

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 771 799**

51 Int. Cl.:

G05B 19/05 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **13.05.2015 E 15167613 (7)**

97 Fecha y número de publicación de la concesión europea: **27.11.2019 EP 2947526**

54 Título: **Aparato para la gestión de programas de controlador lógico programable**

30 Prioridad:

19.05.2014 KR 20140059503

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

07.07.2020

73 Titular/es:

**LSIS CO., LTD. (100.0%)
127 LS-ro, Dongan-gu
Anyang-si, Gyeonggi-do 431-848, KR**

72 Inventor/es:

LEE, SANG HUN

74 Agente/Representante:

SÁNCHEZ SILVA, Jesús Eladio

ES 2 771 799 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

Aparato para la gestión de programas de controlador lógico programable

5 Campo técnico

Las enseñanzas de acuerdo con realizaciones ejemplares y no limitantes de la presente descripción se refieren generalmente a un aparato para la gestión de programas de usuario de PLC (Controlador lógico programable) y un módulo de CPU (Unidad de procesamiento central) de PLC, y más particularmente, a un aparato para la gestión de programas de usuario de PLC configurado para usar de manera más eficiente un espacio de memoria mediante la asignación de un bloque de memoria adecuado al tamaño del programa de usuario, y para modificar/agregar un programa de usuario en ejecución sin detener un PLC operativo.

15 Técnica anterior

Un PLC (controlador lógico programable) es un factor esencial para la automatización de fábrica, y puede usarse de manera variable en campos industriales. Un programa de usuario para PLC puede modificarse o agregarse según sea necesario, y el programa de usuario modificado debe ejecutarse descargándose al PLC. En este momento, es necesario modificar o agregar funciones del programa de usuario sin detener el sistema de control. Para este fin, se dispone un espacio (asignación o espacio libre) en un bloque de memoria montado en cada programa de usuario, se carga una porción modificada en el espacio para permitir que el programa de usuario se modifique incluso durante la operación del PLC.

25 Sin embargo, el tamaño del bloque de memoria asignado a cada programa de usuario se fija convencionalmente, independientemente del tamaño de cada programa de usuario, con la desventaja de una utilización deteriorada del espacio no utilizado, y con la desventaja de no permitir más la modificación del programa cuando no se obtiene espacio para modificar programas.

30 La publicación US 2010/0174876 A1 describe un método para administrar una región de almacenamiento de memoria utilizada por un procesador. El procesador está conectado a la memoria que almacena los datos a los que se accede mientras se ejecuta una tarea. El método de gestión de memoria incluye las etapas de: dividir el área de memoria de la memoria en bloques que tienen una pluralidad de tamaños diferentes; seleccionar un bloque que tenga un tamaño que coincida con el tamaño de los datos a los que se accede mientras se ejecuta la tarea; y almacenar los datos a los que se accede mientras se ejecuta la tarea en el bloque seleccionado.

35 La publicación US 7 110 837 B2 describe un sistema de control que incluye un controlador programable que tiene una unidad de CPU que realiza operaciones cíclicas de procesos. Se proporciona una memoria de usuario temporal adicional además de una memoria de usuario normal. Cuando se va a realizar una edición en línea de un programa de usuario, el programa se almacena en ambas memorias y un motor de ejecución de instrucciones llama al programa almacenado en la memoria temporal. Después de completar la edición en línea en el programa de usuario almacenado en la memoria del usuario, la ejecución del programa se basa en el programa de usuario editado almacenado en la memoria del usuario.

45 Descripción

Problema técnico

Por consiguiente, la presente invención se proporciona para resolver los problemas/desventajas mencionados anteriormente y, por lo tanto, es un objeto de la presente invención proporcionar un aparato para la gestión de programas de usuario de PLC configurado para mejorar la utilización del espacio no utilizado mediante la formación de tamaños variables de bloques de memoria asignados a cada programa de usuario, y realizar modificaciones o adiciones de programas flexibles.

55 Solución técnica

El problema antes mencionado se resuelve con las características de la reivindicación independiente 1. Otras realizaciones son el tema de las reivindicaciones dependientes.

60 Para lograr el objetivo anterior, en un aspecto general de la presente descripción, se proporciona un aparato para la gestión de programas de PLC (Controlador lógico programable), el aparato comprende:

un medio de compilación configurado para generar un código de ejecución compilando un código fuente;
un medio de asignación de bloques configurado para asignar un bloque de memoria adecuado al tamaño del código de ejecución, basado en la información de configuración de memoria con respecto a qué bloques de memoria se forma una memoria;
65 medios de enlace configurados para organizar el código de ejecución en una posición de un bloque de memoria

asignado a través de los medios de asignación de bloques;
medios de generación de tabla de ejecución configurado para generar tabla de ejecución registrada con orden de ejecución e información de posición de cada código de ejecución; y medios de descarga configurados para descargar el código de ejecución y la tabla de ejecución procesada por los medios de enlace al PLC.

5 Preferiblemente, pero no necesariamente, el tamaño del bloque de memoria puede formarse en un número entero de tamaño básico preestablecido (N).

10 Preferiblemente, pero no necesariamente, los medios de asignación de bloques pueden asignar el bloque de memoria tal como es, o un nuevo bloque de memoria en respuesta al tamaño del código de ejecución modificado y el tamaño del bloque de memoria asignado al código de ejecución antes de la modificación cuando se modifica un código fuente.

15 Preferiblemente, pero no necesariamente, el tamaño del bloque de memoria puede formarse con un número entero de tamaño básico preestablecido (N), donde el medio de asignación de bloques busca un bloque de memoria efectivo aumentando el tamaño hasta en una unidad, comenzando desde 'CEIL (tamaño modificado del código de ejecución + N) x N', cuando CEIL (x) es una función que representa un número entero mínimo mayor que x,

20 Preferiblemente, pero no necesariamente, los medios de asignación de bloques pueden reasignar un bloque de memoria con respecto a cada código de ejecución, en base a la información de configuración de memoria, cuando no se puede encontrar un nuevo bloque de memoria.

25 Preferiblemente, pero no necesariamente, el aparato puede comprender además un módulo de CPU de PLC configurado para recibir el código de ejecución y la tabla de ejecución desde los medios de descarga, en donde el módulo de CPU ejecuta secuencialmente el código de ejecución de la memoria con referencia a la tabla de ejecución.

Efectos ventajosos

30 Las realizaciones ejemplares de un aparato para la gestión de programas de PLC de acuerdo con la presente descripción tienen un efecto ventajoso porque se configura un bloque de memoria de programa que tiene tamaños respectivamente diferentes, tales como N, 2N y 4N, y un bloque de memoria de un tamaño adecuado para el tamaño de se asigna el programa de usuario. Por lo tanto, la utilización del espacio sin uso se puede mejorar sobre un método de asignación convencional de tamaño fijo.

35 Otro efecto ventajoso es que un área de ejecución del programa puede configurarse de manera discontinua utilizando una tabla de ejecución, por lo que la ejecución selectiva del programa está habilitada para modificar o agregar un programa de usuario durante la operación del PLC.

Descripción de dibujos

40 La Figura 1 es un diagrama de bloques esquemático que ilustra un aparato para la gestión de programas de PLC y un módulo de CPU de acuerdo con una realización ejemplar de la presente descripción.

La Figura 2 es un ejemplo de un programa de usuario.

La Figura 3 es un ejemplo para explicar la información de configuración de memoria.

La Figura 4 es un ejemplo para explicar la asignación de bloques de memoria y la tabla de ejecución.

45 Las Figuras 5 y 6 son ejemplos detallados de un método para asignar bloques de memoria cuando se modifica un programa de usuario.

Mejor modo

50 A continuación, se describirá en detalle un aparato para la gestión de programas de usuario de PLC de acuerdo con realizaciones ejemplares de la presente descripción con referencia a los dibujos adjuntos.

55 Con referencia a la Figura 1, un PLC (Controlador lógico programable) está interconectado con un aparato para la gestión de programas de usuario (10, en lo sucesivo denominado "aparato"), que es un dispositivo externo separado.

El aparato (10) en la presente descripción define un dispositivo configurado para gestionar un programa aplicable por un usuario a un PLC (20) independientemente de su nomenclatura, y un programa de usuario puede descargarse al PLC (20). El aparato (10) puede realizar diversas funciones para operar y gestionar el PLC (20) siempre que el aparato pueda realizar una función relacionada con la gestión de programas explicada más adelante.

60 El aparato (10) puede instalarse con una herramienta de aplicación configurada para realizar funciones tales como la preparación del programa de escalera por un usuario y la habilitación de la descarga al PLC (20) compilando el programa preparado.

65 El aparato (10) puede configurarse mediante un dispositivo informático, como una computadora de escritorio o una computadora portátil. El aparato (10) puede instalarse con herramientas de aplicación configuradas para realizar

funciones tales como la preparación de un programa de escalera por un usuario, y la habilitación de la descarga al PLC (20) compilando el programa preparado.

5 El PLC (20) puede incluir varios tipos de módulos, como un módulo de CPU (21), un módulo de alimentación (22) y un módulo de E/S (entrada/salida).

10 El módulo de CPU (21) realiza una función de una unidad básica de PLC (20), y puede incluir un procesador de operación como una CPU (Unidad de procesamiento central (21-1) y una memoria (21-2) grabada con código de ejecución de un programa de usuario. El módulo de CPU (21) define un módulo de control principal configurado para gestionar el PLC (20) ejecutando un programa de operación principal del PLC (20) independientemente de su nomenclatura. La Figura 1 ilustra un ejemplo en donde el aparato (10) y el módulo de CPU (21) están interconectados a través de cada módulo de comunicación (10-7, 21-7). El aparato (10) y el módulo de CPU (21) pueden comunicarse a través de varios métodos de comunicación tales como comunicación en serie y Ethernet.

15 Con referencia a la Figura 2, un programa de usuario (18) del PLC (20) generalmente puede incluir varios programas de bajo nivel tales como un programa 1, un programa 2, un programa 3 y un programa 4, y un código fuente de un programa de usuario preparado por un usuario en un entorno de desarrollo de programa PLC puede descargarse al PLC (20) compilándose en un código de ejecución. El aparato (10) según la presente descripción puede incluir un medio de compilación (11), un medio de asignación de bloques (12), un medio de enlace (13), un medio de generación de tabla de ejecución (14) y un medio de descarga (15).

20 Los medios de compilación (11) pueden funcionar para generar un código de ejecución (10-2) compilando el código fuente (10-1) de un programa de usuario, donde el código de ejecución define un tipo de código procesable por un procesador de operación del PLC (20).

25 Los medios de asignación de bloque (12) pueden funcionar para asignar un bloque de memoria en un código de ejecución con referencia a la información de configuración de memoria (10-3). Es decir, los medios de asignación de bloques (12) pueden buscar un bloque de memoria adecuado al tamaño del código de ejecución compilado por los medios de compilación (11), y asignar el bloque de memoria buscado a un código de ejecución relevante.

30 La información de configuración de memoria puede definir información que indica con qué bloques de memoria se debe configurar una memoria. Cada bloque de memoria puede tener varios tamaños en lugar de tener el mismo tamaño. Para ser más específico, el tamaño del bloque de memoria puede configurarse con un número entero de tamaño básico preestablecido (N).

35 La Figura 3 es un ejemplo para explicar la información de configuración de memoria, donde el bloque de memoria tiene un tamaño de N, 2N, 4N, etc., y la información de configuración de memoria puede configurarse para tener una estructura que incluye cuatro bloques de memoria (31-34) cuyo tamaño es N, 2 bloques de memoria (35, 36) cuyo tamaño es 2N, y un bloque de memoria (37) cuyo tamaño es 4N en el orden de la dirección de la memoria.

40 La información de configuración de memoria puede configurarse de forma variable de cualquier manera según la estructura que configura una memoria por unidad de bloque.

45 Los medios de enlace (13) pueden funcionar para organizar un código de ejecución en una posición de un bloque de memoria asignado a través de los medios de asignación de bloques (12). Es decir, un código de ejecución de cada programa de usuario puede estar dispuesto de modo que se coloque en una dirección de memoria realmente posicionada a través de los medios de enlace (13). Por ejemplo, cuando el bloque de memoria (35) del primer tamaño 2N de la Figura 3 se asigna a un cierto código de ejecución, el medio de enlace (13) está configurado de tal manera que el código de ejecución se coloca en el bloque de memoria (35).

50 Los medios de generación de tabla de ejecución (14) pueden generar una tabla de ejecución (10-4) que contiene el orden de ejecución y la información de posición de cada código de ejecución. Aunque cada código de ejecución está dispuesto en un bloque de memoria asignado por los medios de asignación de bloques (12), la posición del bloque de memoria asignada por los medios de asignación de bloques (12) puede determinarse arbitrariamente por el tamaño del código de ejecución y el estado actual de uso de la memoria, y, por lo tanto, no hay garantía de que el código de ejecución de cada programa de usuario esté organizado en el orden de ejecución del mismo. Sin embargo, el código de ejecución debe ejecutarse en un orden deseado por un usuario cuando el PLC lo ejecuta realmente (20) y, por lo tanto, se requiere información sobre el orden de ejecución del programa y la posición de ejecución de cada programa, cuya función es realizada por la tabla de ejecución.

60 Ahora, haciendo referencia a la Figura 4, se describirá un ejemplo de tabla de ejecución.

65 Un programa de usuario en la tabla de ejecución se forma con un programa 1, un programa 21, un programa 3 y un programa 4, y los medios de asignación de bloques (12) asignan el programa 1, el programa 21, el programa 3 y el programa 4 a un bloque de memoria 5, un bloque de memoria 7, un bloque de memoria 1 y un bloque de memoria 2 respectivamente.

Entonces, no hay posibilidad de que los programas 1-4 se ejecuten en orden, porque el medio de enlace (13) organiza el programa 3 en el bloque de memoria 1, el programa 2 en el bloque de memoria 7, el programa 3 en la memoria bloque 1 y el programa 4 al bloque de memoria 2 respectivamente.

Para este fin, la tabla de ejecución (10-4) se registra con la información de posición del programa para la ejecución del orden de ejecución del programa y el orden relacionado, donde cada programa puede ejecutarse secuencialmente según el deseo de un usuario mediante el uso de la información. La tabla de ejecución (10-4) puede incluir varios datos necesarios para designar la orden y la posición de ejecución del programa. Por ejemplo, la tabla de ejecución (10-4) puede incluir la posición de inicio del bloque de memoria e información sobre el tamaño del código de ejecución dispuesto también en el bloque de memoria relevante. Los medios de descarga (15) pueden descargar el código de ejecución procesado por los medios de enlace (13) y la tabla de ejecución generada por los medios de generación de tabla de ejecución (14) al PLC (20). Mientras tanto, cuando se modifica el programa de usuario ya asignado con los bloques de memoria, los medios de asignación de bloques (12) pueden comparar el tamaño del código de ejecución modificado con el tamaño del bloque de memoria asignado al código de ejecución antes de la modificación, pueden asignar la memoria bloquear como es, o puede asignar un nuevo bloque de memoria.

Con referencia a la Figura 5, se ejemplificará una operación de medios de asignación de bloques (12) cuando se modifica un programa de usuario.

Los medios de asignación de bloques (12) determinan si hay un espacio suficiente para cargar un código de ejecución modificado en un bloque de memoria asignado a un programa de usuario relevante (S112), cuando el código de ejecución modificado se cambia a un código de ejecución a través de los medios de compilación (11) para ser asignado en bloque (S111).

Como resultado de la determinación en S112, cuando hay un espacio suficiente para un bloque de memoria actual, el bloque de memoria actual se asigna como está (S113), y cuando no hay espacio suficiente para el bloque de memoria actual, un proceso de se busca un nuevo bloque de memoria (S114).

Como resultado de realizar el proceso de búsqueda de un nuevo bloque de memoria, cuando se busca un nuevo bloque de memoria (S115), se asigna el nuevo bloque de memoria (S116) y se reorganiza un bloque de memoria cuando el nuevo bloque de memoria no se puede buscar (S117). La reorganización del bloque de memoria significa que todos los bloques de memoria se reasignan en cada código de ejecución.

Primero, el medio de asignación de bloques (12) busca un tamaño de bloque de búsqueda según " $\text{CEIL}(\text{tamaño modificado del código de ejecución} \div N) \times N$ ", donde $\text{CEIL}(x)$ es una función que representa un número entero mínimo

La Figura 6 muestra un ejemplo con respecto al proceso (S114) que busca un nuevo bloque de memoria en la Figura 5, donde se supone que el tamaño del bloque de memoria es un número entero de tamaño básico preestablecido (N) como se ilustra en la Figura 3 mayor que x (S114-1).

Además, se determina si queda un bloque de memoria efectivo (bloque de memoria de estado utilizable) en el tamaño del bloque de búsqueda (S114-2). Cuando se determina que un bloque de memoria efectivo en el tamaño del bloque de búsqueda permanece actualmente, entonces, se realiza la etapa de S116 de la Figura 5. Sin embargo, cuando se determina que un bloque de memoria efectivo en el tamaño del bloque de búsqueda no permanece actualmente, el tamaño del bloque de búsqueda aumenta en una etapa (S114-4), y el flujo continúa a la etapa S114-2, donde el proceso de determinar si aún permanece un bloque de memoria efectivo del tamaño del bloque de búsqueda se repite hasta que un valor del tamaño del bloque de búsqueda sea mayor que un tamaño de bloque máximo regulado (S114-5).

Cuando no se puede buscar la memoria de bloque efectiva, conduce a un fallo en la asignación de bloque de memoria, y se realiza la etapa de S117 de la Figura 5. Es decir, cuando no se busca un nuevo bloque de memoria, los medios de asignación de bloque (12) reasignan un bloque de memoria al código de ejecución de cada programa de usuario.

Cuando se modifica un determinado programa de usuario, la reorganización se realiza mediante los medios de enlace (13), el código de ejecución reordenado y una tabla de ejecución modificada se generan de nuevo y se descargan al PLC (20) a través de los medios de descarga (15).

Mientras tanto, el módulo de CPU (21) puede incluir un ejecutor de programas (21-3) que ejecuta de forma secuencial y repetitiva códigos de ejecución de varios programas de usuario grabados en la memoria (21-2), y el módulo de CPU (21) puede mantener una tabla de ejecución (21-4) descargada del aparato (10). Además, el módulo de CPU (21) puede cargar el código de ejecución del programa de usuario descargado desde el aparato (10) en su propia memoria (21-2). Posteriormente, el ejecutor de programas (21-3) puede ejecutar secuencialmente el código de ejecución de la memoria (21-2) con referencia a la tabla de ejecución (21-4).

Como se explica a través del ejemplo de la Figura 4, el ejecutor de programas (21-3) puede ejecutar el programa de

usuario (código de ejecución) en un área real de ejecución del programa de la memoria (21-2) utilizando la información almacenada en la tabla de ejecución.

5 En vista del ejemplo de la Figura 4 en un aspecto de memoria, aunque el ejecutor de programas (21-3) se carga secuencialmente con el programa 3, el programa 4, el programa 1 y el programa 2, cuando se ejecuta con referencia al orden almacenado en la tabla de ejecución, la ejecución se realiza en el orden del código de ejecución grabado en los bloques de memoria 5, 7, 1 y 2, y la ejecución se realiza en el orden del programa 1 al programa 4 a largo plazo.

10 Es decir, el ejecutor de programas (21-3) siempre puede ejecutar programas en un orden correcto, porque la posición de memoria grabada con los programas a ejecutar por la tabla de ejecución se designa de acuerdo con el orden. Además, la tabla de ejecución puede incluir un área de preservación para bloques de memoria no utilizados y, por lo tanto, el PLC (20) puede agregar fácilmente un nuevo programa de usuario en medio de la operación cuando se utiliza el área de preservación.

15 Por ejemplo, el ejecutor de programas (21-3) puede ejecutar un programa 5 recientemente agregado después de la ejecución del programa 4, cuando, por ejemplo, el nuevo programa 5 agregado se descarga del aparato (10), como en el ejemplo de la Figura 4, para agregarse al bloque de memoria 6 de la memoria (21-2), y se agrega una primera área de preservación de la tabla de ejecución con información de la misma.

20 La descripción ilustrativa anterior de la presente descripción se proporciona para permitir a cualquier persona experta en la técnica hacer o usar la invención. Varias modificaciones a la invención serán fácilmente evidentes para los expertos en la técnica, y los principios genéricos definidos en este documento pueden aplicarse a otras variaciones sin apartarse del alcance de la invención como se define en las reivindicaciones. Por lo tanto, la invención no pretende limitar los ejemplos descritos en el presente documento, sino que debe otorgarse al alcance más amplio consistente
25 con la invención tal como se define en las reivindicaciones.

Aplicabilidad industrial

30 Las realizaciones ejemplares de la presente descripción tienen aplicabilidad industrial en que un espacio de memoria puede usarse eficientemente asignando un bloque de memoria adecuado al tamaño del programa de usuario, y para modificar/agregar un programa de usuario en ejecución sin detener un PLC operativo.

REIVINDICACIONES

1. Un aparato (10) para la gestión de programas de PLC, controlador lógico programable, el aparato que comprende:
 - 5 un medio de compilación configurado para generar un código de ejecución compilando un código fuente (11); un medio de asignación de bloques (12) configurado para asignar un bloque de memoria adecuado al tamaño del código de ejecución, basado en la información de configuración de la memoria en cuanto a qué bloques de memoria se forma una memoria; y
 - 10 medios de enlace (13) configurados para organizar el código de ejecución en una posición de un bloque de memoria asignado a través de los medios de asignación de bloques, en donde, el aparato está caracterizado por:
 - medios de generación de tabla de ejecución (14) configurados para generar la tabla de ejecución grabada con el orden de ejecución y la información de posición de cada código de ejecución, en donde un módulo de CPU del PLC ejecuta el código de ejecución con referencia al orden almacenado en la tabla de ejecución; y
 - 15 medios de descarga (15) configurados para descargar el código de ejecución procesado por los medios de enlace (13) y la tabla de ejecución generada por los medios de generación de tabla de ejecución (14) al PLC.

2. El aparato de la reivindicación 1, caracterizado además porque el módulo de CPU (21) del PLC está configurado para recibir el código de ejecución y la tabla de ejecución desde los medios de descarga (15), en donde el módulo de CPU ejecuta secuencialmente el código de ejecución de la memoria con referencia a la tabla de ejecución.

3. El aparato de la reivindicación 1 o 2, caracterizado porque el tamaño del bloque de memoria se forma en un número entero de tamaño básico preestablecido (N).

4. El aparato de la reivindicación 1 o 2, caracterizado porque los medios de asignación de bloques (12) asignan el bloque de memoria tal como es o un nuevo bloque de memoria en respuesta al tamaño del código de ejecución modificado y al tamaño del bloque de memoria asignado al código de ejecución antes de la modificación cuando se modifica un código fuente.

5. El aparato de la reivindicación 4, caracterizado porque el tamaño del bloque de memoria se forma con un número entero de tamaño básico preestablecido (N), donde el medio de asignación de bloque (12) busca un bloque de memoria efectivo aumentando el tamaño hasta en una unidad, comenzando desde 'CEIL (modificado tamaño del código de ejecución ÷ N) x N', donde CEIL (x) es una función que representa un número entero mínimo mayor que x.

6. El aparato de la reivindicación 4, caracterizado porque los medios de asignación de bloque (12) reasignan un bloque de memoria en relación con cada código de ejecución, en base a la información de configuración de memoria, cuando no se puede encontrar un nuevo bloque de memoria.

Figura 1

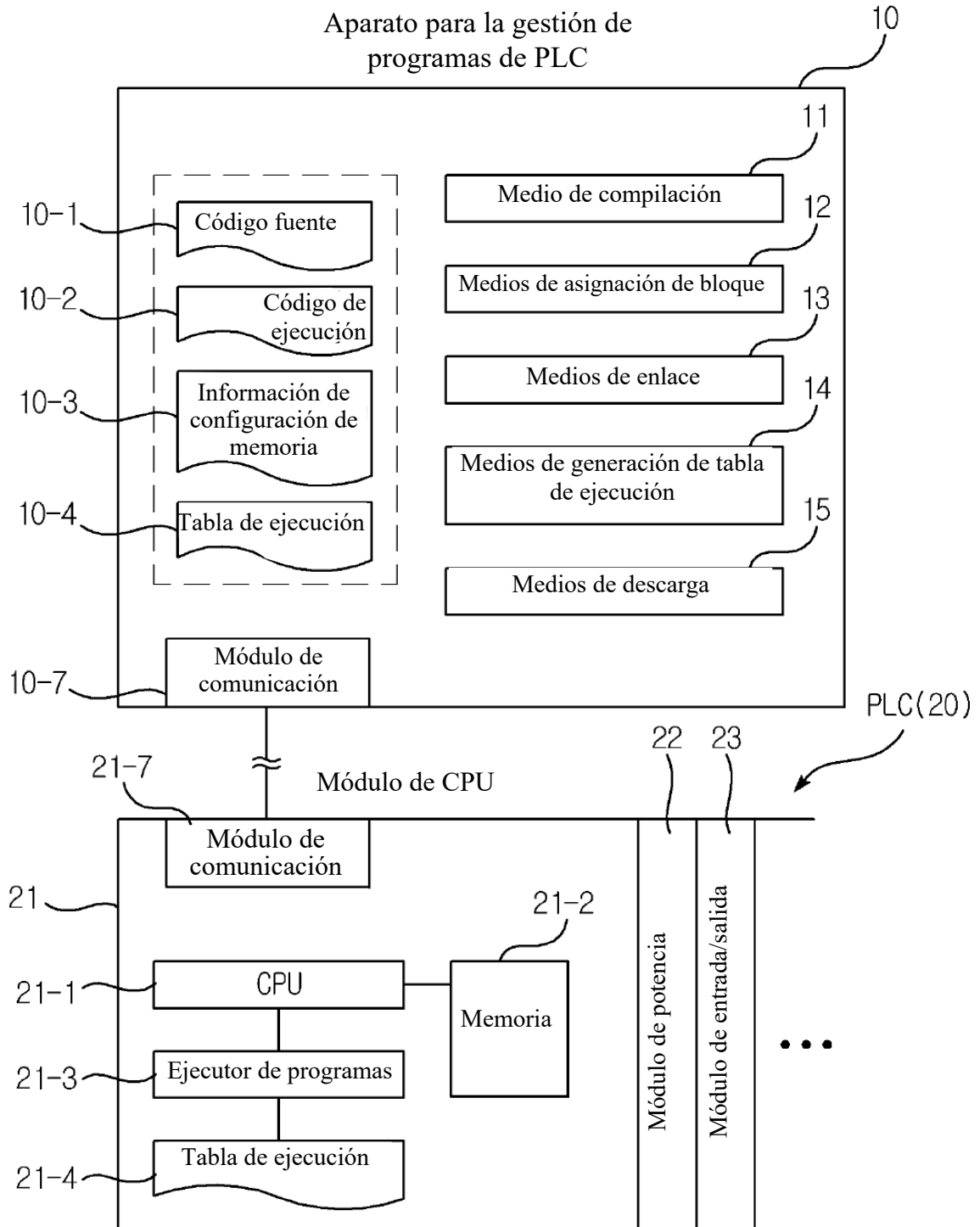


Figura 2

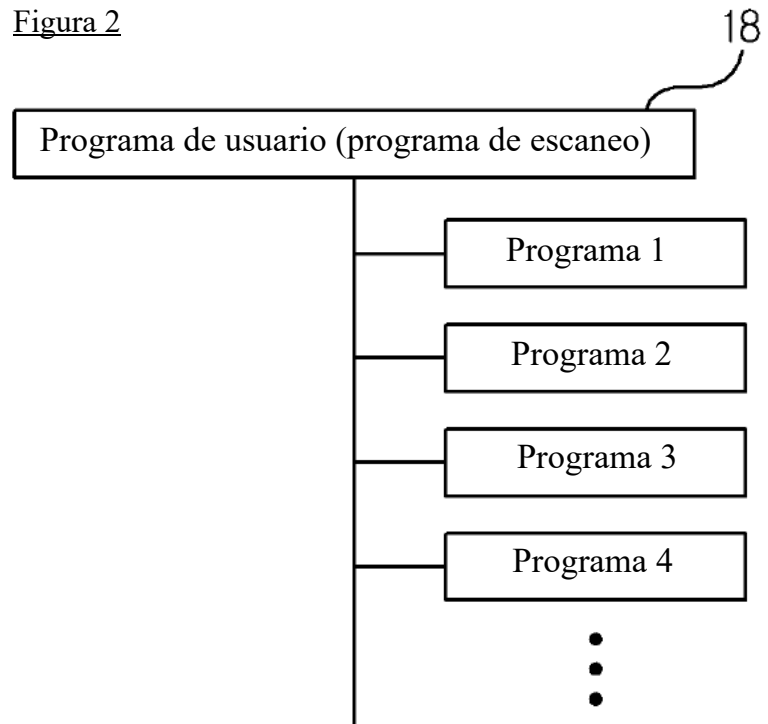


Figura 3

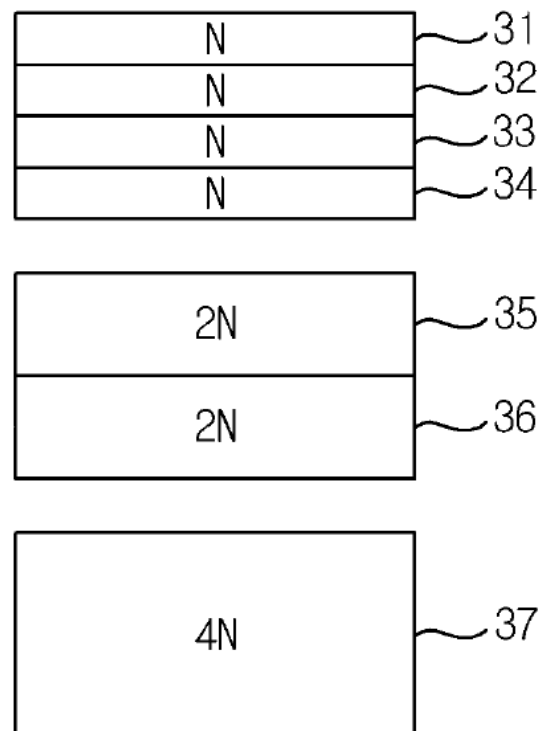


Figura 4

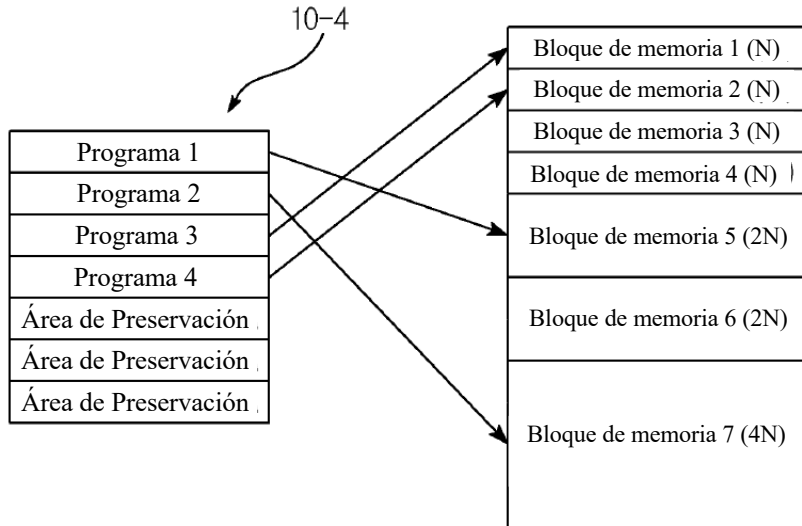


Figura 5

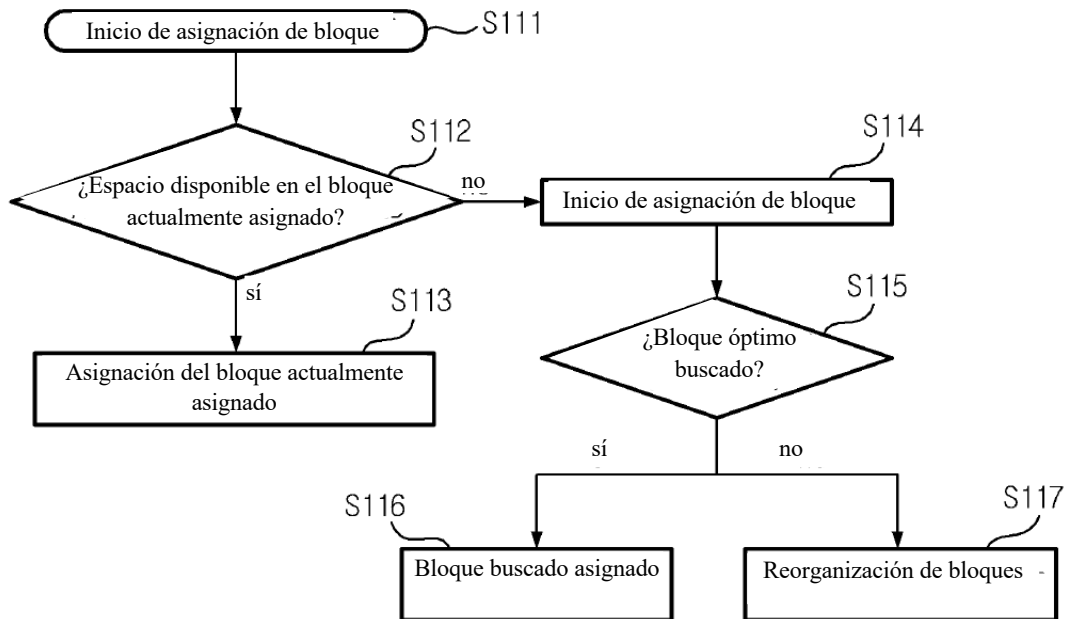


Figura 6

