



19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 335 863**

51 Int. Cl.:  
**G06F 12/02** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **01990601 .5**

96 Fecha de presentación : **20.12.2001**

97 Número de publicación de la solicitud: **1346287**

97 Fecha de publicación de la solicitud: **24.09.2003**

54 Título: **Procedimiento de asignación dinámica de memoria por bloques de memoria elementales en una estructura de datos, y sistema integrado correspondiente.**

30 Prioridad: **21.12.2000 FR 00 16722**

45 Fecha de publicación de la mención BOPI:  
**06.04.2010**

45 Fecha de la publicación del folleto de la patente:  
**06.04.2010**

73 Titular/es: **CP8 TECHNOLOGIES**  
**36-38, rue de la Princesse, B.P. 45**  
**78431 Louveciennes, FR**

72 Inventor/es: **Fougeroux, Nicolas;**  
**Hameau, Patrice y**  
**Landier, Olivier**

74 Agente: **Cañadell Isern, Roberto**

ES 2 335 863 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).

# ES 2 335 863 T3

## DESCRIPCIÓN

Procedimiento de asignación dinámica de memoria por bloques de memoria elementales en una estructura de datos, y sistema integrado correspondiente.

5 El invento se refiere a un método de asignación dinámica de memoria por bloques de memoria elementales en una estructura de datos, en un sistema informático integrado, todavía designado por sistema integrado, y al sistema integrado correspondiente.

10 Los sistemas integrados, u objetos portátiles actuales, como las tarjetas con microprocesador o de chip, las PCMCIA y los asistentes electrónicos tienden a efectuar funciones cada vez más complejas, especialmente en base a la implantación de programas de aplicación múltiples, siendo las tarjetas con microprocesador o microcontrolador, según el caso, designadas por tarjetas multi aplicación.

15 Debido a la polivalencia de los actuales lenguajes de alto nivel y de su adaptabilidad a los elementos materiales más variados en la aplicación concreta de máquinas virtuales, las aplicaciones implantadas en este tipo de objeto pueden ser muy diversas. Especialmente, y en el marco de la evolución de las tarjetas electrónicas con microprocesador o microcontrolador, en este tipo de tarjeta se ha previsto la instalación de un número considerable de aplicaciones tales como portamonedas electrónico PME, control de acceso físico, control de acceso a servicios, así como emisiones televisadas de pago o análogos.

20 La instalación de múltiples aplicaciones en estos objetos portátiles es posible, por un lado, debido a la miniaturización de los circuitos electrónicos y al aumento continuado de la capacidad de memoria de las tarjetas chip y, por otro lado, debido al incremento de la potencia de cálculo de las unidades centrales de tratamiento.

25 La añadidura de las aplicaciones instaladas en estos tipos de materiales plantea, no obstante, el inconveniente principal de una asignación dinámica de memoria optimizada.

30 Efectivamente, como se muestra en la figura 1, se recuerda que el objeto portátil, formado p. ej. de una tarjeta con micro procesador de referencia 10, habitualmente comprende los circuitos de entrada/salida, I/O, referencias 12, de las fuentes de tratamiento de información, referencia 14, formadas por un microcontrolador y unidos a los circuitos de entrada/salida 12. Asimismo, se prevé una memoria no volátil 18 formada por una memoria programable 18a y una memoria de tipo ROM, o memoria de tipo de acceso solo lectura 18b. Estas memorias están vinculadas al microcontrolador o microprocesador 14. Por último, también se ha previsto una memoria de trabajo, memoria RAM, referencia 16, vinculada al microprocesador. Se entiende que dichas conexiones o vínculos son enlaces por BUS.

35 El conjunto es administrado por un sistema de explotación OS, el cual puede implantarse en memoria no volátil 18. Igualmente, en determinados casos, el objeto portátil multiaplicación puede comportar una unidad de cálculo criptográfico SI, referencia 20, unida por sí misma al microprocesador 14.

40 El propio microprocesador puede reemplazarse o completarse por los circuitos lógicos implantados en un chip de semiconducción, pudiendo ser esos circuitos lógicos, circuitos de tipo ASIC (del inglés: Application Specific Integrated Circuit.)

45 En la patente WO 01/16759, registrada previamente a la presente demanda pero publicada *a posteriori*, se describe un método de asignación de memoria dividida en bloques de datos en una tarjeta chip.

El documento EP-A-0 686 918 describe una técnica de atribución, verificación y liberación de objeto que se considera muy cercana al invento.

50 El invento está definido en el repertorio adjunto de reivindicaciones.

55 En el marco de racionalización de la gestión del espacio de memoria disponible en todo el sistema informático, en concreto para un sistema integrado como una tarjeta con microprocesador o microcontrolador, como representado en el figura 1, el invento tiene por objeto la aplicación de un método de asignación dinámica de memoria por bloques de memoria elementales en una estructura de datos; dichas estructuras de datos corresponden ya sea a las aplicaciones como tal, ya sea a los datos del sistema que permiten la aplicación del sistema de explotación, o incluso de los datos específicos de seguridad, como las claves de acceso de cifrado/descifrado u otros. La noción de asignación dinámica recupera la atribución y/o la supresión de memoria atribuida.

60 Otro objetivo del invento, en el marco de la racionalización comentada, es igualmente el acceso más rápido al conocimiento de la pertenencia de cada octeto memorizado en zona memoria en la aplicación o estructura de datos a la que pertenece dicho octeto.

65 Otro objetivo del presente invento es también la disposición de un método dinámico de memoria por bloques de memoria elementales en una estructura de datos que permite evitar o minimizar cualquier fenómeno de diseminación, todavía designado fenómeno de fragmentación, durante el proceso de asignación/borrado de la memoria.

## ES 2 335 863 T3

Otro objeto del presente invento es, finalmente, la aplicación de un método de asignación dinámica de memoria por bloques de memoria elementales en el cual la integridad de la memoria está asegurada por una verificación del valor de cada bloque de memoria elemental asignado.

5 El método de asignación dinámico de memoria en una estructura de datos identificado por un número de identificación y memorizado en formato de paquetes de información numérica en la zona de memoria de un sistema integrado, objeto del presente invento, prevé que, al estar la zona de memoria subdividida en bloques de memoria elementales, este sea aplicado a partir de, al menos, una instrucción de borrado de un bloque de memoria elemental y una instrucción de asignación de un bloque de memoria elemental, por la cual este bloque de memoria elemental está asociado a una referencia de este número de identificación. Para asignar un bloque de memoria elemental, este proceso consiste en atribuir en la referencia del número de identificación el valor de este número de identificación. Para designar un bloque de memoria elemental, el proceso consiste en atribuir en la referencia del número de identificación el valor de este número de identificación. Para borrar un bloque de memoria elemental, este proceso consiste en atribuir un valor arbitrario establecido *a priori*, distinto de cualquier valor de número de identificación.

15 Un sistema integrado multiaplicación, objeto del presente invento, comporta un sistema de explotación que asegure la gestión de los circuitos de entrada/salida, de la memoria viva y de la memoria no volátil mediante un microprocesador. Según la reivindicación 2, puede incluir en memoria no volátil una tabla de gestión de la zona memoria por bloques de memoria elementales, incluyendo dicha tabla, al menos, por cada bloque de memoria elemental, una referencia a un número de identificación de la estructura de datos en la cual el bloque de memoria elemental correspondiente ha sido asignado y un valor de espacio de memoria ocupado en el bloque de memoria elemental asignado correspondiente.

20 El proceso objeto del invento es aplicable en la asignación dinámica de memoria de cualquier sistema informático integrado, más concretamente de cualquier objeto portátil como una tarjeta de micro procesador o micro controlador, en el cual las fuentes de memoria son reducidas, en razón de la medida y el volumen limitados de este último.

El proceso y el sistema integrado multi aplicación objetos del invento se entienden mejor tras efectuar una lectura de la descripción y los dibujos indicados a continuación en los cuales, además de la figura 1 referente al asunto anterior:

30 - la figura 2 representa un organigrama funcional ilustrativo de las etapas que permiten la aplicación del proceso de asignación dinámica de memoria objeto del presente invento;

35 - la figura 2b representa, a título informativo, un organigrama funcional de una variante preferencial de aplicación del proceso de asignación dinámica de memoria, objeto del presente invento, como ilustrado en la figura 2;

- la figura 2c representa, a título informativo, un modo de realización concreto del proceso de cálculo de un valor de verificación (checksum);

40 - la figura 3 representa, a título ilustrativo, una variante preferencial de aplicación del procedimiento objeto del presente invento, en el caso de una operación de borrado de una estructura de datos, como una aplicación;

- las figuras 4a y 4b representan, a título ilustrativo, los diagramas del estado de ocupación de memoria al aplicar el proceso de instalación/desinstalación de aplicaciones, según el procedimiento de asignación dinámica de memoria objeto del presente invento;

45 - la figura 5 representa, a título de ejemplo y sin limitación, un sistema integrado, como una tarjeta de microprocesador o micro controlador, que consta de, grabado en memoria no volátil, una tabla de datos específicos que permiten gestionar el espacio de memoria de este sistema integrado, según el procedimiento objeto del presente invento.

50 Se proporciona una descripción más detallada del procedimiento de asignación dinámica de memoria por bloques de memoria elementales en una estructura de datos, objeto del presente invento, en relación con la figura 2a y las figuras siguientes.

55 La figura 2a representa un organigrama funcional de las etapas fundamentales que permiten la aplicación del proceso de asignación dinámica de memoria objeto del presente invento; dicho organigrama funcional corresponde, de hecho, a un diagrama de estado de susodichas etapas fundamentales.

60 Concretamente, se entiende que la etapa S corresponde a un estado de inicio en el cual una estructura de datos, formada por una aplicación, por los datos sistema, o según el caso por las claves o los valores numéricos específicos, debe ser memorizada, es decir, instalada en el caso de aplicaciones, por ejemplo en la zona de memoria de un sistema integrado, cuya estructura corresponde a la descrita anteriormente en la descripción relacionada con la figura 1.

65 Se entiende especialmente que cada estructura de datos está identificada por un número de identificación señalado  $ID_{A_j}$  donde j puede tener los valores comprendidos entre 1 y n, siendo n el indicador del número total de estructuras de datos susceptibles de estar memorizadas o instaladas en susodicho espacio de memoria.

Además, para la aplicación del proceso objeto del presente invento, se indica que el espacio de memoria sometido al proceso de asignación dinámica de memoria objeto del presente invento, está subdividido de manera óptima en

## ES 2 335 863 T3

bloques de memoria, cada bloque de memoria llamado  $BL_1$ , con la 1 pudiendo variar de  $1=1$  a  $1=L$ , está formado por un número determinado de octetos.

En un modo de realización preferente y sin limitación, se indica que el número de octetos constituidos de cada bloque  $BL_1$  puede ser de una potencia de 2 con el fin de facilitar la redirección de cada bloque de memoria y en cada bloque de memoria así definida. En un modo de aplicación dado a título de ejemplo sin limitación, se indica que cada bloque puede constar de 256 octetos. En la etapa S de inicio, el conjunto de los números de identificación es  $[ID_{A_j}]_{j=1}^n$  y el conjunto de los bloques  $[BL_1]_{l=1}^L$ .

Finalmente, y en el marco de la aplicación del proceso objeto del presente invento, se dispone de un valor numérico arbitrario AAA, el cual tiene una función específica que se detallará a continuación en la descripción, con el fin de permitir la aplicación del método de asignación dinámica de memoria objeto del invento.

De manera general, recordamos que cada estructura de datos, es decir, aplicaciones, datos de sistema, claves o valores numéricos específicos, identificada por el número de identificación  $ID_{A_j}$  puede memorizarse en formato de paquetes de información numérica en la zona de memoria del sistema integrado anteriormente comentado.

Por paquetes de información numérica se entiende, por un lado, los paquetes de datos numéricos representativos de un código objeto que permite la instalación y la ejecución de una aplicación, p. ej. de datos de sistema o de claves, o valores numéricos específicos, respectivamente de los paquetes de datos de aplicaciones generados por una o diversas aplicaciones implantadas sobre el sistema integrado correspondiente, o de datos numéricos intermedios que permiten la gestión del conjunto por el sistema de explotación para asegurar la ejecución de las diferentes funcionalidades del sistema integrado a partir de datos del sistema, de las claves o de susodichos valores numéricos intermedios.

Como muestra la figura 2a, el proceso de asignación dinámico de memoria por bloques de memoria elementales en una estructura de datos, objeto del invento, es aplicado a partir de al menos una instrucción de asignación de un bloque de memoria elemental  $BL_1$ , operación de asignación A por la cual, en el bloque de memoria elemental comentado anteriormente está asociada una referencia al número de identificación de la estructura de datos o aplicación considerada, y a partir de al menos una instrucción de borrado de un bloque de memoria elemental, instrucción correspondiente a la operación E representada en la figura 2a.

En particular, y según un aspecto destacado del proceso de asignación objeto del invento, la operación de asignación A, para asignar un bloque memoria elemental, consiste en atribuir el valor del número de identificación  $ID_{A_j}$  a la referencia del número de identificación asociado al bloque memoria elemental considerado.

Aunque en los procesos de asignación de memoria de los sistemas integrados anteriormente comentados, las zonas de memorización de las estructuras de datos, como las aplicaciones, incluyen una simple referencia al número de identificación de la estructura de datos o aplicación correspondiente realizada bajo la forma de un indicador, indicador de inicio de memorización de la estructura de datos o aplicación e indicador de fin de memorización de la estructura de datos o aplicación considerada, la etapa de asignación A conforme al objeto del presente invento consiste de hecho en atribuir a título de referencia del número de identificación de la estructura de datos o aplicación, el valor del número de identificación antes referido en cada bloque  $BL_1$ , en el cual se memorizan los paquetes de información numérica de código objeto o de datos. Se entiende concretamente, como representado en la figura 2a, que la atribución del valor del número de identificación en la referencia del número de identificación consiste en asociar de manera bi-unívoca el valor de este número de identificación en el bloque memoria elemental correspondiente  $BL_1$ , siendo esta operación:

$$BL_1 (ID_{A_j})$$

Se entiende que la anteriormente comentada puesta en correspondencia bi unívoca puede efectuarse por asociación en cada bloque  $BL_1$  de un campo de datos representativo del valor del número de identificación comentado  $ID_{A_j}$  en una tabla de gestión de asignación dinámica de memoria de la zona memoria del sistema integrado arriba referido.

La estructura completa de susodicha tabla se facilitará en adelante.

Conforme a otro aspecto destacable del proceso de asignación dinámico de memoria por bloques de memoria elementales según el objeto del presente invento, se indica, en referencia a la figura 2a, que la etapa de borrado E incluye, para borrar un bloque de memoria elemental, atribuir en la referencia del número de identificación asociada al bloque de memoria elemental considerada  $BL_1$  un valor arbitrario en vez de y en el lugar del número de identificación de la estructura de datos o aplicación considerada. Este valor arbitrario es por supuesto el valor AAA anteriormente citado en la descripción. Este valor es distinto de cualquier valor de número de identificación atribuido a una estructura de datos considerada.

En la figura 2a, se representan las etapas A de asignación y E de borrado vinculadas al estado de inicio S por una flecha doble, representando cada una de ellas la llamada de función o de la instrucción de asignación A, respectivamente, de la función o de la instrucción de borrado E, las operaciones de asignación, respectivamente de borrado efectuadas conforme a lo descrito anteriormente, y la flecha de retorno que representa una descarga dirigida hacia el estado de inicio S.

## ES 2 335 863 T3

En la figura 2b, se presenta una descripción más detallada de un modo de aplicación preferente del proceso de asignación dinámica de memoria por bloques de memoria elementales conforme al objeto del presente invento.

5 El método objeto del presente invento, en el modo de realización de la figura 2b, puede corresponder a un proceso de asignación de memoria de los datos creados por una aplicación por ejemplo, o, si es el caso, obtenidos por la carga o descarga de actualizaciones de aplicaciones, los paquetes de información numérica, pudiendo corresponder, en ambos casos, a los de una aplicación o una estructura de datos ya instalada.

Por esta razón, se constata:

10

$[ID\_A_j]_{j=1}^n$  conjunto de los números de identificación de estructuras de datos ya instaladas;

$[BL_i]_{i=1}^L$ , conjunto de bloques susceptibles de ser utilizados en la zona de memoria considerada;

15

$ID\_A_k$  número de identificación de la estructura de datos o aplicación por la cual la asignación memoria debe ser efectuada,  $k \in [1, n]$ ;

20

$Q_k$  espacio de memoria necesario en los paquetes de información numérica a los que debe asignarse un espacio de memoria correspondiente.

Por otro lado, se define un índice  $j$  particular, que escribiremos  $j_0$ , al que se le atribuye el valor 0.

25

En estas condiciones, la operación de asignación  $A$ , como muestra la figura 2b, puede constar, para asignar un bloque de memoria elemental dado, llamado  $BL_i$ , del paquete de información numérica  $Q_k$  antes comentado, y, previamente a cualquier etapa consistente en atribuir en la referencia del número de identificación el valor del número de identificación, a comprobar, por cualquier bloque de memoria elemental ya asignado, la identidad de la referencia en el número de identificación y del número de identificación de la estructura de datos correspondiente, en una etapa  $A_1$ . Esta verificación de identidad consiste en efectuar un test mediante el cual debe buscarse el primer bloque de memoria  $BL_i$  cuyo número de identificación asociado  $ID\_A_j$  corresponde al número de identificación  $ID\_A_k$  de la estructura de datos o aplicación para la cual debe efectuarse la asignación.

30

Susodicho test se escribe:

35

$$\exists BL_1 (ID\_A_j) \text{ con } ID\_A_j = ID\_A_k ?$$

$$\text{y } j > j_0$$

40

En caso de resultado negativo del test  $A_1$ , el cual consiste en buscar el próximo bloque perteneciente a la aplicación cuyo número de identificación sea  $ID\_A_k$ , una instrucción de asignación de un bloque de memoria elemental libre es llamado a la etapa  $A_3$ , y ninguno de los bloques eventuales de memoria ya asignados en la aplicación, cuyo número de identificación es  $ID\_A_k$ , no dispone de espacio libre suficiente como para contener el paquete de informaciones  $Q_k$ .

45

En la figura 2b, la llamada de instrucción de asignación del bloque de memoria elemental siguiente es:

$$BL_a(AAAA) = BL_a(ID\_A_k)$$

50

La instrucción de asignación comentada del bloque elemental siguiente permite asignar un bloque de memoria elemental de dirección  $a$ , siendo  $a$  un valor de desajuste cualquiera, y  $a \in \mathbb{N}$  un conjunto de enteros naturales. El bloque de memoria elemental retenido se entiende como un bloque elemental cuyo valor arbitrario  $AAAA$  ha sido atribuido precedentemente y corresponde por este hecho a un bloque borrado, es decir, a un bloque libre y preparado para cualquier operación de asignación y de memorización del paquete de información  $Q_k$  correspondiente.

55

En caso de resultado positivo del test  $A_1$ , significa que el bloque de memoria elemental encontrado es al menos parcialmente asignado en la estructura de datos o aplicación para cuya asignación debe efectuarse.

60

En la figura 2b, la verificación de la existencia de un espacio de memoria suficiente del test  $A_2$  consta de la valorización  $j$  del índice particular  $j_0$  y esto es:

$$LR_j = LB_j - LO_j \text{ et } Q_k \leq LR_j ?$$

65

relación en la que  $LR_j$  asigna el espacio de memoria restante para el bloque de memoria elemental corriente;  $LB_j$  asigna el espacio de memoria total de cada bloque elemental corriente, es decir 256 octetos en el modo de realización anteriormente indicado, y  $LO_j$ , asigna el espacio de memoria ocupado en el bloque de memoria elemental corriente comentado anteriormente.

## ES 2 335 863 T3

En el caso de obtener resultados positivos en el test de verificación  $A_2$ , el paquete de información  $Q_k$ , al requerir un espacio de memoria inferior al espacio de memoria del bloque de memoria elemental corriente, en el paquete de información  $Q_k$  puede asignarse, en una etapa  $A_4$ , el bloque de memoria elemental corriente comentado  $BL_1$  ( $ID_{A_j}$ ), pudiendo ser memorizado (el paquete) en la zona de memoria libre de susodicho bloque de memoria elemental corriente.

Así, la etapa de asignación  $A_4$  consiste en una escritura del paquete de información  $Q_k$  en el bloque  $BL_1$  ( $ID_{A_j}$ ) y una actualización del valor del espacio memoria ocupado en el bloque, escrito:

$$LO_j = LO_j + Q_k$$

En caso de resultado negativo al test de verificación  $A_2$ , el bloque corriente  $BL$  ( $ID_{A_j}$ ) al no disponer de suficiente espacio libre como para contener el paquete de información  $Q_k$ , es necesario volver a la etapa  $A_1$  para volver a buscar un nuevo bloque cuyo número de identificación sea  $ID_{A_k}$  y que no haya sido aún analizado durante esta asignación.

Por supuesto, las etapas de asignación propiamente dichas  $A_3$  y  $A_4$  van seguidas de un regreso al estado de salida  $S$ .

En un modo de realización preferente sin limitación, se indica que *a posteriori* de las etapas de asignación  $A_3$  y  $A_4$ , y por tanto *a posteriori* de la asignación de un bloque de memoria elemental en una aplicación y por supuesto en la memorización por escritura de la información numérica, es decir, del paquete de información  $Q_k$  por escritura de esta información en el bloque de memoria elemental asignado, el método de asignación conforme al objeto del presente invento puede consistir además en calcular, en una etapa  $A_5$  un valor de verificación del bloque memoria elemental asignado, siendo esta operación designada por el cálculo  $CKS$  en la figura 2b.

De una manera general, se indica que para la aplicación de la etapa  $A_5$ , el cálculo del valor de verificación, todavía designado por check sum en inglés, puede ser efectuado gracias a los métodos de cálculo de valor de verificación tradicional, pudiendo tales métodos además corresponder al circuito de cálculo especializado 20 contenido en los sistemas integrados de tipo clásico, así como descrito anteriormente en la descripción. Puede preverse una etapa  $A_6$  de gestión de errores de tipo clásico.

Un modo de realización preferente de la operación de cálculo de un valor de verificación de los bloques de memoria elementales es un modo de verificación asíncrono, como se describe en la figura 2c.

Tal y como representado en la figura arriba referida y con preferencia independientemente de cualquier asignación de un bloque de memoria elemental en una estructura de datos, el método de asignación dinámica de memoria objeto del presente invento puede consistir, para la aplicación de la etapa  $A_5$ , en seleccionar según un criterio de selección aleatorio uno de los bloques de memoria elementales asignados. Esta operación de selección sobre criterio de selección aleatorio aparece representado en la figura 2c por la generación de un valor aleatorio  $A$  en la etapa  $A_{51}$ , pudiendo dicho valor estar comprendido entre  $[1, L]$ , después en una etapa  $A_{52}$  consistente en una llamada a lectura del bloque de memoria elemental de dirección, estando esta operación escrita en la figura 2c:

$$BL_d(X) = BL_a(X)$$

En esta relación,  $X$  designa el contenido del bloque de memoria elemental  $BL_A$  llamada a lectura.

A las etapas  $A_{51}$  y  $A_{52}$  comentadas anteriormente, les sigue una etapa  $A_{53}$  que consiste en efectuar el cálculo de un valor de verificación del bloque de memoria elemental asignado retenido. La etapa  $A_{53}$  se escribe "cálculo de  $CKS$  ( $X$ )". Este valor de verificación se compara con un valor verdadero memorizado en la tabla arriba referida, según el proceso de verificación de tipo check sum comentado anteriormente en la descripción.

En el caso de resultado positivo de la verificación antes mencionada, de acuerdo con un aspecto destacable del método objeto del invento, el proceso de cálculo del valor de verificación puede ser activado periódicamente por una etapa  $A_{54}$  de regreso a la etapa  $A_{51}$ , lo que permite reiniciar la generación de un valor aleatorio distinto, y después la llamada a lectura del bloque de memoria elemental correspondiente por la ejecución de las etapas  $A_{51}$  y  $A_{52}$ .

Contrariamente, en caso de resultado negativo en la etapa  $A_{53}$  del cálculo del valor de verificación, al no estar comprobado el valor de verificación, puede pedirse una etapa  $A_{55}$  de regreso de error para destacar un problema de integridad de datos o de código memorizados. La gestión de la situación de error de tipo clásico, etapa  $A_{56}$ , puede prever el bloqueo del sistema integrado o la tentativa de recuperación de los datos si se incorpora un código corrector de errores como el checksum.

Más adelante, se da un ejemplo de aplicación específica del método de asignación dinámica de memoria por bloques de memoria elementales, según el presente invento.

## ES 2 335 863 T3

El método objeto del invento ha sido aplicado para los bloques de memoria elementales de 256 octetos, en cuyos bloques de memoria elemental  $BL_i$  han sido asociados los siguientes datos:

- Número de identificación  $ID_{A_j}$  de la estructura de datos o de la aplicación: 1 octeto;
- Espacio de memoria ocupada indicado  $LO_j$ : 1 octeto; el espacio de memoria libre según el caso puede ser utilizado;
- Valor de verificación o check sum: 2 octetos.

De manera preferente, como se indicará más adelante en la descripción, los datos anteriores son almacenados por cada bloque de memoria elemental de la siguiente forma:

estructura bloque info {byte  $ID_{A_j}$ , byte  $LR_j$ , word  $CKS$ }

En la relación anterior, se indica que “byte” designa una palabra de un octeto y “word” una palabra de dos octetos.

En lo que se refiere a la codificación del número de identificación asociado a cada bloque de memoria elemental, se ha utilizado la siguiente codificación:

$ID_{A_j} = 00$ : bloque libre,  $AAAA = 00h$ ,

$FFh$ : bloque sistema ilegible,

$FEh$ : bloque de sistema legible,

$FDh$ : valor específico o clave,

$01h-FCh$ : bloque de memoria elemental perteneciente a la aplicación de número de identificación de aplicación  $i$ , con  $i \in [01h, FCh]$ .

En cualquier  $ID_{A_j}$ ,  $LR_j$ , se puede aplicar cualquier valor de  $01h$  a  $FFh$ ; el valor  $00h$  significa además que el bloque de memoria elemental corriente se utiliza por completo, sin dejar ningún espacio de memoria libre disponible en el bloque.

El método de asignación dinámica de memoria por bloques de memoria elementales objeto del invento, permite además borrar cualquier estructura de datos, o aplicación, identificada por un número de identificación de manera particularmente flexible y eficaz, como comentado en adelante y mostrado en la figura 3.

En razón de la atribución de cada bloque de memoria elemental de una referencia correspondiente al número de identificación, el proceso de borrado correspondiente puede consistir, según reivindicación 3, en recoger todos los bloques  $BL_0$ ,  $BL_b$  a  $BL_q$ , cuyo número de identificación  $ID_{A_c}$  corresponda a la aplicación que debe borrarse en la etapa  $E_1$  representada en la figura 3, y después en atribuir a la referencia del número de identificación del conjunto de los bloques de memoria elementales asignados en la aplicación considerada, el valor arbitrario  $AAAA$  anteriormente comentado en la etapa  $E_2$  de la misma figura 3, es decir a susodicho valor  $00h$ .

Efectivamente, habida cuenta de esta atribución, todos los bloques  $BL_0$ ,  $BL_b$  a  $BL_q$ , se consideran en adelante bloques borrados y, por tanto, disponibles para futuras asignaciones, y para la escritura en la totalidad del espacio de memoria reservado para cada una de estas.

En lo que se refiere a la verificación de la pertenencia de un octeto en la zona de memoria asignada según el método objeto de este invento, el número de identificación de la aplicación o de la estructura de datos correspondiente puede obtenerse mediante la siguiente relación:

$$ID_{A_k} = \text{Bloque info } [(dirección\_octeto\_dirección\_inicio\_zona \text{ memoria})/64]$$

esta relación se obtiene teniendo en cuenta el modo de realización descrito para los bloques de 256 octetos por bloque y 4 octetos de información en el bloque info por bloque corriente considerado.

La aplicación del método de asignación dinámico de memoria, según reivindicación 2, permite minimizar o suprimir la dispersión de la zona de memoria gestionada, la cual todavía está designada por fragmentación. Efectivamente, para la creación de un objeto o de un tipo de variable, en la lista bloque info se busca el primer bloque perteneciente a la aplicación o a la estructura de datos, que disponga del espacio libre suficiente para la creación de este objeto o tipo de variable considerada. En estas condiciones, según reivindicación 2, el nuevo bloque no está designado a una

## ES 2 335 863 T3

aplicación o a una estructura de datos salvo que el espacio restante sea insuficiente para el objeto o tipo de variable a crear en, al menos, uno de los bloques ya asignados a esta aplicación o esta estructura de datos. Se esta manera, nunca existe ningún objeto o tipo de variable que no esté repartido en, por lo menos, dos bloques no consecutivos. Estos objetos o tipos de variables al no tener que ser almacenados en un orden concreto, borrado cualquier bloque, que a consecuencia vuelve a estar libre, siempre puede ser restablecido o reasignado (el bloque), ya sea en los datos expedidos de aplicaciones o en el código objeto relativo a una aplicación. El riesgo de dispersión queda entonces limitado a la carga y al borrado habitual de bloques de códigos consecutivos.

El fenómeno de dispersión de la zona de memoria puede ser reducido, además, por un proceso de reasignación de los bloques de memoria elementales en los paquetes de código objeto, mediante reajuste por traslación.

Referente al proceso de verificación de los bloques, se calcula el valor de verificación para cada bloque sometido al proceso de asignación, es decir, en cada escritura del bloque considerado. Contrariamente, con el fin de no ralentizar demasiado el acceso a la memoria, no es necesario verificar los bloques en cada lectura.

En una variante de aplicación, se indica que el valor de verificación puede calcularse en cada lectura para los bloques que contienen datos denominados críticos, tal como los datos sensibles del OS o los valores de claves para las que el valor del número de identificación puede ser FDh.

En cuanto a la repetición periódica del cálculo del valor de verificación en los bloques seleccionados de manera aleatoria, se indica que el periodo puede establecerse en 50 ms, por ejemplo. El cálculo del valor de verificación del conjunto de los bloques también puede aplicarse durante las ejecuciones muy específicas del sistema integrado, tales como el cambio del estado de vida de este último en el dominio de aplicación de tarjetas chip, por ejemplo.

Un ejemplo de instalación y desinstalación de aplicaciones de tipo subprograma o applet en inglés, gracias a la aplicación del método de asignación dinámica de memoria por bloques de memoria elementales según el objeto del presente invento, será ahora dado a lugar con las figuras 4a y 4b. Los subprogramas pueden corresponder a los programas escritos en lenguaje de alto nivel de todo tipo.

Referente a la figura 4a, se consideran 4 subprogramas, asignados por Applet 1, Applet 2, Applet 3 y Applet 4, cada uno de ellos formados por:

TABLA I

Applet 1	150 octetos de código, es decir 1 bloque 600 octetos de datos, es decir 3 bloques
Applet 2	150 octetos de código, es decir 1 bloque 150 octetos de datos, es decir 1 bloque
Applet 3	1100 octetos de código, es decir 5 bloques 200 octetos de datos, es decir 1 bloque
Applet 4	400 octetos de código, es decir 2 bloques 300 octetos de datos, es decir 2 bloques

En referencia a la figura 4b, se indica que la etapa 1 corresponde a la instalación del Applet 1 y del Applet 2; la etapa 2 al borrado del Applet 1; la etapa 3 a la instalación del Applet 3; y la etapa 4 a la instalación del Applet 4.

La tabla II que aparece a continuación indica el valor expresado en valor hexadecimal, del número de identificación ID\_Ak, donde k toma los valores de 1 a 4, y el valor 00 en escritura hexadecimal corresponde al valor arbitrario AAAA.

TABLA II

Etapa 1	01	01	01	01	02	02	00	00	00	00	00	00	00
Etapa 2	00	00	00	00	02	02	00	00	00	00	00	00	00
Etapa 3	03	00	00	00	02	02	03	03	03	03	03	00	00
Etapa 4	03	04	04	04	02	02	03	03	03	03	03	04	00

## ES 2 335 863 T3

En relación con la figura 5, a continuación se describe un sistema integrado específico multi aplicación que permite la aplicación del método de asignación dinámica de memoria conforme al objeto del presente invento.

5 Cabe recordar que el sistema integrado correspondiente está representado a título de ejemplo sin limitaciones bajo la forma de tarjeta con microprocesador o microcontrolador. Este sistema integrado comporta un sistema de explotación que asegura la gestión de los circuitos de entrada/salida de la memoria viva y de la memoria no volátil mediante dicho microprocesador.

10 Referente a la aplicación del método objeto de la reivindicación 8, en memoria programable no volátil 18<sup>a</sup> consta de una tabla de gestión de la zona memoria por bloques de memoria elementales, la cual (la tabla) constituye, al menos para cada bloque de memoria elemental BL<sub>1</sub> anteriormente definido en la descripción, una referencia a un número de identificación de una estructura de datos a la cual ha sido asignado el bloque de memoria elemental correspondiente, y un valor de espacio memoria ocupada en el bloque de memoria elemental correspondiente asignado. La referencia al número de identificación de la estructura de datos se escribe ID\_A<sub>1</sub>, respectivamente ID\_A<sub>2</sub>, y el valor de espacio de memoria ocupada LI<sub>1</sub>, respectivamente LI<sub>2</sub> en la fig. 5.

15 Además, como representado en la figura arriba referida, la tabla puede constar, para cada bloque de memoria elemental correspondiente, de un valor de verificación del bloque elemental asignado, siendo los valores CKS<sub>1a</sub> y CKS<sub>1b</sub>, respectivamente CKS<sub>2a</sub> y CKS<sub>2b</sub>. Se indica que, en el modo de realización de la figura 5, el valor de verificación está codificado sobre 2 octetos.

20 En el caso de aplicar el método objeto del presente invento en un sistema integrado como una tarjeta de chip, y para una capacidad de memoria no volátil 18a, formada por una memoria programable de tipo EEPROM de 32 Koctetos, la tabla comentada ocupa un espacio memoria correspondiente a  $32 \times \text{Koctetos} / 356 \times 4 = 512$  octetos. Cabe destacar que el espacio de memoria utilizado para la gestión de este espacio de memoria, según el método de asignación dinámica de memoria por bloques de memoria elementales objeto del presente invento, ocupa, debido a la memorización necesaria de la tabla arriba referida, un 1,56% de la capacidad de memoria efectivamente disponible. Este valor débil de la zona de memoria ocupada para la aplicación del método objeto del presente invento es aceptable en conjunto, teniendo en cuenta el aumento obtenido en adaptabilidad de uso y de instalación de aplicaciones diversas, cuyo número depende de la capacidad de memoria no volátil 18a.

35

40

45

50

55

60

65

## REIVINDICACIONES

5 1. Método de asignación dinámica de memoria en una o varias estructuras de datos (ID\_Aj) identificada(s) por un número de identificación y memorizada(s) en forma de paquetes de información numérica en una zona de memoria de un sistema integrado, la cual está subdividida en bloques de memoria elementales (BL1) y siendo aplicado dicho método de asignación dinámica a partir de al menos una instrucción de borrado (E) de un bloque de memoria elemental y una instrucción de asignación (A) de un bloque memoria elemental, por la cual en dicho bloque de memoria elemental se ha asociado una referencia en tal número de identificación; el método consta de una serie de etapas que consisten en:

- 10
- para asignar un bloque de memoria elemental:
    - atribuir en dicha referencia de dicho número de identificación el valor de dicho número de identificación, y,
  - 15 • para borrar un bloque de memoria elemental:
    - atribuir en dicha referencia de dicho número de identificación un valor arbitrario, distinto a cualquier valor de número de identificación.
- 20

2. Método según la reivindicación 1, por el cual, para asignar un bloque de memoria elemental, este está formado además, y anteriormente a la etapa que consiste en atribuir en susodicha referencia de dicho número de identificación el valor de dicho número de identificación, para cualquier bloque de memoria elemental asignado, de las etapas que consisten en:

- 25
- comprobar (A1) la identidad de dicha referencia en dicho número de identificación y de dicho número de de identificación, y en caso de resultado positivo de esta comprobación de identidad, susodicho bloque de memoria elemental corriente está al menos parcialmente asignado a esta estructura de datos,
  - 30 - verificar (A2) que existe espacio de memoria suficiente de dicho bloque de memoria elemental corriente, para memorizar dicha información numérica; y llamar una instrucción de asignación de un bloque de memoria elemental siguiente, en caso contrario.
- 35

3. Método según la reivindicación 1 o 2, mediante el cual, para borrar una estructura de datos identificada por un número de identificación (ID\_Ae), se trata de atribuir el valor arbitrario (AAAA), antes comentado, en dicha referencia de dicho número de identificación del conjunto de los bloques de memoria elementales asignados a esta estructura de datos.

4. Método según la reivindicación a, en el cual, después de la asignación de un bloque de memoria elemental en una estructura de datos y después de la memorización por escritura de dicha información numérica en dicho bloque de memoria elemental asignado, consiste además en calcular un valor de verificación (CKS) del bloque de memoria elemental.

5. Método según la reivindicación 1, en el cual, independientemente de cualquier asignación de un bloque de memoria elemental en una estructura de datos, este comprende además las etapas que consisten en:

- 45
- seleccionar (A51), según un criterio de selección aleatorio, uno de los bloques de memoria elemental asignado, y
  - 50 - calcular (A53) un valor de verificación (CKS(X)) de dicho bloque de memoria elemental asignado.

6. Método según la reivindicación 5, en el cual, dicha selección seguida de dicho cálculo de un valor de verificación ambos son activados periódicamente, con un periodo de valor determinado.

7. Sistema integrado formado por una zona memoria asignada de manera dinámica a una o varias estructuras de datos (ID\_Aj) identificadas por un número de identificación y memorizadas en forma de paquetes de información numérica, estando dicha zona de memoria subdividida en bloques de memorias elementales (BL1), y habiendo asociado a cada bloque una referencia a dicho número de identificación, la cual (referencia) posee:

- 60
- el valor de dicho número de identificación cuando el bloque de memoria elemental está asignado,
  - un valor arbitrario, distinto a cualquier valor de número de identificación, cuando el bloque de memoria elemental sea borrado.
- 65

## ES 2 335 863 T3

8. Sistema integrado según reivindicación 7, **caracterizado** por el hecho de que en memoria programable no volátil (18a) comporta una tabla de gestión de la zona memoria por bloques de memoria elementales; y dicha tabla consta de, al menos para cada bloque de memoria elemental:

- 5       - una referencia a un número de identificación de una estructura de datos (ID<sub>Ai</sub>) en la cual el bloque de memoria elemental correspondiente ha sido asignado;
- un valor de espacio de memoria ocupado (L<sub>li</sub>) en dicho bloque de memoria elemental correspondiente asignado.

10

9. Sistema integrado según reivindicación 7, en el cual dicha tabla comporta además para cada bloque de memoria elemental, un valor de verificación (CKS1a a CKS2b) del correspondiente bloque de memoria elemental asignado.

15

20

25

30

35

40

45

50

55

60

65

FIG.1.  
ARTE ANTERIOR

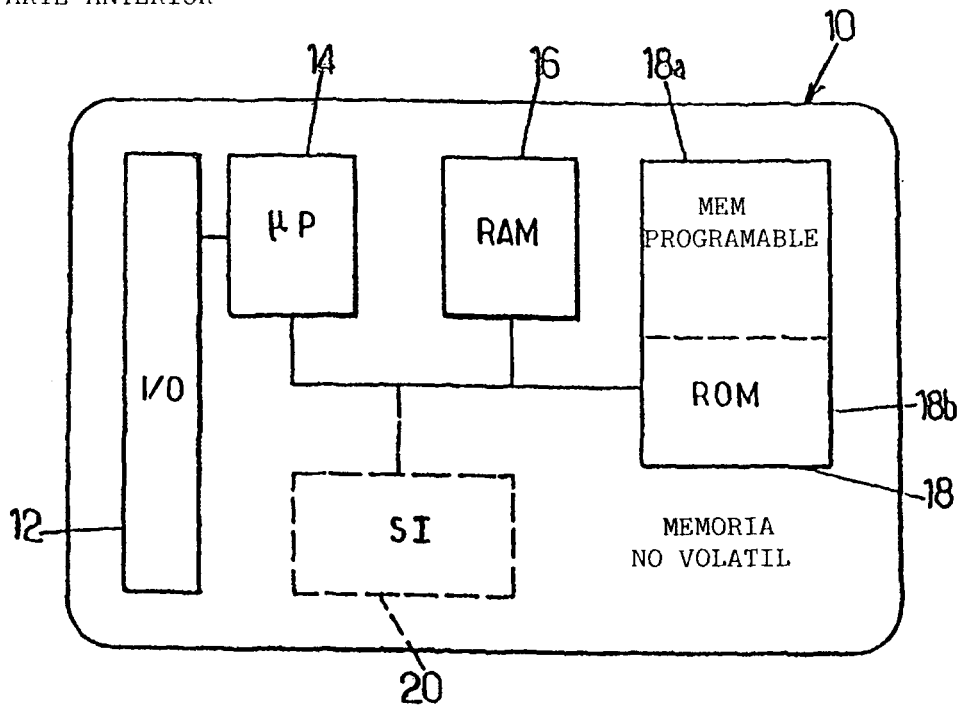
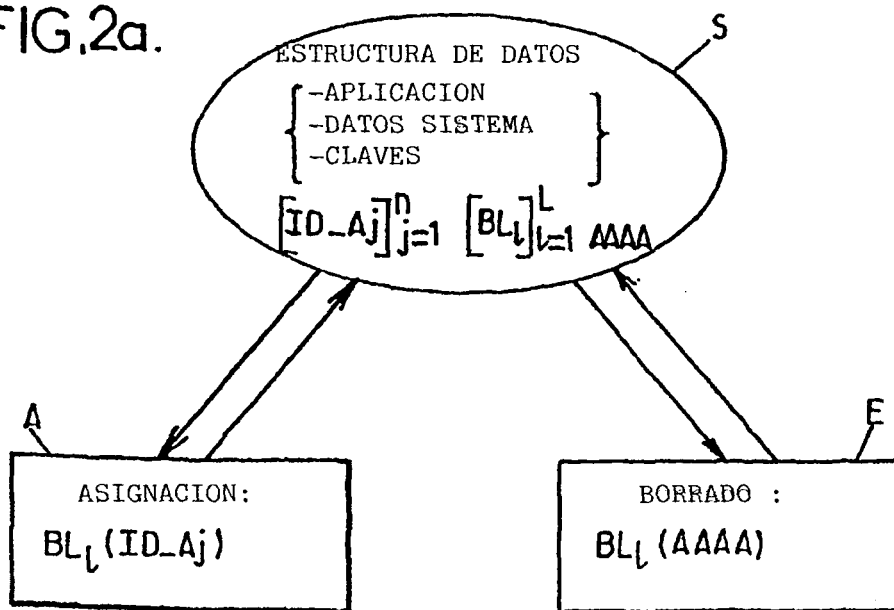


FIG.2a.



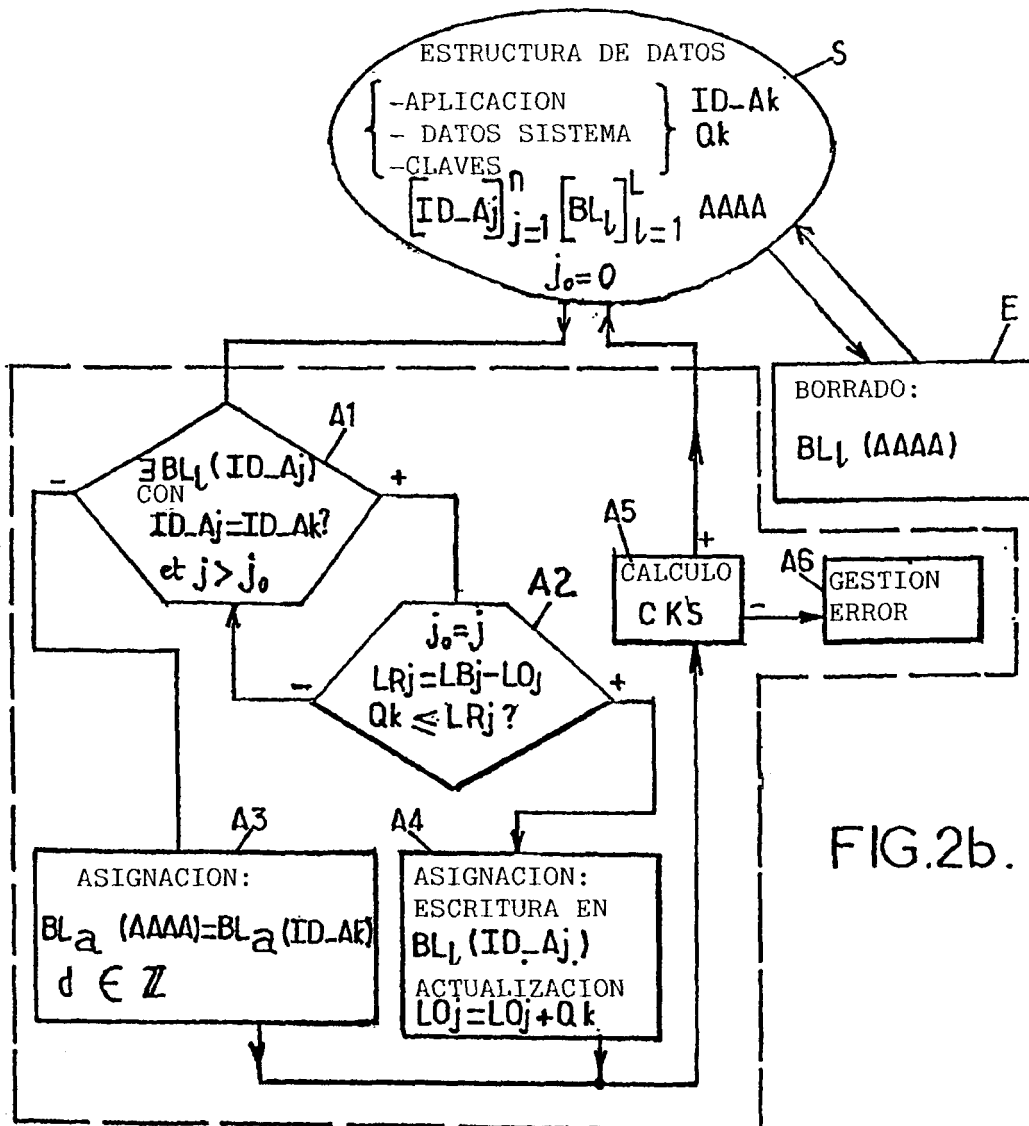


FIG.2b.

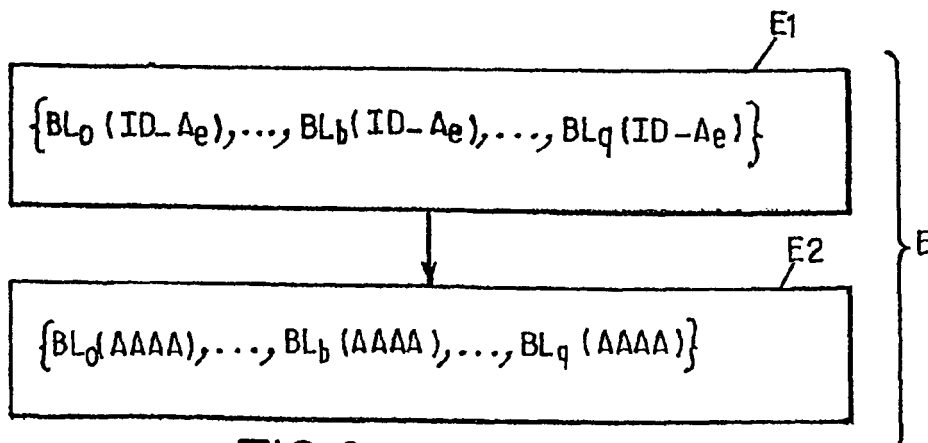


FIG.3. BORRADO APLICACION

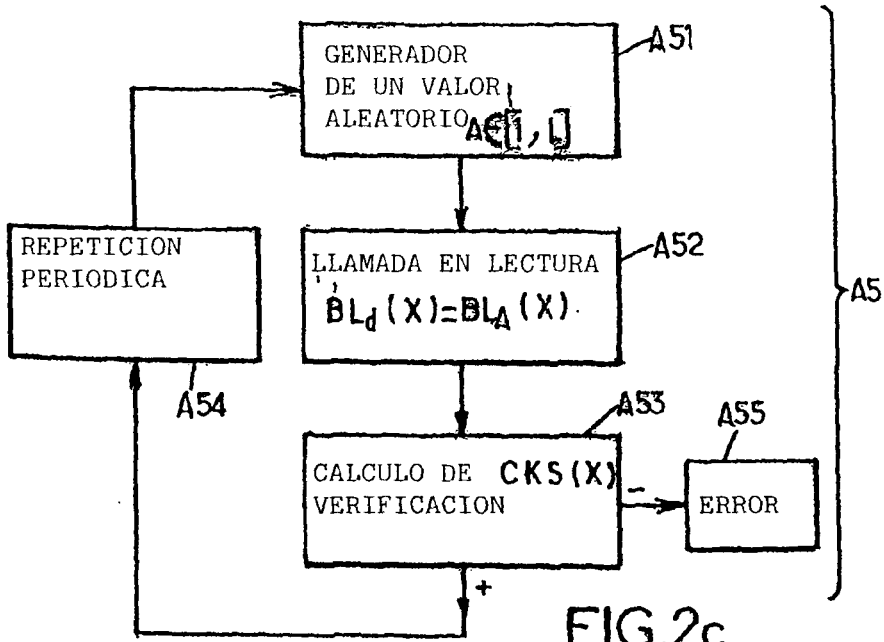


FIG. 2c.

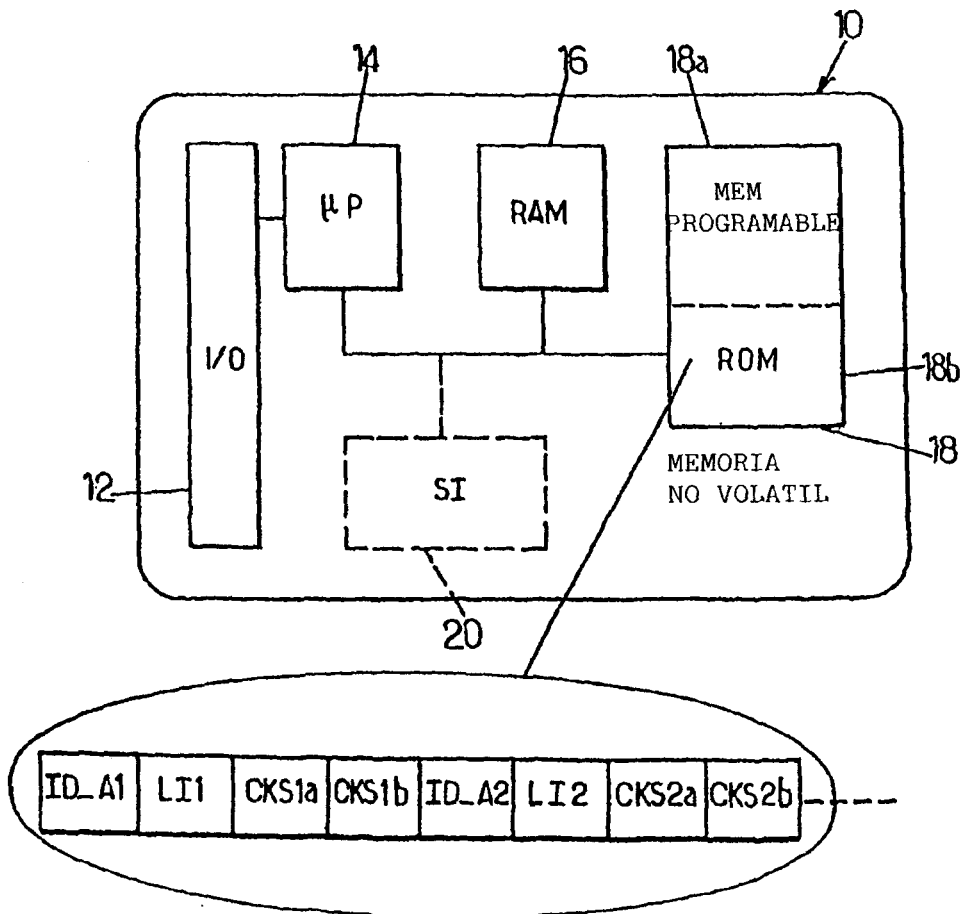


FIG. 5.

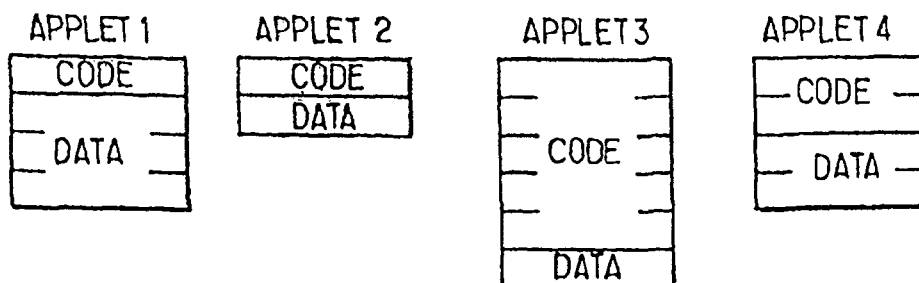


FIG.4a.

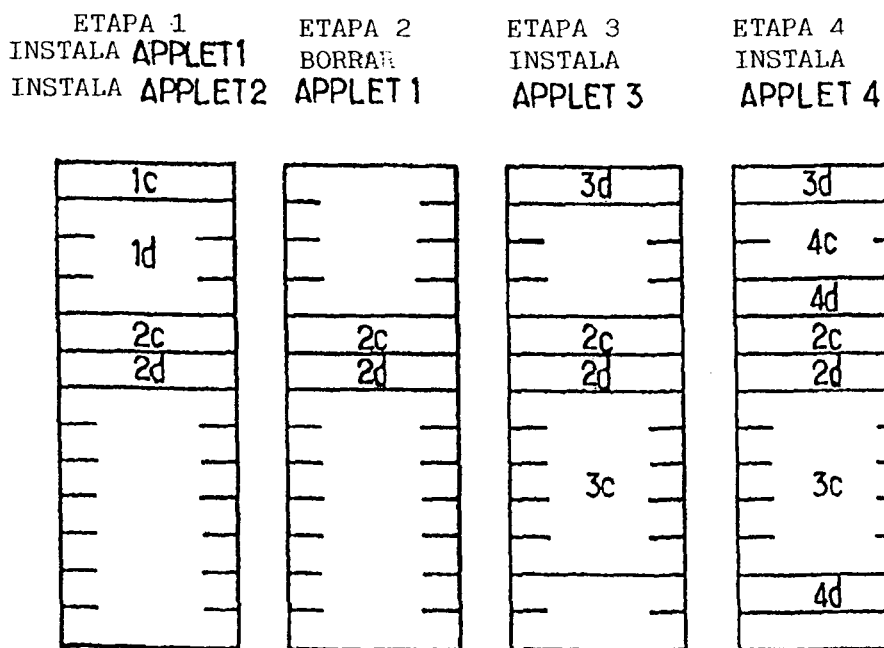


FIG.4b.