(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 665 791**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **90 10297**

(51) Int Cl[5] : G 11 C 16/06

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt** : 13.08.90.

(30) **Priorité** :

(43) **Date de la mise à disposition du public de la demande :** 14.02.92 Bulletin 92/07.

(56) **Liste des documents cités dans le rapport de recherche :** *Se reporter à la fin du présent fascicule.*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s) :** *MAZINGUE Didier* — FR et *REMERY Patrick* — FR.

(72) **Inventeur(s) :** Mazingue Didier et Remery Patrick.

(73) **Titulaire(s) :**

(74) **Mandataire :** Brevatome.

(54) Procédé de mise à jour d'une mémoire EEPROM.

(57) On affecte à l'information à mémoriser plusieurs zones mémoire et l'on mémorise dans ces zones les dernières valeurs de l'information. Un pointeur indique la zone où se trouve la valeur courante. Pour mettre à jour, on efface une zone, on écrit la nouvelle valeur à jour, on modifie le pointeur en prenant garde de ne modifier qu'une donnée à la fois (zone ou pointeur).

Application aux cartes à microprocesseur.

FR 2 665 791 - A1

DESCRIPTION

PROCEDE DE MISE A JOUR D'UNE MEMOIRE EEPROM

La présente invention a pour objet un procédé de mise à jour d'une mémoire EEPROM. Elle trouve une application générale en informatique et, plus spécialement, dans la gestion des cartes à microprocesseur (dites parfois "cartes à puce").

L'avènement des mémoires EEPROM ("Electrical Erasable Programmable Memory"), notamment dans les cartes à microprocesseur, a modifié sensiblement la façon dont sont écrits les programmes informatiques. En effet, si cette technologie permet d'écrire des données et de les lire, elle permet aussi d'effacer et de ré-écrire à une même adresse.

Si ces possibilités ouvrent des applications nouvelles, elles soulèvent également des problèmes relativement aux risques encourus lors des mises à jour. En effet, si, pour une raison quelconque (coupure du secteur, malveillance, etc...), la nouvelle donnée devant remplacer l'ancienne vient à ne pas s'écrire ou à s'écrire de manière inexacte, des erreurs pouvant avoir des conséquences désastreuses dans certaines applications (financières, médicales, contrôle d'accès, sécurité, ...) peuvent se produire.

La présente invention a justement pour but de remédier à cet inconvénient. A cette fin, elle propose un procédé de mise à jour qui évite tout risque de perte de l'information.

Le procédé de l'invention est caractérisé par le fait que :

- pour chaque information I à mémoriser, on utilise une pluralité de p zones d'une mémoire, p étant un entier au moins égal

à 2,

- on mémorise les p dernières valeurs de l'information I_n, I_{n-1}, ..., I_{n-p+1}, dans ces p zones, depuis la valeur courante I_n jusqu'à la (p-1)ième précédente valeur I_{n-p+1},
- on définit par un pointeur P l'adresse de la zone contenant la valeur courante I_n de l'information,
- pour mettre à jour la valeur courante de l'information, on effectue une suite d'opérations n'affectant chacune qu'une des zones mémoire ou que le pointeur, ces opérations comprenant, notamment, l'effacement de la plus ancienne valeur I_{n-p+1}, l'écriture de la nouvelle valeur I_{n+1}, et la modification du pointeur pour définir la nouvelle adresse de la nouvelle zone où se trouve finalement la nouvelle valeur I_{n+1} mise à jour,
- après chaque opération d'écriture dans une zone, on vérifie si le nouveau contenu de la zone est bien égal au contenu que l'on voulait y écrire, et, dans la négative, lorsque l'opération d'écriture a été manquée, on rétablit le contenu antérieur à l'opération manquée et on réitère ladite opération.

Dans une première variante avantageuse, on utilise deux zones mémoire (p=2), la précédente valeur de l'information I_{n-1} étant écrite dans une première zone (a) et la valeur courante I_n dans une seconde zone (b), le pointeur P étant constitué d'un seul bit, dont la valeur indique celle des deux zones mémoire

(a) ou (b) qui contient la valeur courante I_n.

Dans une seconde variante, on utilise huit zones mémoire a, b, c, d, e, f, g, h pour mémoriser les huit dernières valeurs de l'information I_n, I_{n-1}, ..., I_{n-7}, rangées de la gauche vers la droite, le pointeur d'adresse P étant constitué par un octet dont le contenu est une suite de bits à 1, le rang du bit à 1 le plus à droite désignant le rang de la zone où se trouve la valeur courante I_n de l'information, un indicateur à un bit étant utilisé pour valider le pointeur P.

Naturellement, on peut imaginer des variantes à plus de huit zones, la variante à huit zones convenant cependant bien aux dispositifs à microprocesseur à 8 bits.

De toute façon, les caractéristiques de l'invention apparaîtront mieux à la lumière de la description qui va suivre. Cette description porte sur des modes de réalisation du procédé de l'invention, qui sont donnés à titre explicatif et nullement limitatif.

Dans la variante à deux zones, on peut procéder à la mise à jour de plusieurs manières, par exemple en six étapes ou en trois.

1) Mise à jour en six étapes :

- étape 0 (ou état initial) : le pointeur P est à 0, la première zone (a) contient la valeur précédente I_{n-1}, la seconde zone (b) la valeur courante I_n,
- étape 1 : on efface la zone (a),
- étape 2 : on écrit la valeur courante I_n dans la première zone (a),
- étape 3 : on fait passer le pointeur P

à 1,

- étape 4 : on efface la seconde zone (b),
- étape 5 : on écrit la nouvelle valeur à jour I_{n+1} dans la seconde zone (b),
- étape 6 (état final) : on refait passer le pointeur à 0.

Le tableau I suivant résume ces opérations :

TABLEAU I

	(p)	(a)	(b)
état 0 (état initial)	0	I_{n-1}	I_n
état 1	0	0	I_n
état 2	0	I_n	I_n
état 3	1	I_n	I_n
état 4	1	I_n	0
état 5	1	I_n	I_{n+1}
état 6 (état final)	0	I_n	I_{n+1}

Naturellement, le pointeur P pourrait être mis d'abord à 1 (au lieu de 0) si, par convention inverse, la zone (b) était pointée par un 1.

2) Mise à jour en trois étapes :

- étape 0 (état initial) : le contenu du pointeur P est à 0, la première zone (a) comprend la valeur précédente I_{n-1} et la seconde zone (b) la valeur courante I_n,
- étape 1 : on efface la première zone (a),
- étape 2 : on écrit dans la première zone (a) la nouvelle valeur à jour I_{n+1},
- étape 3 : on fait passer le pointeur P à 1.

Le tableau II suivant résume ces opérations :

TABLEAU II

	(p)	(a)	(b)
étape U (état initial)	U	I_{n-1}	I_n
étape 1	0	0	I_n
étape 2	0	I_{n+1}	I_n
étape 3 (état final)	1	I_{n+1}	I_n

Là encore, la convention sur le pointeur peut être inversée (1 au lieu de 0, 0 au lieu de 1).

Dans le mode de réalisation utilisant huit zones et un octet comme pointeur, le procédé de mise à jour comprend, par exemple, les étapes suivantes :

- étape U (état initial) : les huit zones a, b, c, d, e, f, g, h contiennent respectivement les huit dernières valeurs de l'information allant de la valeur courante I_n à la plus ancienne valeur I_{n-7}, le bit à 1 le plus à droite du pointeur indiquant la zone où se trouve la valeur courante I_n, l'indicateur est à 0,
- étape 1 : on efface la zone directement à droite de la zone contenant la valeur courante et qui contient la plus ancienne valeur I_{n-7},
- étape 2 : on écrit la nouvelle valeur à jour I_{n+1} dans la zone qui vient d'être effacée,
- étape 3 (état final) : un 1 supplémentaire est écrit à droite dans le pointeur.

Le tableau III ci-après résume les opérations.

Dans le cas particulier où la valeur à jour I_n est mémorisée dans la zone la plus à droite, le pointeur ayant alors ses huit bits à 1, pour écrire la nouvelle valeur mise à jour, on s'y prent comme suit :

- étape 1 : on efface le contenu de la première zone (a),
- étape 2 : on écrit la nouvelle valeur à jour I_{n+1} dans cette première zone (a),
- étape 3 : on fait passer l'indicateur à i 1,
- étape 4 : on met à 0 les huit bits du pointeur P,
- étape 5 : on met à 1 le premier bit à gauche du pointeur P,
- étape 6 : on remet l'indicateur à i 0.

Le tableau IV ci-après résume ces opérations.

Comme dans le cas de la mise à jour sur deux zones, une seule donnée contenue dans une zone, dans le pointeur ou dans l'indicateur est modifiée à chaque étape.

Là encore, la convention sur l'indicateur peut être inversée (1 au lieu de 0, 0 au lieu de 1).

Si au cours d'une des opérations qui viennent d'être décrites, un quelconque problème faisait que la mise à jour ne pouvait s'effectuer complètement, il serait toujours possible, d'une part, de s'en rendre compte (en effectuant une lecture de la zone qui vient d'être écrite et en la comparant à ce que l'on voulait y écrire) et, d'autre part, de se replacer dans la situation antérieure pour réitérer l'opération manquée.

TABLEAU III

Zones	a	b	c	d	e	f	g	h
Etape initiale :								
Contenu	I_{n-2}	I_{n-1}	I_n	I_{n-7}	I_{n-6}	I_{n-5}	I_{n-4}	I_{n-3}
Pointeur	1	1	1	0	0	0	0	0
Indicateur	0							
Etape 1 :								
Contenu	I_{n-2}	I_{n-1}	I_n	0	I_{n-6}	I_{n-5}	I_{n-4}	I_{n-3}
Pointeur	1	1	1	0	0	0	0	0
Indicateur	0							
Etape 2 :								
Contenu	I_{n-2}	I_{n-1}	I_n	I_{n+1}	I_{n-6}	I_{n-5}	I_{n-4}	I_{n-3}
Pointeur	1	1	1	0	0	0	0	0
Indicateur	0							
Etape 3 :								
Contenu	I_{n-2}	I_{n-1}	I_n	I_{n+1}	I_{n-6}	I_{n-5}	I_{n-4}	I_{n-3}
Pointeur	1	1	1	1	0	0	0	0
Indicateur	0							

TABLEAU IV

Zones	a	b	c	d	e	f	g	h
Etape initiale								
Contenu	I_{n-7}	I_{n-6}	I_{n-5}	I_{n-4}	I_{n-3}	I_{n-2}	I_{n-1}	I_n
Pointeur	1	1	1	1	1	1	1	1
Indicateur	0							
Etape 1								
Contenu	0	I_{n-6}	I_{n-5}	I_{n-4}	I_{n-3}	I_{n-2}	I_{n-1}	I_n
Etape 2								
Contenu	I_{n+1}	I_{n-6}	I_{n-5}	I_{n-4}	I_{n-3}	I_{n-2}	I_{n-1}	I_n
Etape 3								
Indicateur	1							
Etape 4								
Pointeur	0	0	0	0	0	0	0	0
Etape 5								
Pointeur	1	0	0	0	0	00	0	0
Etape 6								
Indicateur	0							

REVENDICATIONS

1. Procédé de mise à jour du contenu d'une mémoire EEPROM, caractérisé par le fait que :

- pour chaque information (I) à mémoriser, on utilise une pluralité de p zones de la mémoire, p étant un entier au moins égal à 2,
- on mémorise les p dernières valeurs de l'information (I_n, I_{n-1}, ..., I_{n-p+1}), dans ces p zones, depuis la valeur courante (I_n) jusqu'à la (p-1)ième précédente valeur (I_{n-p+1}),
- on définit par un pointeur (P) l'adresse de la zone contenant la valeur courante (I_n) de l'information,
- pour mettre à jour la valeur courante de l'information, on effectue une suite d'opérations n'affectant chacune qu'une des zones mémoire ou que le pointeur, ces opérations comprenant, notamment, l'effacement de la plus ancienne valeur (I_{n-p+1}), l'écriture de la nouvelle valeur (I_{n+1}), et la modification du pointeur pour définir la nouvelle adresse de la nouvelle zone où se trouve finalement la nouvelle valeur (I_{n+1}) mise à jour,
- après chaque opération d'écriture dans une zone, on vérifie si le nouveau contenu de la zone est bien égal au contenu que l'on voulait y écrire, et, dans la négative, lorsque l'opération d'écriture a été manquée, on rétablit le contenu antérieur à l'opération manquée et on réitère ladite opération.

2. Procédé selon la revendication 1, caractérisé par le fait que l'on utilise deux zones mémoire (p=2), la précédente valeur de l'information (I_{n-1}) étant écrite dans une première zone (a) et la valeur courante (I_n) dans une seconde zone (b), le pointeur (P) étant constitué d'un seul bit, dont la valeur indique celle des deux zones mémoire ((a) ou (b)) qui contient la valeur courante (I_n).

3. Procédé selon la revendication 2, caractérisé par le fait que la mise à jour s'effectue en six étapes :

- étape 0 (ou état initial) : le pointeur (P) est à p (0 ou 1), la première zone (a) contient la valeur précédente (I_{n-1}), la seconde zone (b) la valeur courante (I_n),
- étape 1 : on efface la zone (a),
- étape 2 : on écrit la valeur courante (I_n) dans la première zone (a),
- étape 3 : on fait passer le pointeur (P) à $\overline{p}$ (1 ou 0),
- étape 4 : on efface la seconde zone (b),
- étape 5 : on écrit la nouvelle valeur à jour (I_{n+1}) dans la seconde zone (b),
- étape 6 (état final) : on refait passer le pointeur à p (0 ou 1).

4. Procédé selon la revendication 2, caractérisé par le fait qu'il comprend les étapes suivantes :

- étape 0 (état initial) : le contenu du pointeur (P) est à p (0 ou 1), la première zone (a) comprend la valeur précédente (I_{n-1}) et la seconde zone (b) la valeur courante (I_n),

- étape 1 : on efface la première zone (a),
- étape 2 : on écrit dans la première zone (a) la nouvelle valeur à jour (I_{n+1}),
- étape 3 : on fait passer le pointeur (P) à $\overline{p}$ (1 ou 0).

5. Procédé selon la revendication 1, caractérisé par le fait que l'on utilise huit zones mémoire (a, b, c, d, e, f, g, h) (p=8) pour mémoriser les huit dernières valeurs de l'information (I_n, I_{n-1}, ..., I_{n-7}), rangées de la gauche vers la droite, le pointeur d'adresse (P) étant constitué par un octet dont le contenu est une suite de bits à 1, le rang du bit à 1 le plus à droite désignant le rang de la zone où se trouve la valeur courante (I_n) de l'information, un indicateur à un bit étant utilisé pour valider le pointeur (P).

6. Procédé selon la revendication 5, caractérisé par le fait qu'il comprend les opérations suivantes :

- étape 0 (état initial) : les huit zones (a, b, c, d, e, f, g, h) contiennent respectivement les huit dernières valeurs de l'information allant de la valeur courante (I_n) à la plus ancienne valeur (I_{n-7}), le bit à 1 le plus à droite du pointeur indiquant la zone où se trouve la valeur courante (I_n), l'indicateur est à i (0 ou 1),
- étape 1 : on efface la zone directement à droite de la zone contenant la valeur courante et qui contient donc la plus ancienne valeur (I_{n-7}),
- étape 2 : on écrit la nouvelle valeur à

jour (I_{n+1}) dans ladite zone, qui vient d'être effacée,

- étape 3 (état final) : un 1 supplémentaire est écrit à droite dans le pointeur.

7. Procédé selon la revendication 6, caractérisé par le fait que, dans le cas particulier où la valeur à jour (I_n) est mémorisée dans la zone la plus à droite, le pointeur ayant alors ses huit bits à 1, pour écrire la nouvelle valeur mise à jour :

- étape 1 : on efface le contenu de la première zone (a),
- étape 2 : on écrit la nouvelle valeur à jour (I_{n+1}) dans cette première zone (a),
- étape 3 : on fait passer l'indicateur à $\overline{i}$ (1 ou 0),
- étape 4 : on met à 0 les huit bits du pointeur (P),
- étape 5 : on met à 1 le premier bit à gauche du pointeur (P),
- étape 6 : on remet l'indicateur à i (0 ou 1).

REPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

2665791
N° d'enregistrement national

RAPPORT DE RECHERCHE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

FR 9010297
FA 445111

DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendications concernées de la demande examinée
Y	EP-A-0 340 981 (SONY) * Colonne 1, ligne 46 - colonne 2, ligne 3; colonne 2, ligne 23 - colonne 4, ligne 29; figures 1-4 *	1
A		2-7

Y	US-A-4 780 855 (IIDA et al.) * Colonne 2, ligne 41 - colonne 3, ligne 47; colonne 4, ligne 28 - colonne 6, ligne 5; figures 1-5 *	1
A		2-7

DOMAINES TECHNIQUES RECHERCHES (Int. Cl.5)

G 11 C 16
G 11 C 7

Date d'achèvement de la recherche: 25-04-1991

Examinateur: LINDQUIST J.W.

CATEGORIE DES DOCUMENTS CITES

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 03.82 (P0413)