

A1

**DEMANDE
DE BREVET D'INVENTION**

②

N° 82 15793

-
- ⑤④ Microprocesseur et procédé pour imbriquer les accès en mémoire de ce microprocesseur.
- ⑤① Classification internationale (Int. Cl.³). G 06 F 13/00.
- ②② Date de dépôt..... 20 septembre 1982.
- ③③ ③② ③① Priorité revendiquée : US, 21 septembre 1981, n° 06/304 017.
- ④① Date de la mise à la disposition du
public de la demande..... B.O.P.I. — « Listes » n° 12 du 25-3-1983.

⑦① Déposant : Société dite : RACAL DATA COMMUNICATIONS, INC. — US.

⑦② Invention de : Philip Frederick Kromer III.

⑦③ Titulaire : *Idem* ⑦①

⑦④ Mandataire : Rinuy et Santarelli,
14, av. de la Grande-Armée, 75017 Paris.

L'invention concerne les microprocesseurs, et plus particulièrement des microprocesseurs utilisant des instructions qui ne nécessitent pas d'extraction en mémoire externe et des instructions qui nécessitent une extraction en mémoire externe. Les instructions ne nécessitant pas d'extraction en mémoire externe seront désignées ci-après instructions de type I tandis que celles nécessitant l'extraction de données dans une mémoire externe seront désignées ci-après instructions du type II. En général, les instructions du type II sont constituées de deux parties : une partie étant la règle Opérande-extraction, l'autre étant la règle Opération. Exprimée par un verbe, une instruction de type I peut être considérée comme, par exemple, "prendre le complément de l'accumulateur", tandis qu'une instruction du type II peut être considérée, par exemple, comme "extraire le contenu de la mémoire désignée par un registre d'index R_3 et ajouter ce nombre à l'accumulateur".

Une mesure très importante de la performance d'un microprocesseur, en particulier dans le cas de processeurs de signaux, est le nombre d'opérations qu'il peut exécuter par seconde. Il est donc souhaitable d'exploiter au maximum chaque cycle d'horloge. Une technique bien connue pour accomplir une telle exploitation est dite technique du "pipeline" dans laquelle une ou plusieurs instructions postérieures sont extraites avant l'achèvement de l'exécution de la première instruction.

Il est également souhaitable d'utiliser peu de bus dans l'architecture d'un microprocesseur afin de réduire le nombre de broches d'entrée/sortie, de simplifier la mémoire externe et de permettre une architecture "Von Neuman" dans laquelle les mémoires d'instruction et de données peuvent partager les mêmes dispositifs.

L'invention s'applique particulièrement à un microprocesseur utilisant des instructions de type I et de type II, la technique du pipeline et deux bus, à savoir un bus d'adresse et un bus d'instructions/données. Dans une telle machine, les différents types d'instructions et de retards présents soulèvent des difficultés pour l'organisation des

opérations de traitement et, par conséquent, un gaspillage du temps précieux de calcul, limitant le nombre d'opérations par seconde qui pourraient être autrement effectuées.

L'invention a pour objet d'améliorer la puissance de calcul des microprocesseurs. L'invention a également pour objet de synchroniser le déroulement des opérations dans une machine utilisant plusieurs types différents d'instructions afin de permettre une utilisation optimale des cycles du microprocesseur. L'invention a plus particulièrement pour objet d'améliorer l'efficacité d'un microprocesseur utilisant une architecture à pipeline, deux bus et deux types d'instructions.

Ces objectifs et avantages, ainsi que d'autres objectifs, et avantages, sont obtenus, conformément à l'invention, par la mise en oeuvre de moyens destinés à tamponner des adresses de données, des moyens de retard d'instructions et un contrôleur destiné à commander ces éléments afin d'effectuer un traitement optimal.

L'invention apporte plusieurs avantages. Tout d'abord, des instructions et des données peuvent être imbriquées sur le même bus. Au cours de chaque cycle, une extraction utile en mémoire est effectuée. Une souplesse maximale est permise pour le codage des instructions. Le rythme d'exécution des instructions est presque aussi uniforme que possible.

L'invention sera décrite plus en détail en regard des dessins annexés à titre d'exemple nullement limitatif et sur lesquels :

la figure 1 est un schéma simplifié d'une forme préférée de réalisation de l'invention ;

la figure 2A montre l'imbrication des instructions et des données, de leurs adresses et de la synchronisation requise de leur apparition sur le bus d'adresses et les bus de données d'instructions de la forme préférée de réalisation ;

la figure 2B montre l'action du signal IRS-1,5 qui sert d'arbitre de bus ;

la figure 2C montre la position dans le temps des signaux d'entrée du décodeur par rapport à d'autres opérations montrées sur les figures 2A à 2E ;

la figure 2D montre la position dans le temps des signaux de sortie du décodeur et du chargement du tampon DAB de bus d'adresses de données par rapport aux autres opérations illustrées sur les figures 2A à 2E ;

5 la figure 2E montre la position dans le temps et le positionnement des contenus du registre d'instruction, du registre d'exécution et du registre X ainsi que le signal de commande IRSO ;

10 les figures 3 et 4 représentent un diagramme d'états et une table de vérité illustrant le fonctionnement de l'élément de mise en séquence d'états de la forme préférée de réalisation de l'invention ; et

la figure 5 est un schéma du circuit logique de l'élément de mise en séquence d'états.

15 La figure 1 représente la forme préférée de réalisation de la structure du microprocesseur selon l'invention. Comme représenté, ce microprocesseur utilise deux bus, à savoir un bus 11 d'adresse et un bus 13 d'instructions/données. Les délais associés en pratique au décodeur et à la logique combinatoire associée du microprocesseur imposent 1,5 cycle pour l'interprétation de la règle d'extraction d'opérande et la génération de l'adresse de l'opérande. 1,5 cycle est également nécessaire à la mémoire pour renvoyer la donnée, l'adresse étant donnée. Le temps
20 total, égal à 3 cycles, correspond à la famille préférée des circuits logiques, à savoir les circuits intégrés à grande échelle du type NMOS c'est-à-dire MOS à canal N, et il constitue le "délai de pipeline" de la forme préférée de réalisation. Ce type de logique impose également la
25 fréquence maximale d'horloge à laquelle travaille le système.

30 La figure 1 représente un ensemble 15 à empilage et compteur d'instruction qui produit des adresses de programme sur un bus d'adresses de programme (PAB) 12 transmises par l'intermédiaire d'un sélecteur 14 à un registre 17 d'adresse, qui adresse des dispositifs de mémoire 19.

35 En réponse à une adresse provenant du compteur 15 d'instruction, le dispositif de mémoire délivre une instruction de type I ou de type II à un registre 23 de sortie de

mémoire par l'intermédiaire du bus instructions/données 13, cette instruction étant transmise à un registre d'entrée 25. Le registre d'adresse est l'un des cinq registres "d'encadrement" 17, 23, 36, 39, 25 montrés sur la figure 1. Ces "registres" n'emmagent pas, mais établissent plutôt un retard ou délai approprié, et mettent en forme et effilent les signaux transférés à travers eux.

L'instruction est appliquée à un décodeur 29 d'instruction d'adresse et elle est introduite dans un registre 27 d'instruction. La ligne 30 de commande aboutissant au compteur d'instruction commande des sauts, des appels de sous-programmes et des retours.

Le décodeur 24 d'instruction d'adresse décode la partie d'extraction d'opérande de l'instruction et applique la sortie décodée par la ligne 31 de commande d'adresse de données à un générateur 33 d'adresse de données. Le générateur 33 délivre une adresse de données au bus d'adresse de données (DAB) 10 qui la transmet à un tampon élastique 35 pouvant emmagasiner une adresse de données pour qu'elle soit ensuite appliquée au dispositif de mémoire 19 par l'intermédiaire du sélecteur 14 et du registre d'adresse 17. Le générateur d'adresse de donnée utilise de préférence la technique d'adressage indirect connue de l'homme de l'art. La forme de réalisation préférée utilise quatre registres d'index pouvant chacun générer une adresse. Le décodeur 29 sélectionne le registre d'index approprié. En cours de fonctionnement, l'instruction considérée peut contenir un ordre de lecture d'un registre d'index et d'une zone de déplacement. L'adresse se trouvant dans le registre d'index indiqué est ensuite appliquée sur le bus d'adresse et son contenu est en même temps incrémenté ou décrétementé suivant le contenu de la zone de déplacement.

Certaines règles d'extraction d'opérande peuvent nécessiter l'exécution d'une opération d'écriture en mémoire. Dans ce cas, le signal 60 d'écriture, transmis par le registre d'adresse 17, conditionne la mémoire pour écrire la donnée présente dans un registre de cumul 43, par l'intermédiaire d'un registre 62. Cette même donnée

apparaîtra sur le bus d'instructions/données de la même manière que le résultat d'un cycle de lecture.

D'autres techniques de génération d'adresses sont connues et utilisables et l'invention n'est pas limitée à
5 une technique particulière.

L'instruction est également placée dans un registre 27 d'instruction pour être finalement exécutée par le micro-processeur.

L'exécution est réalisée par des circuits classiques,
10 comprenant un registre d'exécution 36, un décodeur d'opération 37, un registre 39, une unité logique arithmétique 41, un accumulateur 43 et un registre de données 45. Le décodeur d'opérations décode l'instruction et transmet le signal de sortie décodé au registre d'exécution. Le registre
15 45 de données mémorise un mot de donnée demandé pour l'exécution de l'ordre contenu dans le registre d'exécution 36. Le chargement du registre de données 45 est commandé par un signal IRSO 53. Les autres données d'entrée de l'unité
20 logique arithmétique 41, arrivant par la ligne 47, proviennent de la sortie de l'accumulateur 43.

La section arithmétique peut utiliser plus d'un accumulateur, et de préférence elle en utilise deux. Ainsi, l'entrée "A" de l'unité logique arithmétique 41 peut pro-
25 venir du premier ou du second accumulateur. Il est de préférence prévu une instruction permettant de renvoyer dans la mémoire 19 le signal de sortie de l'un ou l'autre des accumulateurs. L'unité logique arithmétique 41 établit elle-même, de façon typique, plusieurs unités de retard
30 du pipeline. Comme indiqué ci-après, dans la forme préférée de réalisation, les données sont minutées pour arriver à l'entrée B de l'unité logique arithmétique 41 en même temps que l'arrivée de l'ordre approprié contenu dans le registre 39.

Selon l'invention, une imbrication optimale des
35 adresses d'instructions et des adresses de données sur le bus d'adresses 11 et des instructions et des données sur le bus instructions/données est réalisée au moyen d'un générateur DOB (données sur bus) 45. Comme représenté, le

générateur DOB 45 reçoit une entrée du décodeur 29 d'instruction d'adresse. Cette entrée est constituée du signal DAR (demande d'adresse de donnée) qui apparaît en même temps que la production d'une adresse de données par le générateur d'adresses de données.

Le signal DAR indique si l'instruction décodée entraîne une extraction de donnée ou non.

Le générateur DOB 45 produit des signaux de commande sur cinq lignes 49, 51, 52, 53. Lorsque le signal de commande IRSO présent sur la ligne 53 est vrai, le registre d'instruction est décalé. Lorsque le signal de commande IRSO est faux, le registre X 45 est chargé. Le signal de commande présent sur la ligne 49 détermine lorsque le tampon 38 du bus d'adresse de données (DAB) doit être chargé du signal DAB présent sur la ligne 10. Le signal de commande de sélection de bus d'adresse de programme, présent sur la ligne 52 détermine si l'adresse de programme provenant du bus d'adresse de programme (PAB) 15 est transmis au registre d'adresse. Le signal de commande de sélection de bus d'adresse de programme présent sur la ligne 50, détermine si le signal de sortie du générateur d'adresse de données est transmis au registre 17 d'adresse. Le signal de commande de sélection de tampon du bus d'adresse de données, présent sur la ligne 51, détermine si le signal de sortie du tampon 38 du bus d'adresse de données est transmis directement au registre d'adresse 17. Ainsi, les signaux de commande 50 et 51 assurent "l'élasticité" du tampon. De plus, lorsque le générateur DOB émet une adresse de donnée, il inhibe le compteur d'instruction 15.

Le générateur DOB a pour fonctions de générer les créneaux de micro-synchronisation (signal IRS), de servir d'arbitre de bus pour les bus DAB, de tampon DAB et PAB et d'incrémenter le compteur d'instruction.

La figure 3 représente une forme préférée de réalisation du générateur DOB. La figure 4 montre la table d'états pour tous les états possibles et les conditions demandées pour un mouvement d'un état à l'autre.

Le générateur DOB génère les créneaux de synchroni-

sation d'instruction et de données (microsynchronisation) pour le microprocesseur ainsi que pour les circuits de l'unité logique arithmétique. La figure 4 montre la table d'états pour tous les états possibles et les conditions demandées pour passer d'un état à l'autre. Comme mentionné
 5 précédemment, les instructions peuvent être classées en deux types :

A. celles qui extraient un opérande ; c'est à dire créent un créneau de données (type II)

10 B. celles qui n'extraient pas un opérande ; c'est-à-dire qui ne créent pas un créneau de données (type I).

Les règles pour imbriquer les créneaux de données et les extractions d'instructions sont :

1. le créneau de données pour une instruction N, si
 15 nécessaire, est créé immédiatement après l'extraction de l'instruction N + 2.

2. Une extraction d'instruction suit immédiatement chaque créneau de données.

Le générateur DOB sélectionne soit une adresse de
 20 donnée, soit une adresse d'instruction. Le résultat de l'extraction peut être utilisable comme signal d'entrée pour le générateur DOB, en provenance du décodeur principal, trois cycles de bus plus tard. Ceci est dû à la structure en pipeline (délais du matériel) du système.

25 Le décodeur d'instruction d'adresse génère un signal appelé DAR (demande d'adresse de donnée). En réponse à une instruction du type II, ce signal est vrai, et en réponse à une instruction du type I, il est faux.

30 Dans la forme préférée de réalisation, il existe un repositionnement à déclenchement extérieur (par exemple en réponse à la ligne sous tension).

Le cycle de repositionnement laisse le générateur DOB dans l'état 14. Il reste dans cet état 14 tant qu'aucune instruction de type II n'est décodée (DAR = 0).

35 Pendant cet état, le compteur P.C. est incrémenté, son contenu est dirigé vers le bus d'adresse et le signal IRS-1,5 est vrai, indiquant une instruction sur le bus de donnée. Le signal IRS est très important, car il établit les

créneaux d'instruction et de donnée pour les autres circuits du microprocesseur. Si dans l'état 14, $DAR = 1$, le bus DAB est validé et son contenu est transféré directement au bus d'adresse ; le compteur P.C. est inhibé de manière à ne pas effectuer d'incréméntation. Ce signal IRS passe au niveau bas et le générateur passe à l'état 6.

Dans l'état 6, le compteur P.C. est incrémenté. Si $DAR = 0$, le contenu du compteur P.C. est placé sur le bus d'adresse. Le signal IRS est haut et le générateur passe à l'état 10. Si $DAR = 1$, le contenu du compteur P.C. est placé sur le bus d'adresse, IRS est haut, le tampon DAB 38 est chargé du contenu du bus DAB et le générateur passe à l'état 11.

Dans l'état 10, le compteur P.C. est incrémenté. Si $D = 0$, le contenu du compteur P.C. est placé sur le bus d'adresse, IRS est haut et le générateur passe à l'état 12. Si $DAR = 1$, le contenu du compteur P.C. est placé sur le bus d'adresse, et IRS est haut, le tampon DAB 38 est chargé du contenu du bus DAB, et le générateur passe à l'état 13.

Dans l'état 11, le compteur P.C. est inhibé de manière à ne pas pouvoir compter. Si $D = 0$, le contenu du tampon DAB 38 est placé sur le bus d'adresse, IRS reste haut et le générateur DOB passe à l'état 4. Si $DAR = 1$, le contenu du tampon DAB 38 est placé sur le bus d'adresse, cette opération étant suivie du chargement du registre par une nouvelle information présente sur le bus DAB. Le signal IRS est haut et le générateur DOB passe à l'état 5.

Pendant les états 4, 5, 12 et 13 le décodeur 29 est inactif, car les données sont sur le bus I/D 13, du fait que IRSO est bas.

Dans l'état 4, le compteur P.C. est incrémenté et son contenu est placé sur le bus d'adresse. IRS est bas, indiquant qu'une donnée est présente sur le bus de données. Le générateur DOB passe à l'état 10.

Dans l'état 5, le compteur P.C. est incrémenté et

son contenu est placé sur le bus d'adresse. Le tampon DAB 38 est chargé du contenu du bus DAB. IRS est bas et le générateur DOB passe à l'état 11.

5 Dans l'état 13, le compteur P.C. est inhibé de manière à ne pas pouvoir effectuer de comptage. Le contenu du tampon DAB 38 est placé sur le bus d'adresse. IRS est bas et le générateur DOB passe à l'état 6.

10 A l'état 12, le compteur P.C. est incrémenté et son contenu est placé sur le bus d'adresse. IRS est bas et le générateur DOB passe à l'état 14.

15 Par conséquent, les fonctions du générateur DOB sont de générer des créneaux de micro-synchronisation (signal IRS), de servir d'arbitre de bus pour les bus DAB, de tampon DAB et PAB, et d'incrémenter le compteur d'instruction.

La figure 5 représente le circuit logique du générateur DOB. Ce générateur comprend un certain nombre d'inverseurs 55, 56, 57, 58 ; un certain nombre de portes ET 20 60, 61, 62, 63, 64 ; un certain nombre de portes NON-OU 67, 68, 69, 70 ; et quatre bascules 73, 74, 75, 76 de type D, interconnectés comme représenté. L'entrée du générateur est le signal DAR et les signaux de sortie produits comprennent le signal IRS-1,5 et des signaux de commande CHARGE TAMPON DAB ; SELECT TAMPON DAB ; SELECT PAB/
25 INCREMENT PC ; et SELECT DAB. La fonction de ce circuit est illustré par la table de vérité du générateur DOB, montrée sur la figure 4, et par le diagramme de la séquence d'états de la figure 3.

30 Un exemple d'opération sera à présent décrit en regard des figures 2A à 2E et du tableau I annexé. On suppose que l'historique de la machine est tel que trois instructions d'une rangée seront extraites de la mémoire, chacune demandant une extraction de donnée. Le tableau I
35 ci-après montre un segment de programme qui exige une telle série d'instructions.

TABLEAU I

ADRESSE		INSTRUCTION			
0	RRI	+1	< MVP	AD >	
	j1				
5	1	RRI	-1	< ADD	AO >
	j1				
	2	RR3	0	< MVP	A1 >
	j1				
	3	SAUT			
10	4	PCEQ	80	< ADD	A1 >
	5	WRI	0	< ADD	A1 >
	j1				
	6	N			
	7				
15	80	L	< NOUVEAU >	< SUB	A1 >
	81	N			
	82	← IMMATERIEL →			
	83	← IMMATERIEL →			

20 j1 indique qu'un créneau de donnée est associé à l'instruction précédente. Il est ignoré par l'assembleur.

Sur les figures 2A à 2E, I_0 , I_1 , I_2 exigent chacun un créneau de donnée. I_0 est décodé aussi rapidement que possible et l'adresse AD_0 pour le créneau de donnée demandé par I_0 est placé sur DAB, puis sur le bus d'adresse 11 aussi rapidement que possible. Il convient de noter que l'association du retard de la mémoire et du retard minimal du décodeur exige trois cycles d'horloge entre l'instant zéro, auquel l'adresse de I_0 est placée sur le bus d'adresse, et 25 l'instant AD_0 placé sur le bus. I_1 atteint le bus I/D à proximité de I_0 mais conformément à la règle opératoire DOB, l'adresse "3" pour I_3 est placée sur le bus d'adresse avant la mise en place de ADI , à savoir l'adresse de la donnée pour I_1 sur le bus d'adresse. Ceci a pour résultat la présence de quatre 30 unités de retard entre la mise en place de I_1 sur le bus I/D 35

et la donnée D_1 de I_1 sur le bus I/D. De façon similaire, la donnée D_2 de I_2 ne passe sur le bus I/D qu'après l'extraction de I_4 et il en résulte cinq unités de retard entre le moment où I_2 est placé sur le bus I/D et le moment où D_2 est placé sur le même bus I/D.

Entre-temps, les instructions I_0 , I_1 , I_2 pénètrent dans le registre d'instruction, sous la commande du signal IRSO (comme montré sur la figure 2E). Lorsqu'un créneau de donnée est généré, IRSO passe au niveau bas, de sorte qu'il ne se produit aucun décalage lorsque le créneau de donnée apparaît sur le bus I/D. Ainsi, I_0 atteint, par l'intermédiaire du registre d'instruction, le registre d'exécution un demi cycle d'horloge avant l'introduction de la donnée D_0 dans le registre X. Ceci apparaît sur les lignes REG EXEC et REG X de la figure 2E. Ensuite, le signal de sortie du décodeur $C(I_0)$ est appliqué à l'unité logique arithmétique en même temps que D_0 par l'intermédiaire du registre 39 d'encadrement.

Les figures 2D et 2A montrent l'action du tampon élastique conjointement avec celle du générateur DOB. Ceci apparaît notamment en comparant le signal de sortie du bus DAB et le bus I/D. AD_0 apparaît sur le bus DAB un cycle après l'apparition de I_0 sur le bus I/D. AD_1 apparaît sur le bus DAB un cycle d'horloge après l'apparition de I_1 sur le bus I/D et est tamponné par le tampon élastique dans son tampon d'adresse de donnée 38. En même temps que ce tamponnement, l'adresse de l'instruction "3" est placée sur le bus d'adresse. L'action du tampon apparaît, car AD_1 apparaît sur le bus d'adresse seulement après l'arrivée de la valeur "3" sur le bus d'adresse. Ainsi, AD_1 est avancé pas à pas de DAB au tampon DAB jusqu'au bus d'adresse.

AD_2 est retardé dans le tampon DAB 38 (figure 2D) plus longtemps pour permettre à la fois à AD_1 et à "4" d'apparaître sur le bus d'adresse. AD_2 peut rester pendant deux cycles, en raison de l'arrivée de D_0 . Autrement dit, l'arrivée de D_0 invalide le décodeur par l'action de IRSO. Par conséquent, aucune nouvelle adresse de donnée ne peut

être générée.

Le tableau I montre une série d'instructions résultant des combinaisons montrées sur la figure 2. La première instruction I_0 est une instruction RR1 "lecture et registre 1" qui demande une extraction de donnée, combinée avec une opération MVP ("mouvement positif vers A_0 "). I_1 est une instruction RR1-1 et addition à A_0 (il convient de noter que I_0 comprend un + 1 qui est un incrément pour le registre d'index afin que l'extraction suivante à ce registre provienne de la position de mémoire adjacente suivante). I_2 est une instruction RR3 "lecture registre 3" et "mouvement positif vers A_1 ". I_3 est un saut et I_4 donne la destination du saut. I_5 est une instruction "écriture au registre 1". Etant donné que I_1 contient un décrétement, on devrait renvoyer R_1 du registre 1 à la première position.

En raison des retards présents dans la machine, le compteur PC n'est pas affecté par le saut avant que I_4 , c'est-à-dire la destination du saut, apparaisse sur le bus I/D, de sorte que I_5 reste extrait à partir de l'ancienne valeur de PC. Lorsque I_4 apparaît sur le bus I/D, le compteur PC est modifié afin que l'extraction suivante s'effectue pour I_{80} à partir de la position 80. Le "saut" sur les instructions "PC égale" aux adresses 3 et 4 n'exige pas de créneau de donnée. Par conséquent, sous l'action du générateur DOB, un flot de trois instructions formant une rangée sans créneau de donnée intermédiaire apparaît après l'adresse AD_2 . L'instruction "écriture au registre 1", à savoir l'instruction I_5 , a pour résultat l'insertion de AD_5 dans trois créneaux après l'arrivée de la valeur "5" sur le bus d'adresse (voir figures 2A- 2E). En raison de l'action du générateur DOB 45, des instructions sont travaillées aussi uniformément que possible et, de plus, D_2 est disponible pendant plusieurs créneaux.

L'unité logique arithmétique peut effectuer des opérations dépassant un cycle unique. Le signal IRSO et la nature de l'instruction peuvent être utilisés pour inhi-

ber une seconde exécution de IR3. Il est possible d'obtenir de l'élasticité en utilisant le retard du registre I.

Le tableau II montre à titre illustratif des formats d'instructions. La partie supérieure de ce tableau montre une instruction de type II. L'instruction représentée est une instruction de lecture indirecte. Le premier segment de l'instruction indique le type d'extraction de donnée. Le deuxième segment indique le déplacement à ajouter aux contacts du registre d'index. Le troisième segment donne la désignation du registre. Ces trois segments constituent la règle extraction opérande. Le quatrième segment donne l'opération à exécuter.

La partie inférieure du tableau II montre à titre illustratif une instruction de type I. Le premier segment d'information binaire indique qu'un contenu de registre particulier doit être utilisé comme opérande. Il convient de noter qu'une extraction de registre n'exige pas un cycle de bus. Le deuxième segment indique l'identité du registre et le troisième segment indique une opération à exécuter. Par exemple, l'instruction demandée peut être "prendre le contenu d'un registre et l'additionner au contenu de l'accumulateur". Un avantage de l'architecture particulière de l'invention est la souplesse d'utilisation des positions de bits des instructions, c'est-à-dire l'absence des critères rigides imposés aux formats des instructions.

TABLEAU II

INSTRUCTION DE TYPE II - LECTURE INDIRECTE ET REGISTRE D'INDEX

30

OPERATION LECTURE INDIRECTE	DEPLACEMENT A ADDITIONNER AU REG. INDEX	DESIGNATION REGISTRE	< MVP AO > (OPERATION)
-----------------------------------	---	-------------------------	---------------------------

35

INSTRUCTION DE TYPE I - OPERATION LOCALE

OPERATION LOCALE	REGISTRE A SELECT. INTERNE	< ADD. AO > (OPERATION)
------------------	-------------------------------	----------------------------

Il va de soi que de nombreuses modifications peuvent être apportées au microprocesseur décrit et représenté sans sortir du cadre de l'invention.

REVENDICATIONS

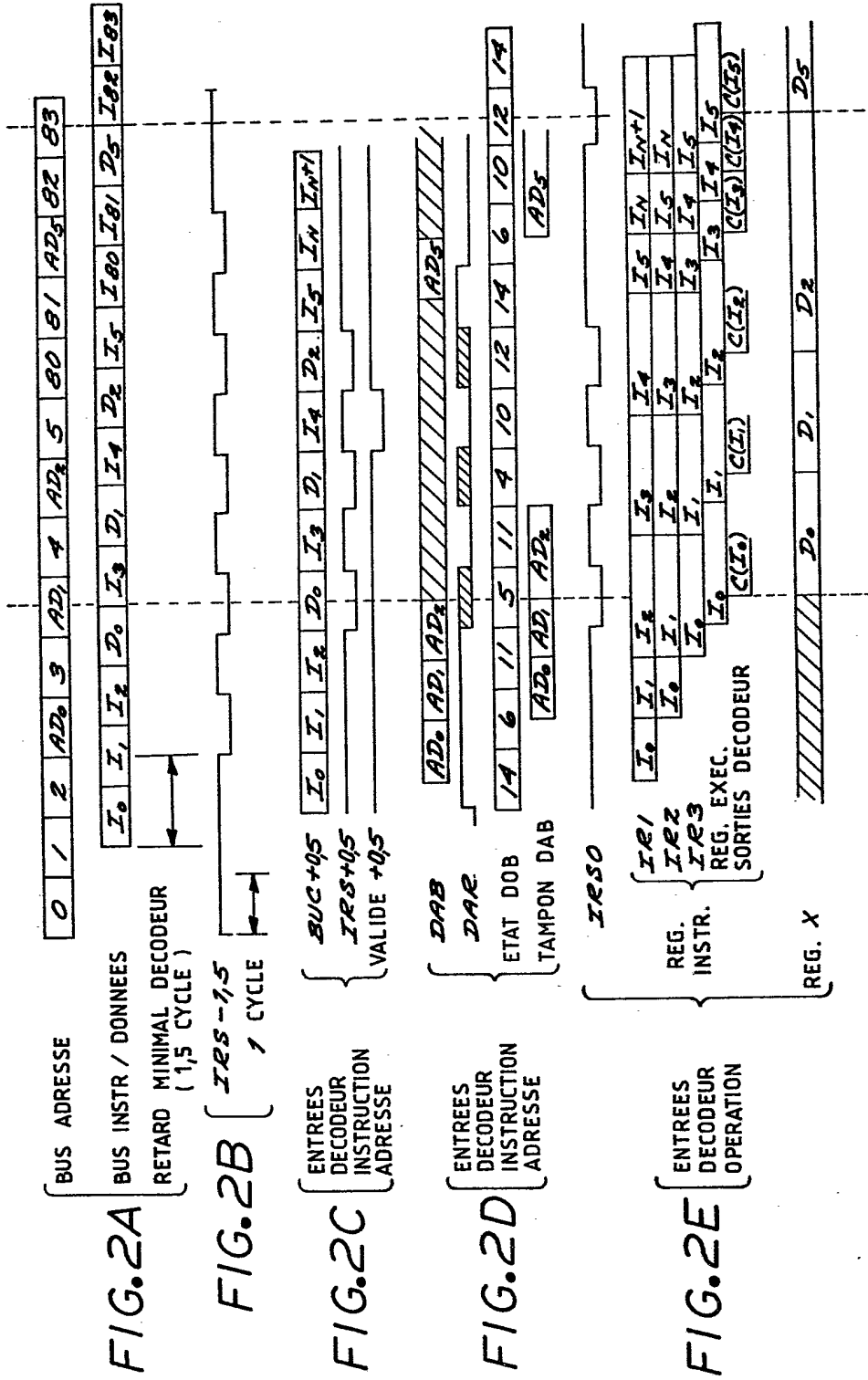
1. Microprocesseur comprenant une mémoire (19) qui peut être adressée par un compteur (15) d'instruction ou une adresse de donnée et qui réagit à ce compteur ou à
5 cette adresse pour produire, respectivement, une instruction ou un mot de données, cette instruction contenant une information indiquant une opération à exécuter par une unité arithmétique (41), les données pouvant être travaillées par cette unité arithmétique, le processeur étant
10 caractérisé en ce qu'il comporte des moyens (45) destinés à imbriquer des adresses d'instruction et des adresses de données présentées à la mémoire et à imbriquer des instructions et des données sortant de la mémoire afin qu'il n'apparaisse aucun créneau de synchronisation perdu et que
15 des ordres et des données correspondantes arrivent à l'instant d'exécution à l'unité logique arithmétique.

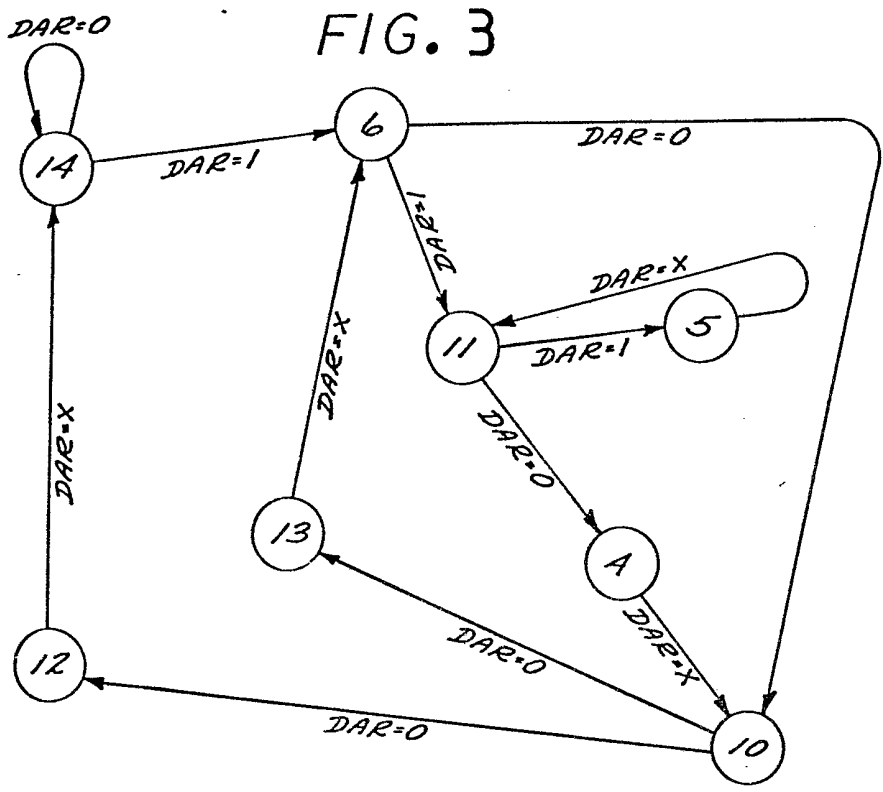
2. Procédé pour imbriquer l'accès à la mémoire d'un microprocesseur, caractérisé en ce qu'il consiste à créer un créneau de synchronisation pour une instruction N, si
20 cela est nécessaire, immédiatement après l'extraction d'une instruction N + K, K étant déterminé par le retard du pipeline, et à extraire une instruction immédiatement après chaque créneau de donnée.

3. Microprocesseur caractérisé en ce qu'il comporte un élément (35) destiné à tamponner des adresses de données vers une mémoire (19), des moyens destinés à mémoriser plusieurs instructions sortant de ladite mémoire et des moyens (45) destinés à commander l'élément de tamponnement et des moyens de mémorisation, ainsi que l'application d'adresses
30 d'instructions et d'adresses de données à ladite mémoire.

4. Microprocesseur caractérisé en ce qu'il comporte une mémoire (19) emmagasinant des données et des instructions, un compteur (15) d'instruction destiné à transmettre des adresses d'instruction à ladite mémoire, des moyens
35 destinés à transmettre des adresses de mémoire à ladite mémoire, des moyens (41) sensibles aux instructions et qui effectuent des opérations arithmétiques sur les données en réponse à une instruction, des moyens destinés à emmagasiner

plusieurs instructions pour les transmettre aux moyens effectuant des opérations arithmétiques, et des moyens (45) destinés à commander si une adresse de donnée ou une adresse d'instruction est appliquée à la mémoire et à commander des moyens d'emmagasinage d'instructions pour synchroniser l'arrivée d'instructions avec les données appropriées aux moyens effectuant une opération arithmétique.





ETAT	ETAT					DAR	ETAT SUIVANT	ETAT SUIVANT					IRS-1/5	SELECT PAB	SELECT DAB	SELECT TAMPON DAB	CHARGE TAMPON DAB
	Q ₁	Q ₂	Q ₃	Q ₄				Q ₁	Q ₂	Q ₃	Q ₄						
4	0	1	0	0	X	10	1	0	1	0	0	1	0	0	0	0	
5	0	1	0	1	X	11	1	0	1	1	0	1	0	0	0	0	
6	0	1	1	0	0	10	1	0	1	0	0	1	0	0	0	0	
						11	1	0	1	1	0	1	0	0	1	1	
10	1	0	1	0	0	12	1	1	0	0	1	1	0	0	0	0	
						13	1	1	0	1	1	1	0	0	1	1	
11	1	0	1	1	0	4	0	1	0	0	1	0	0	1	0	0	
						5	0	1	0	1	1	0	0	1	1	1	
12	1	1	0	0	X	14	1	1	1	0	1	1	0	0	0	0	
13	1	1	0	1	X	6	0	1	1	0	1	0	0	1	0	0	
14	1	1	1	0	0	14	1	1	1	0	1	1	0	0	0	0	
						6	0	1	1	0	1	0	1	0	1	1	

FIG. 4

FIG. 5

