

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 368 829**

51 Int. Cl.:
G06F 11/14 (2006.01)
G07F 7/10 (2006.01)
G06F 17/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **02019027 .8**
96 Fecha de presentación: **27.08.2002**
97 Número de publicación de la solicitud: **1308842**
97 Fecha de publicación de la solicitud: **07.05.2003**

54 Título: **PROCEDIMIENTO Y DISPOSITIVO PARA ADMINISTRAR UN DISPOSITIVO DE MEMORIA.**

30 Prioridad:
28.08.2001 DE 10141926

45 Fecha de publicación de la mención BOPI:
22.11.2011

45 Fecha de la publicación del folleto de la patente:
22.11.2011

73 Titular/es:
**GIESECKE & DEVRIENT GMBH
PRINZREGENTENSTRASSE 159
81677 MÜNCHEN, DE**

72 Inventor/es:
**Vollmann, Siegfried;
Wallhäusser, Curd y
Hosseini, Manucher**

74 Agente: **Arpe Fernández, Manuel**

ES 2 368 829 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 y dispositivo para administrar un dispositivo de memoria

[001] La invención se refiere en general al campo de la administración de una memoria de datos que presenta varias palabras de memoria agrupadas en páginas de memoria y en la que, para escribir o regrabar una palabra de memoria, se requiere un acceso de lectura y/o borrado y/o escritura a toda la página de memoria en la que se halla la palabra de memoria en cuestión. La invención se refiere en particular al procesamiento de transacciones con, respectivamente, al menos una operación de escritura atómica, de tal modo que, en caso de interrumpirse una transacción, el contenido de la memoria se restaura a un estado que existiría si, al menos en lo relativo a sus operaciones de escritura atómicas, la transacción nunca hubiese tenido lugar. El campo de la invención se define más detalladamente en el preámbulo de la reivindicación 1.

[002] Los sistemas compactos de recursos limitados, como los que se emplean por ejemplo en forma de tarjetas chip o módulos de mando para diversos aparatos, deben presentar por una parte un gran rendimiento con bajos costes y pequeñas dimensiones y garantizar por otra parte un funcionamiento seguro incluso en condiciones ambientales desfavorables. El requisito mencionado en segundo lugar motiva en particular que un corte de corriente repentino o cualquier otro fallo de funcionamiento no deba en ningún caso producir una corrupción de datos no prevista. Especialmente en relación con las tarjetas chip, éste es un requisito importante, porque en los sistemas de las tarjetas chip están en juego frecuentemente intereses de seguridad considerables y porque la separación encauzada de la tensión de alimentación de las tarjetas chip constituye un procedimiento de ataque bien conocido.

[003] Las consideraciones de seguridad mencionadas se tratan por ejemplo en la especificación de la JavaCard™ de la empresa Sun Microsystems Inc. El documento "*JavaCard™ 2.1.1 Runtime Environment (JCRE) Specification*", revisión 1.0 del 18 de mayo de 2000 (disponible en <http://java.sun.com/products/javacard>), especifica en la sección 7 una serie de requisitos concernientes a la administración de transacciones y accesos de escritura atómicos. Además, en el documento "*JavaCard™ 2.1.1 Application Programming Interface*", revisión 1.0 del 18 de mayo de 2000 (disponible también en la dirección indicada), se especifican métodos para iniciar, confirmar y cancelar una transacción, así como para la copia y el llenado atómicos y no atómicos de zonas.

[004] Ya se conocen procedimientos de administración de memoria y tarjetas chip correspondientes a este estándar JavaCard™. En un estado actual de la técnica de la solicitante, al menos interno de la empresa, se emplean tarjetas chip cuya memoria de datos está formada por una EEPROM registrable octeto a octeto. Al realizar una operación de escritura atómica durante una transacción se salva en primer lugar en una memoria tampón de recuperación (en inglés: *commit buffer*) el contenido original de las palabras de memoria que se van a sobrescribir. A continuación se sobrescriben octeto a octeto las palabras de memoria de la memoria de datos EEPROM. Si no es posible concluir con éxito este proceso de escritura, por ejemplo debido a un corte repentino de la alimentación de corriente, en el siguiente arranque de la tarjeta chip se utiliza la información almacenada en la memoria tampón de recuperación para restaurar el estado original de la memoria de datos. Si la transacción concluye con éxito, se borra esta información.

[005] Este sistema con memoria de datos EEPROM pone además a disposición las operaciones de escritura no atómicas predefinidas según la especificación JavaCard™ durante una transacción en curso. En estas operaciones de escritura, los datos a escribir se graban sin protección de transacción directamente en las palabras de memoria de la memoria de datos EEPROM. Tales procesos de escritura no atómicos utilizan muy pocos recursos, porque no se requiere un espacio adicional en la memoria tampón de recuperación y además quedan suprimidos los procesos de transferencia de datos, escritura y administración necesarios para la protección de transacción. Una operación de escritura no atómica concluida con éxito dentro de una transacción no se anula ni en el caso de una posible interrupción posterior de la transacción, porque los datos almacenados en la memoria tampón de recuperación se refieren sólo a las palabras de memoria modificadas por las operaciones de escritura atómicas de la transacción. Aquí se presupone no obstante que las áreas de memoria afectadas por los procesos de escritura atómicos y no atómicos dentro de una transacción no se solapan.

[006] Sin embargo, esta ejecución de la interfaz para programas de aplicación JavaCard™ presupone una memoria de datos registrable octeto a octeto y no puede trasladarse a otras técnicas de almacenamiento en las que los bits de una palabra de memoria se ponen individualmente en un único valor binario, por ejemplo el valor "1", pero respectivamente sólo pueden borrarse todos los bits de una página de memoria completa al otro valor binario, por ejemplo el valor "0". Esta limitación se encuentra por ejemplo en tarjetas chip cuya memoria de datos está configurada como una, así llamada, memoria FLASH.

[007] Sin embargo, la utilización de memorias FLASH es deseable en particular para tarjetas chip de alto rendimiento, por ejemplo tarjetas chip que funcionen con técnica de 32 bits, porque, en comparación con las EEPROM convencionales, las memorias FLASH tienen considerables ventajas en lo que se refiere al tiempo de escritura, la tensión de programación necesaria y la densidad de almacenamiento alcanzable. Por lo tanto, existe la necesidad de poner la funcionalidad arriba descrita a disposición también de memorias de datos que empleen tecnología FLASH u otras tecnologías de almacenamiento con características similares.

[0008] Según la nueva versión 2.2 de la especificación JavaCard™, aparecida en junio de 2002, se admiten, en comparación con la versión 2.1.1, libertades considerablemente mayores en el tratamiento de operaciones de escritura no atómicas durante las transacciones en curso. Ahora, en caso de interrumpirse la transacción, ya no es necesario conservar el efecto de una operación de escritura no atómica concluida con éxito. Por regla general es más bien admisible que las áreas de memoria afectadas por operaciones de escritura no atómicas dentro de una transacción presenten un estado cualquiera, indefinido, tras una interrupción de la transacción o un suceso de corte (*tear*). Este comportamiento se describe en las secciones 7.5 y 7.6 del documento "*JavaCard™ 2.2 Runtime Environment (JCRE) Specification*", junio de 2002 (disponible en <http://java.sun.com/products/javacard>).

[0009] Sin embargo, una excepción a esta regla general la constituyen los accesos de escritura a áreas de datos críticas para la seguridad, por ejemplo accesos de escritura a contadores de intentos fallidos como los realizados por el método *javacard.framework.pin.check*. Una llamada de este método hace que un número de identificación personal (PIN = *personal identification number*) introducido por el usuario se compare con un número de identificación almacenado en la tarjeta chip. En caso de un intento fallido, o sea una falta de coincidencia entre los dos números, se disminuye el contador de intentos fallidos, como se describe en la página 98 del documento "*JavaCard™ 2.2 Application Programming Interface*" de junio de 2002.

[0010] Si el acceso de escritura al contador de intentos fallidos se realiza durante una transacción en curso, el nuevo valor indicado por el contador de intentos fallidos no debe volver a modificarse en manera alguna en caso de producirse un suceso *tear* y el proceso de recuperación subsiguiente. Esto es válido también si el contador de intentos fallidos se halla en una página de memoria afectada por procesos de escritura atómicos durante la transacción. Por lo tanto, en la elección de palabras del presente documento, la escritura del contador de intentos fallidos es siempre un proceso de escritura irrevocable y no atómico, incluso aunque para ello se utilice en algunas configuraciones otra designación - por ejemplo *system direct write (sistema de escritura directa)* - por motivos relacionados con la técnica de implementación.

[0011] En la elección de palabras del presente documento, el concepto "operación de escritura no atómica" debe entenderse en el sentido de la versión 2.1.1 de la especificación JavaCard™, o sea como operación de escritura cuyo efecto se conserva en cualquier circunstancia, incluso en caso de una interrupción de la transacción. Por el contrario, las operaciones de escritura que podrían designarse como "no atómicas" según la versión 2.2 de la especificación JavaCard™ se denominan en el presente documento "operaciones de escritura libres". De los accesos de escritura mencionados en la versión 2.2 de la especificación JavaCard™, únicamente los accesos de escritura irrevocables críticos para la seguridad - por ejemplo a contadores de intentos fallidos - constituyen operaciones de escritura no atómicas en la dicción del presente documento.

[0012] Por el artículo "*Implementation of Transactional Mechanisms for Open SmartCard*" de S. Lecomte, G. Grimaud y D. Donsez, publicado en Actes de GDC 1999 Gemplus Developer Conference, París, CNIT, junio de 1999 (actualmente disponible en <http://www.univ-valenciennes.fr/limav/donsez/pub/publi/gdc99.pdf>), se conoce un procedimiento de administración de memoria con las características del preámbulo de la reivindicación 1. En el procedimiento descrito en dicho documento se crea en una memoria tampón de recuperación, antes de cada operación de escritura atómica, una imagen de cada página de memoria afectada por la operación de escritura. Si es necesario recuperar el contenido de la memoria, las entradas de la memoria tampón de recuperación pasarán por orden con el fin de restaurar el estado original de la memoria.

[0013] Sin embargo, en este documento no se menciona la posibilidad de operaciones de escritura no atómicas durante una transacción en curso. En el sistema descrito en el artículo mencionado existe no obstante también la necesidad de soportar operaciones de escritura no atómicas con la funcionalidad arriba indicada por razones de aumento de eficacia y para la implementación de áreas de memoria críticas para la seguridad, como por ejemplo contadores de intentos fallidos.

[0014] En consecuencia, la invención tiene el objetivo de solucionar total o parcialmente los problemas mencionados del estado actual de la técnica. Mediante la invención se pretende en particular poner a disposición una administración de memoria para una memoria de datos registrable sólo página a página que, además de operaciones de escritura atómicas, soporte también operaciones de escritura no atómicas en transacciones. En formas de realización preferidas se persigue el objetivo de, mediante la invención, llevar a cabo el comportamiento predefinido según la especificación JavaCard™ también en tarjetas chip que presenten una memoria de datos prevista con tecnología FLASH.

[0015] Según la invención, este objetivo se logra mediante un procedimiento para administrar una memoria de datos con las características de la reivindicación 1, así como mediante un microcontrolador con las características de la reivindicación 13 y un soporte de datos portátil, en particular una tarjeta chip, con las características de la reivindicación 14. Las reivindicaciones dependientes definen configuraciones preferidas de la invención.

[0016] La invención parte de la idea fundamental de modificar en un proceso de escritura no atómico durante una transacción no sólo el contenido de la memoria de datos, sino en caso dado también el contenido de la memoria tampón de recuperación. Esta idea fundamental es sorprendente ya sólo porque, según el estado actual de la técnica, la memoria tampón de recuperación sirve para salvar el estado original de la memoria y precisamente no debe modificarse. La invención enseña en cambio que, en caso de un acceso de escritura no atómico a una página cuya

imagen se halle ya en la memoria tampón de recuperación, se crea una imagen modificada de la memoria, que resulta de la imagen ya existente mediante una ejecución de la operación de escritura no atómica. De este modo, los efectos de las operaciones de escritura no atómicas concluidas con éxito se conservan también si la transacción se interrumpe a consecuencia de un error o a través de un comando correspondiente y debido a ello se realiza un proceso de recuperación con respecto a las operaciones de escritura atómicas.

5
10
15
20
25
30
35
40
45
50
55

[0017] Mediante la invención es posible realizar la funcionalidad de transacciones con operaciones de escritura atómicas y no atómicas descritas al principio incluso en relación con memorias de datos que permiten únicamente un borrado por páginas y/o una escritura por páginas. Esto abre nuevas posibilidades para el empleo de tales memorias, especialmente en tarjetas chip de alto rendimiento correspondientes por ejemplo a la especificación JavaCard™. La invención permite en particular un tratamiento de operaciones de escritura no atómicas configurado a medida de la administración de datos críticos para la seguridad, como por ejemplo el número de intentos fallidos de introducción de un PIN.

[0018] Según la invención, está previsto crear una imagen de memoria modificada en caso de una operación de escritura no atómica. Con este fin, en distintas formas de realización se modifica directamente una imagen de memoria original ya existente (*before image* [(imagen anterior)]) o bien se crea una copia de la imagen de memoria original ya existente que presenta las modificaciones deseadas. Se prefiere la posibilidad mencionada en segundo lugar, porque de este modo pueden excluirse estados indefinidos de la memoria que, de lo contrario, podrían aparecer posiblemente en caso de un corte repentino de la tensión de alimentación durante la sobrescritura de la imagen de memoria. En algunas formas de realización de la invención, la imagen de memoria original se borra en cuanto se ha creado con éxito en la memoria tampón de recuperación la imagen de memoria modificada. Sin embargo, en ciertas alternativas de realización, tanto el borrado como la reescritura de imágenes de memoria que ya no son necesarias pueden aplazarse hasta que la memoria tampón de recuperación amenace con desbordarse.

[0019] En una forma de realización particularmente sencilla de la invención, la memoria tampón de recuperación contiene en todo momento como máximo una imagen de memoria válida de cada página de memoria. Esto no excluye que, como se acaba de mencionar, la memoria tampón de recuperación contenga imágenes de memoria temporalmente inválidas, es decir imágenes de memoria que se estén creando en ese momento o estén desbloqueadas para su borrado. En configuraciones más complejas están previstas potencialmente varias imágenes de una página de memoria, que se procesan por orden durante el proceso de recuperación.

[0020] Junto con las imágenes de memoria propiamente dichas se crean además respectivamente informaciones de administración correspondientes, que establecen la relación entre las imágenes de memoria salvadas y las imágenes de memoria originales en las que éstas están basadas.

[0021] En el procesamiento de operaciones de escritura no atómicas que afectan a una página de memoria - aún - no escrita de forma atómica dentro de la transacción en curso, se distingue, en formas de realización preferidas, entre las operaciones de escritura realizadas en la totalidad de una página de memoria y las realizadas en una parte de una página de memoria. En el caso mencionado en primer lugar es posible escribir directamente en la memoria de datos, mientras que en el caso mencionado en segundo lugar se realiza preferentemente una protección contra una corrupción del resto de los datos de la página de memoria en caso de un fallo del proceso de escritura.

[0022] La invención pone a disposición preferentemente procedimientos que, al menos en lo que respecta a la administración de transacciones, así como a la escritura y el llenado de zonas de memoria, corresponden a la interfaz para programas de aplicación (*Application Programming Interface - API*) de la especificación JavaCard™. En formas de realización preferidas, la funcionalidad puesta a disposición coincide con la versión 2.1.1 o la versión 2.2 de esta especificación o está perfeccionada según futuras versiones.

[0023] En configuraciones de muy alto rendimiento está prevista una memoria caché para reunir en una sola página de memoria los efectos de múltiples operaciones de escritura atómicas y/o libres. Esta memoria caché puede servir además de memoria intermedia para la creación de la imagen de memoria modificada en la memoria tampón de recuperación. En relación con este proceso, preferentemente se escribe de nuevo en la página de memoria respectivamente correspondiente la entrada existente hasta el momento en la memoria caché.

[0024] La invención es adecuada preferentemente para soportes de datos portátiles y se ejecuta en particular mediante una tarjeta chip. Sin embargo, en el presente texto deben entenderse en general por el concepto "tarjeta chip", además de las formas constructivas usuales con el tamaño de una tarjeta de cheques o en forma de pequeños módulos de tarjeta - por ejemplo SIM (*subscriber identity modules* [módulos de identidad de abonado]) en teléfonos móviles -, también tarjetas chip con otras formas constructivas, por ejemplo en forma de llaveros o anillos. La tarjeta chip presenta preferentemente un microcontrolador monochip que contiene varias unidades funcionales delimitables mutuamente.

[0025] En configuraciones preferidas del microcontrolador según la invención y de la tarjeta chip según la invención, éstos presentan características que corresponden a las características anteriormente descritas o a las definidas en las reivindicaciones de procedimiento dependientes.

[0026] El orden de enumeración de los pasos en las reivindicaciones no debe considerarse como restrictivo. Más bien, están previstas formas de realización de la invención en las que estos pasos se realizan en un orden distinto, o total o parcialmente de forma paralela, o total o parcialmente de forma cuasi paralela o entrelazados entre sí.

[0027] De la descripción siguiente de varios ejemplos de realización y varias alternativas de realización se desprenden otras características, misiones y ventajas de la invención. Los dibujos esquemáticos muestran:

- en la figura 1 una representación de las capas funcionales de una tarjeta chip organizadas unas sobre otras, en un ejemplo de realización de la invención,

- en la figura 2 una representación esquemática del contenido de la memoria de datos y de la memoria tampón de recuperación durante la realización de una operación de escritura atómica,

- en la figura 3 una representación esquemática del contenido de la memoria de datos y de la memoria tampón de recuperación en una operación de escritura no atómica durante una transacción en curso,

- en la figura 4 una representación esquemática del desarrollo de dos operaciones de escritura atómicas, en una alternativa de realización en la que se emplea una memoria caché para aumentar el rendimiento, y

- en la figura 5 una representación esquemática del desarrollo de una operación de escritura atómica y una operación de escritura no atómica, en la alternativa de realización de la figura 4.

[0028] En la figura 1 se muestra una tarjeta chip según el presente ejemplo de realización de la invención en sus capas funcionales organizadas unas sobre otras, en la medida en que se ve afectada la funcionalidad de la tarjeta chip relevante para la invención.

[0029] Una capa de hardware 10 presenta fundamentalmente un microcontrolador monochip que, como módulos funcionales delimitables mutuamente, contiene una unidad de proceso, una memoria y varios módulos auxiliares, como por ejemplo módulos de control, generadores de reloj o similares. En el aspecto físico, la memoria está formada por varias zonas de memoria con distintas tecnologías de almacenamiento, por ejemplo como RAM volátil, como ROM programada por máscara, como memoria FLASH o similar. En el aspecto lógico, la memoria pone a disposición una serie de áreas de memoria que están repartidas por las zonas de memoria físicamente predefinidas. Así, por ejemplo, la memoria de programa está dispuesta en parte en la ROM programada por máscara y en parte en la memoria FLASH. Un área que sirve de memoria de datos 12 y otra que sirve de memoria tampón de recuperación 14 (*commit buffer*) se hallan en la memoria FLASH. Una memoria caché está dispuesta en la RAM y la RAM proporciona también una memoria temporal de trabajo.

[0030] Sobre la capa de hardware 10 está dispuesta una capa de software cercana al hardware 16, que pone a disposición funciones básicas para el control del acceso a la memoria. Además de funciones para leer y escribir palabras de memoria y páginas de memoria, se trata en particular de las funciones "leer_página_antigua", "escribir_página_antigua" y "desechar_imagen de memoria_anterior". La función "leer_página_antigua" (*readOldPage*) restituye la imagen almacenada en la memoria tampón de recuperación 14 de una página ya modificada durante una transacción. Mediante la función "escribir_página_antigua" (*writeOldPage*) se crea de un modo seguro una imagen de memoria modificada, a saber de manera que, incluso en caso de un fallo funcional repentino, no aparezcan valores de datos corrompidos en la antigua imagen de memoria ni en la imagen de memoria modificada. La función "desechar_imagen de memoria_anterior" (*discardBeforeImage*) borra finalmente la imagen de memoria de una página de la memoria tampón de recuperación 14. Con ello se modifica convenientemente sólo la información de administración correspondiente a la página, de modo que la página ya no aparece como página recuperable.

[0031] Como capa siguiente está dispuesta sobre la capa cercana al hardware 16 un administrador de memoria 18. Utilizando las funciones puestas a disposición por la capa cercana al hardware 16, el administrador de memoria 18 realiza los procesos de administración según la invención para implementar los métodos puestos a disposición por una interfaz para programas de aplicación 20.

[0032] En el presente ejemplo de realización, la interfaz para programas de aplicación 20 (*application program interface-API*) pone a disposición métodos según el documento ya citado al principio "*JavaCard™ 2.1.1 Application Programming Interface*". Por lo que se refiere a una descripción de la funcionalidad exacta de estos métodos se incluye en la presente descripción el documento mencionado. Los métodos puestos a disposición por la interfaz para programas de aplicación 20 son en el presente ejemplo de realización los métodos *beginTransaction* (iniciar_transacción), *commitTransaction* (confirmar_transacción), *abortTransaction* (cancelar_transacción), *arrayCopy* (copiar_zona_atómicamente), *arrayCopyNonAtomic* (copiar_zona_no atómicamente) y *arrayFillNonAtomic* (llenar_zona_no atómicamente).

[0033] En resumen, todas las operaciones de escritura atómicas que se presentan dentro de una transacción se consideran como una unidad inseparable según la especificación *JavaCard™*. De este modo se asegura que, una vez terminada una transacción, el estado de la memoria refleje los efectos de todas las operaciones de escritura atómicas o bien no refleje el efecto de ninguna de estas operaciones de escritura.

[0034] Una conclusión con éxito de la transacción consiste en la ejecución del método *commitTransaction* (confirmar_transacción). La transacción falla si se llama el método *abortTransaction* (cancelar_transacción) o bien si durante la transacción se corta la alimentación de corriente u otra conexión con la tarjeta chip (así llamado, suceso *tear*). En ambos casos se restaura el estado de la memoria mediante un proceso de recuperación (*rollback*). En caso de una interrupción de la transacción disparada por software, este proceso de recuperación se realiza al instante, mientras que en el caso de un suceso *tear* disparado, por ejemplo, por una falta de corriente, la interrupción de la transacción se detecta en el siguiente arranque de la tarjeta chip y en consecuencia el proceso de recuperación no puede realizarse hasta ese momento.

[0035] Los métodos no atómicos *arrayCopyNonAtomic* (copiar_zona_no atómicamente) y *arrayFillNonAtomic* (llenar_zona_no atómicamente) no están sujetos a esta protección de transacción. Si se interrumpen éstos durante la ejecución del programa, el área de memoria afectada puede presentar un estado indefinido. Sin embargo, por otra parte, el proceso de recuperación subsiguiente a una interrupción de la transacción o a un suceso *tear* no debe anular el resultado de un método no atómico concluido con éxito durante una transacción en curso.

[0036] La especificación JavaCard™ establece además que los procesos de escritura y copia deben por regla general realizarse de forma atómica también fuera de una transacción. Por lo tanto, tales procesos se desarrollan igualmente utilizando la protección de transacción; pero la problemática en que se basa la invención no se presenta en este caso, porque estas “mini-transacciones” no pueden contener ninguna operación de escritura no atómica.

[0037] En una variante de realización, la interfaz para programas de aplicación 20 está configurada según el documento “*JavaCard™ 2.2 Application Programming Interface*”. En la elección de palabras aquí utilizada, los métodos *arrayCopyNonAtomic* (copiar_zona_no atómicamente) y *arrayFillNonAtomic* (llenar_zona_no atómicamente) no realizan entonces operaciones de escritura no atómicas, sino operaciones de escritura libres. En cambio, el método *javacard.framework.pin.check* provoca una operación de escritura protegida no atómica en un contador de intentos fallidos. Las técnicas explicadas a continuación se emplean para estas operaciones de escritura y otras similares.

[0038] La interfaz para programas de aplicación 20 es utilizada por uno o varios programas de aplicación 22, para realizar las funciones que haya de poner a disposición la tarjeta chip. Según el ejemplo de realización que se está describiendo, la tarjeta chip resulta especialmente adecuada para aplicaciones en las que se cargan en la misma, volúmenes de datos relativamente grandes. Éstos pueden ser, por ejemplo, datos biométricos, por ejemplo una huella dactilar con un tamaño típico de 2 koctetos, o datos de carácter sanitario, por ejemplo recetas en una tarjeta de caja de enfermedad prevista para varias personas de una familia.

[0039] En las figuras 2 y 3 está representado un detalle de la memoria de datos 12 en distintos estados. La memoria de datos 12 está realizada con tecnología FLASH. Presenta un gran número de palabras de memoria, que en el presente ejemplo de realización tienen respectivamente una longitud de 8 bits, es decir de 1 octeto; en las figuras 2 y 3 únicamente tres palabras de memoria están provistas a modo de ejemplo de las referencias 24, 26 y 28. Las palabras de memoria 24, 26, 28 están agrupadas en páginas de memoria 30 y 32, que en el presente ejemplo de realización abarcan respectivamente dieciséis palabras de memoria, es decir dieciséis octetos.

[0040] Debido a la tecnología FLASH utilizada, la memoria de datos 12 no puede modificarse octeto a octeto. En principio, es más bien necesario, para escribir una palabra de memoria, por ejemplo la palabra de memoria 24, leer y almacenar temporalmente en primer lugar toda la página de memoria en la que se halla la palabra de memoria - aquí la página de memoria 30 -, a continuación modificar la palabra de memoria 24 en los datos almacenados temporalmente, borrar por completo la página de memoria 30 y finalmente escribir de nuevo la página de memoria 30 con los datos modificados. Sin embargo, en el procedimiento explicado a continuación con mayor detalle, este principio se modifica repetidas veces en atención a la seguridad y la eficacia.

[0041] La figura 2 muestra, por medio de un ejemplo, los pasos de procedimiento esenciales en la realización de una operación de escritura atómica dentro de una transacción. El objetivo es escribir los valores de octeto “1” y “2”, aquí representados como caracteres ASCII, de forma atómica en las palabras de memoria 24 y 26.

[0042] En un primer paso 34 se crea con este fin en primer lugar en la memoria tampón de recuperación 14 una imagen de memoria 36 (*before image [imagen anterior]*) de la página de memoria 30 en la que se hallan las palabras de memoria a escribir 24, 26. Además de la imagen de memoria 36 propiamente dicha, también se escriben en la memoria tampón de recuperación 14 las informaciones de administración necesarias, en el ejemplo presente la dirección original de la imagen de memoria 36 en la memoria de datos 12 y la longitud de la imagen de memoria 36. Dado que la memoria tampón de recuperación 14 está configurada igualmente como memoria FLASH, este proceso de escritura también se desarrolla en varias etapas. Sin embargo, está previsto escribir la imagen de memoria 36 por completo en una sola página de la memoria tampón de recuperación 14, de modo que aquí puede prescindirse de una lectura previa y un almacenamiento temporal de esta página.

[0043] Una vez creada con éxito la entrada en la memoria tampón de recuperación 14, puede borrarse en el paso 38 la página de memoria afectada por la operación de escritura. El resultado de este proceso de borrado se muestra en la figura 2 en un ejemplo de realización con una codificación de memoria insertada. El borrado físico de las dieciséis palabras de memoria en la página de memoria 30 hace que todos los bits de estas palabras de memoria presenten un

valor homogéneo, es decir bien un “0” lógico o bien un “1” lógico. Si estas palabras de memoria se leyesen y “descodificasen” tras el borrado, se obtendrían datos en apariencia aleatorios, como se muestra en la figura 2 a modo de ejemplo. Dado que además se presentan sumas de comprobación erróneas, esta situación sería detectada por la tarjeta chip y se dispararía un proceso de recuperación.

5 **[0044]** Tras el borrado se escribe de nuevo la página de memoria 30 en el paso 40, con lo que las palabras de memoria 24 y 26 reciben los valores “1” ó “2” de acuerdo con la operación de escritura realizada.

10 **[0045]** Si ahora se confirmase la transacción, por ejemplo ejecutando el método *commitTransaction* (confirmar_transacción), únicamente tendría que borrarse el contenido de la memoria tampón de recuperación 14. Por el contrario, en el caso de una supuesta interrupción de la transacción o en el caso de un suceso *tear* durante los pasos 38 y 40 se realizaría un proceso de recuperación, en el que la imagen de memoria 36 se reescribiría de nuevo en la página de memoria 30 - o sea en la dirección indicada en la memoria tampón de recuperación - para restaurar el estado original de la memoria de datos 12.

15 **[0046]** En la figura 3 está representada una continuación de la transacción, comenzada en la figura 2, con una operación de escritura no atómica. En esta operación, el objetivo es escribir los valores de octeto “5”, “6”, “7”, “8” en la página de memoria 30 en lugar de los valores “K”, “L”, “M”, “N” existentes en la misma. Dado que la página de memoria 30 que ahora ha de escribirse ya ha sido objeto de una operación de escritura atómica durante la transacción en curso - y en consecuencia existe ya como imagen de memoria 36 de esta página de memoria 30 en la memoria tampón de recuperación 14 -, en primer lugar se crea en la memoria tampón de recuperación 14, en el paso 42, una imagen de memoria 44 modificada según la operación de escritura no atómica. Aquí hay que tener en cuenta que la imagen de memoria 36 ya existente no debe borrarse ni sobrescribirse sin salvarla, porque en caso contrario un posible suceso *tear* durante el proceso de escritura podría llevar a una pérdida de la imagen de memoria 36 antes de haberse creado la imagen de memoria modificada 44. En tal caso sería imposible hacer retroceder la transacción en curso.

20 **[0047]** Por lo tanto, la imagen de memoria modificada 44 se crea en primer lugar adicionalmente en la memoria tampón de recuperación 14. Esta imagen de memoria modificada 44 contiene las informaciones de la imagen de memoria original 36, habiéndose modificado no obstante, según la operación de escritura no atómica que ahora se ha de realizar, las palabras de memoria escritas por esta operación. Una vez que se han creado con éxito la imagen de memoria modificada 44 y las informaciones de administración asociadas a la misma, puede borrarse la imagen de memoria original 36 en un paso subsiguiente que no se muestra en la figura 3. Si se produce un suceso *tear* mientras hay varias imágenes de memoria en la memoria tampón de recuperación 14, en el presente ejemplo de realización se evalúa durante el proceso de recuperación subsiguiente sólo la última imagen de memoria, es decir la más reciente, mientras que las imágenes de memoria anteriores se desechan.

25 **[0048]** Por último, en el paso 46, se modifica en la memoria de datos 12 la página de memoria 30 afectada por la operación de escritura no atómica. Éste es de nuevo un proceso de varias etapas, como se ha descrito ya anteriormente en relación con la figura 2. De este modo, la operación de escritura no atómica concluye con éxito. Dado que tanto la palabra de memoria 30 de la memoria de datos 12 afectada por la operación de escritura no atómica como la imagen de memoria 44 correspondiente de la memoria tampón de recuperación 14 han sido modificadas según la operación de escritura no atómica, los efectos de la operación de escritura no atómica se conservan tanto en caso de concluir con éxito la transacción como en caso de una interrupción de esta última disparada por software o por un suceso *tear*.

30 **[0049]** Explicamos ahora además el caso de que - en una continuación de la transacción de las figuras 2 y 3 - se realice una operación de escritura no atómica en una página de memoria, por ejemplo la página de memoria 32, a la que hasta el momento no ha accedido ninguna operación de escritura atómica y de la que, por este motivo, aún no existe ninguna imagen en la memoria tampón de recuperación 14.

35 **[0050]** Si esta operación de escritura no atómica afecta a toda la página de memoria 32, el proceso del borrado con la subsiguiente escritura puede realizarse sin medidas de protección y sin modificar la memoria tampón de recuperación 14. En caso de un posible suceso *tear* durante este proceso, el contenido de la página de memoria 32 estaría indefinido, lo cual es admisible para las operaciones de escritura no atómicas. Si la operación de escritura concluye con éxito, el contenido modificado se halla en la memoria de datos 12 y no se ve influido por un posible futuro proceso de recuperación de la transacción.

40 **[0051]** Por el contrario, si la operación de escritura no atómica afecta sólo a determinadas palabras de memoria de la página de memoria 32, por ejemplo sólo a la palabra de memoria 28, es necesaria una escritura protegida de la página de memoria 32, para no corromper el resto del contenido de la página de memoria 32 en caso de un suceso *tear* durante el proceso de escritura. Con este fin, de forma similar a como se hace en la sección de procedimiento representada en la figura 2, se crea en primer lugar en la memoria tampón de recuperación 14 una imagen de la página de memoria 32 original. A continuación se borra y se escribe de nuevo la página de memoria 32. Si se produce un error durante este proceso, es posible restaurar el estado original de la página de memoria 32, especialmente por lo que se refiere a las palabras de memoria no afectadas por la operación de escritura no atómica, por medio de la imagen existente en la memoria tampón de recuperación 14. En cuanto ha concluido con éxito el proceso de escritura de la página de memoria 32, se desecha la imagen existente en la memoria tampón de recuperación 14 y con ello se descongiona esta última.

[0052] Mediante este procedimiento se logra en suma que en caso de una interrupción de la transacción disparada por software o en caso de un suceso *tear* se conserven los datos escritos con éxito de forma no atómica, mientras que se restauran a su estado original todos los datos escritos hasta el momento de forma atómica dentro de la transacción. Las palabras de memoria que han sido objeto de una operación de escritura no atómica fallida se hallan en un estado indefinido.

[0053] Como ya se ha mencionado, en algunas configuraciones todos los procesos de escritura mediante métodos no atómicos se tratan en la forma recién explicada, mientras que, por ejemplo, en las formas de realización según la versión 2.2 de la especificación JavaCard™ se consideran como operaciones de escritura no atómicas en el sentido arriba indicado sólo las operaciones de escritura en un contador de intentos fallidos o en datos similares del tipo *javacard.framework.OwnerPIN*. Las operaciones de escritura libres, o sea accesos de escritura no atómicos generales en el sentido de la versión 2.2 de la especificación JavaCard™, pueden llevarse a cabo eficazmente, sin modificar una imagen de memoria en caso dado contenida en la memoria tampón de recuperación 14. Sin embargo, si una operación de escritura libre no afecta a toda una página de memoria, es necesario crear al menos una imagen de memoria temporal del estado original de la página de memoria (*before image [imagen anterior]*), para que, en caso de un suceso *tear* o una interrupción de otro tipo, se conserve el resto de la página de memoria no afectado por la operación de escritura libre.

[0054] En una configuración optimizada del ejemplo de realización que se está describiendo, está prevista una memoria caché - véase la capa de hardware 10 en la figura 1 - que permite reducir el número de procesos de escritura necesarios. En esta variante de realización, en la medida en que el tamaño de la memoria caché sea suficiente, se da preferencia a las operaciones de escritura no atómicas frente a las operaciones de escritura atómicas. En este caso, la sobrescritura se realiza en la memoria caché y no en la memoria tampón de recuperación 14 realizada con tecnología FLASH.

[0055] Las figuras 4 y 5 ilustran, a modo de ejemplo, desarrollos en los que se emplea la memoria caché, que se halla en la RAM. Para hacer más clara la representación, el tamaño de una página de memoria - y correspondientemente también el tamaño de cada imagen de memoria en la memoria tampón de recuperación y de cada entrada en la memoria caché - se muestra aquí con ocho palabras de memoria respectivamente, es decir ocho octetos. Cada una de las filas de las figuras 4 y 5 representa un estado de la página de memoria 30, de una entrada 50 en la memoria caché y de una imagen 36 en la memoria tampón de recuperación 14. El orden de las filas de arriba abajo corresponde al orden cronológico de los distintos pasos del procedimiento. Las líneas punteadas entre dos filas indican que el contenido de la página de memoria correspondiente no varía. Las flechas indican la dirección del flujo de información, señalando la punta de la flecha respectivamente la página de memoria modificada.

[0056] El desarrollo a modo de ejemplo según la figura 4 representa la utilización de la memoria caché en una transacción con varios procesos de escritura atómicos. Partiendo de un estado en el que la página de memoria 30 contiene los valores de octeto, aquí representados de nuevo como caracteres ASCII, "A", "B", "C", "D", "E", "F", "G" y "H", se realiza en primer lugar una llamada de método 52, es decir una llamada del método *beginTransaction* (iniciar_transacción) de la interfaz para programas de aplicación 20.

[0057] En la llamada de método 54 siguiente, el objetivo es escribir los octetos "1", "2" en las palabras de memoria que hasta el momento contienen "B" y "C", para lo cual puede utilizarse por ejemplo el método *arrayCopy* (copiar_zona_atómicamente) de la interfaz para programas de aplicación 20. Dado que la memoria caché aún no presenta ninguna entrada para la página de memoria 30, se crea en primer lugar una entrada de caché 50 de este tipo. El contenido de la página de memoria 30 se copia en esta nueva entrada de caché 50 creada. En el paso siguiente, con el fin de proteger la transacción, la entrada de caché 50 se adopta como imagen de memoria 36 en la memoria tampón de recuperación (referencia 14 en las figuras 1 a 3). Por último se modifica la entrada de caché 50 según la instrucción de escritura atómica de la llamada de método 54. Dado que la memoria caché está creada en la RAM, es posible sin problema alguno sobrescribir octeto a octeto palabras de memoria de la entrada de caché 50.

[0058] En una operación de escritura atómica subsiguiente 56 se han de sobrescribir con los valores "5" y "6" las palabras de memoria de la página de memoria 30 con el contenido original "F" y "G". Dado que en la memoria caché ya hay una entrada de caché 50 asignada a la página de memoria 30, únicamente es necesario modificar correspondientemente esta entrada de caché 50 en la RAM. No se modifican ni la página de memoria 30 ni la imagen de memoria 36, situadas ambas en la memoria FLASH. Así pues, la operación de escritura 56 se realiza de forma sumamente rápida y ahorrando recursos.

[0059] Si la transacción concluye con éxito con la llamada 58 del método *commitTransaction* (confirmar_transacción), todo el contenido activo de la memoria caché se escribe de nuevo en las páginas de memoria correspondientes de la memoria FLASH. En el ejemplo de la figura 4 se copia por lo tanto la entrada de caché 50 en la página de memoria 30. En este caso no se necesita la imagen de memoria 36.

[0060] En caso de interrumpirse la transacción a través de una llamada del método *abortTransaction* (cancelar_transacción) o debido a un suceso *tear*, se realiza el proceso de recuperación ya descrito en relación con la figura 2. En este proceso, la página de memoria 30 se sobrescribe con el contenido de la imagen de memoria 36 presente en la memoria tampón de recuperación. El contenido de la memoria caché incluida en la RAM volátil se

desecha; en caso de un suceso *tear*, este contenido de la memoria se hubiera perdido de todos modos a consecuencia de la falta de tensión de alimentación.

[0061] La ejecución de operaciones de escritura libres en las variantes de realización según la versión 2.2 de la especificación JavaCard™ puede efectuarse con una gran eficacia en el contexto que se acaba de describir. Si la página de memoria afectada por una operación de escritura libre está ya presente en la memoria caché, porque ya se haya realizado por ejemplo un acceso de escritura atómico a esta página durante la transacción en curso, sólo se modifica según la operación de escritura libre la entrada de la memoria caché. Este caso se da por ejemplo si, en una variación del desarrollo de la figura 4, la operación de escritura 56 ha de realizarse no de forma atómica, sino como operación de escritura libre. El resultado se muestra entonces como en la figura 4. En este caso, el efecto de la operación de escritura libre se conserva si se confirma la transacción y se pierde si se interrumpe la transacción. Este comportamiento es admisible según la versión 2.2 de la especificación JavaCard™.

[0062] Si, por el contrario, la operación de escritura libre afecta a una página de memoria que aún no se halla en la memoria caché, en primer lugar se escribe la página de memoria en la memoria caché y en la memoria tampón de recuperación 14. A continuación se modifica la entrada de la memoria caché según la operación de escritura libre. El estado así alcanzado corresponde al existente en la figura 4 después de realizar la operación de escritura 54. Sin embargo, en el caso que nos ocupa la entrada de la memoria caché se dota de una marca especial. Esta marca se evalúa al escribirse la entrada de caché de nuevo en la página original de la memoria de datos 12 y hace que, una vez terminado este proceso, se borre la imagen correspondiente existente en la memoria tampón de recuperación 14. Este tipo de liberación de memoria deseable en la memoria tampón de recuperación 14 se realiza también en otros sucesos predeterminados, por ejemplo el inicio de una transacción.

[0063] La representación de la figura 5 muestra otro desarrollo a modo de ejemplo que, como el desarrollo de la figura 4, comienza con la llamada de método 60 *beginTransaction* (iniciar_transacción), seguida de una operación 62 para la escritura atómica de los valores "1" y "2". Sin embargo, en la operación de escritura 64 subsiguiente, los valores "5" y "6" no se escriben de forma atómica como en el desarrollo de la figura 4, sino de forma no atómica. La memoria caché sirve aquí de memoria auxiliar en la modificación de la imagen de memoria 36 en la memoria tampón de recuperación, escribiéndose a su vez de nuevo el contenido de la memoria caché en la página de memoria 30 original.

[0064] El procesamiento de la operación de escritura no atómica 64 comienza con la protección del contenido de la entrada de caché 50. En el presente ejemplo de realización se escribe para ello el contenido de la entrada de caché 50 en la página de memoria 30 asignada, mientras que en alternativas de realización pueden estar previstas otras memorias intermedias. A continuación, la imagen de memoria 36 existente en la memoria tampón de recuperación se carga en la entrada de caché 50 y se modifica en la misma de acuerdo con la operación de escritura no atómica 64. Esta entrada de caché 50 modificada se escribe como imagen de memoria modificada 44 en la memoria tampón de recuperación 14 y sustituye en la misma a la imagen de memoria original 36. Por último se restaura el contenido anterior de la entrada de caché 50 desde la página de memoria 30, en la que se ha almacenado de forma intermedia esta entrada de caché 50. A continuación se modifica la entrada de caché 50 según la operación de escritura no atómica 64, escribiendo los valores "5" y "6" en lugar de los valores originales "F" y "G".

[0065] En caso de una confirmación de la transacción en el paso 66 se escriben en las páginas de memoria correspondientes de la memoria FLASH - como en el desarrollo de la figura 4 - todas las entradas de la memoria caché. Las entradas de la memoria caché escritas de nuevo en la memoria FLASH en respuesta a una confirmación de la transacción se borran de la memoria caché. Un proceso de recuperación eventualmente necesario se realiza también análogamente a la representación de la figura 4, cargando la imagen de memoria modificada 44 en la página de memoria 30.

[0066] Se entiende que, en configuraciones preferidas, la memoria caché no está limitada a una sola entrada de caché 50, sino que más bien presenta espacio para varias entradas, correspondientes respectivamente a una página de memoria. Si la memoria amenaza con desbordarse, es posible por ejemplo escribir de nuevo en la página de memoria asignada la entrada respectivamente más antigua presente en la memoria caché. Mediante esta utilización de la memoria caché se aumenta considerablemente la eficacia del procedimiento, especialmente en el caso de transacciones extensas, en relación con el ejemplo de realización más sencillo según las figuras 2 y 3.

REIVINDICACIONES

1. Procedimiento para administrar una memoria de datos (12) de un soporte de datos portátil, en el que
 - la memoria de datos (12) está prevista con tecnología Flash o con una tecnología con características similares,
 - la memoria de datos (12) presenta un gran número de palabras de memoria (24, 26, 28), que están agrupadas en páginas de memoria (30, 32),
 - la memoria de datos (12) necesita, para la escritura de una palabra de memoria (24, 26, 28), en primer lugar leer y almacenar temporalmente toda la página de memoria en la que se halla la palabra de memoria, a continuación modificar la palabra de memoria en los datos almacenados temporalmente, borrar por completo la página de memoria y finalmente escribir de nuevo la página de memoria con los datos modificados,
 - el procedimiento soporta transacciones con, respectivamente, al menos una operación de escritura atómica,
 - al realizar una transacción se almacena en una memoria tampón de recuperación (14) respectivamente al menos una imagen de memoria (36) de cada página de memoria (30, 32) afectada por una operación de escritura atómica, estando la memoria tampón de recuperación (14) configurada también con tecnología Flash o con una tecnología con características similares y mostrando la imagen de memoria (36) el contenido de la página de memoria (30, 32) sin el efecto de la operación de escritura atómica, para, en caso de una interrupción de la transacción, permitir una recuperación del contenido de la memoria por lo que se refiere a la al menos una operación de escritura atómica de la transacción, **caracterizado porque**
 - el procedimiento soporta además operaciones de escritura atómicas y no atómicas dentro de una transacción,
 - creándose en la memoria tampón de recuperación (14) en el caso de una operación de escritura no atómica en una transacción ya comenzada, al menos cuando exista en la memoria tampón de recuperación (14) una imagen de memoria (36) de una página de memoria (30, 32) afectada por la operación de escritura no atómica dentro de la misma transacción, a partir de la imagen de memoria (36) existente, una imagen de memoria (44) modificada de acuerdo con la operación de escritura no atómica y que contiene las informaciones de la imagen de memoria original (36), habiendo sido modificadas no obstante las palabras de memoria por y según la operación de escritura no atómica a realizar dentro de la transacción y, en caso de un proceso de recuperación, escribiéndose de nuevo en la página de memoria (30, 32) la imagen de memoria modificada (44), de modo que, en caso de una recuperación del contenido de la memoria, se conservan los efectos en el contenido de la memoria de operaciones de escritura no atómicas concluidas con éxito dentro de la transacción.
2. Procedimiento según la reivindicación 1, en el que la imagen de memoria (36) existente en la memoria tampón de recuperación (14) se conserva al menos hasta que se ha creado la imagen de memoria modificada (44) y la misma se ha escrito con éxito en la memoria tampón de recuperación (14).
3. Procedimiento según la reivindicación 1 o la reivindicación 2, en el que la memoria tampón de recuperación (14) contiene en todo momento como máximo una imagen de memoria válida (36, 44) de cada página de memoria (30, 32).
4. Procedimiento según una de las reivindicaciones 1 a 3, en el que, en caso de una operación de escritura no atómica que afecte a una parte de una página de memoria (32) de la que no haya ninguna imagen de memoria en la memoria tampón de recuperación (14), en la memoria tampón de recuperación (14) se almacena una imagen de memoria temporal de la página de memoria (32), para, en caso de un fallo de la operación de escritura no atómica, permitir una recuperación de, al menos, la parte de la página de memoria (32) no afectada por la operación de escritura no atómica.
5. Procedimiento según una de las reivindicaciones 1 a 4, en el que, en caso de una operación de escritura no atómica que afecte a toda una página de memoria (32) de la que no haya ninguna imagen de memoria en la memoria tampón de recuperación (14), la página de memoria (32) afectada se sobrescribe sin ser salvada.
6. Procedimiento según una de las reivindicaciones 1 a 5, que pone a disposición métodos según la especificación JavaCard™ versión 2.2 o versión 2.1.1 en relación con la administración de transacciones y en relación con operaciones de escritura atómicas y no atómicas.
7. Procedimiento según una de las reivindicaciones 1 a 6, en el que, para escribir una parte de una página de memoria (30, 32) de la memoria de datos (12), en dicha la memoria de datos (12) se lee toda la página de memoria (30, 32) y se almacena temporalmente esta última, se borra toda la página de memoria (30, 32), se modifica la parte a escribir de los datos almacenados temporalmente y toda la página de memoria (30, 32) se escribe de nuevo con los datos modificados.
8. Procedimiento según una de las reivindicaciones 1 a 7, en el que está prevista una memoria caché para reunir en una sola página de memoria (30, 32) los efectos de múltiples operaciones de escritura atómicas, escribiéndose de nuevo en la página de memoria (30, 32) correspondiente una entrada (50) existente en la memoria caché a más tardar al producirse una confirmación de la transacción.

- 5 9. Procedimiento según una de las reivindicaciones 1 a 7, que además soporta operaciones de escritura libres durante transacciones en curso, que se realizan con una eficacia mayor que las operaciones de escritura no atómicas correspondientes, y en el que, en caso de una recuperación del contenido de la memoria a consecuencia de una interrupción de la transacción, no se conservan necesariamente los efectos en el contenido de la memoria de operaciones de escritura libres concluidas con éxito.
- 10 10. Procedimiento según la reivindicación 9, en el que está prevista una memoria caché para reunir en una sola página de memoria (30, 32) los efectos de múltiples operaciones de escritura atómicas y/o libres, escribiéndose de nuevo en la página de memoria (30, 32) correspondiente una entrada (50) existente en la memoria caché a más tardar al producirse una confirmación de la transacción.
- 10 11. Procedimiento según la reivindicación 8 o la reivindicación 10, en el que la memoria caché sirve además de memoria intermedia para la creación de la imagen de memoria modificada (44) en la memoria tampón de recuperación (14).
- 15 12. Procedimiento según una de las reivindicaciones 1 a 11, que se realiza mediante una unidad de proceso de un microcontrolador de una tarjeta chip, presentando la tarjeta chip además la memoria de datos (12), la memoria tampón de recuperación (14) y una memoria de programa, que contiene instrucciones de programa para el mando del microcontrolador.
- 20 13. Microcontrolador con una unidad de proceso y varias zonas de memoria, en el que las zonas de memoria ponen a disposición una memoria de datos (12), prevista con tecnología Flash o con una tecnología con características similares, una memoria tampón de recuperación (14), configurada también con tecnología Flash o con una tecnología con características similares, y una memoria de programa y en el que la memoria de programa contiene instrucciones de programa que hacen que la unidad de proceso ejecute un procedimiento según una de las reivindicaciones 1 a 12.
14. Soporte de datos portátil, en particular tarjeta chip, con un microcontrolador según la reivindicación 13.

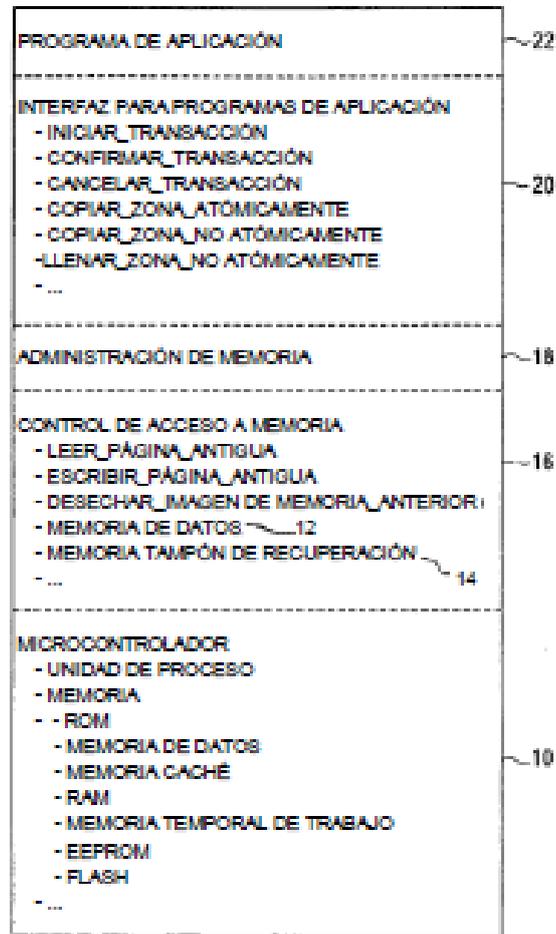


Fig. 1

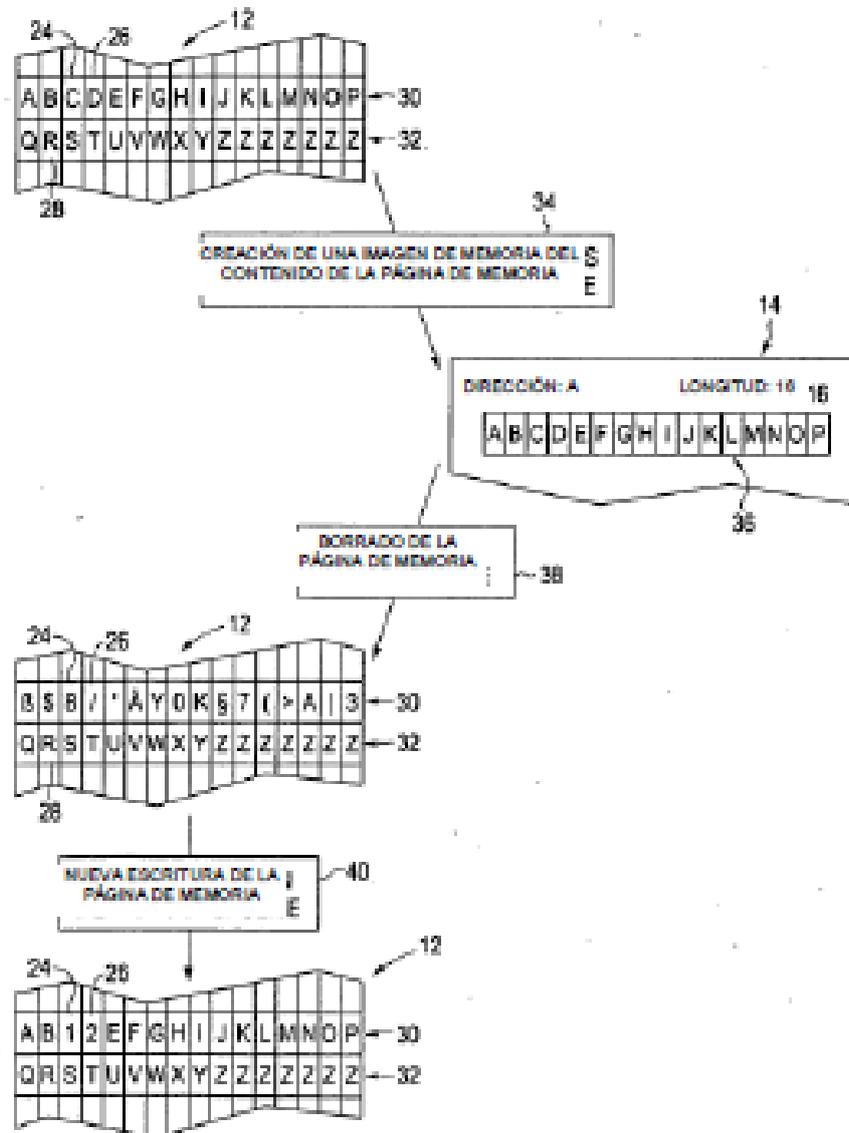


Fig. 2

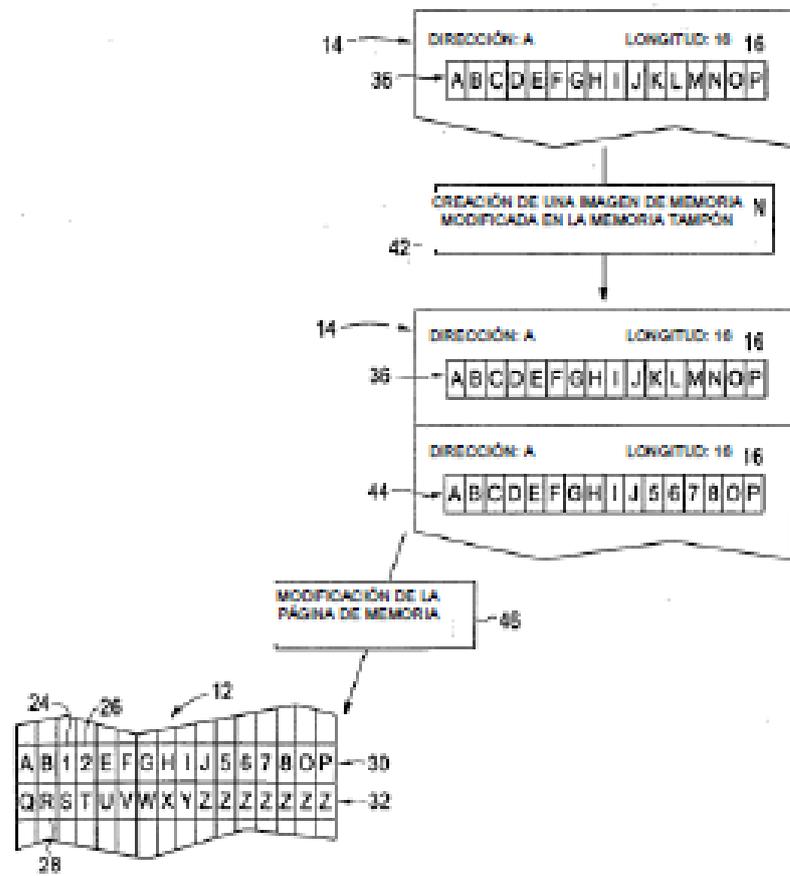


Fig. 3

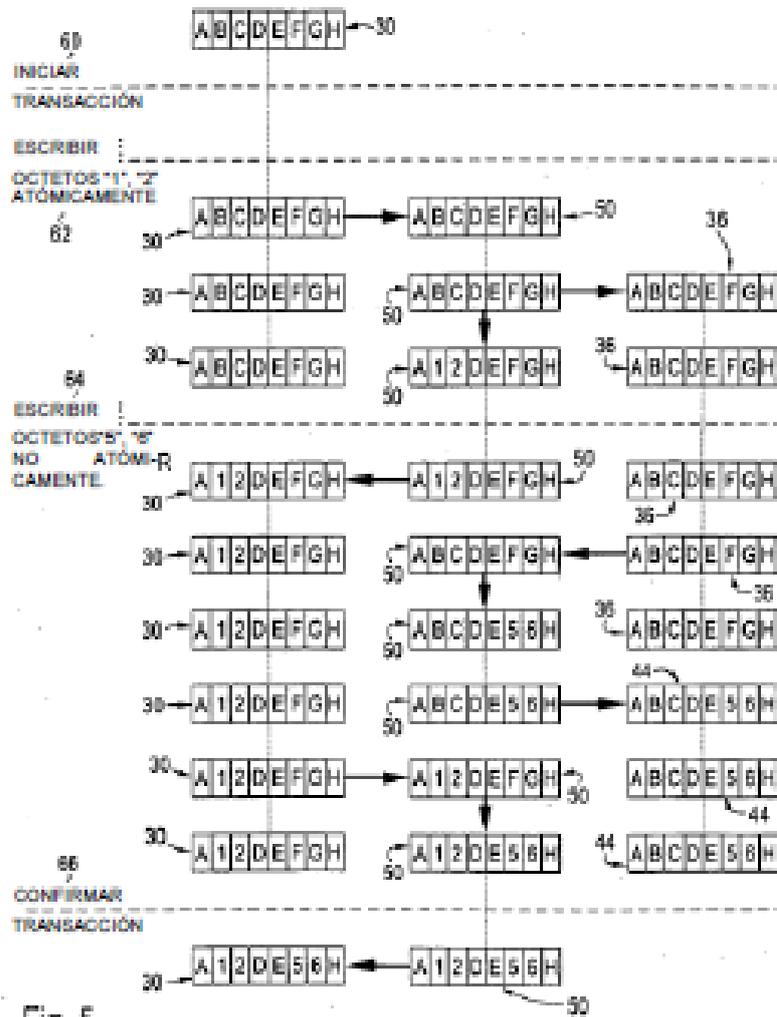


Fig. 5

REFERENCIAS CITADAS EN LA DESCRIPCIÓN

5 La lista de referencias citada por el solicitante lo es solamente para utilidad del lector, no formando parte de los documentos de patente europeos. Aún cuando las referencias han sido cuidadosamente recopiladas, no pueden excluirse errores u omisiones y la OEP rechaza toda responsabilidad a este respecto.

Documentos de patente citados en la descripción