

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 671 336**

51 Int. Cl.:

**G06F 12/02** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **30.05.2008 PCT/IB2008/001397**

87 Fecha y número de publicación internacional: **11.12.2008 WO08149194**

96 Fecha de presentación y número de la solicitud europea: **30.05.2008 E 08776262 (1)**

97 Fecha y número de publicación de la concesión europea: **27.12.2017 EP 2156298**

54 Título: **Método de gestión de la asignación de memoria flash en un token electrónico**

30 Prioridad:

**08.06.2007 EP 07301094**

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

**06.06.2018**

73 Titular/es:

**GEMALTO SA (100.0%)  
6, rue de la Verrerie  
92190 Meudon, FR**

72 Inventor/es:

**DEHLINGER, FRANCK;  
MARSEILLE, FRANÇOIS-XAVIER y  
CASTILLO, LAURENT**

74 Agente/Representante:

**CASANOVAS CASSA, Buenaventura**

**ES 2 671 336 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).

DESCRIPCION

Método de gestión de la asignación de memoria flash en un token electrónico

5 **(Campo de la invención)**

La presente invención se refiere a métodos para gestionar la asignación de memoria flash en un token electrónico. Se refiere particularmente a los métodos de gestión de la asignación de espacios en memoria flash como la memoria flash NOR.

10

**(Técnica anterior)**

15

Los mecanismos de asignación y recolección de basura utilizados con una memoria flash deben cumplir con las restricciones específicas de este tipo de memoria. Por ejemplo, el componente de memoria flash NOR tiene las siguientes características. Una flash NOR se divide en bancos que normalmente tienen un tamaño de 256 Kbytes. Cada banco se divide generalmente en sectores que normalmente tienen un tamaño de 4 Kbytes. Cada sector está dividido en páginas, que normalmente tiene un tamaño de 256 Bytes. Reescribir un byte en una página obliga a borrar primero la página completa que lo contiene. El acceso a la lectura de un byte es directo, ya que no es necesario cargar toda la página para acceder al byte. Para la operación de escritura, la granularidad de acceso es muy alta ya que se puede escribir un único bit. Una operación de escritura corresponde a convertir un bit de 1 a 0. Además, una página se puede programar/borrar un número limitado de veces.

20

25

Al comienzo del ciclo de vida, el contenido de dicha memoria flash se predetermina en un estado virgen, es decir, todos los bytes se inicializan con un único valor preestablecido. En general, este estado virgen corresponde al valor más pequeño o más grande que tomar un byte (0x00h o 0xFFh).

30

Las memorias Flash y EEPROM difieren mucho una de otra. Las EEPROM son lentas, de acceso por bytes para lecturas y escrituras, tipos de memorias, mientras que las memorias flash se pueden programar más rápidamente, pero solo se pueden borrar página a página, y su tamaño de página puede variar mucho.

35

El sistema de asignación diseñado para EEPROM no es adecuado para memorias flash, lo que provoca una gran cantidad de borrados innecesarios y conduce a bajos rendimientos tanto para asignaciones como para liberaciones. Además, esos sistemas de asignación no pueden usar las características de las memorias flash para lograr asignaciones más rápidas.

40

La invención tiene como objetivo minimizar el número de borrados requeridos para la asignación de espacio en una memoria administrada.

La WO00/49488 se refiere a un modo de gestión de asignación de memoria flash.

**(Sumario de la invención)**

45

El objeto de la presente invención es un método para gestionar la asignación de memoria flash en un token electrónico. Dicho token tiene una memoria que comprende un área de lista y un área administrada. Dicha área administrada comprende espacios asignados y al menos un fragmento de memoria libre. Dicha área de lista comprende al menos una entrada válida que referencia a un fragmento de memoria libre. Dicha entrada válida comprende un campo de estado.

50

Dicho método comprende los siguientes pasos:

- seleccionar un fragmento de memoria libre además de una solicitud de asignación, estando dicho fragmento de memoria libre referenciado por una entrada válida anterior,
- identificar un nuevo espacio asignado en el fragmento de memoria libre seleccionado.

55

Se predetermina el campo de estado de dicha entrada válida a estado virgen, y dicho método comprende el paso adicional de invalidar la entrada válida que referencia al fragmento de memoria libre seleccionado.

60

La entrada válida anterior que referencia al fragmento de memoria libre seleccionado se puede invalidar escribiendo un valor predefinido en el campo de estado de dicha entrada válida anterior.

65

Dicho método puede comprender la etapa adicional de solicitar una compactación del área de la lista cuando dicha área de la lista contiene un número predefinido de entradas.

Dichas entradas pueden inicializarse, de forma secuencial, en el área de la lista.

Cada uno de los espacios asignados y cada fragmento de memoria libre se pueden definir por su propio tamaño.

Una entrada válida anterior puede comprender un identificador de fragmento de memoria asociado a un primer fragmento de memoria libre. Dicho método puede comprender el paso adicional de inicializar una nueva entrada válida en el área de la lista si el tamaño del nuevo espacio asignado es menor que el tamaño del primer fragmento de memoria libre seleccionado. En este caso, dicha nueva entrada válida referencia a un segundo fragmento de memoria libre y tiene el mismo identificador de fragmento de memoria que el identificador asociado al primer fragmento de memoria libre.

La solicitud de asignación puede definirse por un tamaño. Dicho método puede comprender la etapa adicional de solicitar una compactación del área administrada cuando el tamaño de la solicitud de asignación es mayor que el tamaño de cada fragmento de memoria libre.

Otro objeto de la invención es un token electrónico que contiene un microprocesador y una memoria. Dicha memoria comprende un área de lista y un área administrada. Dicha área administrada comprende espacios asignados y al menos un fragmento de memoria libre. Dicha área de la lista comprende al menos una entrada válida que referencia a un fragmento de memoria libre. Dicha entrada válida comprende un campo de estado. El campo de estado de dicha entrada válida está predeterminado como en estado virgen. Dicho token comprende medios para identificar un nuevo espacio asignado en el área administrada, para inicializar una nueva entrada en el fragmento de memoria libre, y para invalidar la anterior entrada válida que referencia al fragmento de memoria libre donde se asigna el nuevo espacio.

Una entrada válida del área de lista puede comprender un identificador de fragmento de memoria asociado a un primer fragmento de memoria libre. Cuando se asigna un nuevo espacio en el primer fragmento de memoria libre, se puede inicializar una nueva entrada en el área de la lista. La nueva entrada puede referenciar a un segundo fragmento de memoria libre y tener el mismo identificador de fragmento de memoria que el identificador asociado al primer fragmento de memoria libre.

Cada entrada puede comprender al menos un campo de disponibilidad que indique si dicha entrada se utiliza o no. El valor no utilizado puede corresponder a un estado virgen del campo de disponibilidad.

El número de entradas inicializadas en el área de lista puede limitarse a un umbral preestablecido.

La memoria puede ser una memoria flash NOR.

El token puede ser una tarjeta inteligente, un teléfono móvil, un dispositivo de audio portátil, un dispositivo de navegación portátil o un token de memoria.

**(Breve descripción de los dibujos)**

Otras características y ventajas de la presente invención surgirán más claramente tras una lectura de la siguiente descripción de varias realizaciones preferidas de la invención con referencia a los correspondientes dibujos adjuntos, en los que:

- la Figura 1 representa esquemáticamente la estructura de un token electrónico de tipo de tarjeta inteligente según la invención;
- la Figura 2 representa un ejemplo de estructura para una entrada del área de lista según la invención;
- la Figura 3 es un algoritmo para gestionar la asignación de espacio en la memoria flash según la invención;
- La Figura 4 representa un ejemplo del estado de una memoria flash antes y después de una asignación de espacio de memoria en el área administrada;
- La Figura 5 representa un ejemplo del estado de una memoria flash antes y después de una compactación del área de la lista.

**(Descripción detallada de las realizaciones preferidas)**

La invención se puede aplicar a cualquier tipo de token electrónico. En esta descripción, el token electrónico es una tarjeta inteligente pero podría ser cualquier otro tipo de token electrónico o dispositivo portátil que utilice una memoria flash. El área de la lista se gestiona como una lista de entrada anotada en diario donde las entradas se inicializan secuencialmente en dicha lista.

La Figura 1 muestra la estructura de una tarjeta inteligente como un ejemplo de un token electrónico de acuerdo con una realización preferida de la invención. La tarjeta inteligente ET contiene un microprocesador MP, una interfaz de comunicación IN y una memoria flash ME. La memoria flash ME contiene un área de lista LA y un área administrada MA.

La memoria ME puede consistir en un único circuito o varios circuitos que pueden ser de diferentes tipos.

Como se muestra en la Figura 2, cada entrada del área de la lista puede tener una estructura que comprenda un

campo de estado SF, un campo de disponibilidad AF, un identificador de fragmento de memoria ID, un tamaño de fragmento de memoria CL y un intervalo de fragmento de memoria CO.

5 En una realización preferida, una entrada se codifica en cinco bytes. El campo de disponibilidad AF está codificado en un bit que indica si la entrada se utiliza o no. Por ejemplo, 1 significa que la entrada no se utiliza y 0 significa que se utiliza la entrada. El campo de estado SF se puede codificar en un bit que indica si la entrada es válida o inválida. Por ejemplo, 1 significa que la entrada es válida y 0 significa que la entrada no es válida.

10 El identificador de fragmento de memoria ID se puede codificar en seis bits. Cada entrada válida tiene un identificador único.

15 El tamaño del fragmento de memoria CL corresponde al tamaño del fragmento de memoria libre referenciado. El tamaño del fragmento de memoria CL puede codificarse en dos bytes y el intervalo del fragmento de memoria CO puede codificarse en dos bytes. El intervalo del fragmento de memoria CO corresponde al intervalo en el que se encuentra el fragmento de memoria libre referenciado en el área administrada MA.

20 La Figura 2 muestra un ejemplo de una entrada válida que tiene un identificador de fragmento de memoria ID igual a 5, y que refiere un fragmento de memoria libre de 11 bytes en un intervalo de 38 bytes desde el comienzo del área administrada MA.

25 La Figura 3 muestra un algoritmo para gestionar la asignación de espacio en la memoria flash de acuerdo con la invención. Primero, en el paso S1, se emite una solicitud de asignación. Esta solicitud contiene un tamaño solicitado. Posteriormente, en el paso S2 se realiza una comprobación para verificar si el área de la lista LA se encuentra llena y permite inicializar una nueva entrada.

Si el área de la lista LA está llena, se emite entonces una solicitud de compactación del área de lista LA en el paso S10.

30 Si el área de la lista LA no está llena, en la etapa S3 se inicia una búsqueda de un fragmento de memoria libre que tenga un tamaño igual o superior al tamaño solicitado.

35 Cuando se busca un fragmento de memoria libre, se realiza un escaneo del área de la lista LA. Para cada entrada, primero se verifica el campo de disponibilidad AF. Si el campo de disponibilidad AF se establece en el valor "usado", entonces se realiza una comprobación en el campo de estado SF. Si el campo de estado SF se establece en el valor "válido", se realiza entonces una comparación entre el tamaño del fragmento de memoria CL y el tamaño solicitado. El escaneo gira en bucle en el área de la lista LA hasta que se encuentre una entrada relevante o se llegue al final del área de la lista LA.

40 En el paso S4, si no se ha encontrado ningún fragmento de memoria libre, entonces se emite una solicitud de compactación del área administrada MA durante el paso S11. El área administrada MA puede estar muy fragmentada y potencialmente contiene muchos pequeños fragmentos libres.

45 Si se encuentra un fragmento de memoria libre, entonces el fragmento de memoria encontrado se selecciona en el paso S5.

Seguidamente, en el paso S6, se identifica un nuevo espacio asignado que tiene el tamaño solicitado en el fragmento seleccionado.

50 En el paso S7, se realiza una comprobación para verificar si queda memoria libre en el fragmento de memoria seleccionado que no resulte necesaria para el nuevo espacio de memoria asignado.

Si el tamaño del fragmento de memoria seleccionado es mayor que el tamaño requerido, entonces se inicializa una nueva entrada en el área de la lista LA en el paso S8.

55 A continuación, la entrada anterior que referencia al fragmento seleccionado se invalida en el paso S9, cualquiera que sea el resultado de la comprobación del paso S7.

60 En una realización alternativa, la comprobación del paso S2 se puede realizar después del paso S7 si el tamaño del fragmento seleccionado es mayor que el tamaño solicitado.

65 En la figura 4 se muestra un ejemplo de la división de un fragmento de memoria libre seleccionado. En un primer estado, el área de lista LA contiene dos entradas válidas EN1 y EN3 y una entrada inválida EN2. En este ejemplo, la cantidad máxima de entradas del área de la lista está limitada a cinco. El área administrada MA contiene dos fragmentos de memoria libres FC1 y FC2 y cuatro espacios de memoria asignados AS1, AS2, AS3 y AS4. Si el segundo fragmento libre FC2 se selecciona en el paso S5, y si el fragmento libre seleccionado tiene un tamaño superior al tamaño solicitado, la primera parte del fragmento seleccionado se puede dedicar al espacio AS5

5 asignado recientemente y la última parte del fragmento seleccionado permanece siendo un fragmento de memoria libre FC2. La entrada anterior EN3 que referencia al fragmento seleccionado FC2 se invalida al cambiar el valor de bit del campo de estado SF a 0. Una nueva entrada EN4 se inicializa en el área de lista LA al cambiar el valor de bit del campo de disponibilidad AF a 0. Esta nueva entrada EN4 referencia al fragmento libre restante FC2 con un tamaño de fragmento de memoria CL fijado en el tamaño anterior FC2 – el tamaño solicitado. La nueva entrada EN4 tiene el intervalo de fragmento de memoria CO fijado en el intervalo anterior FC2 + el tamaño solicitado.

10 La Figura 5 muestra un ejemplo de compactación del área de lista LA. En un primer estado, el área de lista LA contiene dos entradas válidas EN1 y EN5 y tres entradas inválidas EN2, EN3 y EN4. El área administrada MA contiene dos fragmentos de memoria libres FC1 y FC2 y cinco espacios de memoria asignados AS1, AS2, AS3, AS4 y AS5.

15 La compactación del área de la lista LA se puede realizar en una memoria de almacenamiento temporal volátil previamente inicializada a 0xFF. Cada entrada válida se carga entonces en la memoria de almacenamiento temporal y se contrae con otras si es necesario. Durante la operación de carga, las entradas se pueden organizar por intervalo referenciado en orden ascendente. El identificador de fragmento de memoria libre ID de cada entrada se modifica para reflejar su posición en la lista ordenada. Seguidamente, la memoria de almacenamiento temporal se vacía en el área de lista LA.

20 En el ejemplo de la figura 5, el área de lista LA contiene finalmente dos entradas válidas EN1 y EN2 y ninguna entrada no válida. El área administrada MA permanece sin cambios. La compactación del área de la lista LA evita el problema de saturación debido al número limitado de entradas en el área de lista LA.

25 Cada vez que se inicializa una nueva entrada en el área de la lista LA, se debe calcular un nuevo identificador de fragmento de memoria libre ID.

30 Si la nueva entrada corresponde a un nuevo fragmento memoria libre, entonces se calcula un nuevo identificador de fragmento de memoria libre ID para que el nuevo identificador de fragmento de memoria libre ID sea más alto que todos los identificadores de fragmentos de memoria libres existentes.

35 Si la nueva entrada corresponde a un fragmento de memoria libre previamente existente cuyo tamaño ha cambiado, entonces el nuevo identificador de fragmento de memoria libre ID es igual al nuevo identificador de fragmento de memoria libre ID correspondiente al fragmento de memoria libre existente previamente. En otras palabras, incluso si el tamaño de un fragmento de memoria libre cambia, el identificador de fragmento de memoria libre asociado ID permanece inalterado.

40 Alternativamente, la consistencia del área de lista LA se puede verificar al reiniciar la tarjeta inteligente o antes de acceder al área de lista LA. Se realiza un escaneo de todas las entradas válidas. Si dos entradas válidas poseen el mismo identificador de fragmento de memoria libre ID, entonces la entrada válida que se inicializó más recientemente en el área de lista LA debe considerarse como inválida. En consecuencia, su indicador de validez se borra a 0. Tal control de la consistencia del área de la lista LA permite evitar los efectos incómodos de una rotura o un corte de corriente.

45 Una ventaja de la invención es permitir un sistema de asignación que está diseñado para asignar atómicamente un nuevo espacio de memoria. En otras palabras, una única operación de programación básica permite validar la asignación de un espacio de memoria. En particular, dicho sistema de asignación evita la apertura de una transacción para garantizar la coherencia del sistema de asignación y evita fugas de memoria en caso de rotura.

50 Cuando el área administrada MA es una cantidad considerable de Java, otra ventaja de la invención es permitir el uso de fragmentos de memoria libres liberados durante la recolección de basura sin requerir una operación de compactación del área administrada.

REIVINDICACIONES

1. Un método para gestionar la asignación de memoria flash en un token electrónico (ET), disponiendo dicho token (ET) de una memoria (ME) que comprende un área de lista (LA) y un área administrada (MA), comprendiendo dicha área administrada (MA) espacios asignados (AS1, AS2, AS3) y al menos un fragmento de memoria libre (FC1, FC2), comprendiendo dicha área de lista (LA) al menos una entrada válida (EN1, EN3) que referencia un fragmento de memoria libre (FC1, FC2) comprendiendo dicha entrada válida (EN1, EN3) un campo de estado (SF), dicho método comprende los siguientes pasos:
- seleccionar (S5) un fragmento de memoria libre (FC2) adicional a una solicitud de asignación, estando referenciado dicho fragmento de memoria libre (FC2) por un entrada válida anterior (EN3),
  - identificar (S6) un nuevo espacio asignado (AS5) en el fragmento de memoria libre seleccionado (FC2),
- caracterizado porque** el campo de estado (SF) de dicha entrada válida (EN1, EN3) es pre-configurado como estado virgen, y **porque** dicho método comprende el paso adicional de invalidar (S9) la entrada válida anterior (EN3) que referencia al fragmento de memoria libre seleccionado (FC2).
2. Un método según la reivindicación 1, en el que la entrada anterior (EN3) que referencia al fragmento de memoria libre seleccionado (FC2) se invalida escribiendo un valor predefinido en el campo de estado (SF) de dicha entrada anterior (EN3).
3. Un método según las reivindicaciones 1 o 2, en el que dicho método comprende el paso adicional de solicitar (S10) una compactación del área de lista (LA) cuando dicha área de lista (LA) contiene un número predefinido de entradas.
4. Un método según las reivindicaciones 1, 2 o 3, en el que dichas entradas (EN1, EN2, EN3) son inicializadas secuencialmente en el área de lista (LA).
5. Un método según las reivindicaciones 1, 2, 3 o 4, en el que cada espacio asignado (AS1, AS2, AS3) y cada fragmento de memoria libre (FC1, FC2) están definidos por su propio tamaño, en el que una entrada válida anterior (EN3) comprende un identificador de fragmento de memoria (ID) asociado a un primer fragmento de memoria libre (FC2), y en que dicho método comprende el paso adicional de inicializar (S8) una nueva entrada válida (EN4) en el área de lista (LA) si el tamaño del nuevo espacio asignado (AS5) es más pequeño que el tamaño del primer fragmento de memoria libre (FC2), dicha nueva entrada válida (EN4) referenciando a un segundo fragmento de memoria libre (FC2) y teniendo el mismo identificador de fragmento de memoria (ID) que el identificador asociado al primero fragmento de memoria libre (FC2).
6. Un método según la reivindicación 5, en el que la solicitud de asignación se encuentra definida por un tamaño, y en el que dicho método comprende el paso adicional de solicitar (S11) una compactación del área administrada (MA) cuando el tamaño de la solicitud de asignación es mayor que el tamaño de cada fragmento de memoria libre (FC1, FC2).
7. Un token electrónico (ET) que contiene
- un microprocesador (MP),
  - una memoria (ME) que comprende un área de lista (LA) y un área administrada (MA), comprendiendo dicha área administrada (MA) espacios asignados (AS1, AS2, AS3) y al menos un fragmento de memoria libre (FC2), comprendiendo dicha área de lista (LA) al menos una entrada válida (EN3) que referencia a un fragmento de memoria libre (FC2), comprendiendo dicha entrada válida (EN3) un campo de estado (SF),
- caracterizado porque** el campo de estado (SF) de dicha entrada válida (EN3) está pre-configurado como estado virgen, y **porque** dicho token (ET) comprende medios para identificar un nuevo espacio asignado (AS5) en el fragmento de memoria libre (FC2), y para invalidar la entrada válida anterior (EN3) que referencia al fragmento de memoria libre (FC2) donde el nuevo espacio (AS5) es asignado.
8. Un token electrónico (ET) según la reivindicación 7, en el que una entrada válida (EN3) del área de lista (LA) comprende un identificador de fragmento de memoria (ID) asociado a un primer fragmento de memoria libre (FC2), y en el que cuando un nuevo espacio (AS5) es asignado en el primer fragmento de memoria libre (FC2), se inicializa una nueva entrada (EN4) en el área de lista (LA), referenciando dicha nueva entrada (EN4) un segundo fragmento de memoria libre (FC2) y teniendo el mismo identificador de fragmento de memoria (ID) que el identificador asociado al primer fragmento de memoria libre (FC2).
9. Un token electrónico (ET) según las reivindicaciones 7 u 8, en el que cada entrada comprende al menos un campo de disponibilidad (AF) que indica si dicha entrada se utiliza o no y en el que el valor no utilizado corresponde a un estado virgen del campo de disponibilidad (AF).

10. Un token electrónico (ET) según las reivindicaciones 7, 8 o 9, en el que el número de entradas inicializadas en el área de lista (LA) está limitado a un umbral pre-establecido.

5 11. Un token electrónico (ET) según las reivindicaciones 7, 8, 9 o 10, en el que la memoria (ME) es una memoria flash NOR.

12. Un token electrónico (ET) según las reivindicaciones 7, 8, 9, 10 u 11, **caracterizado porque** el token (ET) es una tarjeta inteligente, un teléfono móvil, un dispositivo de audio portátil, un dispositivo de navegación portátil o un token de memoria.

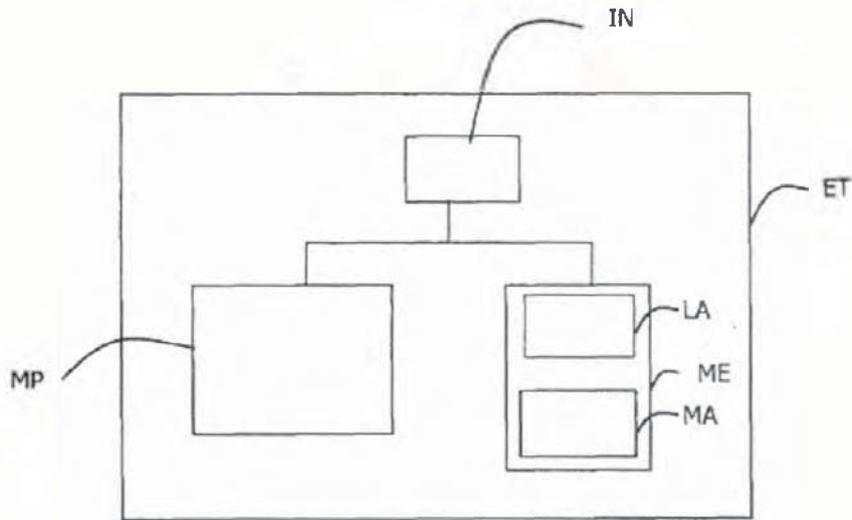


FIG. 1

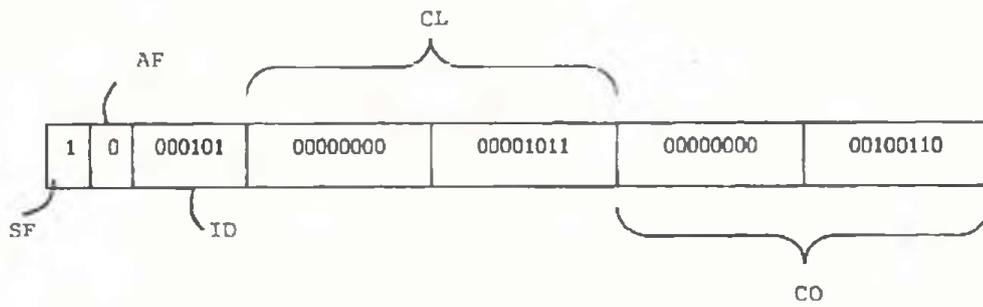


FIG. 2

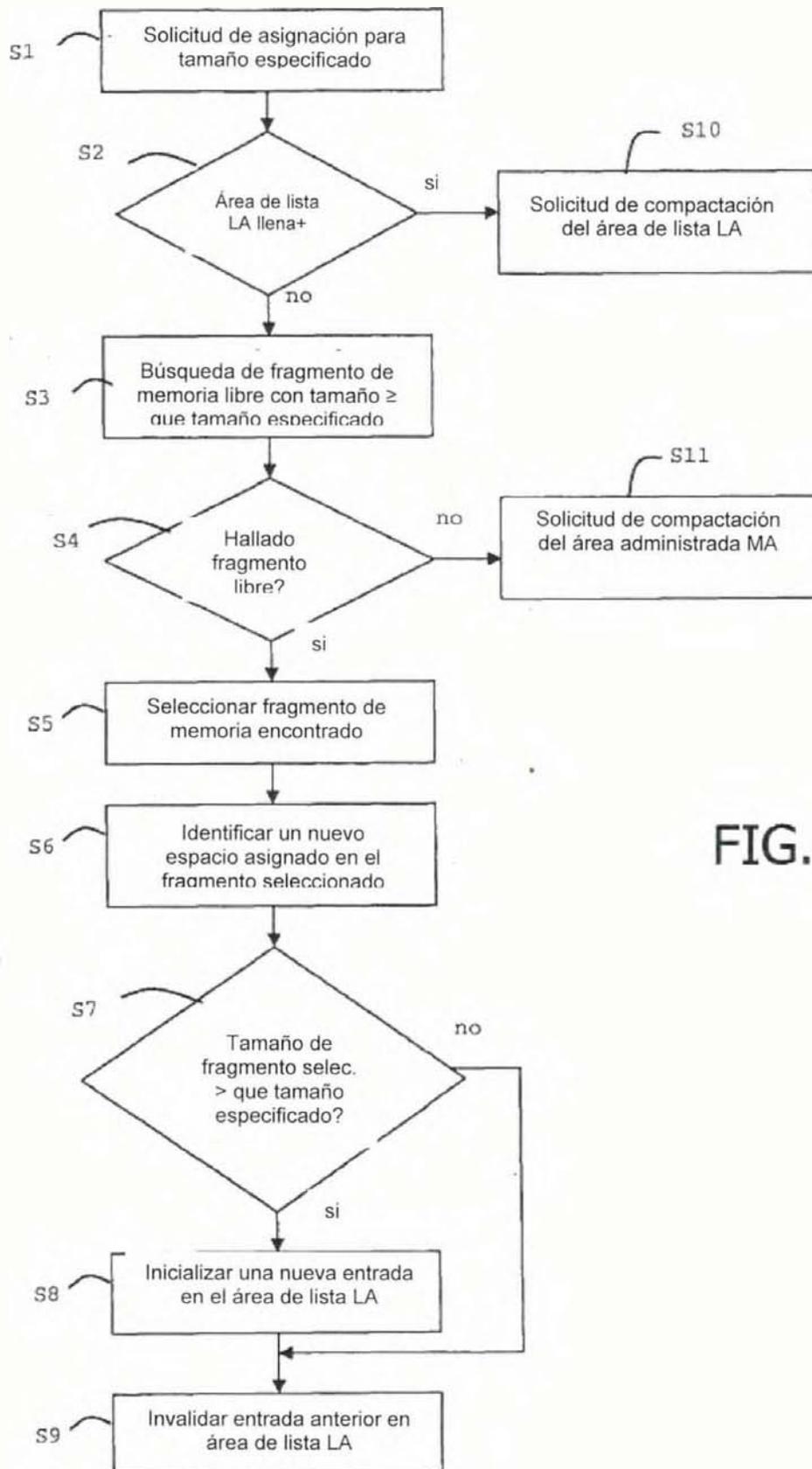


FIG. 3

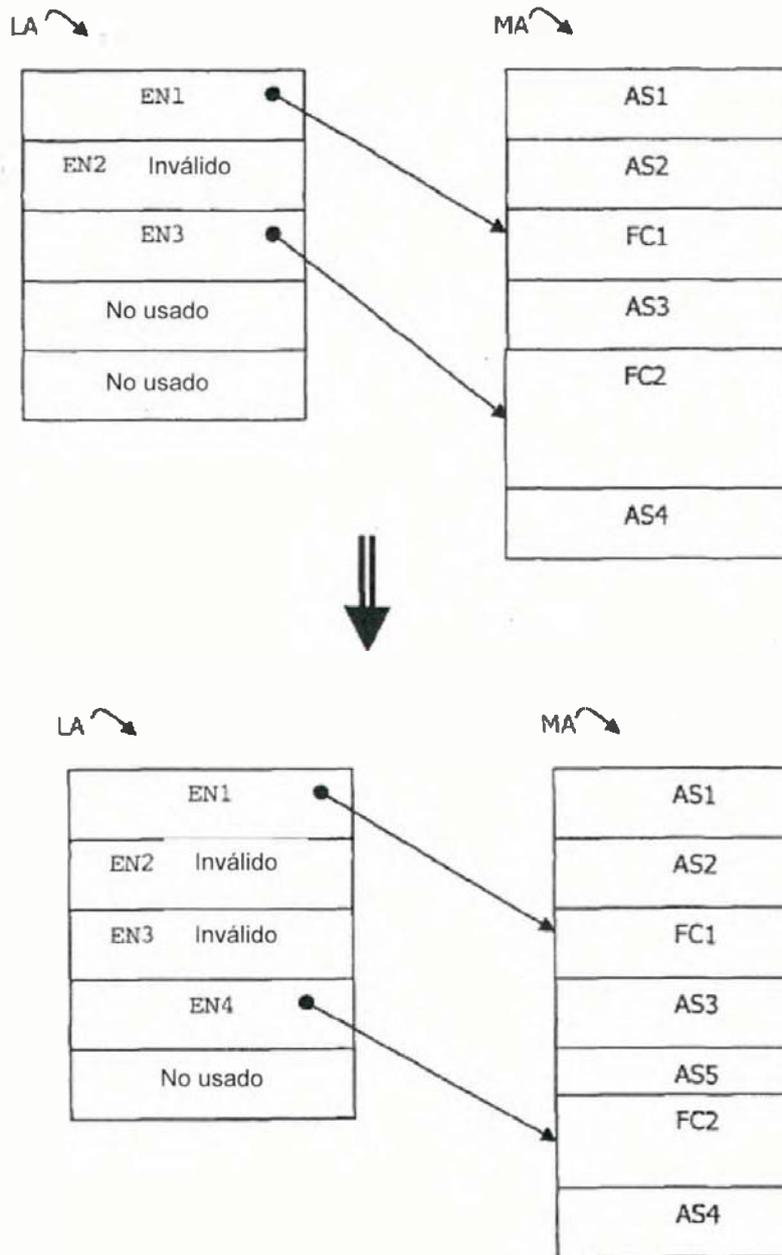


FIG. 4

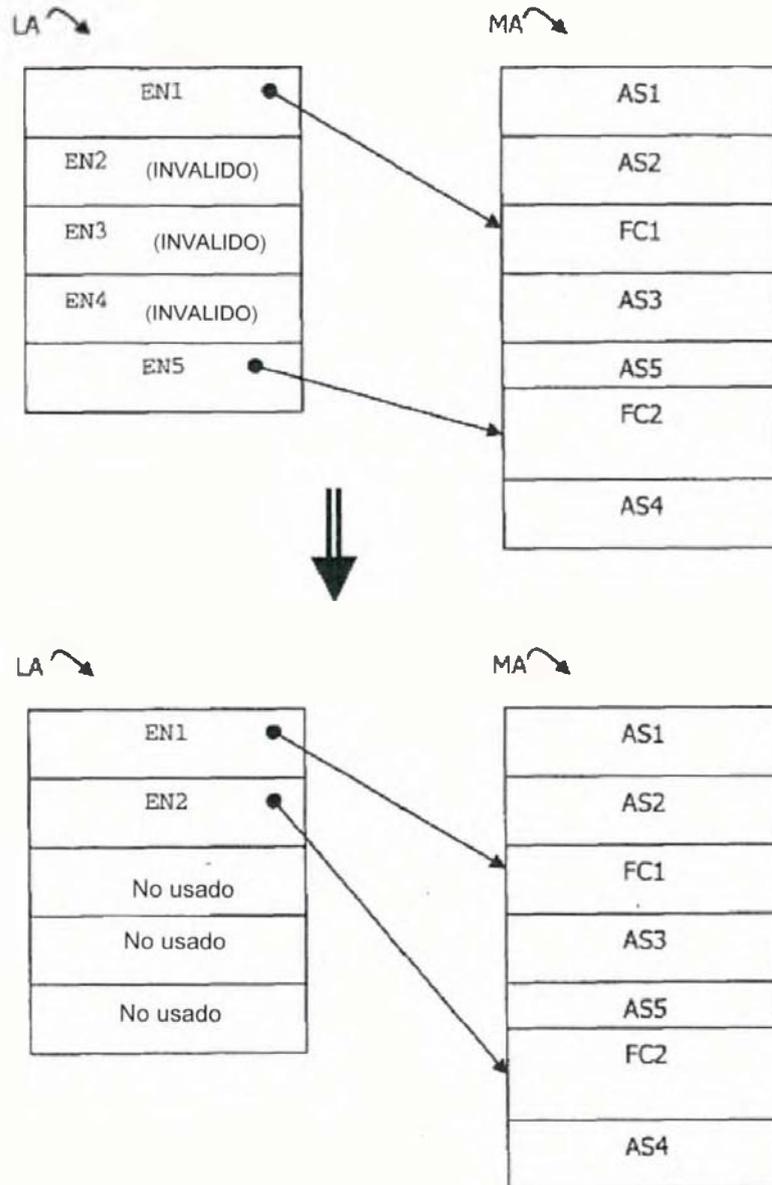


FIG. 5