

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 542 421**

51 Int. Cl.:

G06Q 20/34 (2012.01)

G07F 7/10 (2006.01)

G06F 12/02 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **15.06.2004** **E 04741798 (5)**

97 Fecha y número de publicación de la concesión europea: **29.04.2015** **EP 1634253**

54 Título: **Procedimiento de escritura, de actualización y de asignación de memoria aplicado a la escritura de ficheros en un soporte de memoria tal como una tarjeta con chip**

30 Prioridad:

17.06.2003 FR 0307290

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

05.08.2015

73 Titular/es:

**THALES (100.0%)
45 RUE DE VILLIERS
92200 NEUILLY SUR SEINE, FR**

72 Inventor/es:

D'ATHIS, THIERRY

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 542 421 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

Procedimiento de escritura, de actualización y de asignación de memoria aplicado a la escritura de ficheros en un soporte de memoria tal como una tarjeta con chip

5 La presente invención se refiere a las técnicas de lectura y de escritura de datos en un soporte de memoria. Esta se aplica, concretamente, a la lectura y la escritura de datos en tarjetas con chips, utilizándose estas tarjetas en calidad de títulos de transporte en los transportes públicos. La lectura y la escritura se realizan, por ejemplo, mediante validadores de autobús o a la altura de portillos o de torniquetes de metro.

10 El flujo de viajeros que atraviesan un portillo o un torniquete puede alcanzar en período punta aproximadamente de 30 personas por minuto (en Europa) a 60 personas por minuto (en Asia). Los tiques con banda magnética permiten comunicar (lectura) y actualizar (escritura) datos rápidamente. De hecho, el trayecto del tique sigue al del viajero.

15 Sin embargo, estos tiques no permiten almacenar muchos datos. Un tique con banda magnética de formato Edmonson, por ejemplo, contiene 62 bits útiles solamente. Además, estos tiques no permiten realizar transacciones seguras. Pueden utilizarse tarjetas con chips para almacenar más datos. Estas tarjetas pueden estar equipadas, además, con dispositivos de encriptación y de autenticación, con el fin de hacer seguras las transacciones. Estos dispositivos de encriptación y de autenticación permiten garantizar que las escrituras en la tarjeta se limitan solo a los aparatos que tienen el derecho para ello.

20 Las tarjetas con chips pueden, según su naturaleza, leerse con contacto o sin contacto. Las primeras se leen posicionando el chip de la tarjeta en contacto con una cabeza de lectura magnética. Para ello, a menudo, estas tarjetas se insertan en una ranura de lectura. Esta exigencia mecánica impide que se desplace la tarjeta con el viajero para hacerle seguir el mismo trayecto. Por lo tanto, no es posible utilizar estas tarjetas en el ámbito de los transportes. Por lo tanto, se utilizan tarjetas con chips sin contacto.

25 Las tarjetas con chip sin contacto utilizadas en el ámbito de los transportes responden generalmente a la norma ISO 1443 que se refiere a los intercambios de datos (modulación, portadora). La zona de comunicación entre tarjeta y antena se extiende hasta 10 cm. Con el fin de obtener un movimiento fluido durante la marcha del viajero, la duración de una transacción (llamado tiempo de transacción de billete) no debe exceder de 150 ms. A partir de 200 ms, el viajero debe marcar claramente un tiempo de parada.

30 Si el viajero no deja su tarjeta el suficiente tiempo en la zona de comunicación, se interrumpe la transacción en transcurso. Si la transacción se interrumpe en medio de una escritura, los datos registrados no se integran. Por lo tanto, es deseable que la escritura se realice de tal manera que una tarjeta recupere su estado inicial incluso cuando se interrumpe una transacción. En otras palabras, la tarjeta debería contener los datos iniciales si se interrumpe una transacción.

35 Las tarjetas con chip pueden estar equipadas o no con un microprocesador. Las tarjetas con chip microcableadas están desprovistas de microprocesador. El espacio de memoria de estas tarjetas se divide en bloques. El lector accede a estos bloques, tanto de lectura como de escritura, según sus necesidades. Sin embargo, los datos en este tipo de tarjetas no están estructurados: la noción de fichero lógico no existe. Al contrario, las tarjetas, equipadas con un microprocesador permiten presentar los datos registrados de manera estructurada, en forma de ficheros. Pueden citarse las tarjetas con microprocesador que responden a la norma ISO 7816-4 o a la norma EN 726-3. Sin embargo, estas tarjetas son más caras que las tarjetas microcableadas.

40 La patente de invención francesa FR 2635886A describe unos procedimientos para procesar, es decir escribir, leer y seleccionar, datos en una tarjeta con chip. Un problema que se expone en este documento es disponer de una estructura de datos que permita definir un número de fichero no limitado por el tamaño del espacio de memoria asignado a una FAT, acrónimo anglosajón para *File Allocation Table*. Una FAT es una estructura que contiene datos de definición de fichero. Para resolver el problema citado en este documento, la FAT se inscribe partiendo de un extremo de la zona de memoria, y los datos de los ficheros se inscriben partiendo del otro extremo de la zona de memoria.

Una finalidad de la invención es proporcionar un procedimiento de lectura y un procedimiento de escritura de datos de un fichero en un soporte de memoria (tal como una tarjeta con chip sin contacto microcableada) siendo el procedimiento rápido, sencillo de poner en práctica y que permita que el soporte de memoria recupere su estado inicial cuando se interrumpe una transacción (escritura).

50 La puesta en práctica de la invención requiere asignar previamente un grupo (DHS) de espacios (A0...A6) de memoria en la tarjeta. Se asigna un número de espacios de memoria superior al número de registros (E1...E6) del fichero (FHS) considerado. El número de espacios de memoria asignados en exceso permite efectuar las operaciones de actualización según la invención.

55 Durante una operación de actualización, solo se escriben en la tarjeta los datos (d7) nuevos, sin pérdida de los datos (d1) antiguos. Para ello, los datos (d7) nuevos se escriben en espacios (A2) de memoria no ocupados por datos (d1) antiguos del fichero. Estos espacios de memoria se eligen entre los espacios de memoria previamente asignados.

Una vez escritos los datos nuevos, se escribe un descriptor (IHS2) nuevo de fichero que permite recuperar los registros nuevos del fichero entre los espacios de memoria asignados.

La invención se aplica de manera ventajosa a la escritura de ficheros cíclicos.

5 Otras características y ventajas de la invención se mostrarán con la ayuda de la descripción que sigue hecha con respecto a los dibujos adjuntos:

- las figuras 1 y 2 representan un ejemplo de implantación de los datos en la memoria de una tarjeta sin contacto, representando la figura 2 un detalle de la figura 1;
- las figuras 3, 5 y 7 representan un fichero, un ejemplo de implantación de los datos del fichero en la memoria, y un descriptor de este fichero;
- 10 - las figuras 4, 6 y 8 representan los mismos elementos que las figuras 3, 5 y 7 después de una operación de actualización del fichero;
- las figuras 9 a 14, un ejemplo de actualización de un fichero cíclico;
- las figuras 15a a 15c, un ejemplo de cuadro de permutación que permite realizar una codificación ventajosa de los descriptores de fichero.

15 Ahora, se describe un ejemplo de puesta en práctica de la invención en una tarjeta con chip microcableada. Esta tarjeta es una tarjeta sin contacto utilizada como título de transporte (conocida con el nombre de "*contactless ticketing card*" en la literatura anglosajona).

Ahora, se hace referencia a la figura 1, en la que se representa un ejemplo de implantación de datos en la memoria de una tarjeta con chip microcableada. Los datos de una tarjeta se reagrupan por bloques, reagrupándose los propios bloques por sectores S0, S1,..., S15. Un bloque es la unidad más pequeña que puede abordarse.

Ejemplo 1: Memoria de una tarjeta con chip microcableada

En este ejemplo se describe la tarjeta de memoria "MIFARE® STANDARD" de PHILIPS. La tarjeta comprende 64 bloques. Un bloque comprende 128 bits de datos, es decir 16 octetos. Los bloques se reagrupan por grupo de cuatro, formando cada grupo de bloque un sector. Los bloques de un sector se referencian B0, B1, B2, B3 en la figura 1. Por consiguiente, la tarjeta comprende 16 sectores. Los sectores se referencian S0 a S15.

Fin del ejemplo 1.

Los datos escritos en la tarjeta están protegidos. Solo es posible acceder a los datos con una clave de acceso. Una clave de acceso es un secreto criptográfico requerido para poder acceder al espacio de memoria que este secreto protege.

30 A cada clave de acceso se asocia una información sobre los derechos de acceso a los datos que la clave protege. De esta manera, cada clave de acceso limita más o menos los derechos de acceso a los datos.

Los diferentes derechos de acceso pueden ser los siguientes:

- N: acceso prohibido de lectura o de escritura;
- R: acceso autorizado de solo lectura;
- 35 - D: acceso autorizado de lectura, y acceso de escritura limitado a las operaciones que disminuyen el valor de los datos memorizados;
- W: acceso autorizado de lectura y de escritura.

Unos datos que protege una misma clave pueden tener derechos de acceso diferentes. De esta manera, una persona puede acceder con una clave a ciertos datos con derechos amplios (por ejemplo W), y a otros datos con derechos limitados (por ejemplo R).

Unas claves de acceso diferentes pueden proteger unos mismos datos, lo que permite definir derechos de acceso diferentes para estos mismos datos. De esta manera, varias personas que poseen claves de acceso diferentes pueden acceder a los mismos datos con derechos de acceso más o menos amplios.

45 Las claves de acceso pueden escribirse en la tarjeta. Estas claves no pueden leerse. Pueden modificarse, con la condición de que se conozcan las claves ya escritas.

Ejemplo 2: Protección de los datos en la tarjeta del ejemplo 1

El cuarto bloque B3 de cada sector contiene una clave A de acceso, una clave B de acceso opcional, y los derechos de acceso a los cuatro bloques B0 a B3 de este sector. Cada clave de acceso es un número de 48 bits, es decir 6 octetos. Los derechos de acceso pueden codificarse con los bits del bloque B3 no utilizados para codificar las claves.

Los derechos de acceso a los datos de un bloque de datos (bloques B0 a B2) pueden codificarse con tres bits. El cuadro siguiente, de cinco columnas, da un ejemplo de codificación de los derechos de acceso. Las tres primeras columnas contienen los bits que codifican para los derechos de acceso de un bloque. La cuarta columna contiene los derechos de acceso asociados a la clave A. La quinta columna contiene los derechos de acceso asociados a la clave B.

bit n°1	bit n°2	bit n°3	clave A	clave B
0	0	0	W	W
0	0	1	D	D
0	1	0	R	R
0	1	1	N	W
1	0	0	R	W
1	0	1	N	R
1	1	0	D	W
1	1	1	N	N

Los derechos de acceso a los datos del bloque B3 son particulares. De hecho, la modificación de los datos de este bloque permite modificar el valor de una clave y/o los datos que codifican para los derechos de acceso del sector. Los derechos de acceso a los datos del bloque B3 pueden ser:

- 10 - R1: acceso autorizado de solo lectura a los datos que codifican para los derechos de acceso;
- R2: acceso autorizado de solo lectura a los datos que codifican para los derechos de acceso y a la clave B;
- W1: acceso autorizado de lectura y de escritura a los datos que codifican para los derechos de acceso;
- W2R1: acceso autorizado de escritura pero no de lectura a las claves A y B, acceso autorizado de solo lectura a los datos que codifican para los derechos de acceso;
- 15 - W2R2: acceso autorizado de escritura pero no de lectura a la clave A, acceso autorizado de lectura y de escritura a la clave B, acceso autorizado de solo lectura a los datos que codifican para los derechos de acceso;
- W2W1: acceso autorizado de escritura pero no de lectura a las claves A y B, acceso autorizado de lectura y de escritura a los datos que codifican para los derechos de acceso;
- W3: acceso autorizado de escritura pero no de lectura a la clave A, acceso autorizado de lectura y de escritura a la clave B y a los datos que codifican para los derechos de acceso.
- 20

Los derechos de acceso a los datos del bloque B3 pueden codificarse con tres bits. El cuadro siguiente, de cinco columnas, da un ejemplo de codificación de estos derechos de acceso. Las tres primeras columnas contienen los bits que codifican para los derechos de acceso de este bloque. La cuarta columna contiene los derechos de acceso asociados a la clave A. La quinta columna contiene los derechos de acceso asociados a la clave B.

bit n°1	bit n°2	bit n°3	clave A	clave B
0	0	0	W2R2	N
0	0	1	W3	N
0	1	0	R2	N
0	1	1	R1	W2W1
1	0	0	R1	W2R1
1	0	1	R1	W1
1	1	0	R1	R1
1	1	1	R1	R1

Fin del ejemplo 2.

Diferentes máquinas procesan los títulos de transporte. Un primer grupo de máquinas permite realizar operaciones de venta o de recarga. Las máquinas de este primer grupo se llaman máquinas de venta. Comprenden las máquinas de personalización de título, los robots de venta, las máquinas de venta en taquilla. Durante una operación de venta,

la máquina de venta escribe datos relativos al contrato (características del trayecto, duración del abono...). Durante una operación de recarga, la máquina de venta aumenta un crédito, si la tarjeta permite memorizar un crédito (tarjeta monedero electrónico), encontrándose el valor de la tarjeta aumentado con el crédito recargado.

5 Un segundo grupo de máquinas permite controlar la tarjeta y/o disminuir su valor a cambio de un servicio de transporte. Las máquinas de este primer grupo se llaman máquinas de validación. Comprenden los validadores sin obstáculos, los portillos, los aparatos de control portátiles, las máquinas de información, las consolas de autobús. Las máquinas de validaciones no modifican los datos del contrato, en cambio deben acceder a ellos como lectura. Además, las máquinas de validación pueden modificar el valor de la tarjeta, estando representado este valor por datos llamados datos de validación.

10 Por lo tanto, las máquinas de venta y las máquinas de validaciones acceden a los datos de la tarjeta con derechos de acceso diferentes. Estos dos tipos de máquinas utilizan claves diferentes.

Las máquinas de venta pueden utilizar una primera clave KV. Las máquinas de validaciones pueden utilizar una segunda clave KT, siendo limitados los derechos de acceso asociados a la clave KT con respecto a los asociados a la clave KV.

15 **Ejemplo 3: Aplicación a los transportes públicos**

La clave KV permite acceder a los datos de contrato y a los datos de validación como lectura y como escritura (W). La segunda clave KT también permite acceder a los datos de validación como lectura y como escritura (W). Sin embargo, permite acceder a los datos de contrato solo como lectura (R). Las claves KV y KT corresponden respectivamente a las claves B y A de la tarjeta (véase ejemplo 2).

20 Fin del ejemplo 3.

Como se ha indicado más arriba, unos juegos de claves diferentes pueden proteger los datos de diferentes sectores. El hecho de dividir la tarjeta en varios sectores que unas claves de acceso separadas protegen permite utilizar la tarjeta para aplicaciones diferentes (tarjetas multiaplicaciones).

25 Esto es útil también si varios operadores de transporte utilizan una misma tarjeta como título de transporte. Cada operador de transporte puede utilizar un juego de clave que le es propio para acceder a sus datos (como lectura y escritura) de manera exclusiva. Los diferentes operadores pueden compartir otros datos. Los operadores pueden utilizar un juego de claves que comparten para acceder a estos datos de manera no exclusiva.

Ejemplo 4: Aplicación a los transportes públicos, con varios operadores (alternativa al ejemplo 3)

30 La tarjeta contiene datos privados de tres operadores de transporte, y datos que estos operadores comparten. Cada operador accede a sus datos privados de manera exclusiva. Todos los operadores pueden acceder a los datos que comparten. Al igual que en el ejemplo 3, los datos comprenden datos de validación y datos de contrato.

35 Las máquinas de venta de un operador nº i (con $i = 1, 2$ o 3) utilizan una clave KVi privada para acceder a los datos de validación (como lectura y como escritura) y de contrato (como lectura y como escritura) de este operador. Las máquinas de venta de todos los operadores utilizan una clave KVS que comparten para acceder a los datos que comparten (validación, contrato) entre los operadores. Las máquinas acceden a estos datos que comparten con los mismos derechos de acceso.

40 Asimismo, las máquinas de validación de un operador utilizan una clave KTi privada para acceder a los datos de validación (como lectura y como escritura) y de contrato (como solo lectura) de este operador. Las máquinas de validación de todos los operadores utilizan una clave KTS que comparten para acceder a los datos que comparten con los mismos derechos de acceso.

En la tarjeta se escribe también un histórico de las transacciones (conocido con el nombre de "*history long*" en la literatura anglosajona). Las máquinas de validación y las máquinas de venta de todos los operadores pueden acceder a los datos del histórico como lectura y como escritura utilizando respectivamente las claves KTS y KVS.

Fin del ejemplo 4.

45 Los programas de las máquinas de venta y de las máquinas de validación utilizan ficheros para acceder a los datos. Un fichero es una sucesión ordenada de registros. Cada registro comprende unos datos. Un fichero comprende un número determinado de registros, teniendo cada registro un tamaño determinado.

Ejemplo 5: Organización en ficheros de los datos del ejemplo 4

50 Los datos se reagrupan en nueve ficheros FVi , FTi (con $i = 1, 2$ o 3), FVS , FTS , FHS . Estos ficheros comprenden un número de registros fijos. Cada registro comprende 128 bits de datos, lo que corresponde a un bloque. Por supuesto, la invención no se limita a este ejemplo particular: el tamaño de un registro puede ser diferente del tamaño de un bloque.

El cuadro siguiente comprende cinco columnas. La primera columna contiene el nombre del fichero. La segunda columna contiene el número de registro del fichero. La tercera columna contiene una descripción de los datos del fichero. La cuarta columna recuerda los derechos de acceso de las máquinas de validación a estos datos. Estos derechos de acceso se asocian a las claves KTS, KT1, KT2 o KT3 que corresponden a la clave A de la tarjeta (véase ejemplo 2). La quinta columna recuerda los derechos de acceso de las máquinas de venta a estos datos. Estos derechos de acceso se asocian a las claves KVS, KV1, KV2 o KV3 que corresponden a la clave B de la tarjeta (véase ejemplo 2).

Fichero	Regist.	Datos	clave A	clave B
FTi	1	datos de validación del operador n° i	R	W
FTi	1	datos de contrato del operador n° i	W	W
FVS	1	datos de validación que comparten	R	W
FTS	3	datos de contrato que comparten	W	W
FHS	6	histórico de las transacciones	W	W

Fin del ejemplo 5.

Según la invención, se asignan unos espacios de memoria de la tarjeta para escribir los registros de los ficheros. Cada registro tiene por objeto escribirse en un espacio de memoria asignado. La asignación de espacios de memoria se traduce en una determinación de parámetros que utilizan los programas de las máquinas que ponen en práctica los procedimientos de escritura o de lectura según la invención. De esta manera, se realiza una asignación de espacios de memoria durante la programación de los softwares que equipan estas máquinas.

Se realiza una asignación de espacios de memoria para un grupo de registros seleccionados. Este grupo comprende un número determinado de registros. Cada registro del grupo tiene un tamaño determinado. Los registros de este grupo pueden ser, por ejemplo, los registros de un fichero. Pero la invención no se limita a este ejemplo. El grupo de registros puede estar formado por una parte solamente de los registros de un fichero dado. Por otra parte, los registros del grupo pueden pertenecer a varios ficheros.

Algunos (incluso todos los) registros de este grupo tienen por objeto actualizarse. Esta actualización consiste en sustituir los datos de estos registros por datos nuevos.

Según la invención, se determina un número N de espacios de memoria que se van a asignar. Este número excede el número de registros del grupo. Los espacios de memoria en exceso se asignan para escribir en ellos los datos nuevos durante una actualización. El número de espacios de memoria en exceso es al menos igual al número máximo de registros del grupo de registros susceptibles de actualizarse simultáneamente durante una operación de escritura.

Se determina un grupo de espacios de memoria en el soporte de memoria. Este grupo comprende N espacios de memoria. Cada espacio de memoria tiene una posición determinada y un tamaño determinado. El tamaño de los espacios de memoria es suficiente para escribir en ellos los datos de los registros del grupo.

Ahora, se hace referencia a las figuras 3 y 5 en las que se representan el fichero FTS y un ejemplo de espacios de memoria asignados para este fichero. El fichero FTS comprende tres registros E1, E2, E3. Pueden modificarse como máximo dos de estos registros durante una actualización de los datos de este fichero. Por lo tanto, se asigna un espacio de memoria para cada registro, y dos espacios de memoria suplementarios para actualizar los datos de este fichero. En consecuencia, se asigna en total un grupo DTS de cinco espacios A0, A1, A2, A3, A4 de memoria para el fichero FTS, asignándose dos espacios de memoria en exceso con respecto a los registros del fichero.

La posición de cada espacio de memoria asignado es un parámetro de los programas que ejecutan las máquinas que ponen en práctica los procedimientos de escritura o de lectura según la invención. Se determina la posición de cada espacio de memoria, de manera que cada registro del fichero pueda escribirse en un espacio de memoria entre los espacios de memoria asignados. El tamaño de un espacio de memoria asignado (y la separación entre dos espacios de memoria) es al menos igual al tamaño del registro que está escrito en él o que tiene por objeto escribirse en él.

Ventajosamente, con el fin de que sea compatible con los mecanismos de seguridad que la tarjeta pone en práctica, las mismas claves de acceso protegen los espacios de memoria asignados para un fichero determinado. Ventajosamente, los mismos derechos de acceso se asocian a las claves de acceso que protegen espacios de memoria asignados para un fichero determinado.

Ejemplo 6: Asignación de los espacios de memoria de la tarjeta de los ejemplos 1 y 2 para escribir datos del ejemplo 4

Se hace referencia a las figuras 1 y 2. Se asignan unos grupos DV1, DT1, DV2, DT2, DV3, DT3, DVS, DTS, DHS de espacios de memoria para escribir los registros de los ficheros FV1, FT1, FV2, FT2, FV3, FT3, FVS, FTS, FHS.

- 5 El cuadro siguiente comprende cinco columnas. La primera columna contiene la referencia a un grupo de espacios de memoria asignados. La segunda columna contiene el número de bloques asignados (1 bloque = 1 espacio de memoria en este ejemplo). La tercera columna contiene el nombre del fichero del que los registros tienen por objeto escribirse en estos espacios de memoria. La cuarta columna recuerda el número de registros del fichero. La quinta columna contiene el número de registros máximo que pueden modificarse durante una actualización del fichero.

Espacios asignados	Nº de bloques asignados	Fichero	Nº registros	Actualización
DV _i	2	FT _i	1	1
DT _i	4	FT _i	2	2
DVS	2	FVS	1	1
DTS	5	FTS	3	2
DHS	7	FHS	6	1

- 10 Por supuesto, pueden asignarse espacios de memoria para escribir los registros de varios ficheros, y no de un fichero único. En la medida en que se modifica uno solo de los ficheros durante cada actualización, esto permite reducir el número de espacios de memoria que se van a asignar en exceso.

Fin del ejemplo 6.

- 15 Después de haber asignado espacios de memoria del soporte de memoria, puede realizarse la escritura de los datos según la invención. Se escribe cada registro del fichero en un espacio de memoria entre los espacios de memoria asignados. A continuación, se escribe un descriptor del fichero en el soporte de memoria. El descriptor del fichero es un conjunto de datos que permiten determinar en qué espacio de memoria asignado se escribe cada registro del fichero.

- 20 Se hace referencia a las figuras 3, 5 y 7 para describir un ejemplo de escritura de los datos de un fichero. El fichero es el fichero FTS que se comparte de los ejemplos anteriores. Los espacios A0, A1,... A4 de memoria asignados para escribir los datos de este fichero son en total cinco. El fichero FTS comprende tres registros E1, E2, E3. Se escriben los datos d1, d2, d3 de estos registros en tres espacios entre los cinco asignados. Por ejemplo como se ilustra en la figura 5, los datos d1 del registro E1 se escriben en el espacio A3 de memoria asignado, los datos d2 del registro E2 se escriben en el espacio A1 de memoria asignado, y los datos d3 del registro E3 se escriben en el espacio A4 de memoria asignado. A continuación, se escribe el descriptor ITS1 del fichero FTS. El descriptor ITS1 hace referencia a los espacios A3, A1, A4 de memoria, tomados en este orden. Este permite recuperar los registros del fichero FTS.

- 30 Según la invención, una vez escritos los registros de un fichero, es posible actualizarlos de tal manera que los datos registrados del fichero no cambian si se interrumpe una operación de escritura antes del final de la actualización.

Se hace referencia a las figuras 4, 6 y 7 para describir un ejemplo de actualización del fichero FTS. Esta actualización tiene lugar después de la etapa de escritura descrita en relación con las figuras 3, 5 y 7. Durante esta etapa de actualización, los datos d2 y d3, que los registros E2 y E3 contienen respectivamente, tienen por objeto ser sustituidos por datos d4 y d5 nuevos.

- 35 Antes se lee el descriptor ITS1 de fichero (véase figura 7). Se deducen de él los espacios de memoria, que entre los espacios DTS de memoria asignados, el descriptor ITS no referencia. En este ejemplo, se trata de los espacios A0 y A2 de memoria. Se escriben los datos d4, d5 nuevos en espacios de memoria entre los espacios de memoria no referenciados. En este ejemplo, los datos d4 se escriben en el espacio A0 de memoria y los datos d5 se escriben en el espacio A2 de memoria. Se escribe un descriptor ITS2 nuevo del fichero en el soporte de memoria. El descriptor ITS2 nuevo referencia los espacios A0 y A2 de memoria en lugar de los espacios A1 y A4 de memoria respectivamente. En otras palabras, ITS2 referencia los espacios de memoria en los que se escriben los datos d4, d5 nuevos en lugar de los espacios de memoria en los que están escritos los datos d2, d3 antiguos.

- 40 Un descriptor de fichero, tal como ITS1 o ITS2, puede contener, por ejemplo, un número de orden de cada espacio de memoria asignado. Este número de orden puede variar en este ejemplo de cero a cuatro. En los ejemplos anteriores, el descriptor ITS1 comprende los números tres, uno y cuatro. El descriptor ITS2 comprende, por su parte, los números tres, cero y dos. Cada uno de los números de un descriptor puede codificarse en binario sobre tres bits.

De esta manera, pueden codificarse en binario los tres números que forman el descriptor ITS1 o ITS2 sobre nueve bits.

Este modo de realización, en el que no se referencian los espacios de memoria libres, permite minimizar el tamaño del descriptor de fichero. Sin embargo, esto implica que se lea el conjunto del descriptor de fichero para deducir de él los espacios de memoria libres.

Según otro modo de realización (no representado), los descriptores ITS1 y/o ITS2 referencian no solamente los espacios de memoria utilizados, sino también los espacios de memoria libres. Por ejemplo, los espacios de memoria libres pueden ser los últimos espacios de memoria referenciados en el descriptor de fichero. Este modo de realización permite encontrar más rápidamente los espacios de memoria libres. Además, permite repartir el desgaste entre los bloques de memoria, encargándose de que no se escriban siempre los datos nuevos en los mismos espacios de memoria.

No obstante, este modo de realización es interesante en la medida en que no impone que se escriba el descriptor de fichero sobre un bloque de datos suplementario. De hecho, si hay que leer un bloque más, se pierde más tiempo en leer datos (tiempo de acceso) que en determinar los espacios de memoria libres (no referenciados). En consecuencia, se preferirá en este caso el primer modo de realización descrito.

Según un modo de realización ventajoso, los registros de un fichero, para los que se asigna un grupo de espacio de memoria, tienen todos el mismo tamaño. Esto permite limitar el tamaño de los espacios de memoria que se van a asignar. Ventajosamente, el tamaño de estos registros es igual a un número entero de bloques de memoria. Ventajosamente, los espacios de memoria tienen exactamente el mismo tamaño que los registros. Cuando todos los registros del fichero tienen el mismo tamaño, el descriptor de fichero permite encontrar los datos del fichero realizando una permutación de los datos registrados en los espacios de memoria asignados para este fichero.

Con el fin de comprobar si se han integrado unos datos registrados, puede utilizarse un sello. Un sello es un valor que varía en función de los datos que se van a comprobar. Este sello se registra en la tarjeta. Se leen los datos registrados y se calcula el valor asociado a estos datos. Este valor calculado se compara con el valor del sello registrado. Si estos valores son idénticos, se considera que los datos registrados se han integrado. Una forma clásica de realizar un sello es calcular un código de redundancia cíclica (CRC).

No obstante, la utilización de un CRC no es satisfactoria para comprobar la integridad de los datos registrados en una tarjeta usada (tarjeta con chip o con banda magnética). Con el tiempo, ciertos bits pueden conservar un valor constante, que corresponde al estado de menor energía (cero o uno según la tecnología empleada). Estos bits defectuosos pueden ser tanto para bits del sello como bits de datos. En consecuencia, puede encontrarse uno en una situación en la que se compensan errores de escritura en el sello y los datos, de tal manera que se considera que unos datos erróneos se han integrado.

Según un modo de realización de la invención, se determina un sello mediante recuento del número de ceros registrados. Por ejemplo, si unos datos registrados sobre ocho bits contienen el número 240 (o sea "11110000" en binario), el valor del sello asociado es 4 (o sea "100" en binario). Si el nivel de energía escasa es cero, ciertos bits del número registrado son susceptibles de permanecer con un cero. El valor del sello registrado será estrictamente inferior al número de ceros registrados. Aunque unos bits del sello sean erróneos (cero en lugar de uno), el valor del sello registrado permanecerá estrictamente inferior al número de ceros registrados en los datos. El recuento del número de cero permite obtener el mismo resultado si el nivel de energía escasa es uno (el valor del sello permanece estrictamente superior al número de ceros registrados en los datos). En consecuencia, la comprobación de integridad mediante el sello permanece fiable, aunque el sello contenga bits erróneos. En otras palabras, el desgaste de la tarjeta no puede provocar errores que se compensen.

Por supuesto, el sello puede ser el número de bits con un cero al que se añade una constante aditiva. Puede multiplicarse este valor por cualquier constante estrictamente positiva. De una manera general, el valor del sello es una función creciente del número de ceros contenidos en los datos que se van a comprobar.

El sello puede aplicarse a los datos del fichero. Según un modo de realización ventajoso, el sello se aplica a los datos del descriptor de fichero, y no a los datos del propio fichero. De esta manera, puede comprobarse la integridad del descriptor. Se considera que la operación de escritura se ha desarrollado correctamente si el descriptor se escribe correctamente, en la medida en que la etapa de escritura del descriptor sigue a la etapa de escritura de los datos del fichero.

Durante la lectura de los datos del fichero, se verifica la integridad de los datos leídos con el sello. De esta manera, puede comprobarse si se ha desarrollado bien la última transacción (operación de escritura). En caso de error (transacción interrumpida de manera inesperada, por ejemplo), la invención permite recuperar los datos antiguos del fichero. De esta manera, pueden preservarse los datos antiguos registrados en la tarjeta, incluso cuando se ha interrumpido una transacción de manera inesperada.

El descriptor del fichero que corresponde a los datos antiguos (fichero antes de la actualización) puede conservarse en memoria. Por lo tanto, para un mismo fichero se utilizan dos descriptores (antiguo y nuevo). En cada operación

de actualización de un fichero, puede escribirse el descriptor nuevo en lugar del más antiguo.

Con el fin de reconocer el antiguo del nuevo descriptor, puede registrarse un número de secuencia que permite datar los descriptors. Para reconocer el antiguo del nuevo descriptor, es suficiente con que se codifique el número de secuencia sobre dos bits (número comprendido entre cero y tres).

- 5 El cuadro siguiente da el descriptor más reciente en función de los números de secuencia asociados a dos descriptors A y B:

Número de secuencia asociado a A	Número de secuencia asociado a B	Descriptor más reciente
0	1	B
0	3	A
1	0	A
1	2	B
2	1	A
2	3	B
3	0	B
3	2	A

Las combinaciones no representadas (dos números de secuencia de misma paridad) corresponden a estados anormales.

- 10 Según una variante de realización, un número de secuencia puede utilizarse para comprobar la edad de la tarjeta. Cuando este número alcanza un valor determinado, que corresponde a un estado de desgaste medio predeterminado, puede ponerse en práctica un mantenimiento preventivo que consiste en sustituir la tarjeta antigua por una tarjeta nueva. En otras palabras, gracias al número de secuencia puede detectarse el momento en el que debe sustituirse una tarjeta.

- 15 Cuando las tarjetas se utilizan como títulos de transporte, puede realizarse una operación llamada de ratificación. Esta operación consiste en inscribir datos en la tarjeta, probando estos datos que la transacción (actualización) que ha operado una máquina se ha desarrollado bien. Se dice que una tarjeta está ratificada cuando se ha efectuado una ratificación. Una tarjeta solo puede ratificarse si, como continuación a una transacción, la máquina ha podido verificar que esta se ha desarrollado bien.

- 20 Según un modo de realización de la invención, se realiza una ratificación copiando el descriptor nuevo sobre el descriptor antiguo. Una tarjeta ratificada posee dos descriptors idénticos. Si los descriptors son diferentes, la tarjeta no se ha ratificado.

Entonces, ya no es necesario inscribir un número de secuencia para reconocer el nuevo y el antiguo descriptor. Para ello, puede dedicarse una primera ubicación de memoria a la escritura de los descriptors nuevos, conteniendo la otra ubicación la copia realizada durante las ratificaciones. Durante las actualizaciones, el descriptor nuevo continúa todavía escrito en la primera ubicación de memoria. Al final de la ratificación, se recopia en la otra ubicación. Por lo tanto, esta otra ubicación contiene todavía el descriptor antiguo si la ratificación no tiene lugar.

Ahora, se hace referencia a las figuras 9 a 14 en las que se representa un ejemplo de actualización de un fichero cíclico.

- 30 La figura 9 representa el fichero FHS (histórico de transacciones). Este fichero consta de seis registros referenciados E1 a E6. Estos registros se inscriben en memoria en el grupo DHS de siete espacios de memoria. El grupo DHS de espacio de memoria (véase figura 11), asignado para el fichero FHS, contiene inicialmente datos referenciados d6 a d1. Un primer descriptor IHS1 de fichero comprende, por ejemplo, un número de orden de cada espacio de memoria que corresponde a cada registro. El descriptor IHS1 representado (véase figura 13) comprende la secuencia 3; 1; 5; 6; 0; 4, lo que significa que los datos de los registros E1 a E6 se memorizan respectivamente en los espacios A3, A1, A5, A6, A0 y A4 de memoria.

Como se ilustra en la figura 10, durante la operación de actualización de FHS, se añade un dato d7 en el registro E1, se desplazan los datos d6 a d2 de un registro, y se suprime el dato d1 (antiguamente en el registro E6).

- 40 Para ello, se escribe el dato d7 nuevo en el espacio A2 de memoria libre, es decir el espacio de memoria que IHS1 no referencia (véase figura 12) en este ejemplo de realización. Se inscribe un descriptor IHS2 nuevo de fichero (véase figura 14). Se señala que el descriptor nuevo de fichero se deduce muy sencillamente del antiguo. El

descriptor nuevo comprende la secuencia 2; 3; 1; 5; 6; 0. Esta secuencia se deduce del descriptor antiguo añadiendo el número 2 en la cabecera de secuencia, efectuando un desplazamiento de los otros números, y suprimiendo el último número de la secuencia.

5 De esta manera, la invención permite actualizar los datos del fichero cíclico de manera muy sencilla. Es suficiente con que se escriban sencillamente los datos d7 nuevos, y se escriba el descriptor IHS2 nuevo.

Según un modo de realización ventajoso, un número que representa un índice en una tabla predeterminada codifica los descriptores de fichero. Esta tabla contiene todas las colocaciones posibles de los registros del fichero en los espacios de memoria asignados.

10 Ahora, se hace referencia a las figuras 15a a 15c. Estas figuras representan un ejemplo de tabla para codificar los descriptores ITS1 o ITS2.

15 La tabla comprende una primera columna ITS, que es el índice de colocación. Las tres columnas siguientes contienen la referencia del espacio de memoria asociado a cada uno de los tres registros E1, E2, E3 del fichero FTS. El número de colocaciones posibles de tres registros en cinco espacios de memoria es de sesenta ($5! / 2! = 60$). Por lo tanto, el número único que forma el descriptor ITS1 o ITS2 puede codificarse en binario con seis bits en lugar de nuevo anteriormente. El descriptor ITS1, que contenía la secuencia 3; 1; 4 puede sustituirse por el índice 41. El descriptor ITS2, que contenía la secuencia 3; 0; 2, puede sustituirse por el índice 37.

De una manera general, el número de colocaciones posibles es igual a $N!/P!$, donde N es el número total de espacios de memoria asignados y P es igual al número de espacios de memoria asignados en exceso con respecto al número de registros seleccionados.

20

REIVINDICACIONES

1. Procedimiento de escritura y de actualización de un fichero (FTS) escrito en un soporte de memoria que utiliza dos descriptores del fichero, comprendiendo el fichero un número determinado de registros (E1, E2, E3) de tamaños determinados, teniendo por objeto una parte de estos registros (E2, E3) actualizarse con datos (d4, d5) nuevos, que
5 substituyen datos (d2, d3) antiguos, asignándose un número determinado de espacios (A0, A1, A2, A3, A4) de memoria del soporte de memoria para escribir los datos de los registros del fichero, determinándose la posición de cada espacio de memoria, escribiéndose los datos de cada registro del fichero en un espacio de memoria entre los espacios de memoria asignados, escribiéndose un primer descriptor (ITS1) del fichero en el soporte de memoria, referenciando el descriptor del fichero, en el orden de los registros del fichero, espacios de memoria entre los
10 espacios de memoria asignados para permitir que se determine en qué espacio de memoria asignado se escriben los datos de cada registro del fichero, excediendo el número de espacios de memoria asignados el número total de registros del fichero, siendo el número de espacios de memoria en exceso al menos igual al número máximo de registros de un grupo de registros susceptibles de actualizarse simultáneamente durante una operación de escritura, comprendiendo el procedimiento las siguientes etapas:

15 - se lee el descriptor del fichero,
 - se deducen de él los espacios (A0, A2) de memoria asignados libres,
 - se escriben los datos (d4, d5) nuevos en espacios de memoria entre los espacios de memoria asignados libres,
 - a continuación, se escribe un descriptor (ITS2) nuevo del fichero en el soporte de memoria, referenciando el
20 descriptor nuevo los espacios (A0, A2) de memoria en los que se escriben los datos (d4, d5) nuevos en lugar de los espacios (A1, A4) de memoria en los que están escritos los datos (d2, d3) antiguos,
 - se conserva en memoria el primer descriptor del fichero,
 - en cada operación de actualización del fichero posterior, se escribe un descriptor nuevo del fichero en lugar del descriptor más antiguo de los dos descriptores del fichero.

25 2. Procedimiento de escritura y de actualización según la reivindicación 1, **caracterizado porque** los registros de un fichero, para los que se asigna un grupo de espacio de memoria, tienen todos el mismo tamaño.

3. Procedimiento de escritura y de actualización según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** el descriptor está codificado por un número que representa un índice de colocación en una
30 tabla predeterminada, conteniendo dicha tabla todas las colocaciones posibles de los registros del fichero en los espacios de memoria asignados.

4. Procedimiento de escritura y de actualización según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** se registra un sello, asociado a los datos de cada descriptor, siendo el sello una función creciente del número de bits con un cero del descriptor.

5. Procedimiento de escritura y de actualización según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** se registra un número de secuencia, asociado a cada descriptor.
35

6. Procedimiento de escritura y de actualización según la reivindicación 5, **caracterizado porque** el número de secuencia se codifica sobre dos bits.

7. Procedimiento de escritura y de actualización según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** los espacios de memoria libres no se referencian para minimizar el tamaño del descriptor de
40 fichero.

8. Procedimiento de escritura y de actualización según una cualquiera de las reivindicaciones 1 a 4 o 7, **caracterizado porque** se copia el descriptor (ITS2) nuevo sobre el primero (ITS1) para efectuar una ratificación.

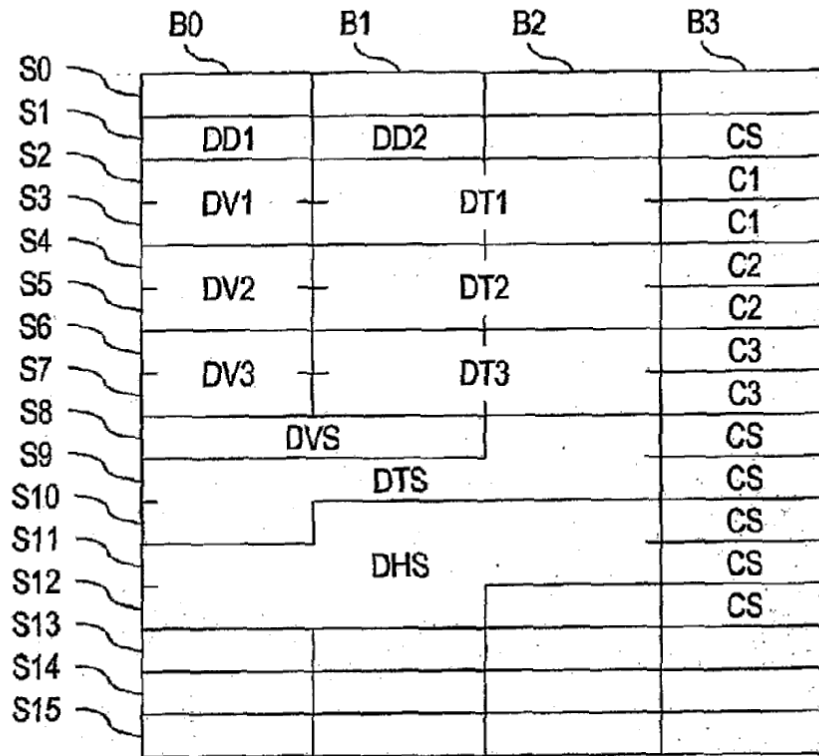


FIG. 1

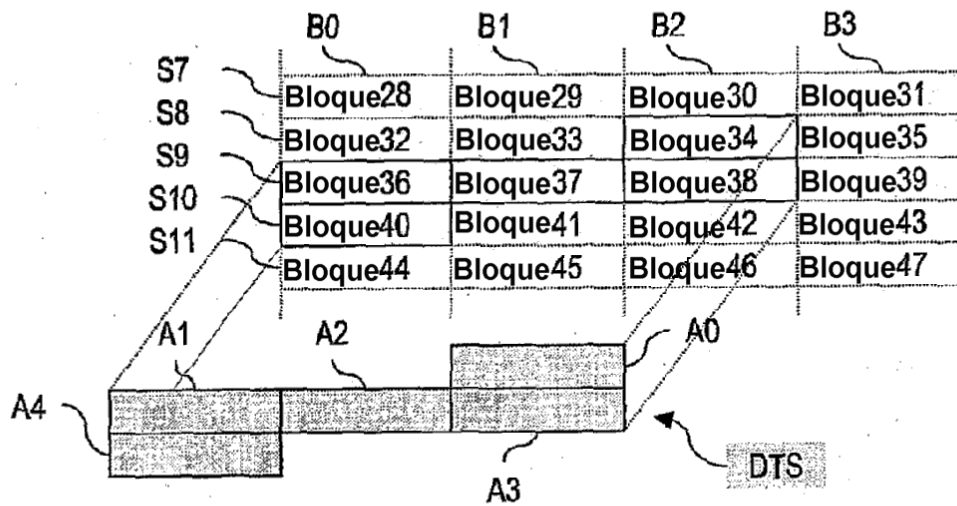


FIG. 2

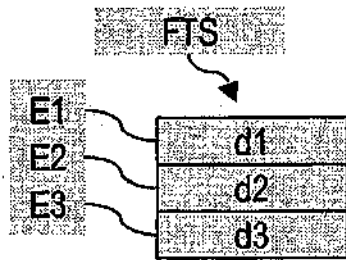


FIG. 3

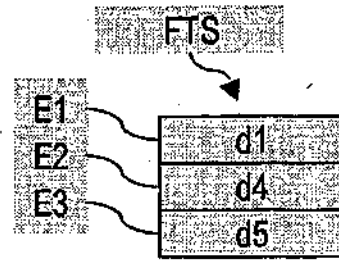


FIG. 4

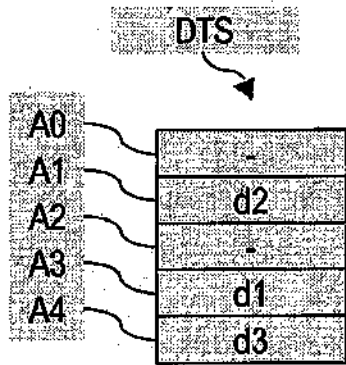


FIG. 5

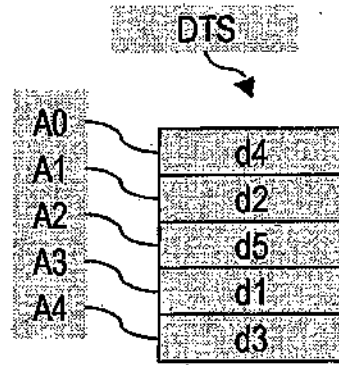


FIG. 6

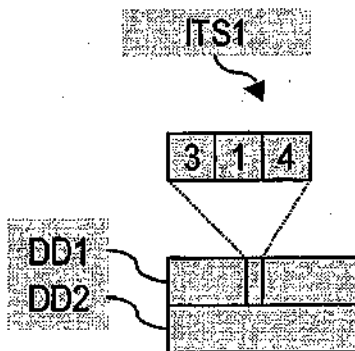


FIG. 7

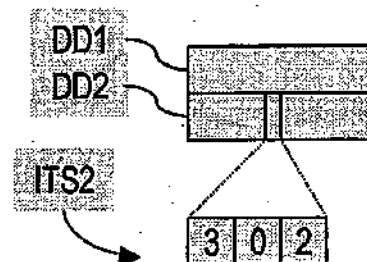


FIG. 8

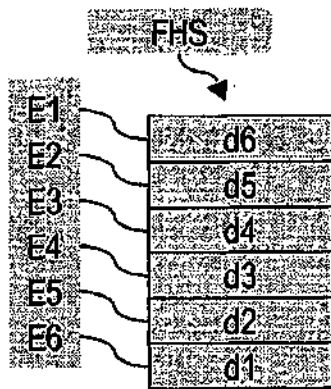


FIG. 9

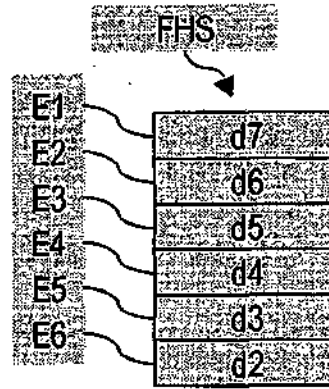


FIG. 10

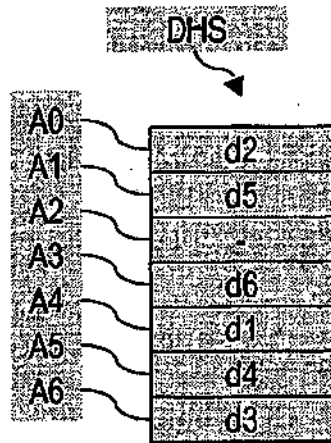


FIG. 11

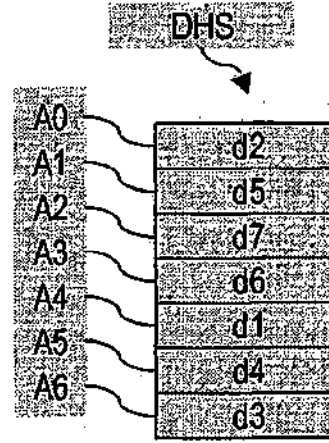


FIG. 12

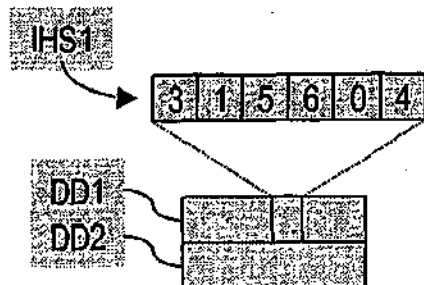


FIG. 13

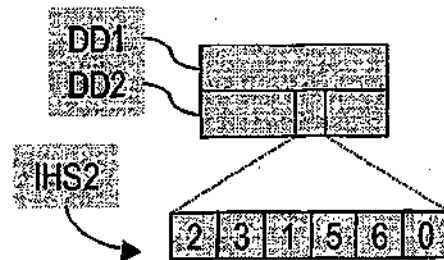


FIG. 14

ITS	E1	E2	E3
0	A0	A1	A2
1	A0	A1	A3
2	A0	A1	A4
3	A0	A2	A1
4	A0	A2	A3
5	A0	A2	A4
6	A0	A3	A1
7	A0	A3	A2
8	A0	A3	A4
9	A0	A4	A1
10	A0	A4	A2
11	A0	A4	A3
12	A1	A0	A2
13	A1	A0	A3
14	A1	A0	A4
15	A1	A2	A0
16	A1	A2	A3
17	A1	A2	A4
18	A1	A3	A0
19	A1	A3	A2

FIG. 15a

ITS	E1	E2	E3
20	A1	A3	A4
21	A1	A4	A0
22	A1	A4	A2
23	A1	A4	A3
24	A2	A0	A1
25	A2	A0	A3
26	A2	A0	A4
27	A2	A1	A0
28	A2	A1	A3
29	A2	A1	A4
30	A2	A3	A0
31	A2	A3	A1
32	A2	A3	A4
33	A2	A4	A0
34	A2	A4	A1
35	A2	A4	A3
36	A3	A0	A1
37	A3	A0	A2
38	A3	A0	A4
39	A3	A1	A0

FIG. 15b

ITS	E1	E2	E3
40	A3	A1	A2
41	A3	A1	A4
42	A3	A2	A0
43	A3	A2	A1
44	A3	A2	A4
45	A3	A4	A0
46	A3	A4	A1
47	A3	A4	A2
48	A4	A0	A1
49	A4	A0	A2
50	A4	A0	A3
51	A4	A1	A0
52	A4	A1	A2
53	A4	A1	A3
54	A4	A2	A0
55	A4	A2	A1
56	A4	A2	A3
57	A4	A3	A0
58	A4	A3	A1
59	A4	A3	A2

FIG. 15c