al sistema 116 de ordenadores. Si las CRC no coinciden, el módulo procesador 102 que opera en el modo primario no transmite al DCS 112 ni al sistema 116 de ordenadores en ese momento y, en vez de ello, realiza un procedimiento de diagnóstico para determinar por qué los dos módulos procesadores 102 calcularon CRC diferentes. Este evento puede indicar algún error, y esta funcionalidad promueve la corrección del error antes de propagar el error más allá del controlador 109 de procesos doblemente redundante. Se entiende que el cuerpo del mensaje y la propia CRC transmitida por el módulo procesador 102 puede estar encapsulada dentro de un cuerpo de mensaje por un nodo de comunicaciones en la red 110 y una CRC calculada por ese nodo de comunicaciones unida al nuevo mensaje para soportar una comunicación fiable entre este nodo de red y otros nodos de red en la red 110. La CRC determinada por el módulo procesador 102, no obstante, puede ser usada por el DCS 112 y/o el sistema 116 de ordenadores para detectar errores introducidos en el cuerpo del mensaje producido por el módulo procesador 102 cuando este cuerpo del mensaje transita por la red 110.

En una realización, los mensajes que han de transmitirse desde el controlador 109 de procesos doblemente redundante al DCS 112 y/o al sistema 116 de ordenadores pueden ser transmitidos desde una cola de mensajes en el módulo procesador 102. Los mensajes que han pasado la prueba de comparación de CRC descrita más arriba pueden acumularse en la cola de mensajes en el módulo procesador 102, y los módulos procesadores 102 pueden ser capaces de devolver al procesamiento las instrucciones de la aplicación de control en vez de esperar que se transmita cada mensaje. Los mensajes pueden ser transmitidos por el enlace de comunicaciones entre el módulo procesador 102 que opera en el modo primario y la red 110 según permita el ancho de banda del enlace de comunicaciones, por ejemplo mediante una tarea de menor prioridad de la aplicación de control y/o por un chip transceptor que proporciona una cola de mensajes.

Pasando ahora a la FIG. 2, se exponen detalles adicionales de una realización de la FPGA 126 y el controlador de HDLC 128. Cuando el módulo procesador 102 que opera en el modo primario procesa una salida que ha de transmitirse al dispositivo 104 en servicio, la FPGA 126 forma un cuerpo de mensaje y una CRC calculada en el cuerpo del mensaje y envía al controlador de HDLC 128 tanto el cuerpo del mensaje como la CRC. El controlador de HDLC 128 da formato al cuerpo del mensaje y a la CRC creando una trama de HDLC que entonces transmite por el bus 106 de ES de proceso al dispositivo 104 en servicio. En una realización, el controlador de HDLC 128 recibe concurrentemente la misma trama de HDLC que transmite compara el valor de CRC en la trama de HDLC recibida con el valor de CRC que recibió de la FPGA 126. Si los valores de CRC no coinciden, el controlador de HDLC 128 alerta a la FPGA 126, y la FPGA 126 puede realizar un procedimiento de recuperación de errores. Esta comprobación de las CRC de la trama de HDLC puede promover una corrección más rápida de un error y una restauración más inmediata de la comunicación normal con el dispositivo 104 en servicio.

Se entenderá que cada una de las varias innovaciones presentadas más arriba contribuye a promover una informática tolerante a fallos. Además, las características y las técnicas específicas descritas no dependen de un soporte físico especializado, sino que pueden ser implementadas usando componentes de serie, a lo que puede denominarse uso de soporte físico genérico. Aunque en una realización cada una de las varias innovaciones descritas puede estar incorporada en un controlador 109 de procesos doblemente redundante, se entiende que la presente divulgación también contempla otras realizaciones de un controlador 109 de procesos doblemente redundante que incorporan una innovación o una selección reducida de las varias innovaciones descritas.

Pasando ahora a la FIG. 3, se expone un procedimiento 200. En el bloque 202, el primer módulo procesador 102a que opera en el modo primario forma y/o compone un primer mensaje para transmitir desde el controlador 109 de procesos doblemente redundante al DCS 112 y/o al sistema 116 de ordenadores. El primer mensaje comprende una primera carga útil y una primera comprobación de redundancia cíclica (CRC). El primer módulo procesador 102a calcula la primera CRC con la primera carga útil. En el bloque 204, el segundo módulo procesador 102b que opera en el modo de copia forma y/o compone un segundo mensaje. El segundo mensaje comprende una segunda carga útil y una segunda comprobación de redundancia cíclica (CRC). El segundo módulo procesador 102b calcula la segunda CRC con la segunda carga útil. En una realización, el segundo módulo procesador 102b puede transmitir la segunda CRC al primer módulo procesador 102a, por ejemplo en un mensaje de sincronización transmitido a la ubicación 124a de la memoria del estado de sincronización del primer módulo procesador 102a. Se entiende que el procesamiento de los bloques 202 y 204 puede ocurrir sustancialmente al mismo tiempo o que el procesamiento del bloque 204 puede ocurrir ligeramente antes que el procesamiento del bloque 202.

En el bloque 206, el primer módulo procesador 102a compara la primera CRC con la segunda CRC. En el bloque 208, la ejecución del primer módulo procesador 102a se bifurca a uno de dos rutas de procesamiento en función del resultado de la comparación de las CRC. Si las CRC coinciden, el procesamiento prosigue hasta el bloque 210, en el que el primer módulo procesador 102a transmite la primera carga útil y la primera CRC al DCS 112 y/o al sistema 116 de ordenadores. Si las CRC no coinciden, el procedimiento prosigue al bloque 212, en el que el primer módulo procesador 102a inicia diagnósticos. Se entiende que la expresión carga útil empleado con referencia a la descripción del procedimiento 200 corresponde a la expresión cuerpo de mensaje usado en la descripción anterior con referencia a la FIG. 1.

Aunque en la presente divulgación se han proporcionado varias realizaciones, debe entenderse que los sistemas y los procedimientos dados a conocer pueden ser implementados de muchas otras formas específicas sin apartarse del ámbito de la presente divulgación. Ha de considerarse que los presentes ejemplos son ilustrativos y no restrictivos, y que la invención no ha de estar limitada a los detalles dados en el presente documento. Por ejemplo, los diversos elementos o componentes pueden combinarse o integrarse en otro sistema, o ciertas características pueden ser omitidas o no implementadas.

Además, los sistemas, los subsistemas, las técnicas y los procedimientos descritos e ilustrados en las diversas realizaciones de forma diferenciada o separada pueden combinarse o integrarse con otros sistemas, módulos, técnicas o procedimientos sin apartarse del ámbito de la presente divulgación. Otros elementos mostrados o expuestos como acoplados directamente o comunicándose entre sí pueden estar acoplados indirectamente o comunicarse a través de algún dispositivo, interfaz o componente intermedio, ya sea eléctrica, mecánicamente o de otra manera. Otros ejemplos de cambios, sustituciones y alteraciones son determinables por un experto en la técnica y podrían ser realizados sin apartarse del ámbito dado a conocer en el presente documento.

En consecuencia, la invención también versa sobre un controlador de procesos doblemente redundante. El controlador puede comprender una aplicación de control de procesos que se ejecuta en unos módulos primero y segundo. Cuando es ejecutada por el primer módulo, una primera instancia de la aplicación escribe en el segundo módulo una primera información de sincronización, lee en el primer módulo una segunda información de sincronización y, cuando la segunda información de sincronización no coincide con la primera después del transcurso de un intervalo de tiempo de espera, lleva a cabo una función de sincronización; y, cuando es ejecutada por el segundo módulo, la segunda instancia de la aplicación escribe en el primer módulo la segunda información de sincronización, lee en el segundo módulo la primera información de sincronización y, cuando la primera información de sincronización no coincide con la segunda después del transcurso del intervalo de tiempo de espera, lleva a cabo la función de sincronización. La primera instancia de la aplicación llama a la función de sincronización proporcionada por el sistema operativo multitarea en tiempo real antes de invocar una función de eventos establecidos proporcionada por un sistema operativo multitarea en tiempo real.

REIVINDICACIONES

1. Controlador (109) de procesos doblemente redundante que comprende:

un primer procesador (102a);
una primera memoria (122a);
una primera instancia de un sistema operativo multitarea en tiempo real (RTOS);
una primera instancia de una aplicación de control de procesos almacenada en la primera memoria;
un segundo procesador (102b);
una segunda memoria (122b);
una segunda instancia del sistema operativo multitarea en tiempo real; y
una segunda instancia de la aplicación de control de procesos almacenada en la segunda memoria (122b),
en el que, cuando es ejecutada por el primer procesador (102a) en un contexto proporcionado por la primera instancia del sistema operativo multitarea en tiempo real, la primera instancia de la aplicación de control de procesos:

escribe en la segunda memoria (122b) una primera información de sincronización usando una función de sincronización proporcionada por la primera instancia del sistema operativo multitarea en tiempo real,
lee de la primera memoria (122a) una segunda información de sincronización,
realiza una función de resincronización cuando la segunda información de sincronización no coincide con la primera información de sincronización después del transcurso de un intervalo predeterminado de tiempo de espera, y
llama a la función de sincronización proporcionada por la primera instancia del sistema operativo multitarea en tiempo real antes de invocar una función de eventos establecidos proporcionada por la primera instancia del sistema operativo multitarea en tiempo real, y

en el que, cuando es ejecutada por el segundo procesador (102b), la segunda instancia de la aplicación de control de procesos:

escribe en la primera memoria (122a) la segunda información de sincronización usando la función de sincronización proporcionada por la segunda instancia del sistema operativo multitarea en tiempo real,
lee de la segunda memoria (122b) la primera información de sincronización, y
realiza la función de resincronización cuando la primera información de sincronización no coincide con la segunda información de sincronización después del transcurso de un intervalo predeterminado de tiempo de espera,

caracterizado porque la primera información de sincronización comprende uno de varios valores de estado que proporciona una indicación de qué instrucción de la aplicación de control de procesos ha ejecutado recientemente el primer procesador (102a), y
la segunda información de sincronización comprende uno de varios valores de estado que proporciona una indicación de qué instrucción de la aplicación de control de procesos ha ejecutado recientemente el segundo procesador (102b).

2. Controlador (109) de procesos doblemente redundante según la reivindicación 1, en el que la primera información de sincronización se escribe en la segunda memoria (122b) mediante una transmisión por Ethernet desde el primer procesador (102a) al segundo procesador (102b) y en el que la segunda información de sincronización se escribe en la primera memoria (122a) mediante una transmisión por Ethernet desde el segundo procesador al primer procesador.

3. Controlador (109) de procesos doblemente redundante según cualquiera de las reivindicaciones precedentes, en el que la aplicación de control de procesos comprende un secuenciador de estados, en el que la primera instancia de la aplicación de control de procesos ejecuta un primer secuenciador de estados que rastrea el estado de ejecución de la primera instancia de la aplicación de control de procesos, y en el que la segunda instancia de la aplicación de control de procesos ejecuta un segundo secuenciador de estados que rastrea el estado de ejecución de la segunda instancia de la aplicación de control de procesos.

4. Controlador (109) de procesos doblemente redundante según la reivindicación 3, en el que la primera instancia del secuenciador de estados determina la primera información de sincronización y en el que la segunda instancia del secuenciador de estados determina la segunda información de sincronización.

5. Controlador (109) de procesos doblemente redundante según cualquiera de las reivindicaciones precedentes, en el que la función de sincronización proporcionada por la primera instancia del sistema operativo multitarea en tiempo real promueve la sincronización en un ciclo de reloj generado por la primera instancia del sistema operativo multitarea en tiempo real.

6. Controlador (109) de procesos doblemente redundante según la reivindicación 5,

en el que, cuando es ejecutada por el primer procesador (102a) en un contexto proporcionado por la primera instancia del sistema operativo multitarea en tiempo real, la primera instancia de la aplicación de control de procesos escribe una primera información de sincronización que indica a la segunda memoria (122b) una sincronización de ciclo de reloj usando una función de sincronización proporcionada por la primera instancia del sistema operativo multitarea en tiempo real, y
en el que, cuando es ejecutada por el segundo procesador (102b), la segunda instancia de la aplicación de control de procesos ajusta el reloj en función de la primera información de sincronización almacenada en la segunda memoria.

7. Controlador (109) de procesos doblemente redundante según cualquiera de las reivindicaciones precedentes que, además, comprende un controlador (128a) de comunicaciones de control de enlace de datos de alto nivel (HDLC) acoplado al primer procesador (102a), en el que, además, la primera instancia del sistema operativo multitarea en tiempo real:

forma un primer mensaje que comprende una primera carga útil de datos y una primera comprobación de redundancia cíclica (CRC) y
transmite el primer mensaje al controlador de comunicaciones de control de enlace de datos de alto nivel, y

en el que el controlador (128a) de comunicaciones de control de enlace de datos de alto nivel está configurado para:

recibir el primer mensaje,
transmitir el primer mensaje a un dispositivo en servicio,
recibir el primer mensaje transmitido,
calcular una segunda comprobación de redundancia cíclica en función de la recepción del primer mensaje transmitido, y
transmitir un mensaje de error al primer procesador (102a) cuando la segunda comprobación de redundancia cíclica es diferente de la primera comprobación de redundancia cíclica.

8. Controlador (109) de procesos doblemente redundante según cualquiera de las reivindicaciones precedentes en el que, cuando es ejecutada por el primer procesador (102a) en un contexto proporcionado por la primera instancia del sistema operativo multitarea en tiempo real, la primera instancia de la aplicación de control de procesos:

escribe en la segunda memoria (122b) una segunda información de sincronización usando la función de sincronización proporcionada por el sistema operativo multitarea en tiempo real,
lee de la primera memoria (122a) una tercera información de sincronización, y
realiza una función de resincronización cuando la tercera información de sincronización no coincide con la segunda información de sincronización después del transcurso de un intervalo predeterminado de tiempo de espera.

9. Controlador (109) de procesos doblemente redundante según la reivindicación 8, en el que se intercambian la segunda información de sincronización y la tercera información de sincronización para sincronizar el estado entre las instancias primera y segunda de la aplicación de control de procesos.

10. Controlador (109) de procesos doblemente redundante según la reivindicación 9, en el que la aplicación de control de procesos comprende un secuenciador de estados, en el que la primera instancia de la aplicación de control de procesos ejecuta un primer secuenciador de estados que rastrea el estado de ejecución de la primera instancia de la aplicación de control de procesos, y en el que la segunda instancia de la aplicación de control de procesos ejecuta un segundo secuenciador de estados que rastrea el estado de ejecución de la segunda instancia de la aplicación de control de procesos.

11. Controlador (109) de procesos doblemente redundante según cualquiera de las reivindicaciones precedentes, en el que la primera información de sincronización se escribe en la segunda memoria (122b) mediante una transmisión por Ethernet desde el primer procesador (102a) al segundo procesador (102b).

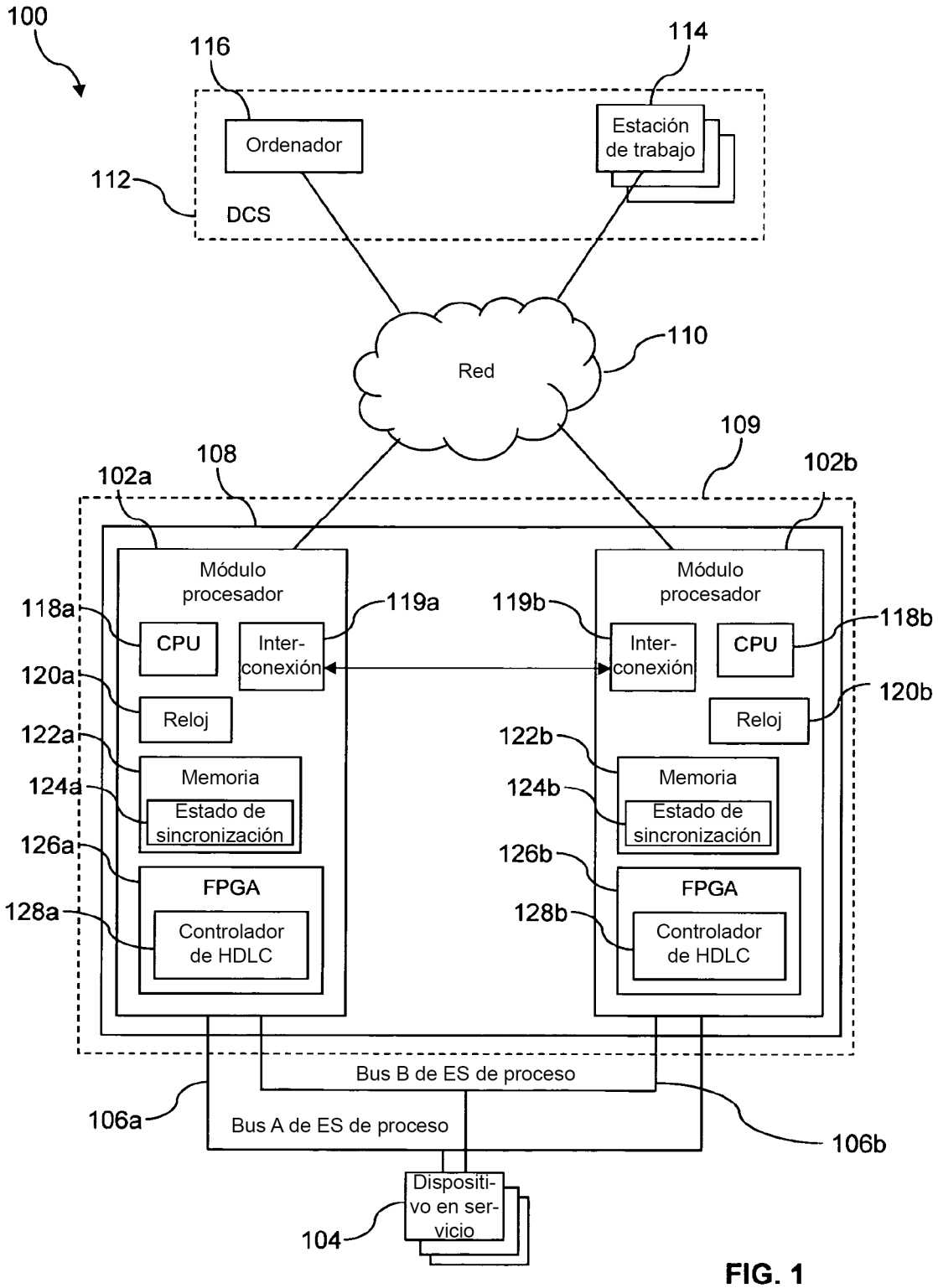
100
116
114
Ordenador
Estación de trabajo
112
DCS
Red
110
109
108
102a
102b
Módulo procesador
Módulo procesador
118a
119a
119b
118b
CPU
Inter-conexión
Inter-conexión
CPU
120a
120b
Reloj
Reloj
122a
122b
Memoria
Memoria
124a
124b
Estado de sincronización
Estado de sincronización
126a
126b
FPGA
FPGA
128a
128b
Controlador de HDLC
Controlador de HDLC
Bus B de ES de proceso
106a
Bus A de ES de proceso
106b
Dispositivo en servicio
104

FIG. 1

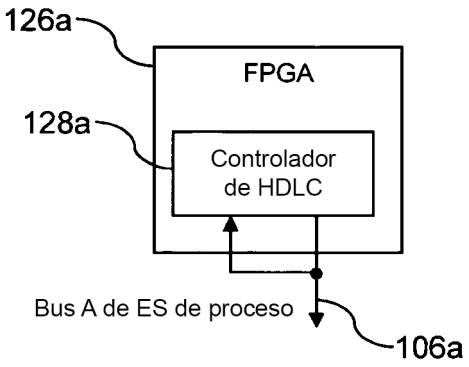
126a
FPGA
128a
Controlador
de HDLC
Bus A de ES de proceso
106a

FIG. 2

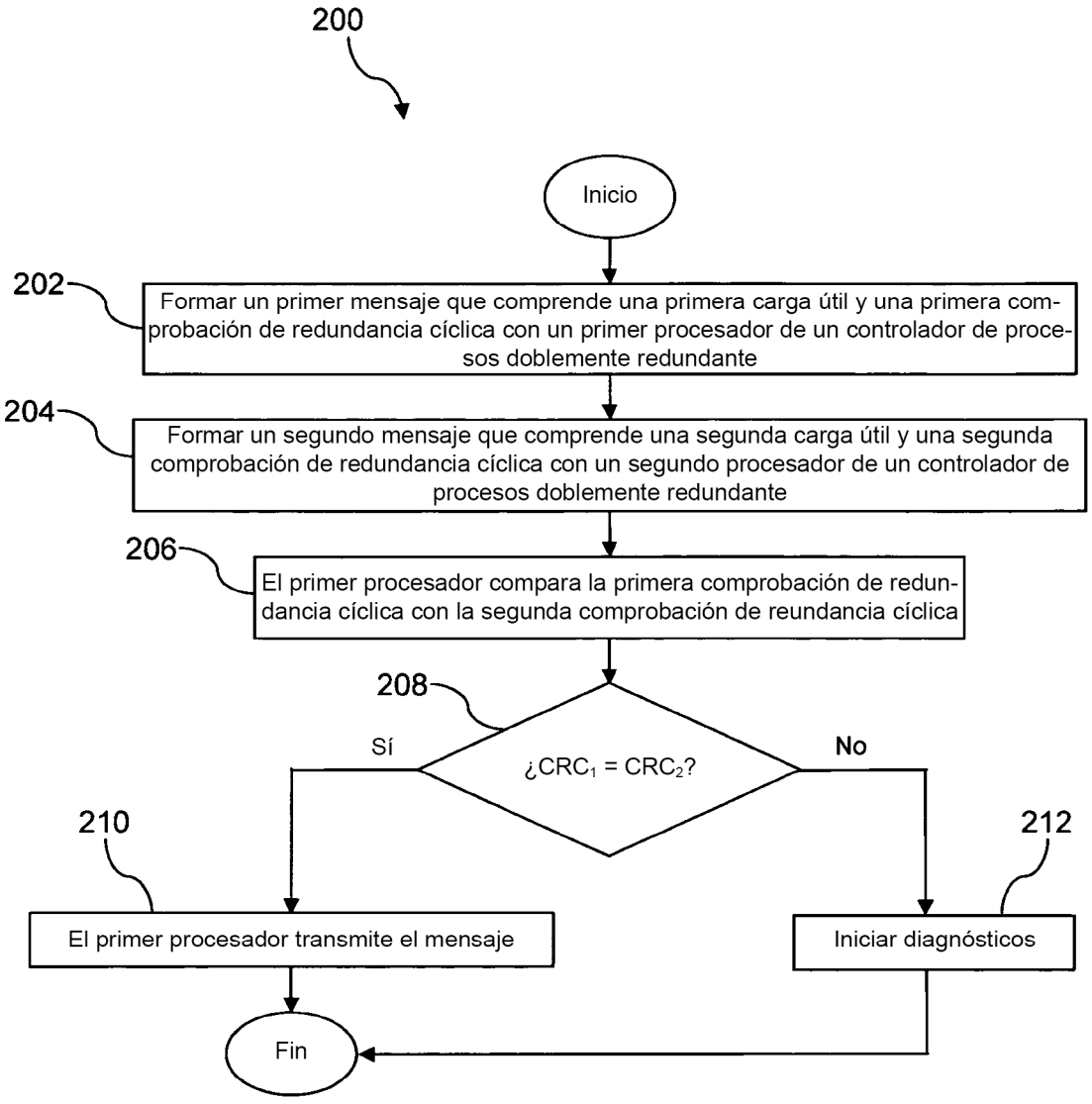
200
Inicio
202
Formar un primer mensaje que comprende una primera carga útil y una primera comprobación de redundancia cíclica con un primer procesador de un controlador de procesos doblemente redundante
204
Formar un segundo mensaje que comprende una segunda carga útil y una segunda comprobación de redundancia cíclica con un segundo procesador de un controlador de procesos doblemente redundante
206
El primer procesador compara la primera comprobación de redundancia cíclica con la segunda comprobación de reundancia cíclica
208
¿CRC1 = CRC2?
Sí
No
210
El primer procesador transmite el mensaje
212
Iniciar diagnósticos
Fin

FIG. 3

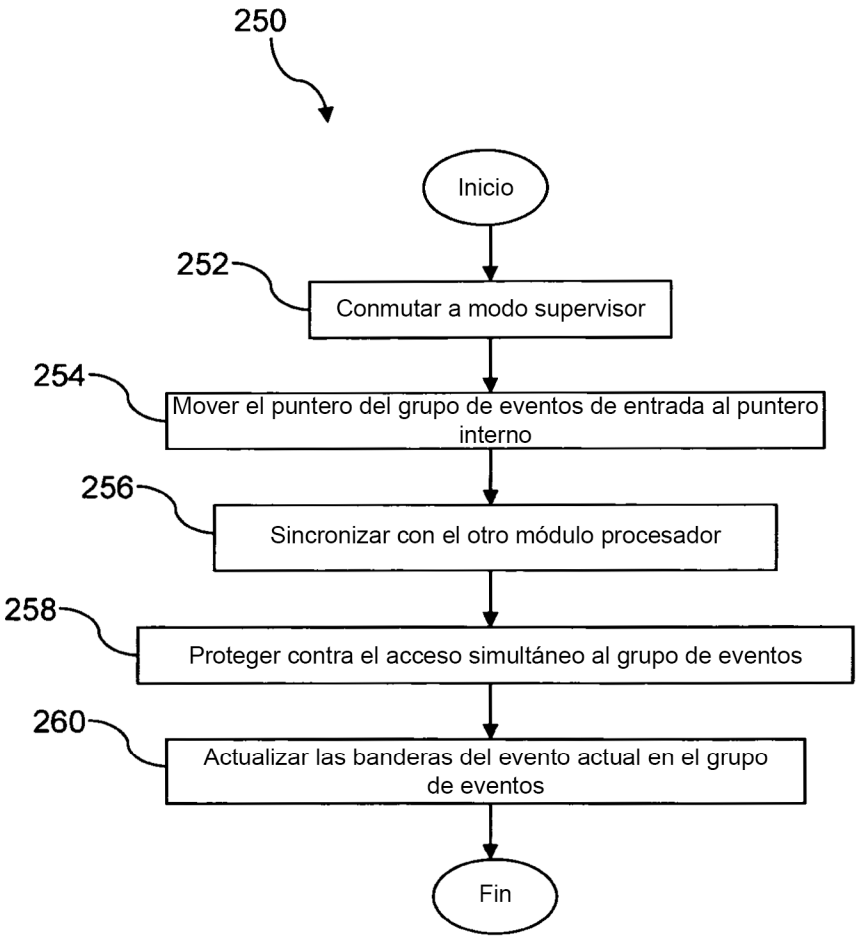
250
Inicio
252
Conmutar a modo supervisor
254
Mover el puntero del grupo de eventos de entrada al puntero interno
256
Sincronizar con el otro módulo procesador
258
Proteger contra el acceso simultáneo al grupo de eventos
260
Actualizar las banderas del evento actual en el grupo de eventos
Fin

FIG. 4