

12) **DEMANDE DE BREVET D'INVENTION**

A1

22) Date de dépôt : 20.04.01.

30) Priorité :

43) Date de mise à la disposition du public de la demande : 25.10.02 Bulletin 02/43.

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

60) Références à d'autres documents nationaux apparentés :

71) Demandeur(s) : **STMICROELECTRONICS SA**
Société anonyme — FR et **INFINEON TECHNOLOGIES AG** — DE.

72) Inventeur(s) : HENRIQUES JEAN PAUL et DEVAUX FABRICE.

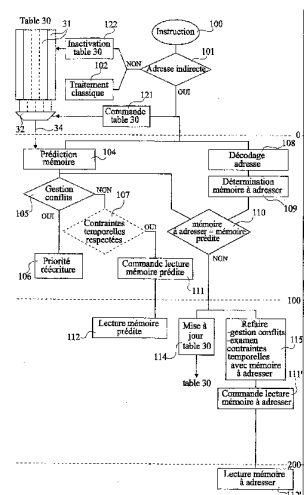
73) Titulaire(s) :

74) Mandataire(s) : CABINET MICHEL DE BEAUMONT.

54) **PROCEDE D'ADRESSAGE DE MEMOIRE OPTIMISE.**

57) L'invention concerne un procédé pour adresser en lecture un emplacement d'une parmi plusieurs mémoires à l'aide d'une adresse codée provenant d'une instruction. Le procédé comprend les étapes suivantes

- a) prédire (104) la mémoire correspondant à l'emplacement à adresser,
- b) décoder (108) l'adresse de l'emplacement à adresser et déterminer (109) la mémoire à adresser,
- c) gérer (105) un éventuel conflit de lecture et de réécriture en supposant que la mémoire prédite est la mémoire à adresser,
- d) commander (111) l'adressage de la mémoire prédite à l'issue de l'étape de gestion (105),
- e) à l'issue de l'étape b), déterminer (110) si la mémoire à adresser correspond à la mémoire prédite, et
- f) si la mémoire à adresser ne correspond pas à la mémoire prédite, gérer (115) un éventuel conflit de lecture et de réécriture dans la mémoire à adresser et adresser l'emplacement de la mémoire à adresser.



FR 2 823 874 - A1



PROCÉDÉ D'ADRESSAGE DE MÉMOIRE OPTIMISÉ

La présente invention concerne l'adressage de mémoire et plus particulièrement la recherche en mémoire de données par un dispositif informatique tel qu'un microcontrôleur.

La figure 1 représente schématiquement la structure d'un microcontrôleur classique. Le microcontrôleur 1 comprend un contrôleur de programme FPC ("Fetch and Pipeline Controller" ou "contrôleur de capture et de séquençement"), un module d'adresse ADU ("Address Unit" ou "module d'adresse") et un module FXU ("Fixed point Unit" ou "module de calcul").

Le contrôleur de programme FPC est relié à une mémoire de programme 3 externe au microcontrôleur par l'intermédiaire d'un contrôleur de mémoire de programme 5, lui aussi externe au microcontrôleur. Le contrôleur de programme FPC comprend un module de précapture 7 qui, au cours d'un cycle d'horloge, stocke l'instruction courante et assure le séquençement du microcontrôleur. Le contrôleur de programme FPC comprend aussi un module de capture et de décodage 9. A chaque nouveau cycle d'horloge, le contenu du module 7 est transféré dans le module 9. Ainsi, au cours d'un cycle n, le module 9 contient l'instruction du cycle précédent, n-1, et la décode. Le décodage de l'instruction du cycle n-1 permet de connaître la nature de l'opération à effectuer, l'adresse sous forme codée du ou des opérandes à

utiliser dans l'opération et l'adresse sous forme codée de l'emplacement mémoire où le résultat de l'opération doit être placé.

Le module d'adresse ADU comprend d'abord un module 11 de décodage d'adresse. A chaque nouveau cycle, le module de capture et de décodage 9 transfère son contenu au module de décodage d'adresse 11. Le module 11 décode les adresses codées qu'il reçoit, c'est-à-dire qu'il détermine, à partir de ces adresses, l'adresse physique ou adresse réelle des opérandes à récupérer en mémoire, ainsi que l'adresse physique de l'emplacement où doit être stocké le résultat de l'opération.

Le module d'adresse ADU comprend ensuite un module 13 qui permet d'accéder en lecture au support mémoire contenant les opérandes. De façon classique, le microcontrôleur 1 comporte deux mémoires internes de données, une mémoire vive 15 accessible par l'intermédiaire d'un contrôleur mémoire 17 et un banc de registres 19 accessibles directement. Une partie de la mémoire 15 est affectée à une pile adressée par un pointeur de pile. Dès que le module 11 a fini de décoder les adresses à utiliser (et de réaliser divers traitements qui seront vus par la suite), il transfère son contenu au module d'accès en lecture 13. Le module 13 adresse alors en lecture, au début d'un cycle, la mémoire appropriée et récupère les opérandes à utiliser. Cette opération, qui dure un cycle, est symbolisée par les flèches 21 et 22 allant respectivement du contrôleur 17 et du banc de registres 19 au module 13.

Au cycle suivant, le module 13 transfère son contenu à un module d'exécution 23 qui constitue l'unique module du module FXU. Le module 23 réalise la ou les opérations arithmétiques ou logiques requises par l'instruction et fournit le résultat de l'opération. Le module 23 transfère ensuite le résultat de l'opération et l'adresse physique à laquelle le résultat doit être stocké à un module de réécriture 25. Le module 23 peut accéder directement à un registre du banc de registres 19 en écriture, ce qui est symbolisé par la flèche 24. Par exemple, le

module 23 peut modifier, si cela est requis, le contenu d'un registre du banc utilisé comme pointeur d'adresse.

Le module de réécriture 25 fait aussi partie du module d'adresse ADU. Il adresse la mémoire 15 en écriture, ce qui est symbolisé par la flèche 26, afin d'y stocker le résultat de l'opération. Le module 25 peut aussi adresser un registre du banc de registres 19, comme cela est symbolisé par la flèche 27, pour y stocker le résultat si cela est requis. Le module 25 réalise sa tâche en un cycle d'horloge.

Les modules 7, 9, 11, 13, 23 et 25 du microcontrôleur (on pourrait nommer ces modules "étages") réalisent une structure dite "pipeline", c'est-à-dire une structure dans laquelle le contenu d'un module est transféré séquentiellement au module suivant. Le microcontrôleur contient ainsi six instructions en même temps, à divers stades de traitement. Si chaque module n'utilisait qu'un cycle d'horloge pour accomplir sa tâche, le traitement d'une instruction ne prendrait que six cycles et le microcontrôleur fonctionnerait de façon optimale.

En pratique cependant, une limitation vient du module d'adresses ADU, et en particulier du module de décodage d'adresse 11. En effet, comme on le verra par la suite, certaines adresses nécessitent un temps de décodage élevé et le module 11 a besoin de deux cycles d'horloge pour fournir les adresses physiques correspondantes. Il en résulte la perte d'un cycle et le fonctionnement du microcontrôleur n'est pas optimal.

Un objet de la présente invention est de prévoir un procédé d'adressage de mémoire moins long que dans l'art antérieur.

Pour atteindre cet objet, la présente invention prévoit un procédé pour adresser en lecture un emplacement d'une parmi plusieurs mémoires à l'aide d'une adresse codée provenant d'une instruction. Le procédé comprend les étapes suivantes :

a) prédire, parmi les plusieurs mémoires, la mémoire correspondant à l'emplacement à adresser,

b) décoder l'adresse de l'emplacement à adresser et déterminer la mémoire à adresser,

c) à l'issue de l'étape a), gérer un éventuel conflit de lecture et de réécriture en supposant que la mémoire prédite est la mémoire à adresser,

d) commander l'adressage de la mémoire prédite à l'issue de l'étape de gestion c),

e) à l'issue de l'étape b), déterminer si la mémoire à adresser correspond à la mémoire prédite, et

f) si la mémoire à adresser ne correspond pas à la mémoire prédite, gérer un éventuel conflit de lecture et de réécriture dans la mémoire à adresser et adresser l'emplacement de la mémoire à adresser.

Selon un mode de réalisation de la présente invention, l'étape a) utilise une table contenant une historique des mémoires adressées lors des instructions précédentes.

Selon un mode de réalisation de la présente invention, la table renferme un nombre de registres égal au nombre de pointeurs utilisés pour déterminer une adresse, chacun desdits registres étant affecté à un pointeur particulier, et dans lequel chacun desdits registres contient une information permettant de déterminer laquelle des mémoires a été adressée par ledit pointeur lors de sa dernière utilisation.

Selon un mode de réalisation de la présente invention, l'étape de décodage et de détermination b) débute en même temps que l'étape de prédiction a).

Selon un mode de réalisation de la présente invention, en cas de conflit de lecture et réécriture dans une même mémoire, la priorité est donnée à la réécriture.

Selon un mode de réalisation de la présente invention, une étape d'examen de contraintes temporelles est réalisée après la gestion de conflit de lecture et de réécriture des étapes c) ou f), afin d'examiner si la lecture de la mémoire à lire peut être effectuée au cours du cycle d'horloge suivant.

Selon un mode de réalisation de la présente invention, les adresses codées appartiennent à un des types suivants : adresse courte, adresse longue ou adresse indirecte.

Selon un mode de réalisation de la présente invention, le procédé comprend une étape de détermination de type d'adresse et les étapes de prédiction a), de gestion c), d'adressage d) et de comparaison e) ne sont réalisées que si l'adresse codée de l'emplacement à adresser est une adresse indirecte.

Selon un mode de réalisation de la présente invention, la table est invalidée si l'adresse codée de l'emplacement à adresser n'est pas une adresse indirecte.

Selon un mode de réalisation de la présente invention, le procédé est réalisé par un dispositif de type microcontrôleur ou microprocesseur, et les mémoires pouvant être adressées sont une mémoire de type SRAM interne audit dispositif, une mémoire de type DPRAM interne audit dispositif et une mémoire externe audit dispositif.

Ces objets, caractéristiques et avantages, ainsi que d'autres, de la présente invention seront exposés en détail dans la description suivante de modes de réalisation particuliers faite à titre non-limitatif en relation avec les figures jointes parmi lesquelles :

la figure 1, déjà décrite, représente schématiquement la structure d'un microcontrôleur classique ;

les figures 2A, 2B, 2C1 et 2C2 représentent des types d'adresse utilisés dans la présente invention ;

la figure 3 représente un diagramme temporel illustrant le fonctionnement du module d'adresse dans l'art antérieur ;

les figures 4A et 4B représentent chacune un diagramme temporel illustrant le fonctionnement du module d'adresse selon la présente invention ; et

la figure 5 représente schématiquement un organigramme résumant la mise en oeuvre de la présente invention.

Le microcontrôleur utilisé dans la présente invention a une structure globalement semblable à celle du microcontrôleur de

la figure 1 et de mêmes références correspondent à des éléments identiques ou réalisant, du moins en partie, la même fonction. Les particularités du microcontrôleur de la présente invention par rapport à celui de la figure 1 apparaîtront au cours de la description, notamment lors de la description de la figure 5.

Les adresses physiques utilisées dans la présente invention sont des adresses de 24 bits. Comme cela a été mentionné, les instructions reçues par le microcontrôleur ne comportent pas d'adresses physiques, mais des adresses codées plus courtes permettant de déterminer les adresses physiques correspondantes.

De manière classique, dans la présente invention, les adresses codées sont de trois types.

En figure 2A, est représentée une adresse physique de type "court". Cette adresse, dite courte, comporte 8 bits, b0 à b7. Les bits b0 à b7 correspondent à l'adresse d'un registre particulier comportant 24 bits déterminés. Les 24 bits du registre déterminé par l'adresse courte forment une base fixe qui doit être ajoutée à l'adresse courte pour obtenir l'adresse physique correspondante.

En figure 2B, est représentée une adresse codée de type "long". Cette adresse, dite longue, comporte 16 bits, b0 à b15. Les 16 bits de l'adresse longue forment les 16 bits de poids faible de l'adresse physique correspondante. Les 8 bits de poids fort de l'adresse physique sont déterminés à partir de bits particuliers de l'adresse longue. A titre d'exemple non limitatif, dans une mise en oeuvre de la présente invention, les deux bits de poids fort b15, b14 de l'adresse longue sont utilisés pour adresser un parmi quatre registres programmables de 8 bits représentant les 8 bits de poids fort de l'adresse physique correspondante.

En figure 2C1, est représentée une adresse codée de type "indirect". Cette adresse, dite "indirecte", comporte un nombre de bits réduit qui représentent l'adresse d'un registre particulier R. Le registre R sert de pointeur d'adresse. Son

contenu, non fixe, sert à déterminer l'adresse physique souhaitée. A titre d'exemple non limitatif, dans une mise en oeuvre de la présente invention, les adresses indirectes comportent quatre bits, b0 à b3. Ces quatre bits permettent d'adresser un de seize pointeurs d'adresse. Comme cela est illustré en figure 2C2, chaque pointeur d'adresse est un registre de 16 bits, b0 à b15. A partir des 16 bits du pointeur d'adresse, l'adresse physique est déterminée de la même manière que pour les adresses longues. Du fait de sa grande souplesse et du faible nombre de bits qu'il requiert, ce mode d'adressage, dit indirect, est très généralement utilisé.

L'étape de décodage de l'adresse codée est réalisée par le module 11. Les adresses indirectes sont les plus longues à décoder et nécessitent une durée égale à environ un quart de cycle d'horloge.

Après avoir déterminé l'adresse physique de 24 bits, le module 11 détermine quel est le support mémoire à adresser. Dans la présente invention, en plus des mémoires internes au microcontrôleur, le module d'adresse ADU peut accéder à une ou plusieurs mémoires externes au microcontrôleur. Par ailleurs, ce qui, dans la présente invention, correspond à la mémoire vive de l'art antérieur comprend une mémoire de type SRAM ("Synchronous Random Access Memory" - "mémoire vive synchrone"), une mémoire de type DPRAM ("Dual Port Random Access Memory" - "mémoire vive à double accès"), une pile pouvant faire partie d'une des deux mémoires précédentes et des registres, comme les registres permettant de déterminer les adresses physiques à l'aide des adresses codées.

Pour déterminer la mémoire à adresser, le module 11 examine les bits de l'adresse physique, par exemple en commençant par les bits de poids fort. En effet, les adresses des emplacements de chaque mémoire particulière se suivent en général et vont d'une adresse inférieure ou minimale à une adresse supérieure ou maximale. Par exemple, les adresses (en notation hexadécimale) allant de 008000 à 00FBFF correspondent à des

emplacements dans la mémoire SRAM, les adresses de 00FC00 à 00FDFE correspondent à des emplacements dans la mémoire DPRAM, etc. Si l'adresse ne correspond à aucun emplacement d'une mémoire interne au microcontrôleur, on en déduit que la mémoire à adresser est une mémoire externe. S'il y a plusieurs mémoires externes, on procède comme précédemment pour déterminer de quelle mémoire il s'agit. Cette étape de détermination de la mémoire à adresser est longue. Il peut être nécessaire d'examiner un grand nombre de bits de l'adresse physique, 12 par exemple, avant de connaître la mémoire à adresser. La durée de cette étape est de l'ordre d'un quart de cycle d'horloge.

Ensuite, le module 11 procède à la gestion d'un éventuel conflit entre la lecture et la réécriture dans un même support mémoire. Au cours de cette étape, aussi nommée étape de gestion de conflit de bande passante, le module 11 détermine si l'instruction présente dans le module 25 nécessite une réécriture dans la même mémoire que celle qui doit être adressée en lecture par l'instruction qu'il contient. Si c'est le cas, la priorité est en général donnée à la réécriture dans la mémoire, car l'instruction correspondante est plus ancienne que l'instruction demandant la lecture. Cette étape de gestion est en général moins longue que les précédentes, et a une durée égale à environ 15 % d'un cycle d'horloge.

Après l'étape de gestion, le module 11 réalise une quatrième étape consistant à déterminer si les contraintes temporelles liées à l'accès de la mémoire à adresser sont respectées. En effet, les mémoires, les mémoires synchrones en particulier, doivent recevoir les commandes d'adresses à lire en un temps déterminé avant le début (front montant de l'horloge, par exemple) du cycle au cours duquel a lieu la lecture. Ce temps déterminé permet en particulier à la mémoire de décoder les commandes fournies (si cela n'est pas respecté, la mémoire a un comportement non prévisible et la valeur lue ne peut être utilisée). Cette quatrième étape dure environ 10 % d'un cycle d'horloge. Si les contraintes temporelles sont respectées, le

module 11 transfère son contenu dans le module 13 qui prépare l'adressage et réalise la lecture de la mémoire au cycle suivant. Si les contraintes temporelles ne sont pas respectées, le module 11 transfère son contenu au module 13 au début du cycle suivant.

5 Le module 13 dispose alors d'un cycle pour préparer l'adressage et la lecture n'est faite qu'au cycle d'après.

Selon le type d'adresse codée utilisé par l'instruction, la durée des étapes réalisées par le module 11 est plus ou moins longue. Dans le cas d'adresses courtes ou longues, dans

10 l'art antérieur comme dans l'invention, les contraintes temporelles sont généralement respectées et la lecture de la mémoire à adresser peut être faite au cycle suivant. Par contre, dans le cas des adresses indirectes, les contraintes temporelles ne sont jamais respectées dans l'art antérieur et deux cycles

15 d'horloge sont requis avant la lecture de la mémoire.

La figure 3 représente trois cycles consécutifs d'horloge, et les diverse étapes réalisées par le module d'adresse dans l'art antérieur, dans le cas d'une adresse indirecte.

20 Au cours du premier cycle, allant de 0 à 100, le module 11 de l'art antérieur réalise les étapes de décodage d'adresse, de détermination de la mémoire à adresser, de gestion de conflit et d'examen des contraintes temporelles. Ces quatre étapes successives, notées A, B, C, D, durent respectivement 25 %, 25 %, 25

25 15 % et 10 % du temps du premier cycle. Le module 11 a terminé son traitement au temps noté 75, correspondant à 75 % du premier cycle. Il ne reste que 25 % de la durée d'un cycle pour terminer le premier cycle, et cette durée est insuffisante pour permettre la préparation de l'adressage mémoire par le module 13. En consé-

30 quence, le module 11 ne transfère son contenu dans le module 13 qu'au début du deuxième cycle d'horloge, allant de 100 à 200. La lecture dans la mémoire ne débute alors qu'au début du troisième cycle, allant de 200 à 300, et l'opérande recherché est disponible au temps 300.

On voit que, dans l'art antérieur, la lecture nécessite trois cycles d'horloge dans le cas d'une adresse indirecte, et que le deuxième cycle est très sous-utilisé. Dans certains micro-contrôleurs de l'art antérieur, d'ailleurs, les étapes C et D sont réalisées directement au début du deuxième cycle dans le cas des adresses indirectes.

L'invention va être maintenant décrite en relation avec les figures 4A et 4B qui illustrent le traitement, selon la présente invention, du même cas que celui envisagé dans la figure 3. Comme cela est représenté en figure 4A, le module 11 de la présente invention réalise les étapes de décodage d'adresse A et de détermination de mémoire à adresser B. Comme en figure 3, ces étapes ont une durée totale égale à 50 % de la durée du premier cycle, allant de 0 à 100.

En même temps que l'étape A, le module 11 réalise une étape de prédiction A'. Cette étape de prédiction consiste à prédire la mémoire qui doit être adressée. L'étape A', représentée en figure 4B, commence au début du premier cycle et a une durée approximative égale à 20 % du premier cycle. La mémoire prédite est supposée être la mémoire à adresser. Pour déterminer la mémoire prédite, on peut utiliser diverses tables plus ou moins complexes contenant l'historique des instructions précédentes, ou du moins l'historique des mémoires précédemment adressées, afin de faire des pronostics sur la mémoire qui doit être adressée lors de l'instruction courante. On peut aussi simplement utiliser comme mémoire prédite la mémoire qui a été adressée lors de l'instruction précédente. A la première instruction d'un programme, on peut choisir une mémoire quelconque comme mémoire prédite.

A la suite de l'étape A', le module 11 réalise une étape de gestion de conflit C' (figure 4B). Au cours de cette étape, il est examiné si la mémoire prédite correspond à la mémoire en attente de réécriture par le module 25. Supposons dans un premier temps que ce ne soit pas le cas. Après l'étape de gestion C', le module 11 réalise une étape D' (toujours en figure

4B) d'examen pour savoir si les contraintes temporelles sont respectées. Les étapes C' et D' durent respectivement 15 % et 10 % de la durée d'un cycle et l'ensemble des étapes A', C' et D' a une durée égale à environ 45 % du premier cycle. On notera que, dans ce contexte, l'étape D' conduit toujours à un respect des contraintes temporelles et que cette étape pourrait être supprimée.

A la fin de l'étape B, la mémoire à adresser est connue. Le module 11 réalise alors une étape E (figure 4A) pour déterminer si la mémoire prédite correspond à la mémoire à adresser. Cette étape dure environ 10 % du temps d'un cycle.

Si la prédiction est bonne, c'est-à-dire si la mémoire prédite correspond effectivement à la mémoire à adresser, l'ensemble du traitement réalisé par le module 11 dans la présente invention a une durée totale égale à 60 % du premier cycle. La durée restante du premier cycle (40 % du cycle) est alors suffisante pour que le module 11 présente la demande d'adressage au module 13 au cours du premier cycle. Le module 13 prépare l'adressage de la mémoire à adresser au cours du premier cycle et la lecture a lieu au cours du deuxième cycle d'horloge, allant de 100 à 200. Par rapport à l'art antérieur, la présente invention permet le gain d'un cycle d'horloge. Il n'y a pas de "temps mort" dans le traitement des instructions et le fonctionnement du microcontrôleur est optimisé.

Si la mémoire prédite ne correspond pas à la mémoire à adresser, ce qui est connu à la fin de l'étape E, le module 11 réalise, à la suite de l'étape E ou au début du deuxième cycle, une nouvelle étape de gestion de conflit. Au cours de cette étape, le module 11 examine si la mémoire à adresser correspond à la mémoire en attente de réécriture dans le module 25. Si ce n'est pas le cas, le module 11 réalise une nouvelle étape, optionnelle, d'examen des contraintes temporelles, dans ce cas toujours respectées. Il fait ensuite préparer l'adressage de la mémoire à adresser par le module 13. Ces étapes ont lieu au cours du deuxième cycle et la lecture de la mémoire intervient au cours

du troisième cycle. La durée requise pour traiter l'instruction dans ce cas est identique à celle de l'art antérieur.

Dans les programmes utilisés par les microcontrôleurs, un grand nombre d'instructions consécutives concernent souvent l'adressage d'un même support mémoire. Aussi, dans un programme, un même pointeur d'adresse est souvent alloué à un même support mémoire. La prédiction de la mémoire à adresser en fonction d'une historique des mémoires adressées précédemment a de grandes chances de succès et le gain de temps permis par la présente invention est considérable.

Dans la présente invention comme dans l'art antérieur, en cas de conflit entre une lecture et une réécriture dans la mémoire à adresser, la priorité est en général donnée à la réécriture.

La figure 5 représente un organigramme résumant la mise en oeuvre de la présente invention et permettant un exposé plus détaillé de la manière dont peut être réalisée la prédiction.

Tout d'abord, au niveau du contrôleur de programme FPC, on examine (étape 101) si l'instruction 100 concerne au moins une adresse indirecte. Si ce n'est pas le cas (pas d'adresse requise ou que des adresses courtes ou longues), le traitement de l'instruction est réalisé de façon classique (étape 102), c'est-à-dire que le décodage des adresses, la détermination de la mémoire à adresser, la gestion d'éventuels conflits, l'examen des contraintes temporelles et la transmission au module 13 sont normalement effectués par le module 11 au cours d'un seul cycle.

Si l'adresse est une adresse indirecte, le module 11 commence, à l'instant 0 (début du premier cycle où le module d'adresse ADU commence à traiter l'instruction), deux traitements simultanés. D'une part, il détermine la mémoire prédite (étape 104), à l'aide d'une information qu'il reçoit d'une table 30. Ensuite, le module 11 réalise la gestion d'un éventuel conflit de priorité concernant la mémoire prédite (étape 105). Comme cela a été dit, s'il y a conflit, la réécriture sera réalisée prioritairement à la lecture (bloc 106). S'il n'y a pas de conflit, le

module 11 examine en 107, de manière optionnelle on l'a vu, si les contraintes temporelles sont respectées, puis il commande (étape 111) la lecture de la mémoire prédite à l'aide de l'adresse calculée, comme on le verra par la suite, au cours
5 d'une étape 108 de décodage de l'adresse codée. La lecture de la mémoire prédite est réalisée au cours du deuxième cycle (bloc 112).

D'autre part, à l'instant 0, le module 11 commence le décodage (calcul) de l'adresse indirecte (étape 108). Ensuite, il
10 détermine la mémoire à adresser effectivement (étape 109). Après cela, il détermine en 110 si la mémoire prédite correspond effectivement à la mémoire à adresser. Si c'est le cas, aucune action n'est prise.

Si la mémoire prédite ne correspond pas à la mémoire à
15 adresser, la donnée lue dans la mémoire prédite ne sera pas prise en compte. Le module 11 met d'une part à jour la table 30 (étape 114) et d'autre part réalise une nouvelle étape de gestion de conflit de réécriture-lecture, cette fois en ce qui concerne la mémoire à adresser (étape 115). Cette gestion de conflit est de
20 préférence réalisée par les mêmes éléments matériels que ceux qui sont utilisés pour gérer les conflits de la mémoire prédite. La mise à jour de la table et la gestion des conflits concernant la mémoire à adresser peuvent être faits dès que l'on sait que la
25 deuxième cycle, comme cela est représenté. En cas de conflit, la priorité est donnée à la réécriture. S'il n'y a pas de conflit, il est procédé (étape 115) à l'examen des contraintes temporelles, optionnel car ces contraintes sont toujours respectées dans ce cas. La commande en lecture de la mémoire à
30 adresser a lieu à l'étape 111' et la lecture de la mémoire à adresser a lieu au cours du troisième cycle (étape 112').

On va maintenant décrire plus précisément le contenu de la table 30 et la manière dont elle est utilisée. On a vu que, dans la présente invention, il y a seize pointeurs d'adresse et
35 un pointeur de pile. La table 30 contient dix-sept registres 31,

correspondant à chacun des dix-sept pointeurs utilisés. Chacun des registres 31 renferme une information indiquant la mémoire qui a été adressée par le pointeur concerné la dernière fois où ce pointeur a été utilisé dans une instruction. Les registres 31
5 peuvent être des registres de trois bits, ce qui suffit pour permettre la discrimination de huit mémoires. Si les seules mémoires adressées par les pointeurs sont les mémoires SRAM, DPRAM et une seule mémoire externe, on pourra utiliser deux bits seulement pour les registres 31 ou utiliser trois bits avec un
10 codage simple, par exemple les bits "001" correspondant à la mémoire SRAM, les bits "010" correspondant à la mémoire DPRAM, et les bits "100" correspondant à la mémoire externe.

Lorsqu'une adresse indirecte est détectée au niveau du contrôleur FPC, cette adresse est examinée au niveau d'un bloc
15 121 pour déterminer le pointeur à utiliser pour le décodage de l'adresse codée. Cette information est utilisée pour commander un multiplexeur 32 associé à la table 30. Le multiplexeur 32 a dix-sept entrées, dont chacune est reliée à un des registres 31. La commande du multiplexeur 32 consiste à sélectionner le registre
20 correspondant au pointeur utilisé dans l'instruction. Le multiplexeur 32 fournit alors, sur sa sortie 34, une information correspondant à la mémoire qui a été adressée par ce pointeur lors de sa dernière utilisation. Cette information est fournie au module 11. Le module 11 décode cette information au cours de
25 l'étape 104 et choisit comme mémoire prédite la mémoire adressée précédemment par le pointeur utilisé.

En cas d'insuccès de la prédiction, une mise à jour de la table 30 est nécessaire et est réalisée de la manière suivante. Le code correspondant à la mémoire à adresser est
30 déterminé (par exemple, "001" pour la mémoire SRAM dans l'exemple donné ci-dessus) et il est stocké dans le registre 31 correspondant au pointeur concerné, à la place de l'ancien code.

Au début de chaque nouveau programme, on peut remplir tous les registres de la table 30 avec un même code, par exemple
35 celui correspondant à la mémoire SRAM.

Dans une variante de la présente invention, la table 30 est invalidée (bloc 122) lorsque l'instruction courante n'utilise pas d'adresse indirecte. L'invalidation de la table 30 peut être réalisée au niveau de la table 30 elle-même ou au niveau du multiplexeur 32. Lorsque la table 30 est invalidée, la sortie 34 du multiplexeur ne délivre aucune information et l'étape de prédiction A' n'a pas lieu. On peut aussi, lorsque la table 30 est invalidée, prévoir par exemple que la sortie du multiplexeur 32 délivre un code ne correspondant à aucun support mémoire.

10 La présente invention est susceptible de diverses variantes et modifications qui apparaîtront à l'homme de l'art. En particulier, le dispositif mettant en oeuvre la présente invention pourra être un microcontrôleur, un microprocesseur, ou tout autre dispositif approprié.

15 La structure des adresses physiques, la manière dont est codée, dans une adresse physique, la mémoire à adresser, le nombre de pointeurs utilisés, le nombre des registres ainsi que le nombre des mémoires adressables ne sont donnés qu'à titre d'exemple, et l'homme du métier est à même de modifier ces éléments dans le cadre de la présente invention.

20 La table 30 pourra contenir une historique différente que celle qui a été décrite à titre d'exemple. Par exemple, pour prédire la mémoire à adresser, on pourra considérer les mémoires adressées par les trois dernières utilisations du pointeur concerné par l'instruction courante et choisir la mémoire qui a été adressée le plus souvent par ce pointeur. On pourra aussi, pour prédire la mémoire à adresser, prendre en compte non seulement la ou les mémoires adressées par le pointeur concerné, mais aussi la mémoire adressée lors de l'instruction précédente, même si cette instruction ne concerne pas le même pointeur.

30 La présente invention peut être mise en oeuvre par tout moyen matériel approprié, les moyens matériels décrits ne l'ayant été qu'à titre d'exemple.

REVENDICATIONS

1. Procédé pour adresser en lecture un emplacement d'une parmi plusieurs mémoires à l'aide d'une adresse codée provenant d'une instruction, comprenant les étapes suivantes :

5 a) prédire (104), parmi les plusieurs mémoires, la mémoire correspondant à l'emplacement à adresser,

b) décoder (108) l'adresse de l'emplacement à adresser et déterminer (109) la mémoire à adresser,

10 c) à l'issue de l'étape a), gérer (105) un éventuel conflit de lecture et de réécriture en supposant que la mémoire prédite est la mémoire à adresser,

d) commander (111) l'adressage de la mémoire prédite à l'issue de l'étape de gestion c),

e) à l'issue de l'étape b), déterminer (110) si la mémoire à adresser correspond à la mémoire prédite, et

15 f) si la mémoire à adresser ne correspond pas à la mémoire prédite, gérer (115) un éventuel conflit de lecture et de réécriture dans la mémoire à adresser et adresser l'emplacement de la mémoire à adresser.

2. Procédé selon la revendication 1, dans lequel 20 l'étape a) utilise une table (30) contenant une historique des mémoires adressées lors des instructions précédentes.

3. Procédé selon la revendication 2, dans lequel la table (30) renferme un nombre de registres (31) égal au nombre de 25 registres étant affecté à un pointeur particulier, et dans lequel chacun desdits registres contient une information permettant de déterminer laquelle des mémoires a été adressée par ledit pointeur lors de sa dernière utilisation.

30 4. Procédé selon l'une quelconque des revendications 1 à 3, dans lequel l'étape de décodage et de détermination b) débute en même temps que l'étape de prédiction a).

5. Procédé selon l'une quelconque des revendications 1 à 4, dans lequel, en cas de conflit de lecture et réécriture dans une même mémoire, la priorité est donnée à la réécriture (106).

6. Procédé selon l'une quelconque des revendications 1 à 5, dans lequel une étape (107) d'examen de contraintes temporelles est réalisée après la gestion de conflit de lecture et de réécriture des étapes c) ou f), afin d'examiner si la lecture de la mémoire à lire peut être effectuée au cours du cycle d'horloge suivant.

7. Procédé selon l'une quelconque des revendications 1 à 6, dans lequel les adresses codées appartiennent à un des types suivants : adresse courte, adresse longue ou adresse indirecte.

10 8. Procédé selon la revendication 7, comprenant une étape (101) de détermination de type d'adresse et dans lequel les étapes de prédiction a), de gestion c), d'adressage d) et de comparaison e) ne sont réalisées que si l'adresse codée de l'emplacement à adresser est une adresse indirecte.

15 9. Procédé selon les revendications 2 et 8, dans lequel la table (30) est invalidée (122) si l'adresse codée de l'emplacement à adresser n'est pas une adresse indirecte.

20 10. Procédé selon l'une quelconque des revendications 1 à 9, réalisé par un dispositif de type microcontrôleur ou microprocesseur, et dans lequel les mémoires pouvant être adressées sont une mémoire de type SRAM interne audit dispositif, une mémoire de type DPRAM interne audit dispositif et une mémoire externe audit dispositif.

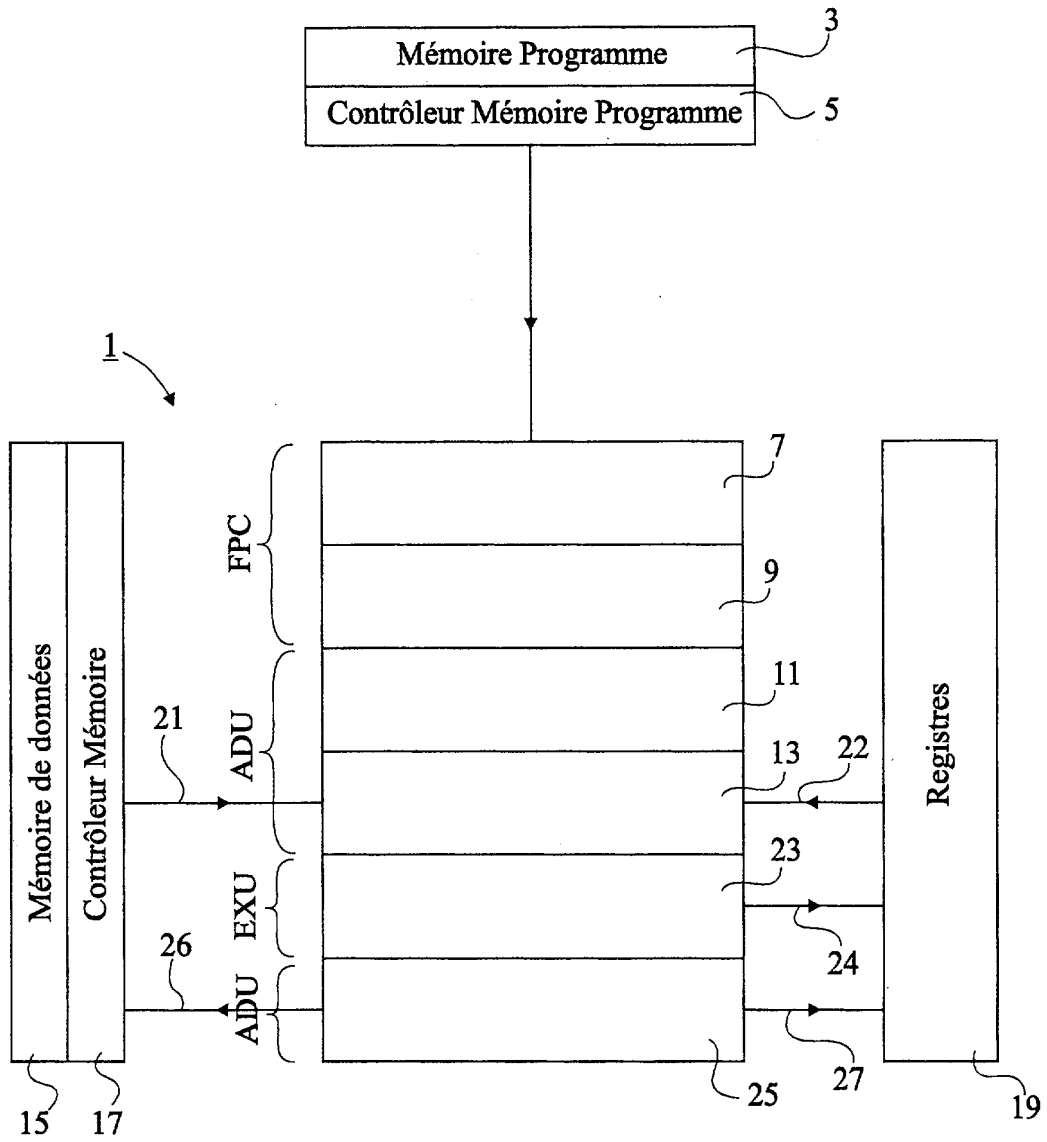


Fig 1

Fig 2A

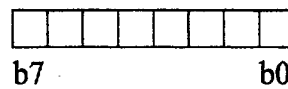


Fig 2B

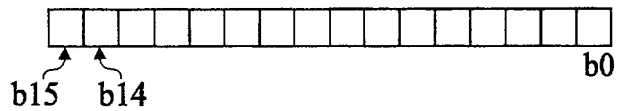


Fig 2C1

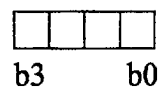


Fig 2C2

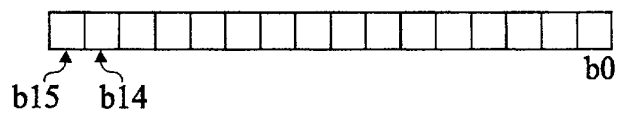




Fig 3

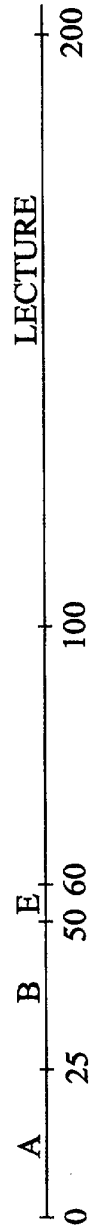


Fig 4A

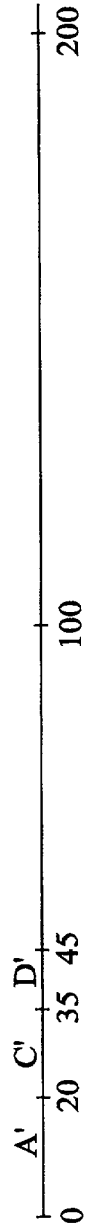


Fig 4B

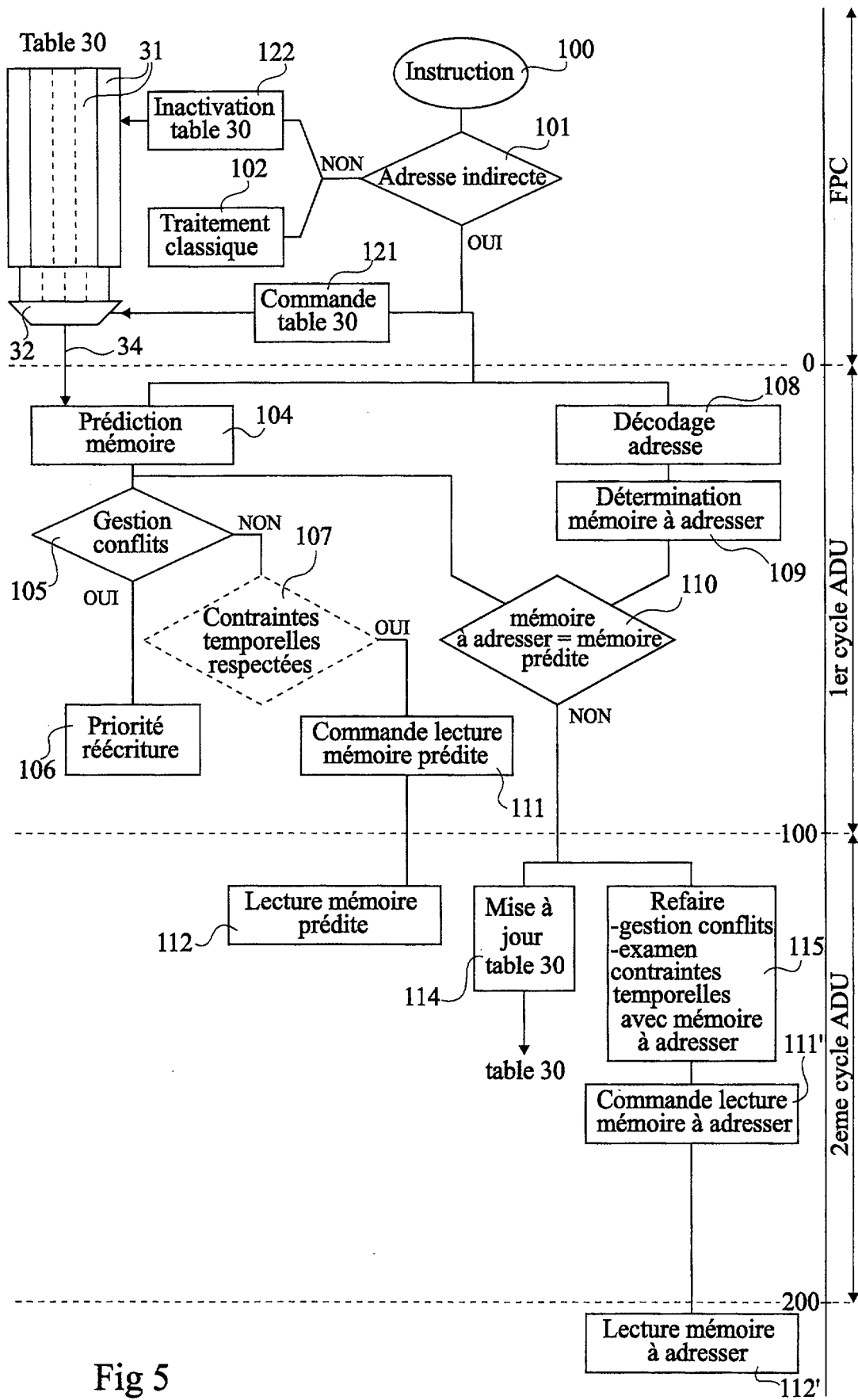


Fig 5



2823874

**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

N° d'enregistrement
national

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

FA 604250
FR 0105405

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
A	EP 0 332 910 A (BULL HN INFORMATION SYST) 20 septembre 1989 (1989-09-20) * le document en entier *	1	G06F12/02 G11C11/413
A	US 5 710 914 A (FETTWEIS GERHARD ET AL) 20 janvier 1998 (1998-01-20)		
			DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7)
			G06F
		Date d'achèvement de la recherche	Examineur
		15 janvier 2002	Daskalakis, T
CATÉGORIE DES DOCUMENTS CITÉS		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	
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire			

1

EPO FORM 1503 12.99 (P04C14)

**ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE
RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0105405 FA 604250**

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 15-01-2002
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
EP 0332910 A	20-09-1989	IT 1216087 B	22-02-1990
		DE 68916563 D1	11-08-1994
		DE 68916563 T2	22-12-1994
		EP 0332910 A2	20-09-1989
		US 5060188 A	22-10-1991

US 5710914 A	20-01-1998	AUCUN	
