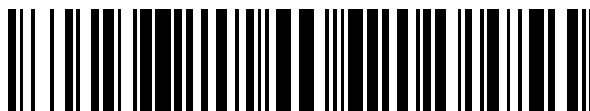


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 680 622**

51 Int. Cl.:

G05B 19/042 (2006.01)

G05B 19/05 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **14.04.2015 E 15163449 (0)**

97 Fecha y número de publicación de la concesión europea: **06.06.2018 EP 2933697**

54 Título: **Sistema de PLC que tiene una pluralidad de módulos de CPU y método de control del mismo**

30 Prioridad:

15.04.2014 KR 20140044648

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

10.09.2018

73 Titular/es:

**LSIS CO., LTD. (100.0%)
127 LS-ro, Dongan-gu
Anyang-si, Gyeonggi-do 431-080, KR**

72 Inventor/es:

**LEE, SOO GANG y
KWON, DAE HYUN**

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 680 622 T3

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

DESCRIPCIÓN

Sistema de PLC que tiene una pluralidad de módulos de CPU y método de control del mismo

5 Campo técnico

La presente divulgación se refiere a un sistema de PLC que tiene una pluralidad de módulos de CPU y un método de control del mismo.

10 Antecedentes de la técnica

En general, la comunicación se realiza a través de un circuito en paralelo conocido como un panel posterior para cada módulo en un PLC (Controlador Lógico Programable) usado en una FA (Automatización Industrial). Es decir, un panel posterior se usa para comunicar desde un módulo a otro módulo, donde un módulo que solicita una comunicación debe obtener un derecho de control sobre el panel posterior. Cuando varios módulos intentan acceder a un panel posterior sin derecho de control, se genera una señal de conflicto para deshabilitar un acceso normal.

15

En general, en una comunicación entre un módulo maestro que solicita una comunicación y un módulo esclavo que responde a la petición del módulo maestro, un derecho de control sobre un recurso común emite un testigo de acuerdo con una regla predeterminada, y aunque métodos de acceso son iguales cuando un maestro predeterminado posee un testigo (es decir, que tiene un derecho de control), es difícil realizar un rendimiento satisfactorio en control en tiempo real, porque se requiere un tiempo para intercambiar testigos y un tiempo para procesar los testigos de acuerdo con una regla predeterminada para determinar una propiedad de testigo.

20

25 SIMATIC S5-135U/155U SYSTEM MANUAL-RELEASE 06, diciembre de 1998, en particular 6-1 a 6-28 divulga un Controlador Lógico Programable (PLC), que puede usarse tanto en operación de un único procesador como de múltiples procesadores con hasta cuatro CPU.

30 La Figura 1 es una vista esquemática que ilustra un método ilustrativo para obtener un derecho de control basado en testigos de acuerdo con un sistema convencional de múltiples maestros.

Haciendo referencia a la Figura 1, una pluralidad de maestros (100~400) pueden poseer un derecho de control llamado testigo en un método de orden cíclico. El maestro que posee el testigo tiene una autoridad para acceder a un panel posterior y puede acceder a un módulo esclavo a través del panel posterior. Es decir, cuando existe una necesidad para que un maestro controle a un esclavo, el maestro comprueba si tiene un testigo, y accede al esclavo a través de un panel posterior si tiene el testigo, y cuando el maestro no tiene un testigo, y el maestro no puede acceder al esclavo sino que tiene que esperar. El maestro puede acceder al esclavo una vez que tiene un testigo y puede transferir el testigo a otros maestros de acuerdo con un algoritmo predeterminado.

35

Aunque la Figura 1 ha ilustrado un método de orden cíclico, es posible recibir y transferir el testigo en otros diversos métodos de acuerdo con una regla predeterminada por un realizador. Sin embargo, este método tiene un límite en rendimiento en tiempo real, porque requiere un algoritmo de control de testigo y tiempo de procesamiento de recepción/transmisión de testigo.

40

La Figura 2 es una vista esquemática que ilustra un método ilustrativo para obtener un derecho de control de acuerdo con un intervalo de tiempo en un sistema convencional de múltiples maestros, donde el sistema está disponible con n número de maestros.

45

Haciendo referencia a la Figura 2, un tiempo continuo se divide por intervalo de tiempo usando el número de maestros o más del número de maestros, donde el maestro puede acceder a un panel posterior únicamente dentro de un tiempo asignado al mismo.

50

Cuando existe una necesidad de controlar un esclavo, un maestro opera de una manera de tal forma que el maestro primero comprueba si es un intervalo de tiempo asignado a sí mismo y comienza a acceder al panel posterior, y el maestro espera durante intervalo de tiempo no asignado a sí mismo o intenta de nuevo después de un periodo de tiempo predeterminado.

55

Este método tiene un efecto ventajoso en que puede reducirse un tiempo de procesamiento porque no hay intercambio directo de testigos pero existe una posibilidad de colisión porque puede intentarse simultáneamente el acceso por error de medición de tiempo de cada maestro. Es decir, una pluralidad de sistemas maestros se accionan mediante respectivos relojes y cuando se genera un error de tiempo en los relojes, existe una posibilidad de colisión debido a error de reloj en el mismo momento.

60

Divulgación

Problema técnico

5 La presente divulgación se ha hecho para resolver las desventajas/problemas anteriores de la técnica anterior y por lo tanto un objeto de ciertas realizaciones de la presente invención es proporcionar un sistema de PLC que tiene una pluralidad de CPU configuradas para obtener un derecho de control de panel posterior para acceder a un módulo esclavo en tiempo real y un método de control del mismo.

10 Solución técnica

En un aspecto general de la presente divulgación, se proporciona un sistema de PLC que comprende:

15 una pluralidad de módulos de CPU, en el que:

la pluralidad de módulos de CPU comprende un módulo de CPU maestro, en el encendido, el módulo de CPU maestro averigua el número y atributos de todos los módulos de CPU presentes dentro del sistema de PLC y establece recuentos de tiempo que definen un respectivo intervalo de tiempo para acceder a un panel posterior para cada uno de los módulos de CPU, el módulo de CPU maestro transmite los recuentos de tiempo a todos los otros módulos de CPU, cada módulo de CPU incluye un detector configurado para detectar una señal de reloj generada por un módulo de CPU que accede al panel posterior, cada módulo de CPU accede al panel posterior durante intervalo de tiempo asignado si la señal de reloj no se detecta por el detector, y cuando cada módulo de CPU finaliza su acceso al panel posterior, detiene la generación de la señal de reloj y el módulo de CPU maestro sincroniza los recuentos de tiempo iniciando desde cero el recuento de tiempo, en el que los recuentos de tiempo definen "recuento máximo de intervalo de tiempo", "recuento actual de intervalo de tiempo", un respectivo "recuento de inicio de intervalo de tiempo" y un respectivo "recuento de fin de intervalo de tiempo" para cada módulo de CPU.

30 Preferentemente, pero no necesariamente, el primer módulo de CPU y la pluralidad de los segundos módulos de CPU que intentan acceder al panel posterior no acceden al panel posterior cuando se detecta una señal de reloj por el detector.

35 Preferentemente, pero no necesariamente, el primer módulo de CPU sincroniza recuentos de intervalo de tiempo con un recuento de inicio cuando el primer módulo de CPU y la pluralidad de los segundos módulos de CPU que accedieron al panel posterior detienen la generación de una señal de reloj.

40 Preferentemente, pero no necesariamente, el primer módulo de CPU determina el intervalo de tiempo teniendo en cuenta el número y atributos del primer módulo de CPU y la pluralidad de segundos módulos de CPU.

Preferentemente, pero no necesariamente, el primer módulo de CPU transmite un recuento máximo de intervalo de tiempo, un recuento actual de intervalo de tiempo, un recuento de inicio de intervalo de tiempo y un recuento de fin de intervalo de tiempo a la pluralidad de segundos módulos de CPU.

45 En otro aspecto general de la presente divulgación, se proporciona un método de control de sistema de PLC que tiene una pluralidad de módulos de CPU, comprendiendo el método:

50 determinar un intervalo de tiempo asignado para indicar un recuento accesible al panel posterior; averiguar una señal de reloj cuando el recuento es un recuento que corresponde a un intervalo de tiempo asignado; acceder al panel posterior; y sincronizar recuentos en un momento cuando finaliza el acceso al panel posterior.

55 Preferentemente, pero no necesariamente, el método puede comprender adicionalmente no acceder al panel posterior cuando se detecta una señal de reloj en un recuento que corresponde al intervalo de tiempo asignado.

Preferentemente, pero no necesariamente, el método puede comprender adicionalmente esperar para acceder al panel posterior en un intervalo de tiempo de siguiente recuento.

60 Preferentemente, pero no necesariamente, el método puede comprender adicionalmente generar una señal de reloj en un momento de acceso al panel posterior.

65 Preferentemente, pero no necesariamente, el método puede comprender adicionalmente finalizar la generación de una señal de reloj en un momento cuando finaliza el acceso al panel posterior.

Preferentemente, pero no necesariamente, el intervalo de tiempo puede determinarse teniendo en cuenta el número y atributos de la pluralidad de módulos de CPU conectados al panel posterior.

5 En aún otro aspecto general de la presente divulgación, se proporciona un método de control de sistema de PLC que tiene una pluralidad de CPU, comprendiendo el método:

10 averiguar una señal de reloj cuando un recuento es un recuento que corresponde a un intervalo de tiempo asignado por un módulo de CPU maestro de la pluralidad de módulos de CPU;
 acceder a un panel posterior y generar una señal de reloj mediante la pluralidad de módulos de CPU; y
 10 finalizar la generación de una señal de reloj en un momento cuando finaliza el acceso al panel posterior mediante la pluralidad de módulos de CPU.

15 Preferentemente, pero no necesariamente, el método puede comprender adicionalmente no acceder al panel posterior cuando se detecta una señal de reloj en un recuento que corresponde al intervalo de tiempo asignado.

Preferentemente, pero no necesariamente, el método puede comprender adicionalmente esperar para acceder al panel posterior en un intervalo de tiempo de siguiente recuento.

20 Preferentemente, pero no necesariamente, el método puede comprender adicionalmente sincronizar, mediante el módulo de CPU maestro, el recuento en un momento de finalización de generación de señal de reloj.

Efectos ventajosos

25 Las realizaciones ilustrativas de la presente divulgación tienen efectos ventajosos en que puede evitarse la colisión averiguando (comprobando o confirmando) si otros módulos de CPU se conectan incluso si corresponden a un intervalo de tiempo del propio módulo de CPU, y un módulo de CPU maestro sincroniza recuentos en un momento de finalización, mediante un cierto módulo de CPU, de un acceso a un módulo esclavo para reducir de este modo un tiempo en espera para el acceso.

30 Descripción de los dibujos

La Figura 1 es una vista esquemática que ilustra un método ilustrativo para obtener un derecho de control basado en testigos de acuerdo con un sistema convencional de múltiples maestros.

35 La Figura 2 es una vista esquemática que ilustra un método ilustrativo para obtener un derecho de control de acuerdo con un intervalo de tiempo en un sistema convencional de múltiples maestros.

La Figura 3 es una vista esquemática que ilustra un sistema de PLC de acuerdo con la presente divulgación.

La Figura 4 es un diagrama de bloques esquemático que ilustra conectividad en la Figura 3.

40 La Figura 5 es una vista de bloques esquemática que ilustra parámetros transmitidos a CPU1, CPU2 y CPU3 siendo generados por CPU0 de la Figura 3 de acuerdo con una realización ilustrativa de la presente divulgación.

La Figura 6 es una vista esquemática que ilustra intervalos de tiempo generados por CPU0 de acuerdo con una realización ilustrativa de la presente divulgación.

La Figura 7 es un diagrama de flujo que ilustra un método para controlar un sistema de PLC de acuerdo con la presente divulgación.

45 Mejor modo

50 Diversas realizaciones ilustrativas se describirán de forma más completa en lo sucesivo con referencia a los dibujos adjuntos, en los que se muestran algunas realizaciones ilustrativas. El presente concepto inventivo, sin embargo, puede incorporarse de muchas formas diferentes y no debería interpretarse como limitado a los ejemplos de realizaciones expuestos en este documento. En su lugar, el aspecto descrito se concibe para incluir todas tales alteraciones, modificaciones y variaciones que pertenecen al alcance e idea novedosa de la presente divulgación.

55 En lo sucesivo, se proporcionará una explicación para la obtención del derecho de control de acuerdo con un sistema convencional de múltiples maestros y a continuación con las realizaciones ilustrativas de la presente divulgación con referencia a los dibujos adjuntos.

La Figura 3 es una vista esquemática que ilustra un sistema de PLC de acuerdo con la presente divulgación, y la Figura 4 es un diagrama de bloques esquemático que ilustra conectividad en la Figura 3.

60 Haciendo referencia a la Figura 3, un sistema de PLC de acuerdo con una realización ilustrativa de la presente divulgación puede incluir una pluralidad de ranuras (A), donde una pluralidad de módulos de CPU (Unidades de Procesamiento Central) (10, 20, 30, 40) pueden asignarse a una ranura predeterminado, y una pluralidad de módulos esclavos controlados por la pluralidad de módulos de CPU (10, 20, 30, 40) pueden asignarse a una ranura cerca de cada uno de la pluralidad de módulos de CPU (10, 20, 30, 40).

65

En este momento, el módulo esclavo puede incluir diversos módulos que incluyen un módulo de comunicación, un módulo de entrada/salida y un módulo de recuento, por ejemplo, y debería ser evidente que la presente divulgación no se limita por los tipos de módulos esclavos.

5 Es decir, módulos esclavos (11, 12) de un módulo CPU0 (10) pertinente pueden asignarse a un intervalo cerca del módulo de CPU 0 (10), y respectivos módulos esclavos también pueden asignarse a otros módulos de CPU (20, 30, 40). Aunque la realización ilustrativa de la presente divulgación ha ilustrado un caso en el que se ilustran dos módulos esclavos que corresponden a un módulo de CPU, la presente divulgación no se limita a los mismos, y debería ser evidente a los expertos en la técnica que tipos y número de módulos esclavos pueden determinarse según requieran las necesidades.

10 En lo sucesivo, por conveniencia, el módulo de CPU 0 (10) se designará como 'CPU0' como se ilustra en los dibujos, y módulos esclavos de CPU0 (10) se designarán como SO_1 (11) y SO_2 (12). Análogamente, módulos esclavos de CPU1 (20) se designarán como S1_1 (21) y S1_2 (22), módulos esclavos de CPU2 (30) se designarán como S2_1 (31) y S2_2 (32), y módulos esclavos de CPU3 (40) se designarán como S3_1 (41) y S3_2 (42). Adicionalmente, la CPU0 (10) se designará como un módulo de CPU maestro configurado para controlar todo un sistema de PLC, y otras CPU1, CPU2 y CPU3 (20, 30, 40) se designarán como módulos maestros de CPU de bus. Sin embargo, debería ser evidente que la presente divulgación no se limita a las designaciones anteriores.

20 Como se observa a partir de lo anterior, módulos asignados dentro de una pluralidad de intervalos (A) de un sistema de PLC se conectan a un panel posterior, la configuración de lo cual se ilustra en la Figura 4 que a su vez ilustra miméticamente un ejemplo de la Figura 3.

25 Es decir, puede saberse que cada uno de CPU0 (10), SO_1 (11), SO_2 (12), CPU1 (20), S1_1 (21), S1_2 (22), CPU2 (30), S2_1 (31), S2_2 (32), CPU3 (40), S3_1 (41) y S3_2 (42) se asignan todos a través de un panel posterior (B).

30 En el sistema de PLC anterior, el módulo de CPU maestro que es CPU0 (10) asigna intervalos de tiempo en respuesta a un tiempo predeterminado para controlar una pluralidad de módulos de CPU, donde cada módulo de CPU no accede incondicionalmente a un módulo esclavo a través del panel posterior (B) incluso si se correspondiera con su intervalo de tiempo, el módulo de CPU accede a un módulo esclavo a través del panel posterior (B) evitando colisión usando un método de averiguar si se hace el acceso mediante otros módulos de CPU durante intervalo de tiempo, y puede evitarse la colisión fundamental sincronizando, mediante un módulo de CPU maestro, recuentos de intervalo de tiempo basándose en un tiempo cuando el acceso de cada módulo de CPU finaliza, una descripción detallada a lo mismo se proporcionará en lo sucesivo.

35 En la descripción de la presente divulgación, debería mencionarse que un significado de 'acceso a módulo esclavo a través del panel posterior (B)' es el mismo que el de 'acceso a módulo esclavo a través del panel posterior (A)'.

40 Cuando se aplica una potencia eléctrica a un sistema de PLC ilustrado en las Figuras 3 y 4, la CPU0 (10) puede averiguar (comprobar o confirmar) el número de módulos de CPU presentes dentro de un sistema de PLC y determinar un intervalo de tiempo y transmitir la determinación a la CPU1, CPU2 y CPU3 (20, 30, 40). En este momento, el intervalo de tiempo puede determinarse teniendo en cuenta el número de módulos de CPU y atributos de módulos de CPU pertinentes, con lo que una longitud de intervalo de tiempo asignado a cada módulo de CPU puede ser el mismo o diferente.

La Figura 5 es una vista de bloques esquemática que ilustra parámetros transmitidos a CPU1, CPU2 y CPU3 siendo generados por CPU0 de la Figura 3 de acuerdo con una realización ilustrativa de la presente divulgación.

50 Haciendo referencia a la Figura 5, el parámetro de la presente divulgación puede incluir un recuento máximo de intervalo de tiempo (5A), un recuento actual de intervalo de tiempo (5B), un recuento de inicio de intervalo de tiempo (5C) y un recuento de fin de intervalo de tiempo (5D).

55 El recuento máximo de intervalo de tiempo (5A) puede establecerse igual a o mayor que un tamaño en el que se añaden tamaños de intervalos de tiempo asignados a cada módulo de CPU. El recuento actual de intervalo de tiempo (5B) puede ser un funcionamiento sin sincronizar desde cero (0) hasta un recuento máximo de intervalo de tiempo, en el que la CPU0 (10) puede sincronizarse a cero (0) en un tiempo cuando finaliza el acceso a respectivos módulos esclavos de cada módulo de CPU, con lo que todos los módulos de CPU son iguales en el momento de cero. El recuento de inicio de intervalo de tiempo (5C) es un tiempo de intervalo de tiempo accesible, por cada módulo de CPU, al panel posterior (B), y el recuento de fin de intervalo de tiempo (5D) indica un final de intervalo de tiempo accesible, por cada módulo de CPU, al panel posterior (B).

60 Es decir, la CPU0 (10) puede averiguar el número y atributos de los módulos de CPU (10-40) que accedieron al panel posterior (B) cuando el sistema de PLC se enciende o reinicia, para establecer un intervalo de tiempo

accesible por cada módulos de CPU (10~40), para establecer el recuento máximo de intervalo de tiempo (5A) y para establecer el recuento de inicio de intervalo de tiempo (5C) y el recuento de fin de intervalo de tiempo (5D).

5 Sucesivamente, la CPU0 (10) puede transmitir el recuento máximo de intervalo de tiempo (5A), el recuento actual de intervalo de tiempo (5B), el recuento de inicio de intervalo de tiempo (5C) y el recuento de fin de intervalo de tiempo (5D) a otros módulos de CPU (10~40). El recuento máximo de intervalo de tiempo (5A) y el recuento actual de intervalo de tiempo (5B) son iguales en relación con todos los módulos de CPU (10~40), pero el recuento de inicio de intervalo de tiempo (5C) y el recuento de fin de intervalo de tiempo (5D) pueden ser diferentes para cada uno de los módulos de CPU (10~40).

10 Por ejemplo, la CPU0 (10) puede establecerse accesible a un intervalo de tiempo que corresponde a cuando un intervalo de tiempo de CPU0 (10) es un recuento desde cero (0) a nueve, cuando la CPU1 (20) es un recuento desde 10 a 19, y cuando la CPU2 (30) es un recuento desde 20 a 29, en ese momento, el recuento de inicio de intervalo de tiempo (5C) de la CPU0 (10) puede ser cero (0) y el recuento de fin de intervalo de tiempo (5D) puede ser 9.

15 El sistema de PLC de acuerdo con una realización ilustrativa de la presente divulgación como se ilustra en la Figura 3 puede compartir un reloj. Cada módulo de CPU (10~40) puede incluir un detector configurado para detectar un reloj y cada módulo de CPU (10~40) puede generar una señal de reloj en respuesta al reloj del sistema de PLC.

20 La Figura 6 es una vista esquemática que ilustra intervalos de tiempo generados por CPU0 de acuerdo con una realización ilustrativa de la presente divulgación.

25 Ahora, haciendo referencia de nuevo al ejemplo anterior, como en (a), la CPU0 (10) puede acceder a un intervalo de tiempo 1 (TS1), la CPU1 (20) puede acceder a un intervalo de tiempo 2 (TS2), la CPU2 (30) puede acceder a un intervalo de tiempo 3 (TS3) y la CPU3 (40) puede acceder a un intervalo de tiempo 4 (TS4) respectivamente a través del panel posterior (B).

30 En este momento, la CPU0 (10) puede acceder a SO_1 (11) o SO_2 (12) a través del panel posterior (B) en un recuento que corresponde a TS1, en el que la CPU0 (10) que accedió al SO_1 (11) o SO_2 (12) a través del panel posterior (B) puede generar una señal de reloj como en (b). Puede saberse averiguando a través de cada señal de reloj detector que el equilibrio de módulos de CPU (20, 30, 40) es de tal forma que un cierto módulo de CPU arbitrario se conecta al módulo esclavo a través del panel posterior (B) cuando una señal de reloj está presente en el sistema de PLC. Es decir, como en (c), la CPU0 (10) que accedió en recuento T1 al SO_1 (11) o SO_2 (12) a través del panel posterior (B) puede finalizar un acceso pertinente.

35 Cuando la CPU1 (20) intenta conectar a un módulo esclavo en recuento T3 a través del panel posterior (B), la CPU1 (20) puede averiguar una señal de reloj de sistema de PLC a través de su detector de señal de reloj. Puede averiguarse que es posible el acceso a un recuento T3 pertinente a través del panel posterior (b) porque la CPU1 (20) puede averiguar una señal de reloj generada por la CPU0 (10) en recuento T3 en un sistema de PLC, y ponerse en espera para que se acceda a un siguiente intervalo de tiempo.

40 La CPU0 (10) puede finalizar el acceso en recuento T2 y simultáneamente finalizar la generación de señal de reloj. En este caso, la CPU0 (10) del sistema de PLC de acuerdo con la presente divulgación puede sincronizar recuentos en el recuento T2. Es decir, como en (c), el recuento puede iniciarse de nuevo desde cero (0) en un recuento que corresponde a T2. Es decir, TS1 puede iniciarse de nuevo desde el principio.

45 Sin embargo, aunque la realización ilustrativa de la presente divulgación ha ilustrado y explicado que el módulo de CPU que finaliza el acceso y el módulo de CPU que sincroniza los recuentos son el mismo, puede decirse que el módulo de CPU que sincroniza los recuentos es CPU0 (10) que es un módulo de CPU maestro y el módulo de CPU que finaliza el acceso corresponde a todos los módulos de CPU.

50 La CPU1 (20) que espera su propio orden puede acceder a un módulo esclavo a través del panel posterior en recuento T4 (es decir, 10) cuando su propio intervalo de tiempo se recuenta de nuevo después de la sincronización. La CPU1 (10) que accedió al módulo esclavo puede generar una señal de reloj predeterminada, en la que otro módulo de CPU que ha averiguado la señal de reloj no accede al panel posterior (B) pero puede esperar hasta que su próximo intervalo de tiempo se recuenta.

55 La CPU1 (20) puede acceder a su propio módulo esclavo hasta el recuento T5 en la realización ilustrativa de la presente divulgación, y aunque no se ilustra, la CPU1 (20) puede recontar el intervalo de tiempo de nuevo desde el principio sincronizando recuentos de nuevo como en (c).

60 Por lo tanto, puede reducirse un tiempo en espera de otros módulos de CPU sincronizando los recuentos en un momento de finalización el acceso mediante el módulo de CPU de acuerdo con la presente divulgación. Por ejemplo, como en (b), cuando los recuentos no se sincronizan incluso cuando la CPU0 (10) finaliza el acceso, la CPU1 (20)

debe esperar su intervalo de tiempo hasta recuento T6, pero el acceso puede habilitarse en T4 que es antes de T6, con lo que el tiempo en espera puede reducirse.

5 La Figura 7 es un diagrama de flujo que ilustra un método para controlar un sistema de PLC de acuerdo con la presente divulgación.

10 Haciendo referencia a la Figura7, la CPU0 (10) que es un módulo maestro en el sistema de PLC de acuerdo con la presente divulgación puede determinar intervalo de tiempo configurado para determinar un recuento accesible, por cada módulo de CPU, a un módulo esclavo a través del panel posterior (B) teniendo en cuenta el número y atributos de los módulos de CPU cuando el sistema de PLC se enciende o reinicia (S71, S72). En este momento, la CPU0 (10) puede determinar el recuento máximo de intervalo de tiempo (5A), el recuento actual de intervalo de tiempo (5B), el recuento de inicio de intervalo de tiempo (5C) a cada módulo de CPU, y el recuento de fin de intervalo de tiempo (5D) y transmitir a cada módulo de CPU.

15 Posteriormente, considérese un caso en el que un módulo de CPU, por ejemplo, la CPU1 (20), requerido para acceso desde un módulo esclavo. Sin embargo, debería ser evidente que operaciones de la presente divulgación no se limitan únicamente a la CPU1 (20) y pueden ser aplicables a otros módulos de CPU.

20 La CPU1 (20) puede averiguar si un recuento actual es un recuento que corresponde a su propio intervalo de tiempo cuando existe una petición de acceso desde S1_1 (21) o S1_2 (22) (S73). Es decir, la CPU1 (20) puede averiguar si el recuento actual corresponde al recuento 10 a 20 en la Figura 6 (S74)

25 Como resultado de S74, si se determina que el recuento no es su propio intervalo de tiempo, la CPU1 (20) puede esperar durante un tiempo predeterminado a un tiempo que corresponde a su propio intervalo de tiempo (S76). La CPU1 (20) puede notificar que el acceso ha fallado (S77) cuando no alcanza un recuento que corresponde a su propio intervalo de tiempo incluso si ha transcurrido un tiempo predeterminado

30 Como resultado de S74, si se determina que el recuento es su propio intervalo de tiempo, es decir, cuando el recuento actual corresponde al recuento 10 a 20, la CPU1 (20) puede averiguar si una señal de reloj está disponible (S75). La generación de una señal de reloj actual puede confirmarse mediante la detección por un detector de señal de reloj incluido en la CPU1 (20). Es decir, por ejemplo, cuando la CPU2 (30) accede a través del panel posterior (B) en un recuento actual, la CPU 2 (30) puede generar un recuento actual, en el que el detector de señal de reloj de la CPU1 (20) puede confirmar la generación de la cuenta actual, la CPU1 (20) puede determinar que es imposible acceder al panel posterior (B) en un recuento actual y puede esperar un tiempo predeterminado para corresponder a un siguiente intervalo de tiempo propio (S76). La CPU1 (20) puede notificar que el acceso ha fallado cuando no alcanza un recuento que corresponde a su propio intervalo de tiempo incluso si ha transcurrido un tiempo predeterminado (S77).

40 La CPU1 (20) puede acceder a S1_1 (21) o S1_2 (22) a través del panel posterior (B) cuando no hay señal de reloj generada en correspondencia con su propio intervalo de tiempo (S78). Al mismo tiempo, la CPU1 (20) puede generar una señal de reloj y notificar que accede a un módulo esclavo a través del panel posterior (B) (S79). Aunque S78 y S79 se ilustran en una relación secuencial, que es por conveniencia, y debería ser evidente que S78 y S79 son de una relación realizada simultáneamente. Sin embargo, debería ser evidente que la presente divulgación no se limita a únicamente un caso en el que la generación de señal de reloj de la CPU1 (20) se realiza simultáneamente junto con un módulo esclavo a través del panel posterior (B). La generación de señal de reloj en S79 puede realizarse de forma continua durante un tiempo cuando se accede al módulo esclavo a través del panel posterior (B).

50 Sucesivamente, la CPU1 (20) detiene la generación de señal de reloj (S81) al mismo tiempo cuando el acceso finaliza (S80), y la CPU0 (10) en el sistema de PLC puede sincronizar recuentos en un momento de finalización la señal de reloj y transmitir los recuentos sincronizados a cada módulo de CPU (S82).

55 Las realizaciones ilustrativas de la presente divulgación tienen efectos ventajosos en que puede evitarse la colisión averiguando (comprobando o confirmando) si otros módulos de CPU se conectan incluso si corresponden a un intervalo de tiempo del propio módulo de CPU, y un módulo de CPU maestro sincroniza recuentos en un momento de finalización, mediante un cierto módulo de CPU, de un acceso a un módulo esclavo para reducir de este modo un tiempo en espera para el acceso.

60 Aunque la presente divulgación se ha descrito en detalle con referencia a las realizaciones anteriores y ventajas, muchas alternativas, modificaciones y variaciones serán evidentes para expertos en la materia dentro de los requisitos y límites de las reivindicaciones. Por lo tanto, debería entenderse que las realizaciones anteriormente descritas no se limitan por ninguno de los detalles de la descripción anterior, a no ser que se especifique de otra manera, sino que debería interpretarse ampliamente dentro del alcance como se define en las reivindicaciones adjuntas.

65

Aplicabilidad industrial

5 Las realizaciones ilustrativas de la presente divulgación tienen aplicabilidad industrial en que puede evitarse la colisión averiguando (comprobando o confirmando) si otros módulos de CPU se conectan incluso si corresponden a un intervalo de tiempo del propio módulo de CPU, y un módulo de CPU maestro sincroniza recuentos en un momento de finalización, mediante un cierto módulo de CPU, de un acceso a un módulo esclavo para reducir de este modo un tiempo en espera para el acceso.

REIVINDICACIONES

1. Un sistema de PLC que comprende una pluralidad de módulos de CPU (10, 20, 30, 40), caracterizado por que:

5 la pluralidad de módulos de CPU (10, 20, 30, 40) comprende un módulo de CPU maestro (10),
 en el encendido, el módulo de CPU maestro (10) está adaptado para averiguar el número y atributos de todos los
 módulos de CPU (10, 20, 30, 40) presentes dentro del sistema de PLC y para establecer recuentos de tiempo
 que definen un respectivo intervalo de tiempo para acceder a un panel posterior para cada uno de los módulos
 de CPU,
 10 el módulo de CPU maestro (10) está adaptado para transmitir los recuentos de tiempo a todos los otros módulos
 de CPU (20,30,40),
 cada módulo de CPU (10, 20, 30, 40) incluye un detector configurado para detectar una señal de reloj generada
 por un módulo de CPU que accede al panel posterior, cada módulo de CPU (10, 20, 30, 40) está adaptado para
 acceder al panel posterior durante intervalo de tiempo asignado si la señal de reloj no es detectada por el
 15 detector, y cada módulo de CPU (10, 20, 30, 40) que finaliza su acceso al panel posterior está adaptado para
 detener la generación de la señal de reloj después de lo cual el módulo de CPU maestro (10) está adaptado para
 sincronizar los recuentos de tiempo iniciando desde cero el recuento de tiempo, en el que los recuentos de
 tiempo definen "recuento máximo de intervalo de tiempo", "recuento actual de intervalo de tiempo", un respectivo
 "recuento de inicio de intervalo de tiempo" y un respectivo "recuento de fin de intervalo de tiempo" para cada
 20 módulo de CPU (10, 20, 30, 40).

2. El sistema de PLC de la reivindicación 1, en el que un módulo de CPU que intenta acceder al panel posterior no
 accede al panel posterior cuando una señal de reloj es detectada por el detector.

25 3. El sistema de PLC de la reivindicación 1 o 2, en el que el módulo de CPU maestro (10) sincroniza recuentos de
 intervalo de tiempo con un recuento de inicio cuando uno de los módulos de CPU (10, 20, 30, 40) que accedió al
 panel posterior finaliza su acceso y detiene la generación de una señal de reloj.

30 4. El sistema de PLC de una cualquiera de las reivindicaciones 1 a 3, en el que el módulo de CPU maestro (10)
 determina el intervalo de tiempo teniendo en cuenta el número y atributos de los módulos de CPU (10, 20, 30, 40)
 presentes dentro del sistema de PLC.

35 5. El sistema de PLC de una cualquiera de las reivindicaciones 1 a 4, en el que el módulo de CPU maestro (10)
 transmite los recuentos de tiempo que definen un recuento máximo de intervalo de tiempo y un recuento actual de
 intervalo de tiempo a todos los otros módulos de CPU y transmite un respectivo recuento de inicio de intervalo de
 tiempo y un respectivo recuento de fin de intervalo de tiempo a cada de los módulos de CPU.

40 6. Un método de control de sistema de PLC que tiene una pluralidad de módulos de CPU (10, 20, 30, 40), el método
 caracterizado por que comprende:

un módulo de CPU maestro (10) que establece recuentos de tiempo que definen un respectivo intervalo de
 tiempo para acceder a un panel posterior para cada uno de los módulos de CPU (10, 20, 30, 40) basándose en el
 número y atributos de todos los módulos de CPU (10, 20, 30, 40) presentes dentro del sistema de PLC (S72); y
 cada módulo de CPU (10, 20, 30, 40) accediendo al panel posterior durante intervalo de tiempo asignado
 45 basándose en los recuentos de tiempo recibidos desde el módulo de CPU maestro (10), en el que cada módulo
 de CPU (10, 20, 30, 40) incluye un detector configurado para detectar una señal de reloj generada por un módulo
 de CPU que accede al panel posterior,
 cada módulo de CPU (10, 20, 30, 40) accediendo al panel posterior (S78) durante intervalo de tiempo asignado si
 la señal de reloj no es detectada por el detector (S75),
 50 cuando cada módulo de CPU (10, 20, 30, 40) finaliza su acceso al panel posterior (S80), detener la generación
 de la señal de reloj (S81) y sincronizar mediante el módulo de CPU maestro (10) los recuentos de tiempo
 iniciando desde cero el recuento de tiempo (S82), y definiendo los recuentos de tiempo "recuento máximo de
 intervalo de tiempo", "recuento actual de intervalo de tiempo", un respectivo "recuento de inicio de intervalo de
 tiempo" y un respectivo "recuento de fin de intervalo de tiempo" para cada módulo de CPU (10, 20, 30, 40).

55 7. El método de la reivindicación 6, que comprende adicionalmente: por cada módulo de CPU (10, 20, 30, 40) que no
 accede al panel posterior cuando una señal de reloj es detectada por el detector en un recuento que corresponde al
 intervalo de tiempo asignado (S77).

60 8. El método de la reivindicación 7, que comprende adicionalmente: después de no acceder al panel posterior
 cuando una señal de reloj es detectada, esperar para acceder al panel posterior en un intervalo de tiempo de
 siguiente recuento.

65 9. El método de una cualquiera de las reivindicaciones 6 a 8, comprendiendo además generar una señal de reloj en
 un momento de acceso al panel posterior para cada módulo de CPU (S79).

10. El método de la reivindicación 9, comprendiendo además finalizar la generación de señal de reloj en un momento cuando finaliza el acceso al panel posterior.

5 11. El método de una cualquiera de las reivindicaciones 6 a 10, en el que el intervalo de tiempo se determina teniendo en cuenta el número y atributos de la pluralidad de módulos de CPU conectados al panel posterior.

FIG.1

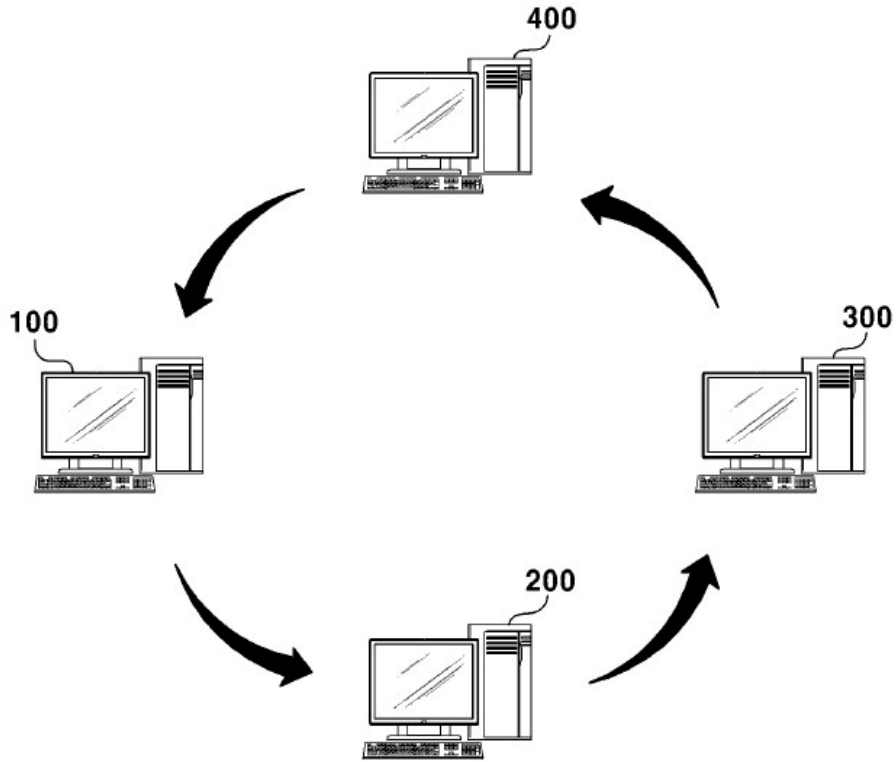


FIG.2

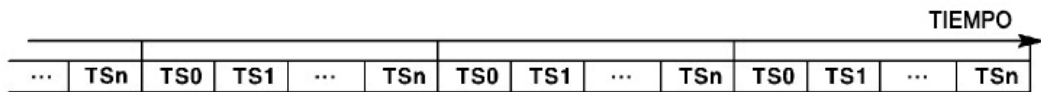


FIG.3

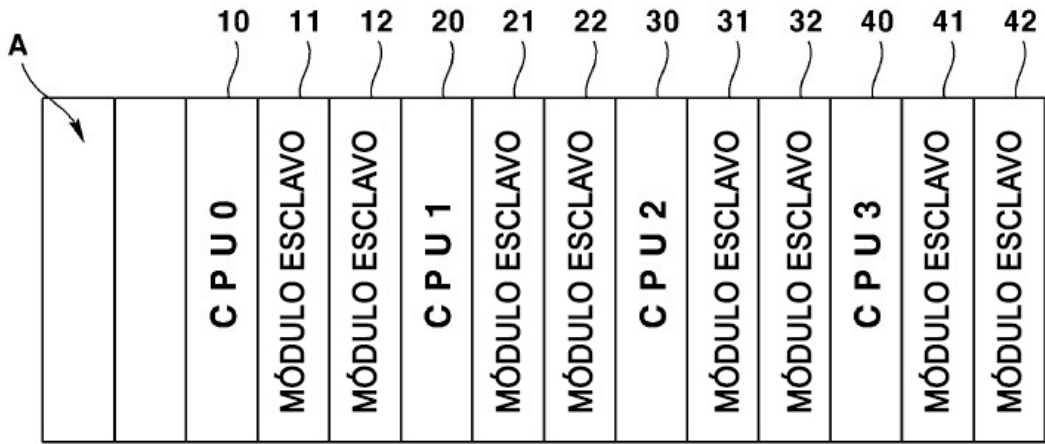


FIG.4

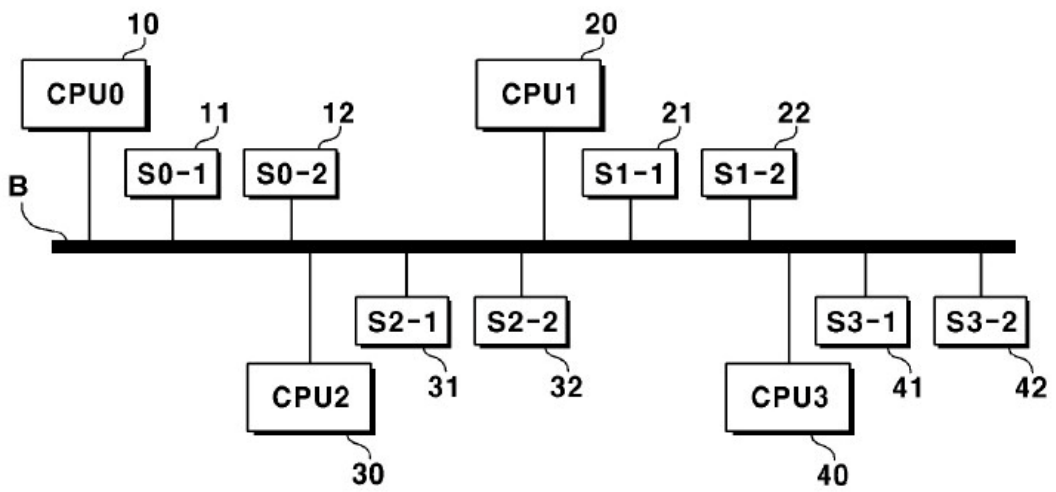


FIG.5

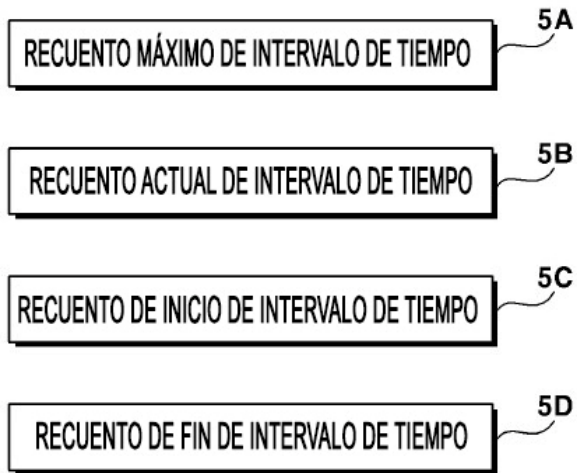


FIG.6

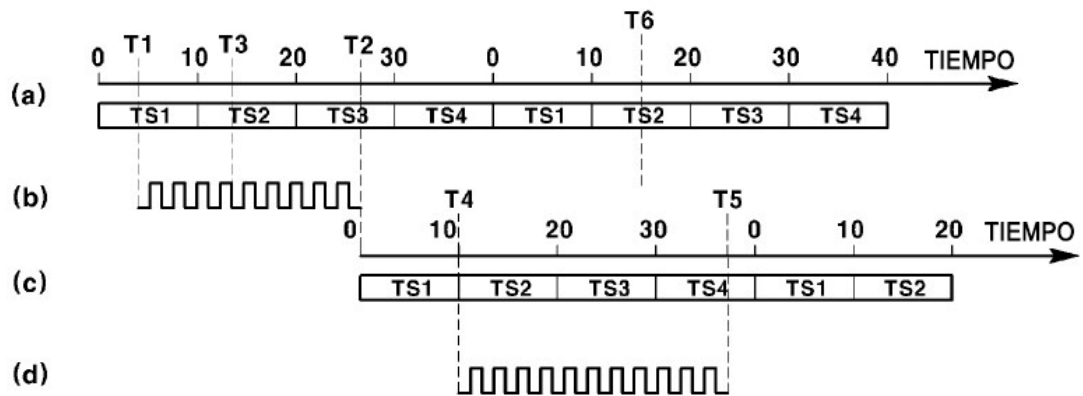


FIG.7

