(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 700 554 B1

(12) FASCICULE DE BREVET EUROPEEN

(45) Date de publication et mention de la délivrance du brevet:
08.10.1997 Bulletin 1997/41

(21) Numéro de dépôt: **94917057.5**

(22) Date de dépôt: **25.05.1994**

(51) Int Cl.[6]: **G07F 7/10**

(86) Numéro de dépôt international:
PCT/FR94/00618

(87) Numéro de publication internationale:
WO 94/28521 (08.12.1994 Gazette 1994/27)

(54) PROCEDE D'ECRITURE D'INFORMATIONS DANS UNE MEMOIRE NON-VOLATILE

VERFAHREN ZUM DATENSCHREIBEN IN EINEM NICHT FLÜCHTIGEN SPEICHER

METHOD OF WRITING INFORMATION IN A NON-VOLATILE MEMORY

(84) Etats contractants désignés:
DE ES GB IT

(30) Priorité: **26.05.1993 FR 9306323**

(43) Date de publication de la demande:
13.03.1996 Bulletin 1996/11

(73) Titulaire: **GEMPLUS CARD INTERNATIONAL**
F-13420 Gémenos (FR)

(72) Inventeur: **GORDONS, Edouard**
F-13400 Aubagne (FR)

(74) Mandataire: **Schmit, Christian Norbert Marie**
Cabinet BALLOT-SCHMIT,
16, avenue du Pont Royal
94230 Cachan (FR)

(56) Documents cités:
EP-A- 0 319 799
FR-A- 2 600 444
FR-A- 2 612 316

Il est rappelé que: Dans un délai de neuf mois à compter de la date de publication de la mention de la délivrance du brevet européen, toute personne peut faire opposition au brevet européen délivré, auprès de l'Office européen des brevets. L'opposition doit être formée par écrit et motivée. Elle n'est réputée formée qu'après paiement de la taxe d'opposition. (Art. 99(1) Convention sur le brevet européen).

EP 0 700 554 B1

Printed by Jouve, 75001 PARIS (FR)

Description

L'invention concerne les cartes à mémoire, et plus particulièrement les cartes comportant une mémoire non-volatile dont le contenu est géré (écriture, effacement, mise à jour d'enregistrements) par un microprocesseur.

Un des buts que se propose la présente invention est d'assurer une meilleure intégrité et une meilleure cohérence des données inscrites dans la mémoire non volatile de la carte.

Lorsqu'on développe des applications nouvelles sous forme de programmes mis en mémoire dans la carte et exécutés par le microprocesseur, il est important de faire en sorte que les données conservées dans la carte soient cohérentes et aient une signification incontestable par rapport à l'application. L'exemple le plus simple est celui des cartes servant de moyen de paiement : si une donnée inscrite dans la carte correspond à un montant de débit ou de crédit il est impératif qu'il n'y ait pas d'erreur sur cette donnée.

Les données enregistrées dans la mémoire non volatile de la carte sont disposées dans des fichiers selon des méthodes connues. Le fichier comporte des enregistrements, les enregistrements sont placés à des positions physiques déterminées de la mémoire; une table d'allocation de fichiers détermine les emplacements de mémoire occupés. Les enregistrements peuvent être chaînés, c'est-à-dire que chaque enregistrement comporte deux parties : une donnée et une adresse de l'enregistrement suivant du fichier. A l'intérieur d'un même fichier, les données sont en principe cohérentes entre elles, et en tout cas on peut en vérifier la cohérence en examinant le contenu du fichier.

Mais parfois, on doit en outre assurer une cohérence de données inscrites dans plusieurs fichiers différents. Cette cohérence n'est pas facile à assurer dans tous les cas de fonctionnement défectueux.

Un des risques de fonctionnement défectueux est par exemple l'arrachage de la carte hors du lecteur pendant une opération d'écriture. La donnée risque d'être perdue. De tels problèmes liés à des erreurs pouvant intervenir pendant la transmission d'informations ont été décrits dans les documents FR-A-2 612 316 et EP-A-0 319 799. On a déjà tenté de prévoir des procédures de sécurité (verrouillage de séquences critiques) pour éviter ce risque : on écrit en deux temps, et on ne valide l'écriture que si elle s'est déroulée correctement. Mais si l'arrachage se produit pendant l'écriture, on conserve en mémoire non-volatile le fait qu'une interruption anormale a eu lieu et on reconstitue la donnée perdue lors de la remise sous tension de la carte.

Cependant, cette procédure d'écriture, bien qu'elle soit plus sophistiquée qu'une procédure simple (et donc plus longue), ne permet pas d'assurer la cohérence des données inscrites dans deux fichiers différents. Un exemple simple peut être donné : un fichier de débit et un fichier de crédit sont présents et sont liés par le fait que toute augmentation d'un côté doit correspondre à une diminution égale de l'autre.

On comprend qu'il y a un risque qu'une interruption d'alimentation intempestive puisse détruire la cohérence des données inscrites dans les deux fichiers : par exemple, on a eu le temps d'inscrire correctement le débit d'un côté et pas le crédit de l'autre.

Par ailleurs, pour assurer la cohérence de données qui ne sont pas écrites par une seule opération, il peut être nécessaire de lier plusieurs opérations d'une manière ou d'une autre.

Un but de l'invention est d'améliorer la sécurité de fonctionnement des cartes à puces, du point de vue de l'intégrité et de la cohérence des données écrites dans la mémoire non volatile de ces cartes.

Selon l'invention, on propose un procédé d'écriture, mise à jour, et effacement d'informations dans une carte à mémoire non volatile, qui est caractérisé en ce qu'il comporte :

- l'exécution d'une commande de groupage pour l'allocation d'un espace mémoire dit espace de transaction destiné à stocker des données temporaires de sauvegarde de N opérations successives d'ajout, mise à jour, et/ou effacement, N étant un nombre fini supérieur à 1,
- puis l'exécution de ces opérations successives et le stockage simultané des données de sauvegarde correspondant à chaque opération dans l'espace de transaction,
- puis soit une opération de validation globale des N opérations, soit une opération d'invalidation globale des N opérations, la commande de validation comportant une libération des espaces mémoires occupés auparavant par les enregistrements effacés ou mis à jour durant les N opérations, et la commande d'invalidation comportant principalement l'utilisation des données de sauvegarde pour reconstituer l'état de la mémoire antérieurement aux N opérations.

Ainsi, on va pouvoir s'assurer de la cohérence de données en liant indissolublement l'exécution de certaines opérations d'écriture de sorte que si ces opérations sont erronées on puisse les annuler globalement et non individuellement, alors que si elles sont justes on va les valider globalement et non individuellement.

L'invention propose donc également une carte à mémoire non-volatile à microprocesseur qui comporte :

- des moyens pour que le microprocesseur reçoive et exécute une instruction de commande de groupage pour l'allocation d'un espace mémoire dit espace de transaction destiné à stocker des données temporaires de sauvegarde de N opérations successives d'ajout, mise à jour et/ou effacement d'enregistrement, N étant un nombre fini supérieur à 1,
- des moyens pour exécuter les N opérations succes-

sives et pour stocker simultanément des données de sauvegarde correspondant à chaque opération dans l'espace de transaction,

- des moyens pour valider globalement les N opérations et des moyens pour invalider globalement les N opérations, les moyens de validation permettant une libération des espaces mémoires occupés auparavant par les enregistrements effacés ou mis à jour durant les N opérations, et les moyens d'invalidation permettant principalement l'utilisation des données de sauvegarde pour reconstituer l'état de la mémoire antérieurement aux N opérations, et les moyens de validation et d'invalidation sont activés lorsque l'instruction de commande de groupage a été reçue préalablement à l'exécution des N opérations.

L'invention est particulièrement applicable dans le cas d'enregistrements organisés en fichiers chaînés, chaque enregistrement contenant à la fois une donnée et une adresse d'enregistrement suivant; les informations de chaînage d'enregistrements sont placés dans l'espace des transactions pendant les opérations d'ajout, mise à jour, et effacement de ces enregistrements.

Il est très avantageux que les informations de chaînage placées dans l'espace des transactions soient les anciens chaînages correspondant à la succession d'enregistrements avant ajout ou mise à jour. Dans ce cas, chaque opération d'ajout d'enregistrement comporte de préférence d'une part l'écriture dans le fichier de mémoire non volatile d'une nouvelle donnée avec un nouveau chaînage, et d'autre part l'écriture dans l'espace des transactions d'une information sur l'ajout et sur l'ancien chaînage. L'opération d'effacement d'enregistrement comporte d'une part l'écriture d'un nouveau chaînage dans la mémoire non volatile et d'autre part l'écriture d'une information de suppression et d'une information d'ancien chaînage dans l'espace des transactions. La mise à jour d'un enregistrement comporte l'ajout d'un enregistrement et la libération de l'espace précédemment occupé par 1' enregistrement, avec modification correspondante des chaînages pour que le nouvel enregistrement se substitue au précédent dans la chaîne. La mise à jour comporte alors l'écriture dans l'espace des transactions d'une information sur l'ancien chaînage, ainsi qu'une information de mise à jour.

L'écriture dans l'espace des transactions est de préférence une écriture de type protégée contre les interruptions intempestives d'alimentation, c'est-à-dire qu'elle s'effectue de préférence avec une étape préliminaire de sauvegarde de ce qui doit être écrit, puis la mise en place d'un verrou, l'écriture proprement dite, et l'enlèvement du verrou.

D'autres caractéristiques et avantages de l'invention apparaîtront à la lecture de la description détaillée qui suit et qui est faite en référence aux dessins annexés dans lesquels :

- la figure 1 représente les étapes pouvant être exécutées lors d'un ajout d'enregistrement;
- la figure 2 représente les étapes pouvant être exécutées lors de la suppression d'un enregistrement;
- la figure 3 représente les étapes pouvant être exécutées lors d'une mise à jour d'enregistrement.

L'invention va maintenant être décrite plus précisément à propos d'un exemple simple dans lequel une commande de groupage d'opérations est associée à une commande de validation globale ou invalidation globale des opérations groupées.

L'exemple qui suit représente une succession d'opérations ainsi groupées.

a) écriture dans la mémoire non volatile de données D1 et D2 dans un fichier
b) sélection d'un autre fichier
c) écriture dans la mémoire non volatile de données D3, D4 et D5
d) sélection d'un autre fichier
e) mise à jour de la donnée D2
f) sélection d'un autre fichier
g) effacement de la donnée D1

puis, en principe validation de l'ensemble de ces opérations,
ou au contraire invalidation de l'ensemble, c'est-à-dire annulation de l'effacement de D1, de la mise à jour de D2, de l'écriture de D5, D4, et D3, et enfin de l'écriture de D2 et D1,
ou enfin possibilité d'interruption anormale des opérations ci-dessus, par exemple arrachement de la carte après l'étape e) de mise à jour de D2, avec nécessité de revenir à l'état de la mémoire non volatile qui existait avant l'étape a).

Pour réaliser cette dernière possibilité, il faut d'abord que les mécanismes d'écriture d'une donnée dans la carte comportent des moyens de protection contre une interruption anormale d'alimentation pendant une opération d'écriture. Le principe général d'une telle protection consiste à placer un verrou dans la mémoire non volatile, ce verrou étant positionné dans un état logique déterminé ("état verrouillé") au début de l'exécution d'une opération d'écriture et étant remis à l'état initial ("déverrouillé") à la fin de l'opération; d'autre part, des informations de sauvegarde de la donnée en cours d'écriture sont placées dans la mémoire non volatile avant la mise en place du verrou. Au moment de la remise sous tension on examine systématiquement l'état du verrou; si on constate qu'il est dans l'état verrouillé, c'est qu'il y a eu interruption anormale et les informations de sauvegarde sont utilisées pour terminer la procédure d'écriture. Par exemple, avant verrouillage, on a écrit dans une zone de la mémoire non volatile l'information à écrire et l'adresse à laquelle elle doit être écrite. Ce n'est qu'après la mise en place du verrou qu'on

écrit définitivement l'information.

On supposera donc que les mécanismes d'écriture sont protégés par verrou pour tenir compte de ce que la carte peut être arrachée (ou l'alimentation interrompue pour une autre raison).

Selon l'invention on prévoit que le microprocesseur de la carte, qui exécute les séquences d'ajout, mise à jour, et effacement d'enregistrement de fichier lorsqu'il reçoit de sa mémoire de programme des instructions correspondantes, peut recevoir une instruction de commande de groupage d'opérations d'écriture. En l'absence de cette commande de groupage les instructions ne peuvent pas être invalidées globalement.

On peut prévoir que l'instruction de groupage est paramétrable pour permettre de choisir le nombre N d'opérations successives qui peuvent être validées ou invalidées globalement.

L'instruction est conçue de manière que son lancement impose qu'on lance ultérieurement une commande de validation ou une commande d'invalidation après l'exécution des N opérations.

La commande de groupage peut être une commande lancée avant chaque groupe d'opérations qui doivent être validées ou invalidées globalement; ou alors elle peut être une commande résidente systématique, c'est-à-dire que toute opération d'écriture, effacement ou mise à jour est exécutée uniquement par groupe de N opérations successives.

L'exécution de l'instruction de groupage commence par la désignation d'un espace de mémoire non volatile qu'on appellera espace des transactions TS; dans cet espace des transactions seront sauvegardées des données nécessaires en cas d'invalidation des opérations, ou éventuellement en cas d'interruption anormale de l'alimentation de la carte pendant la succession de N opérations. Ensuite, l'exécution de la commande de groupage modifie les séquences d'écriture, mise à jour, ou effacement par rapport aux séquences exécutées en l'absence de commande de groupage. En d'autre mots, le lancement de la commande de groupage provoque l'utilisation de sous-programmes spécifiques d'écriture, effacement, ou mise à jour pour les opérations d'écriture, mise à jour ou effacement qui sont lancées ultérieurement.

L'espace de mémoire peut être une zone fixe de la mémoire non volatile, mais il est cependant préférable de prévoir que cette zone varie d'un groupe de N opérations au suivant, pour éviter l'utilisation trop fréquente de la même zone de mémoire non volatile. On peut prévoir par exemple que l'espace de transactions est un espace désigné de manière aléatoire ou pseudo-aléatoire parmi les espaces libres de la mémoire; ou alors cet espace varie selon une règle pré-établie.

De préférence, le mécanisme d'écriture de données dans l'espace des transactions est un mécanisme protégé contre les interruptions anormales d'alimentation pendant une opération d'écriture. Le mécanisme peut être un mécanisme à verrou tel que décrit ci-dessus.

On suppose dans la suite que les enregistrements de la mémoire non volatile sont organisés en fichiers chaînés.

De préférence, pour toute opération d'ajout d'un nouvel enregistrement au fichier, qui comporte la désignation d'un nouvel espace mémoire utilisé et la création d'un chaînage, la séquence d'écriture exécutée est la suivante : stockage dans l'espace des transactions du chaînage correspondant à l'état de la mémoire avant l'exécution de la séquence d'ajout d'enregistrement; puis écriture proprement dite du nouvel enregistrement (donnée et chaînage). On stocke aussi dans l'espace des transactions une information sur l'ajout d'un enregistrement à une adresse donnée.

Pour toute opération de mise à jour d'enreqistrement, qui comporte la modification d'un chaînage avec désignation d'un nouvel espace mémoire et la libération d'un ancien espace mémoire, et l'enregistrement d'une donnée et d'un chaînage dans le nouvel espace, la séquence de mise à jour exécutée est la suivante : stockage dans l'espace des transactions de la donnée et du chaînage correspondant à l'état de la mémoire avant mise à jour; stockage d'une information sur un ajout d'enregistrement et d'une information sur une suppression d'enregistrement; puis exécution de la mise à jour (donnée et chaînage).

Pour toute opération d'effacement d'enregistrement, qui comporte la libération d'un emplacement de mémoire et la modification d'un chaînage, la séquence d'effacement exécutée est la suivante : on stocke dans l'espace des transactions l'ancien chaînage et une information sur l'emplacement de l'enregistrement modifié; puis on modifie le chaînage.

Après l'exécution d'une succession de N opérations différentes, trois cas peuvent se présenter :

- on doit valider la succession d'opérations; l'espace des transactions est alors parcouru; on analyse son contenu et on déclare libres les emplacements de mémoire non volatile occupés par les enregistrements qui étaient à effacer ou à mettre à jour lors de la succession de N opérations. Cela consiste à mettre à zéro (si un bit "0" indique un emplacement libre et un bit "1" un emplacement occupé) un ou plusieurs bits correspondants d'une table d'allocation de fichier représentant l'occupation de la mémoire non volatile. Les chaînages et données de nouveaux enregistrement ajoutés lors d'opérations d'ajout ou de mise à jour sont déjà inscrits dans la mémoire.
- ou alors on doit invalider la succession d'opérations, par exemple parce qu'une erreur manuelle a été commise ou pour toute autre raison; on parcourt encore l'espace des transactions, en remontant la chronologie des opérations effectuées; on analyse le contenu de cet espace, et on déclare libres (mise à zéro de bits dans la table d'allocation de fichier) les nouveaux emplacements de mémoire occupés

par les enregistrements inscrits lors des opérations d'ajout et de mise à jour, et on restitue progressivement les anciens chaînages à partir de valeurs conservées pour chaque opération dans l'espace des transactions.

- enfin, la dernière possibilité est le cas où une interruption anormale d'alimentation a eu lieu pendant la succession de N opérations. Tout d'abord la remise sous tension de la carte va permettre la détection d'un verrou à l'état verrouillé; on va rétablir les données qui ont pu être incorrectement enregistrées dans l'espace des transactions; puis on va procéder exactement comme dans le cas d'une invalidation, la succession d'opérations devant être globalement annulée si une interruption anormale a eu lieu.

Quel que soit l'opération exécutée (validation ou invalidation), l'espace des transactions peut alors être libéré.

En résumé, on utilise trois commandes spécifiques; la première notifie la création d'un espace des transaction en même temps qu'elle arme le processus d'indivisibilité de N opérations successives d'ajout, effacement, ou mise à jour; la deuxième permet la validation de la totalité des actions dont l'historique est conservé dans l'espace des transactions; et la troisième permet l'invalidation de toutes les actions dont l'historique est conservé dans l'espace des transactions.

Dans la technique préférentielle choisie, l'espace des transactions stocke les anciens chaînages, mais les nouveaux enregistrements et chaînages sont déjà inscrits dans la mémoire avant la validation. C'est pourquoi il faut interdire toute opération de lecture dans la mémoire tant que la commande de validation n'est pas effectuée, sous peine de risquer de lire des informations non encore valides.

Des techniques supplémentaires de protection contre différents défauts de fonctionnement peuvent être utilisées dans le cadre de l'invention. Par exemple, on peut se prémunir contre les écritures infructueuses dans la mémoire, dues à une dégradation de celle-ci : pour cela on déplace l'enregistrement incriminé lorsqu'il y a écriture infructueuse d'un chaînage.

On remarquera que l'espace de transactions, utilisé pour sauvegarder des données nécessaires à la reconstitution du contexte précédent de la mémoire, est de faible dimension; chaque opération d'écriture nécessite essentiellement la conservation d'une paire adresse/donnée. De plus cette paire peut être écrite et lue par un seul accès à la mémoire non volatile, ce qui limite le temps consommé par les procédures spéciales d'écriture utilisées selon l'invention.

Le choix d'écrire dans l'espace des transactions les anciens chaînages et données enregistrées, après avoir écrit dans la mémoire non volatile les nouvelles données, permet de minimiser le temps global nécessaire à l'écriture et à la validation. Le temps total serait plus grand si l'on stockait dans l'espace des transactions les données finales à écrire plutôt que les données anciennes.

En effet, la validation est en quelque sorte anticipée dans le cas préférentiel choisi ici : la validation s'effectue en deux étapes. Pour chaque commande, la première étape consiste à réaliser les chaînages souhaités et conserver les valeurs précédentes (paire adresse/donnée). La seconde étape, exécutée plus tard, est la validation finale de toutes les opérations; les emplacements mémoire devenus inutilisés sont alors libérés. Dans le cas où le mécanisme de validation n'est pas choisi, les commandes d'écriture de données en mémoire non volatile auraient effectué sensiblement le même nombre d'opérations, c'est-à-dire : établissement des chaînages puis libération des espaces mémoire occupés par des enregistrements à effacer ou mettre à jour. Le temps mis pour écrire dans l'espace des transactions est par conséquent à ajouter au temps d'exécution de la commande (dans le cas d'une validation) pour obtenir le temps total d'exécution par commande.

Exemple illustratif détaillé

Pour mieux faire comprendre les mécanismes mis en jeu dans le procédé selon l'invention, on a représenté aux figures 1, 2 et 3, respectivement des actions d'ajout, effacement et mise à jour d'enregistrement.

Sur chaque figure on a représenté l'état du fichier avant l'opération, après l'opération, après la validation, et après l'invalidation.

Sur les figures on a représenté des enregistrements avec leurs numéros (exemple E10 pour le dixième enregistrement), leur adresse (exemple A10 pour l'adresse du 10ème enregistrement), leur contenu comportant d'une part une valeur de chaînage qui est l'adresse de l'enregistrement suivant et d'autre part une donnée (D10 pour le dixième enregistrement). On a représenté également le contenu de la table d'allocation de fichier (FAT) à l'emplacement correspondant à chaque adresse d'enregistrement; un contenu "1" signifie que l'emplacement est occupé, un contenu "0" signifie au contraire que l'emplacement est libre.

Enfin, on a représenté l'espace des transactions TS avec les informations qu'il va contenir après chaque opération.

figure 1 : ajout d'enregistrement E11 après le dernier enregistrement E10 du fichier.

a) avant ajout, l'état du fichier est le suivant :

Le chaînage inscrit dans E10 à l'adresse A10 est une indication de fin de fichier (EOF). L'emplacement A10 est alloué ("1" dans la table FAT); l'emplacement A11 est libre ("0" dans la table FAT).

b) après l'opération d'ajout : l'adresse A11 est allouée ("1" dans la table); le chaînage est modifié dans E10 (A11 est indiqué comme adresse d'enregistrement suivant); un chaînage (fin de fichier

EOF) et une donnée (D11) sont inscrits à l'adresse A11 pour constituer l'enregistrement ajouté E11. Parallèlement on stocke dans l'espace des transactions les informations suivantes : enregistrement ajouté; ancien chaînage EOF dans l'enregistrement E10 à l'adresse A10.

c) opération de validation : il faut comprendre que la validation n'a lieu qu'après plusieurs opérations (ajouts, effacements, mise à jour), mais pour faire comprendre simplement le fonctionnement de l'invention on considère que la validation est immédiate (et de même si c'est une invalidation). On voit sur la figure 1, partie c) que la validation ne change rien à l'état du fichier ni de la table FAT par rapport à ce qu'ils sont immédiatement après l'opération d'ajout. On vérifie donc qu'on a bien réalisé une sorte de validation anticipée.

d) opération d'invalidation, au lieu d'une opération de validation : l'examen de l'espace des transactions fait apparaître un ajout d'enregistrement; on sait que l'enregistrement ajouté est l'enregistrement A11 qui contient une indication de fin de fichier EOF; le bit correspondant à l'adresse A11 est mis à zéro dans la table FAT, libérant l'emplacement A11; par ailleurs, l'ancien chaînage inscrit dans l'espace des transactions est restauré : l'indication de fin de fichier EOF est remise dans l'enregistrement à l'adresse A10.

L'état initial du fichier est donc restauré par l'action d'invalidation.

figure 2 : effacement d'un enregistrement E3 entre deux enregistrements E2 et E4.

a) avant effacement de E3, l'état du fichier est le suivant : adresses A2, A3, A4 occupées ("1" dans la table FAT pour chacune) respectivement par des enregistrements E2, E3, E4, chaînages inscrits dans les enregistrements : A3/A2 (A3 dans l'enregistrement E2 à l'adresse A2), A4/A3, et A5/A4; données D2, D3, D4 dans les enregistrements E2, E3, E4 respectivement.

b) après l'opération d'effacement (non encore validée) : les chaînages sont modifiés; A4 est indiqué comme adresse d'enregistrement suivant dans l'enregistrement E2. Parallèlement, l'espace des transactions enregistre les informations suivantes : opération de suppression, et ancien chaînage : A3/A2. La table d'allocation FAT n'est pas modifiée à ce stade.

c) validation : l'espace des transactions est parcouru et fait apparaître une opération d'effacement à l'adresse A3. Le bit correspondant à A3 est donc mis à zéro dans la table FAT pour libérer l'emplacement à l'adresse A3. Le fichier est modifié et tient compte de la suppression de l'enregistrement E3.

d) invalidation : l'espace des transactions est examiné et fait apparaître une suppression et un ancien chaînage A3/A2 qui indique que l'enregistrement effacé est A3. Cet ancien chaînage est restauré en replaçant l'adresse A3 dans l'enregistrement E2 situé à l'adresse A2. Aucune action n'est effectuée sur la table d'allocation de fichier qui n'avait pas été modifiée lors de la suppression d'enregistrement. Le fichier se retrouve dans l'état initial.

figure 3 : mise à jour d'enregistrement E7 entre deux enregistrements E6 et E8.

a) avant mise à jour, l'état du fichier est le suivant : enregistrement E6 à l'adresse A6, comprenant un chaînage vers l'adresse A7 et une donnée D6, avec "1" dans la table d'allocation puisque l'adresse est utilisée; enregistrement E7 à l'adresse A7, comportant un chaînage vers une adresse A8 et une donnée D7, et "1" dans la table FAT. Une adresse A12, disponible ("0" dans la table d'allocation), est également représentée.

b) la mise à jour consiste à utiliser l'adresse A12 pour y placer une donnée D'7 en remplacement de la donnée D7 et à modifier les chaînages pour que l'enregistrement E'7 à l'adresse A12 se substitue à l'enregistrement E7 à l'adresse A7 dans la suite des enregistrements chaînés. Après mise à jour, la table d'allocation de fichier comporte des "1" aux emplacements correspondants aux adresses A6, A7 et A12. L'enregistrement à l'adresse A6 comporte maintenant un chaînage vers l'adresse A12 (A12/A6) et non plus A7; l'enregistrement E'7 à l'adresse A12 comporte un chaînage vers l'adresse A8 pour qu'il se substitue complètement à l'enregistrement E7 dans la chaîne. L'espace des transactions a enregistré l'existence d'une mise à jour, avec un ancien chaînage A7/A6 (adresse A7 dans l'enregistrement à l'adresse A6).

c) opération de validation : l'espace des transactions fait apparaître une mise à jour, avec ancien chaînage A7/A6; ceci signifie qu'il faut supprimer l'enregistrement en A7, et cela impose de mettre à "0" le bit corespondant à A7 dans la table d'allocation, libérant ainsi l'espace de mémoire à cette adresse.

d) opération d'invalidation : l'espace des transactions fait apparaître une mise à jour avec ancien chaînage A7/A6; ceci implique qu'il faut examiner le contenu à l'adresse A6, y trouver l'adresse de chaînage qui y est inscrite (A12), et supprimer alors l'enregistrement à l'adresse A12 (E'7). Un bit "0" est donc mis dans la table d'allocation à la position correspondant à l'adresse A12. Parallèlement, l'ancien chaînage est restauré, c'est-à-dire que l'adresse A7 est replacée dans le contenu de l'enregistrement E6 à l'adresse A6. D'autre part, comme on n'a pas modifié le contenu A8+D7 à l'ancienne adresse A7, l'enregistrement E7 est intact. Le bit de la table d'allocation en A7 était par ailleurs resté à "1". On est

bien revenu à l'état initial.

Les exemples ainsi détaillés montrent bien que chaque opération n'utilise qu'un très petit espace de mémoire TS; l'espace des transactions qui peut changer d'emplacement à chaque nouvelle série de N opérations est donc de taille réduite et la durée de séquences est limitée.

On notera qu'on n'a pas représenté sur les figures l'effacement des informations de l'espace des transactions après chaque opération de validation ou invalidation.

Revendications

1. Procédé d'écriture, mise à jour, et effacement d'informations dans une carte à mémoire non volatile, caractérisé en ce qu'il comporte

   - l'exécution d'une commande de groupage pour l'allocation d'un espace mémoire dit espace de transaction destiné à stocker des données temporaires de sauvegarde de N opérations successives d'ajout, mise à jour, et/ou effacement, N étant un nombre fini supérieur à 1,
   - puis l'exécution de ces opérations successives et le stockage simultané des données de sauvegarde correspondant à chaque opération dans l'espace de transaction,
   - puis soit une opération de validation globale des N opérations, soit une opération d'invalidation globale des N opérations, la commande de validation comportant une libération des espaces mémoires occupés auparavant par les enregistrements effacés ou mis à jour durant les N opérations, et la commande d'invalidation comportant principalement l'utilisation des données de sauvegarde pour reconstituer l'état de la mémoire antérieurement aux N opérations.

2. Procédé selon la revendication 1, caractérisé en ce que les enregistrements de la mémoire sont organisés en fichiers chaînés, les anciens chaînages étant stockés dans l'espace des transactions et les mises à jour et ajouts d'enregistrements étant exécutés lors de l'exécution des N opérations.

3. Procédé selon l'une des revèndications 1 à 2, caractérisé en ce que les enregistrements dans la mémoire sont organisés en fichiers chaînés.

4. Procédé selon la revendication 3, caractérisé en ce que chaque opération d'ajout d'enregistrement comporte l'écriture d'une nouvelle donnée et un nouveau chaînage dans la mémoire non volatile et l'écriture d'une information sur l'ancien chaînage dans- l'espace des transactions.

5. Procédé selon l'une des revendications 3 et 4, caractérisé en ce que l'opération d'effacement d'enregistrement comporte l'écriture d'un nouveau chaînage dans la mémoire non volatile et l'écriture de l'ancien chaînage dans l'espace des transactions.

6. Procédé selon l'une des revendications 3, 4, et 5, caractérisé en ce que la mise à jour d'un enregistrement comporte l'ajout d'un enregistrement et la libération de l'espace précédemment occupé par 1' enregistrement, avec modification correspondante des chaînages pour que le nouvel enregistrement se substitue au précédent dans la chaîne, la mise à jour comportant alors une mise en mémoire dans l'espace des transactions de l'ancien chaînage.

7. Procédé selon l'une des revendications 1 à 6, caractérisé en ce que l'écriture dans l'espace des transactions est une écriture de type protégée contre les interruptions intempestives d'alimentation.

8. Carte à mémoire non-volatile à microprocesseur qui comporte

   - des moyens pour que le microprocesseur reçoive et exécute une instruction de commande de groupage pour l'allocation d'un espace mémoire dit espace de transaction destiné à stocker des données temporaires de sauvegarde de N opérations successives d'ajout, mise à jour et/ou effacement d'enregistrement, N étant un nombre fini supérieur à 1,
   - des moyens pour exécuter les N opérations successives et pour stocker simultanément des données de sauvegarde correspondant à chaque opération dans l'espace de transaction,
   - des moyens pour valider globalement les N opérations et des moyens pour invalider globalement les N opérations, les moyens de validation permettant une libération des espaces mémoires occupés auparavant par les enregistrements effacés ou mis à jour durant les N opérations, et les moyens d'invalidation permettant principalement l'utilisation des données de sauvegarde pour reconstituer l'état de la mémoire antérieurement aux N opérations, et les moyens de validation et d'invalidation sont activés lorsque l'instruction de commande de groupage a été reçue préalablement à l'exécution des N opérations.

Patentansprüche

1. Verfahren zum Schreiben, Aktualisieren und Löschen von Informationen in einer Karte mit nichtflüchtigem Speicher, dadurch gekennzeichnet, daß es enthält:

- Ausführen eines Blockbildungsbefehls für die Zuweisung eines Transaktionsraum genannten Speicherraums, der dazu vorgesehen ist, temporäre Sicherungsdaten von N aufeinanderfolgenden Hinzufügungs-, Aktualisierungs- und/oder Löschungsoperationen zu speichern, wobei N eine endliche Zahl größer als 1 ist,
- dann Ausführen dieser aufeinanderfolgenden Operationen und gleichzeitiges Speichern der jeder Operation entsprechenden Sicherungsdaten im Transaktionsraum,
- dann Ausführen entweder einer Operation zur globalen Freigabe der N Operationen oder einer Operation der globalen Sperrung der N Operationen, wobei der Freigabebefehl eine Leerung der Speicherräume enthält, die vorher durch die Datensätze belegt waren, welche während der N Operationen gelöscht oder aktualisiert wurden, und wobei der Sperrbefehl hauptsächlich die Verwendung der Sicherungsdaten umfaßt, um den Zustand des Speichers vor den N Operationen wiederherzustellen.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Datensätze des Speichers in verknüpften Dateien organisiert sind, wobei die früheren Verknüpfungen im Raum der Transaktionen gespeichert sind und die Aktualisierungen und Hinzufügungen von Datensätzen während der Ausführung der N Operationen ausgeführt werden.

3. Verfahren nach einem der Ansprüchen 1 bis 2, dadurch gekennzeichnet, daß die Datensätze im Speicher in verknüpften Dateien organisiert sind.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß jede Operation einer Hinzufügung eines Datensatzes das Schreiben einer neuen Dateneinheit und einer neuen Verknüpfung in den nichtflüchtigen Speicher und das Schreiben einer Information bezüglich der früheren Verknüpfung in den Raum der Transaktionen umfaßt.

5. Verfahren nach einem der Ansprüche 3 und 4, dadurch gekennzeichnet, daß die Operation des Löschens des Datensatzes das Schreiben einer neuen Verknüpfung in den nichtflüchtigen Speicher und das Schreiben der früheren Verknüpfung in den Raum der Transaktionen umfaßt.

6. Verfahren nach einem der Ansprüche 3, 4 und 5, dadurch gekennzeichnet, daß die Aktualisierung eines Datensatzes die Hinzufügung eines Datensatzes und die Freigabe des vorher von dem Datensatz belegten Raums mit entsprechender Modifikation der Verknüpfungen, damit der neue Datensatz in der Verknüpfung den vorhergehenden ersetzt, umfaßt, wobei die Aktualisierung dann die Speicherung der früheren Verknüpfung in den Raum der Transaktionen enthält.

7. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, daß das Schreiben in den Raum der Transaktionen ein Schreiben des Typs ist, der gegen fehlerhafte Unterbrechungen der Stromversorgung geschützt ist.

8. Karte mit nichtflüchtigem Speicher mit Mikroprozessor, die enthält:

- Mittel, damit der Mikroprozessor einen Blockbildungsbefehl empfängt und ausführt, um einen Transaktionsraum genannten Speicherraum zuzuweisen, der dazu vorgesehen ist, temporäre Sicherungsdaten von N aufeinanderfolgenden Datensatzhinzufügungs-, Datensatzaktualisierungs- und/oder Datensatzlöschungsoperationen zu speichern, wobei N eine endliche Zahl größer als 1 ist,
- Mittel zum Ausführen der N aufeinanderfolgenden Operationen und zum gleichzeitigen Speichern von jeder Operation entsprechenden Sicherungsdaten im Transaktionsraum,
- Mittel zum globalen Freigeben der N Operationen und Mittel zum globalen Sperren der N Operationen, wobei die Freigabemittel eine Leerung von Speicherräumen ermöglichen, die vorher durch die Datensätze belegt waren, die während der N Operationen gelöscht oder aktualisiert werden, und wobei die Sperrmittel hauptsächlich die Verwendung der Sicherungsdaten ermöglichen, um den Zustand des Speichers vor den N Operationen wiederherzustellen, wobei die Freigabe- und Sperrmittel aktiviert werden, wenn der Blockbildungsbefehl vor der Ausführung der N Operationen empfangen worden ist.

Claims

1. A method of writing, updating and erasing data in a card with a non-volatile memory, characterised in that it consists in

- running a batching command to allocate a space in the memory which will be a transaction space for storing temporary back-up data for N successive transactions added, updated and/or erased, where N is a finite number greater than 1,
- then running these successive transactions and simultaneously storing the back-up data corresponding to each transaction in the transaction space,

- followed by a process of globally validating the N transactions or a process of globally invalidating the N transactions, the validation command consisting in freeing memory space previously occupied by the records erased or updated during the N transactions and the invalidation command essentially consisting in using the back-up data to restore the memory to its status prior to the N transactions.

2. A method as claimed in claim 1, characterised in that the records in the memory are organised in strings of files, the old strings being stored in the transaction space and the records are updated and added when the N transactions are run.

3. A method as claimed in one of claims 1 to 2, characterised in that the records are organised in strings of files in the memory.

4. A method as claimed in claim 3, characterised in that each transaction whereby a record is added consists in writing a new datum and a new string in the non-volatile memory and writing a datum on the old string in the transaction space.

5. A method as claimed in one of claims 3 and 4, characterised in that the transaction whereby a record is erased consists in writing a new string in the non-volatile memory and writing the old string in the transaction space.

6. A method as claimed in one of claims 3, 4 and 5, characterised in that the process of updating a record consists in adding a record and freeing space previously occupied by the record along with the corresponding amendment to the strings so that the new record takes the place of the old one in the string, whereby the update makes a memory entry in the transaction space of the old string.

7. A method as claimed in one of claims 1 to 6, characterised in that the process of writing in the transaction space is of the type where what is written is protected against untimely interruptions due to loading.

8. A non-volatile memory card with a micro-processor which consists of

- means enabling the micro-processor to receive and run a batching command instruction to allocate a memory space which will be a transaction space for storing the temporary back-up data for N successive transactions whereby a record is added, updated and/or erased, where N is a finite number greater than 1,
- means for running N successive transactions and for simultaneously storing the back-up data corresponding to each transaction in the transaction space,
- means for globally validating the N transactions and means for globally invalidating the N transactions, whereby the validation means allow memory spaces previously occupied by records that have been erased or updated during N transactions to be freed whilst the invalidation means essentially allow the back-up data to restore the memory to its status prior to the N transactions and the validation and invalidation means are triggered when the batch command instruction is received before the N transactions are run.

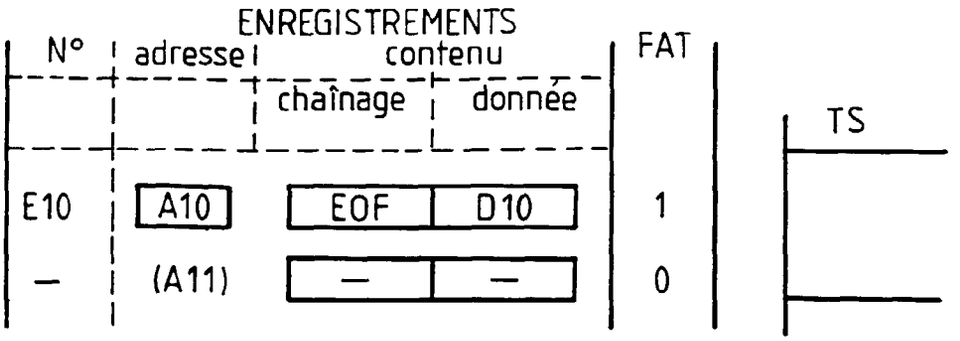


FIG_1a avant ajout E11

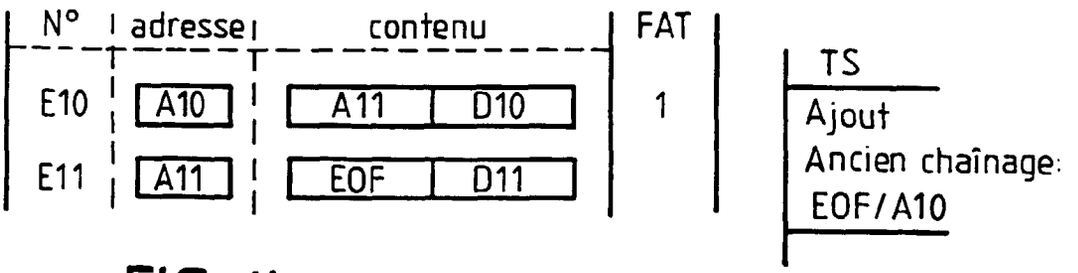


FIG_1b après ajout E11

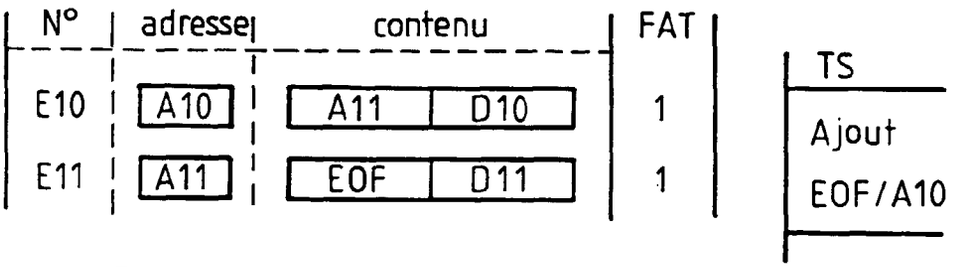


FIG_1c après validation

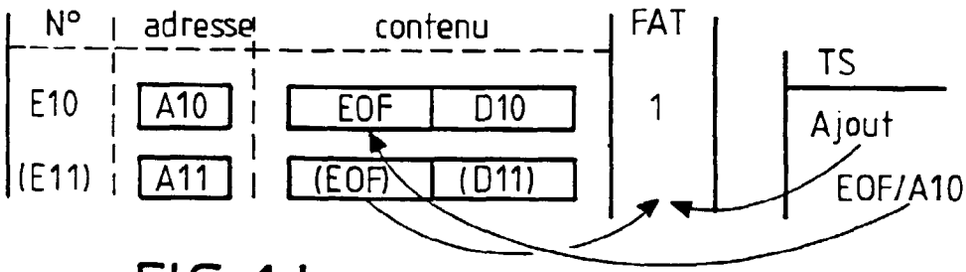


FIG_1d après invalidation

AJOUT D'ENREGISTREMENT

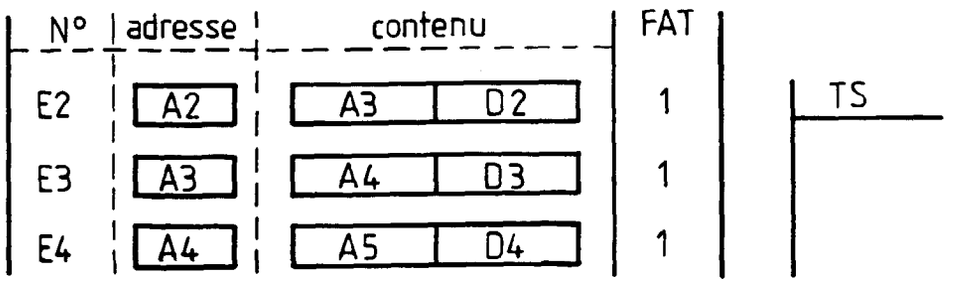


FIG_2a avant suppression E3

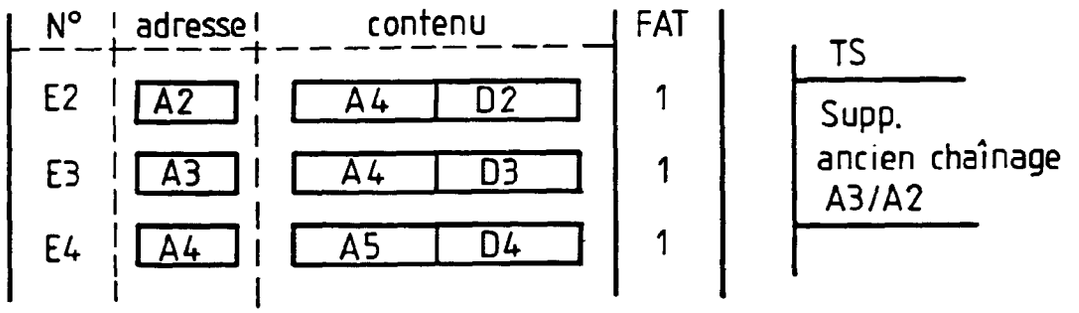


FIG_2b après suppression E3

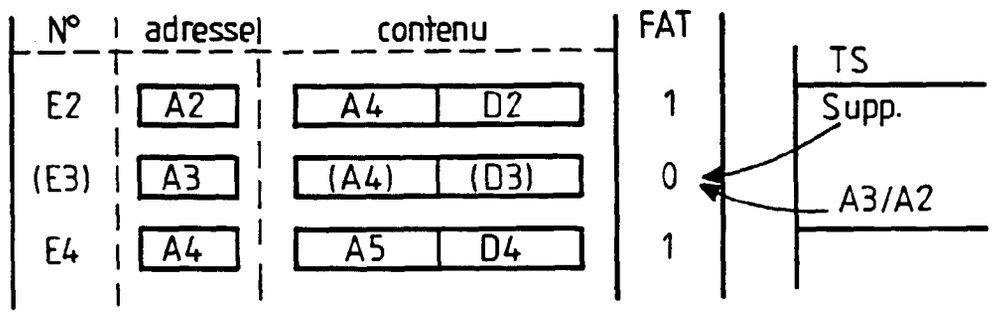


FIG_2c après validation

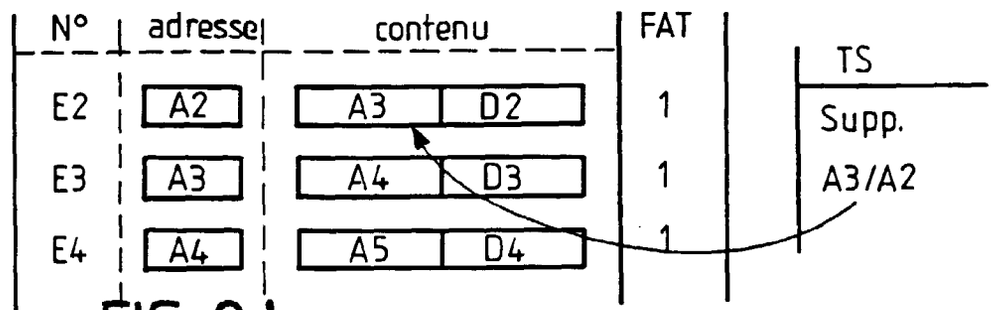


FIG_2d après invalidation

SUPPRESSION D'ENREGISTREMENT

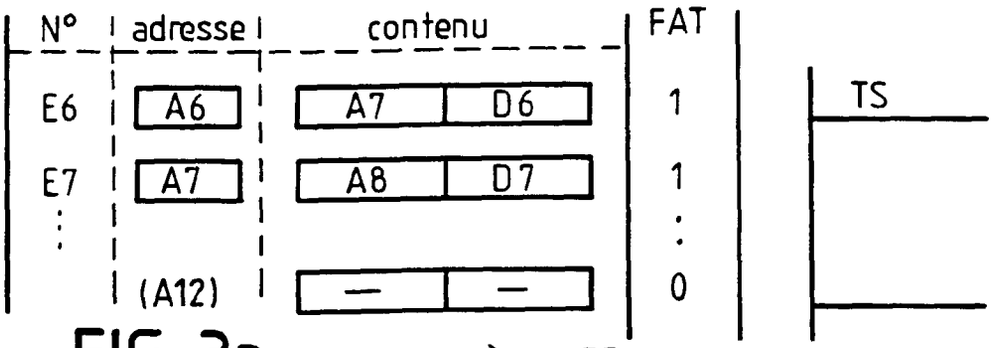


FIG_3a avant mise à jour E7

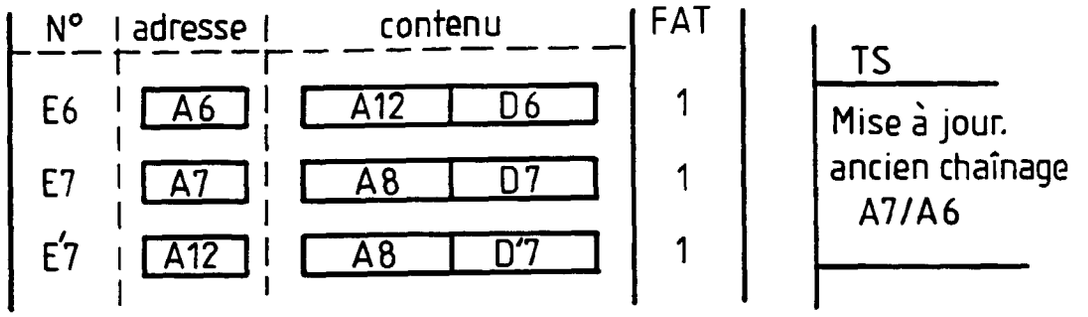


FIG_3b après mise à jour E7

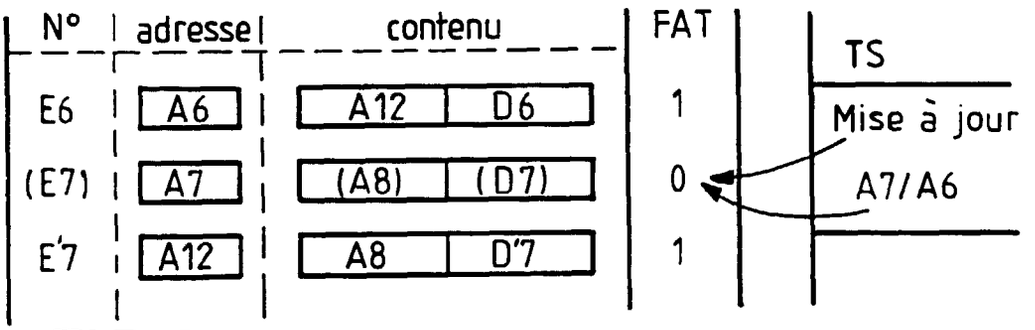


FIG_3c après validation

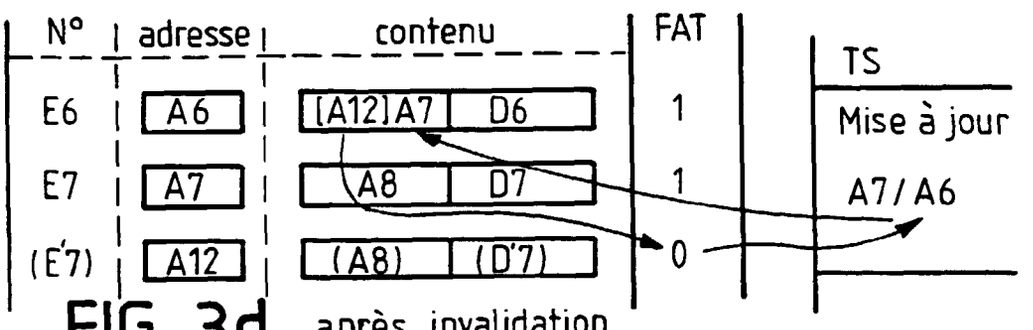


FIG_3d après invalidation

MISE A JOUR D'ENREGISTREMENT