

La présente invention concerne de façon générale un microcalculateur. Plus spécialement, elle concerne un procédé d'exploitation d'un microcalculateur à un cycle d'instruction perfectionné, qui peut aider à accélérer le traitement dans le microcalculateur.

Les microcalculateurs monolithiques modernes utilisent souvent une architecture en pipeline qui permet que des cycles d'extraction et des cycles d'exécution d'instruction se chevauchent. Dans le cycle d'extraction, des instructions sont extraites de la mémoire. Chaque instruction est constituée par un code opération (code OP) qui est emmagasiné dans une mémoire morte, par exemple une mémoire morte programmable (PROM), et un opérande, également emmagasiné dans une ROM, qui peut pointer une valeur variable emmagasiné dans une mémoire vive (RAM). Dans les architectures en pipeline classiques, l'extraction, ou prise en charge, du code OP et l'extraction de l'opérande sont effectuées au cours de cycles d'exécution d'instruction différents, qui sont ci-après appelés "cycles d'instruction". Par exemple, pour extraire une instruction de 2 bits contenant un code OP de 1 bit et un opérande de 1 bit dans un microcalculateur de 4 bits, il faut deux cycles d'instruction. Ainsi, dans ce cas, chaque cycle d'instruction prend trois cycles d'horloge machine. Cette architecture en pipeline classique n'est pas vraiment efficace.

En général, le cycle de prise en charge, ou extraction, servant à l'extraction de codes OP ou d'opérandes est plus court que le cycle d'instruction. Ceci signifie qu'il y a une durée du cycle d'exécution pendant laquelle aucune extraction n'est effectuée. Si la composante d'instruction suivante peut être extraite pendant cette durée, il devient possible d'abrèger le temps machine.

Ainsi, c'est un but de l'invention de proposer un microcalculateur présentant un cycle d'extraction amélioré qui abrège le cycle d'instruction.

Pour réaliser les buts ci-dessus énoncés, ainsi que d'autres buts, un microcalculateur selon l'invention comporte un décodeur d'instruction et un compteur de programme. Le décodeur d'instruction décode des instructions extraites et délivre

un signal de commande ordonnant l'exécution de l'instruction prise extraite. Le signal de commande venant du décodeur d'instruction comporte une composante qui commande les cycles d'extraction, laquelle déclenche un cycle d'extraction au début de chaque cycle d'instruction afin d'extraire l'opérande relatif à l'instruction pendant le cours de son exécution et à mi-durée de chaque cycle d'instruction afin d'extraire le code OP relatif à l'instruction suivante.

Le compteur de programme répond au déclenchement de chaque cycle d'extraction en incrémentant sa valeur de comptage de façon à maintenir cette dernière cohérente avec l'adresse à laquelle il est accédé pendant chaque cycle d'extraction.

Selon un aspect de l'invention, un procédé d'exploitation d'un microprocesseur ayant une architecture en pipeline, qui permet que les données d'instruction relatives à un cycle d'exécution d'instruction ultérieur soient extraites de la mémoire pendant un cycle d'instruction en cours, comprend les opérations suivantes :

- préprogrammer des codes d'opération d'instruction (code OP) pour des instructions demandant plus d'un byte de données d'instruction avec une information supplémentaire demandant une extraction, comprimée dans le temps, de données d'instruction ; et
- extraire plus d'un byte de données d'instruction de la mémoire pendant le cycle d'instruction faisant immédiatement suite à une demande d'extraction comprimée dans le temps.

La donnée d'instruction est constituée d'un byte de code OP et d'un byte d'opérande, et le byte du code OP est extrait immédiatement avant le début d'un cycle d'instruction, l'opération d'extraction comprenant l'opération qui consiste à extraire le byte de l'opérande associé au byte de code OP extrait au même instant que le début du cycle d'instruction. Selon une autre possibilité, la donnée d'instruction consiste en un byte de code OP et deux bytes d'opérande, et le byte du code OP est extrait immédiatement avant le début d'une première instruction, l'opération d'extraction comprenant les opérations consistant à extraire un premier byte

d'opérande associé au byte de code OP extrait au moment du début du premier cycle d'instruction et à extraire un deuxième byte d'opérande associé au premier byte d'opérande pendant le premier cycle d'instruction faisant suite à l'extraction du premier byte d'opérande.

Le procédé comprend en outre l'opération consistant à exécuter l'instruction spécifiée par les bytes du code OP et de l'opérande dans le cycle d'instruction immédiatement avant lequel le byte du code OP a été extrait.

La description, conçue à titre d'illustrations de l'invention, vise à donner une meilleure illustration de ses caractéristiques et avantages ; elle s'appuie sur les dessins annexés, parmi lesquels :

- les figures 1(A) et 1(B) sont des diagrammes temporels montrant respectivement les cycles d'instruction selon la technique antérieure et selon l'invention pour l'exécution d'instructions de 2 bytes ;

- les figures 2(A) et 2(B) sont des diagrammes temporels montrant respectivement des cycles d'instruction selon la technique antérieure et selon l'invention pour l'exécution d'instructions de 3 bytes ;

- la figure 3 est un schéma de principe simplifié d'un système de microcalculateur selon l'invention ;

- la figure 4 compare des diagrammes temporels de procédures classique et préférée d'exécution d'instruction, où (B) montre la valeur du compteur de programme pendant l'exécution d'instructions de 1 byte et (C) montre cette même valeur pour des instructions de 2 bytes ;

- les figures 5(A) et 5(B) illustrent les durées de cycles d'instructions associés à l'exécution d'instructions de 2 bytes et de 3 bytes selon les procédés de la technique antérieure et de l'invention ; et

- les figures 6(A) et 6(B) sont des illustrations de décodeurs d'instruction selon la technique antérieure et selon l'invention.

On se reporte maintenant aux dessins, en particulier aux figures 1(A) et 1(B), où les idées générales et fondamentales du procédé préféré selon l'invention sont présentées en opposition avec un procédé classique. La figure 1(A) montre un procédé typique de la technique antérieure, tandis que la figure 1(B) montre le procédé préféré selon l'invention. Les deux diagrammes s'appliquent à l'exécution d'une instruction de deux bytes qui comportent un code d'opération (code OP) de huit bits, tels que LDA, et un opérande de huit bits qui, dans ce cas, est un code d'adresse identifiant l'adresse de la RAM à laquelle on doit accéder. Dans l'exemple représenté, l'instruction ordonne le transfert ou la charge du contenu d'une adresse particulière de la RAM dans l'accumulateur.

Dans le procédé classique de la figure 1(A), le code OP LDA relatif au deuxième cycle d'instruction est extrait avant le début du premier cycle d'instruction. Dans le cycle d'instruction suivant coïncidant avec le début du premier cycle d'instruction, la donnée se trouvant à l'adresse de la RAM identifiée par l'opérande est extraite. La donnée extraite dans le cycle d'extraction commençant en même temps que le premier cycle d'instruction et le code OP LDA extrait avant le premier cycle d'instruction sont utilisés dans le deuxième cycle d'instruction.

Dans le processus classique ci-dessus présenté, un compteur de programme est incrémenté d'une unité à la fin de chaque cycle d'instruction. Ainsi, au début du cycle d'extraction du code OP, le compteur de programmes a pour valeur PC. Au début du premier cycle d'instruction, le compteur de programmes est incrémenté de 1, si bien que la valeur de comptage devient PC + 1. De la même façon, le compteur de programmes est incrémenté d'une unité au début du deuxième cycle d'instruction, si bien que la valeur de comptage devient PC + 2.

Dans le procédé selon l'invention présenté sur la figure 1(B), le code OP du premier cycle d'instruction est extrait avant le début du premier cycle d'instruction, comme dans le processus précédent. Toutefois, ce code OP de 2 bytes comporte une demande visant à comprimer le cycle d'extraction. Pendant la durée

de M_2 à M_3 du premier cycle d'instruction, la donnée se trouvant à l'adresse de la RAM identifiée par l'opérande de la RAM est extraite. La donnée d'opérande extraite pendant la durée M_2 à M_3 du premier cycle d'instruction et le code OP extrait avant le premier cycle d'instruction sont utilisés pour exécuter le premier cycle d'instruction. Dans le procédé selon l'invention, la donnée se trouvant à l'adresse de la RAM qui est extraite pendant la durée M_2 à M_3 n'est pas nécessaire à l'exécution de l'instruction du premier cycle d'instruction jusqu'à la durée M_0 à M_1 suivante.

Pendant le processus ci-dessus exposé, le compteur de programme est incrémenté d'une unité à la fin de chaque cycle d'instruction. Le compteur de programme est également incrémenté d'une unité en réponse à chaque impulsion d'horloge M_3 . En d'autres termes, à la fin du cycle d'extraction relatif au code OP, la valeur du compteur de programme est incrémenté de PC à $PC + 1$. En réponse à l'impulsion d'horloge M_3 du premier cycle d'instruction, le compteur de programme est incrémenté de 1, si bien que la valeur de comptage passe de $PC + 1$ à $PC + 2$. Ensuite, le compteur de programme est de nouveau incrémenté de 1, en réponse à une impulsion d'horloge M_1 apparaissant à la fin du premier cycle d'instruction.

Pendant la période M_0 à M_1 du premier cycle d'instruction, le code OP relatif au cycle d'instruction suivant peut être extrait.

Ainsi, comme on l'aura compris, il suffit d'un seul cycle d'instruction pour exécuter l'instruction de 2 bytes dans le processus préféré selon l'invention.

Les figures 2(A) et 2(B) montrent un autre exemple se rapportant à l'exécution d'une instruction de 3 bytes. L'exemple représenté vise un appel de sous-programme par lequel la commande de programme saute à une adresse de programme spécifiée par 2 bytes, c'est-à-dire un byte d'adresse de niveau haut P_H et un byte d'adresse de niveau bas P_L . L'instruction comporte donc un code OP de 1 byte (8 bits) CALL et les deux bytes d'opérande P_H et P_L .

Dans le processus classique de la figure 2(A), le code OP relatif à l'instruction à exécuter dans le troisième cycle

d'instruction est extrait avant le début du premier cycle d'instruction. Dans le cycle d'extraction, ou prise en charge, suivant qui a lieu en même temps que le premier cycle d'instruction, le byte d'adresse de niveau haut P_H est extrait. De la même façon, dans le
 5 deuxième cycle d'instruction, le byte d'adresse de niveau P_L est extrait. Les bytes d'adresse haut P_H et bas P_L extraits lors des cycles d'extraction ayant lieu en même temps que les premier et deuxième cycles d'instruction et le code OP CALL
 10 extrait avant le premier cycle d'instruction sont utilisés pour exécuter l'instruction d'appel de sous-programme dans le troisième cycle d'instruction.

Pendant le processus ci-dessus exposé, le compteur de programme est incrémenté d'une unité à la fin de chaque cycle d'instruction. Par conséquent, à la fin du cycle d'extraction du
 15 code OP, la valeur du compteur de programme est incrémentée de PC à PC + 1. A la fin du premier cycle d'instruction, le compteur de programme est incrémenté de 1 et, par conséquent, la valeur de comptage augmente de PC + 1 à PC + 2. De même, le compteur de programme est incrémenté de 1 à la fin des deuxième et troisième
 20 cycles d'instruction.

Dans le processus selon l'invention de la figure 2(B), le code OP CALL relatif à l'instruction à exécuter lors du deuxième cycle d'instruction est extrait avant le début du premier cycle d'instruction. Le code OP CALL comporte une demande visant à
 25 comprimer les cycles d'extraction suivants. Pendant la durée M_2 à M_3 du premier cycle d'instruction, le byte d'adresse haut P_H est extrait. Pendant la durée de M_0 à M_1 du premier cycle d'instruction, le byte d'adresse P_L est extrait. Les bytes d'adresse P_H et P_L extraits lors du premier cycle d'instruction et le code OP
 30 CALL extrait avant le premier cycle d'instruction sont utilisés pour exécuter l'appel de sous-programme lors du deuxième cycle d'instruction.

Pendant le processus ci-dessus exposé, le compteur de programme est incrémenté d'une unité à la fin de chaque cycle d'instruction. De plus, le compteur de programme est incrémenté
 35 de 1 en réponse à chaque impulsion d'horloge M_3 qui tombe à mi-parcours

de chaque cycle d'instruction. A la fin du cycle d'extraction relatif au code OP, la valeur du compteur de programme augmente de PC à PC + 1. En réponse à l'impulsion d'horloge M_3 du premier cycle d'instruction, le compteur de programme est de nouveau
 5 incrémenté de 1 si bien que la valeur de comptage devient PC + 2. Ensuite, le compteur de programme est incrémenté de 1 en réponse à l'impulsion d'horloge M_1 survenant à la fin du premier cycle d'instruction, et ainsi de suite.

Au début du deuxième cycle d'instruction, le code
 10 OP relatif à l'instruction suivante à exécuter peut être extrait.

Ainsi, comme on l'aura compris, il suffit de deux cycles d'instruction pour exécuter une instruction de 3 bytes dans le processus préféré selon l'invention.

La figure 3 montre un système de microprocesseur
 15 conçu pour exécuter le processus préféré selon l'invention. Le mode de réalisation préféré d'un microcalculateur comprend une ROM programmable (PROM) 1, un tampon de données 2, un registre d'instruction 3, un décodeur d'instruction 4 qui peut reposer sur un réseau logique programmable (PLA) par exemple, un compteur
 20 de programme 5, un registre 6 d'adresse de RAM, et des transistors à effet de champ (FET) 7 à 9.

On va maintenant décrire en relation avec la figure 4 le fonctionnement du mode de réalisation préféré du microcalculateur constitué de la manière ci-dessus indiqué. Sur la figure 4, (A)
 25 indique des impulsions d'horloge M_2 , M_3 , M_0 et M_1 produites pendant chaque cycle d'instruction. (B) indique la valeur de comptage du compteur de programme 5 pendant l'exécution d'instructions de 1 byte séquentielles. (C) montre les variations de la valeur de comptage du compteur de programme 5 pendant l'exécution d'instructions
 30 de 2 bytes séquentielles.

Le décodeur d'instruction délivre un signal PCUP d'incrémentation du compteur de programme et plusieurs signaux de commande C_1 à C_3 , le dernier d'entre eux commandant le passage de l'information dans les éléments 1 à 6 du microprocesseur. Les
 35 signaux de commande C_1 , C_2 et C_3 subissent une intersection logique avec les impulsions d'horloge M_3 , M_1 et M_3 de nouveau de façon à

commander respectivement les FET 7, 8 et 9. Les FET 7 et 8 permettent que des données, soit des opérandes, soit des codes OP venant du programme ROM, soient verrouillées par le tampon de données 2 ou le registre d'instruction 3, pour l'état conducteur. Le FET 9 permet que l'opérande se trouvant dans le tampon de données 2 passe sur le bus de données pour l'état conducteur, si bien que l'opérande peut être chargée dans le registre 6 d'adresse de la RAM en ce qui concerne les chargements, les mises en mémoire et les ordres analogues, ou dans le compteur de programme 5 en ce qui concerne les appels de sous-programme, etc.

Lorsqu'un code OP contient une demande d'extraction d'opérande en compression dans le temps, le décodeur d'instruction 4 délivre à la fois une impulsion PCUP et des impulsions C_2 et C_3 , ce qui permet de charger l'opérande dans le tampon de données 2 et de l'en faire sortir en réponse à l'impulsion d'horloge M_1 suivante et d'incrémenter le compteur de programme afin d'indiquer le code OP suivant.

Dans l'exemple représenté sur la figure 4(B), une instruction de 1 byte est exécutée au cours d'un unique cycle d'instruction. L'extraction des instructions se trouvant dans la ROM 1 est déclenchée par le flanc postérieur de l'impulsion d'horloge M_1 en présence du signal de sortie C_1 venant du décodeur d'instruction 4. Au même instant, le compteur de programme 5 est incrémenté de 1 en réponse au signal PCUP. L'instruction extraite (code OP) est lue dans le registre d'instruction 3, puis est délivrée au décodeur d'instruction 4. Le décodeur d'instruction 4 décode l'instruction extraite et délivre les signaux appropriés C_1 , C_2 et C_3 .

Le flanc postérieur du signal d'horloge M_1 indique le début du cycle d'instruction pendant lequel l'instruction extraite lors du cycle d'instruction précédent est exécutée. En même temps que l'exécution de l'instruction extraite lors du cycle précédent, l'instruction de 1 byte suivante est extraite pendant le cycle d'instruction en cours. L'adresse de la PROM à laquelle il faut accéder en ce qui concerne l'instruction est

représentée par la valeur de comptage (PC, PC + 1, PC + 2) qui est incrémentée d'une unité sur chaque flanc postérieur du signal d'horloge M_1 par le signal PCUP.

Pendant l'exécution d'instructions de 1 byte,
 5 puisqu'il n'y a pas de demande d'extraction d'opérande comprimée, les signaux de sortie C_2 et C_3 du décodeur d'instruction 4 restent inactifs. En résultat, les FET 8 et 9 restent non conducteurs. De plus, à ce moment, puisque le signal PCUP du décodeur d'instruction 4 n'est pas produit à mi-durée du cycle d'instruction, le
 10 compteur de programme 5 n'est incrémenté qu'en réponse au flanc postérieur du signal d'horloge M_1 .

Comme représenté sur la figure 4(C), lorsque les instructions de 2 bytes comportant un code OP de 1 byte et une
 15 opérande de 1 byte doivent être exécutées, les impulsions C_2 et C_3 sont délivrées par le décodeur d'instruction 4 après extraction du code OP relatif à l'instruction suivant le flanc postérieur du signal d'horloge M_1 . Les signaux de sortie C_2 et C_3 rendent conducteurs les FET 8 et 9 de manière à décharger l'opérande de la ROM 1 et la charger dans le tampon de données 2, le compteur
 20 de programme 5 et, ou bien, le registre 6 d'adresse de la RAM.

Sur le flanc postérieur du signal d'horloge M_3 , le compteur de programme 5 est incrémenté d'une unité en réponse au signal PCUP venant du décodeur d'instruction 4. Au même instant, par exemple, la donnée contenue dans le registre 6 d'adresse
 25 de la RAM est lue et est utilisée pour l'exécution de l'instruction pendant la durée M_0 à M_1 .

Pendant la durée M_0 à M_1 , l'instruction suivante est extraite de la PROM. Dans ce cas, puisque le compteur de programme 5 est incrémenté par PCUP en réponse au flanc postérieur de l'impulsion d'horloge M_3 , l'adresse de la PROM représentée par la valeur
 30 de comptage correspond à l'adresse du code OP relatif à l'instruction suivante.

Comme on l'aura compris, pour permettre la compression du cycle d'extraction selon le processus préféré ci-dessus exposé,
 35 il est nécessaire que les deuxième et, ou bien, troisième bytes de données ne contiennent pas de composantes du code OP, mais, plutôt,

contiennent seulement des opérandes telles que données ou adresses. Puisque, dans la plupart des microcalculateurs à 4 bits, les instructions ne contiennent qu'un code OP de 1 byte et un ou plusieurs bytes d'opérande, le processus préféré ci-dessus exposé est applicable à la plupart des microcalculateurs de ce type.

Comme représenté sur les figures 5(A) et 5(B), selon le processus préféré de l'invention, des instructions de 2 bytes qui, classiquement, demandent 2 cycles d'instruction pour être exécutés, peuvent être exécutés en un seul cycle d'instruction dans une architecture en pipeline. De la même façon, des instructions de 3 bytes qui, classiquement, demandent 3 cycles d'instruction pour être exécutés, peuvent être exécutés en 2 cycles d'instruction. Ainsi, comme on l'aura facilement compris, le procédé préféré selon l'invention abrège efficacement la durée d'exécution.

De plus, les figures 6(A) et 6(B) illustrent schématiquement la zone nécessaire pour le décodeur d'instruction dans le système de la technique antérieure et le système selon l'invention, où (A) indique le système classique et (B) indique celui selon l'invention. Sur les figures 6(A) et 6(B), les lignes verticales représentent des lignes de sélection d'éléments ET et des lignes horizontales représentent des lignes de sélection d'éléments OU. En outre, M_1 représente un signal sélecteur d'un état 1 machine, M_2 représente un signal d'entrée sélecteur d'un état 2 machine, et M_3 représente un signal d'entrée sélecteur d'un état 3 machine.

Comme on l'aura compris en observant les figures 6(A) et 6(B), dans le système classique, trois états machine sont nécessaires pour exécuter une instruction de 3 bytes demandant 3 cycles d'instruction. Toutefois, selon le processus préféré de l'invention, pour exécuter des instructions de 3 bytes, il ne faut que 2 cycles d'instruction. Par conséquent, il ne faut que deux états machine du décodeur d'instruction pour exécuter des instructions de 3 bytes. Ainsi, la ligne de sélection d'éléments OU qui sert à commander une ligne de signal pour l'état 3 machine n'est plus nécessaire. Ceci conduit à une réduction notable de la longueur des lignes de sélection d'éléments ET et permet donc

une réduction concomitante de la taille du décodeur d'instruction.

Ainsi, l'invention a atteint les buts et avantages ci-dessus avancés.

- 5 Bien entendu, l'homme de l'art sera en mesure d'imaginer, à partir du procédé dont la description vient d'être donnée à titre simplement illustratif et nullement limitatif, diverses variantes et modifications ne sortant pas du cadre de l'invention.

REVENDEICATIONS

1. Procédé d'exploitation d'un microprocesseur ayant une architecture en pipeline, qui permet que les données d'instruction relatives à un cycle d'exécution d'instruction ultérieur
5 soient extraites de la mémoire pendant un cycle d'instruction en cours, caractérisé en ce qu'il comprend les opérations suivantes :
- préprogrammer des codes opération d'instruction (codes OP) pour des instructions demandant plus d'un byte de donnée d'instruction avec une information supplémentaire demandant une
10 extraction, comprimée dans le temps, de la donnée d'instruction ; et
 - extraire plus d'un byte de donnée d'instruction hors de la mémoire pendant le cycle de l'instruction faisant immédiatement suite à une demande d'extraction comprimée dans le temps.
- 15 2. Procédé selon la revendication 1, caractérisé en ce que ladite donnée d'instruction consiste en un seul byte de code OP et un seul byte d'opérande, et ledit byte du code OP est extrait immédiatement avant le début d'un cycle d'instruction, ladite opération d'extraction comprenant l'opération qui consiste
20 à extraire ledit byte d'opérande associé avec ledit byte de code OP extrait en même temps que le début dudit cycle d'instruction.
3. Procédé selon la revendication 1, caractérisé en ce que ladite donnée d'instruction consiste en un seul byte de code OP et deux bytes d'opérande, et ledit byte du code OP est
25 extrait immédiatement avant le début d'une première instruction, ladite opération d'extraction comprenant les opérations consistant à extraire un premier byte de l'opérande associé avec ledit byte du code OP extrait en même temps que le début dudit premier cycle d'instruction et à extraire un deuxième byte de l'opérande associé
30 avec ledit premier byte de l'opérande pendant ledit premier cycle d'instruction suivant l'extraction dudit premier byte de l'opérande.
4. Procédé selon la revendication 2, caractérisé en ce qu'il comprend en outre l'opération consistant à exécuter
l'instruction spécifiée par lesdits bytes du code OP et de l'opé-
35 rande dans ledit cycle d'instruction immédiatement avant lequel ledit byte du code OP a été extrait.

5. Procédé selon la revendication 3, caractérisé en ce qu'il comprend en outre l'opération consistant à exécuter l'instruction spécifiée par le byte du code OP et lesdits premier et deuxième bytes de l'opérande dans un deuxième cycle d'instruction faisant immédiatement suite audit premier cycle d'instruction.

1/4

FIG. 1 (A)

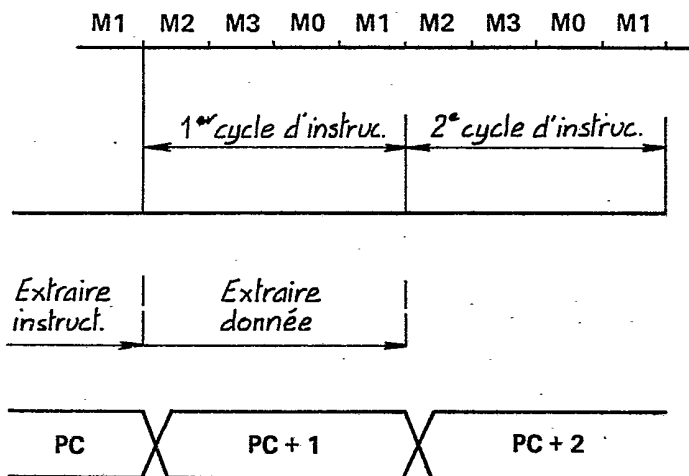
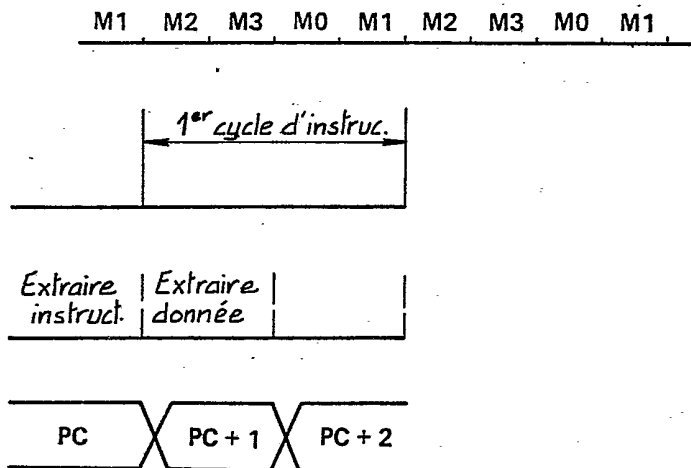


FIG. 1 (B)



2/4

FIG.2(A)

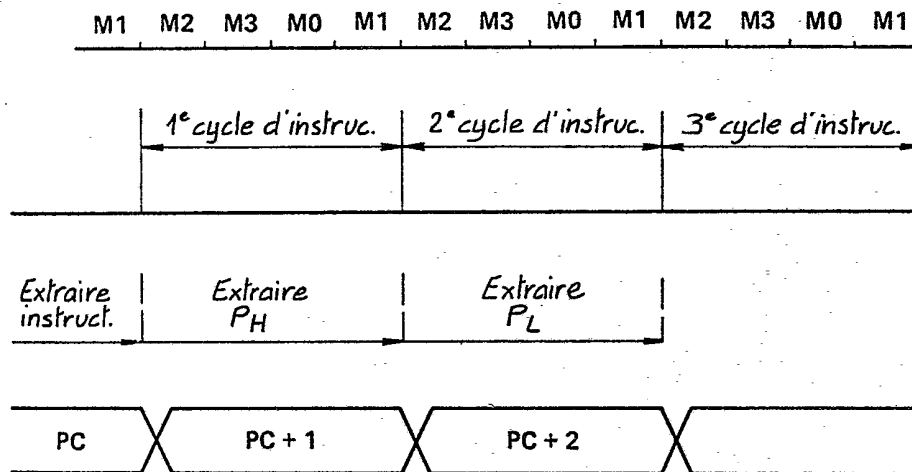
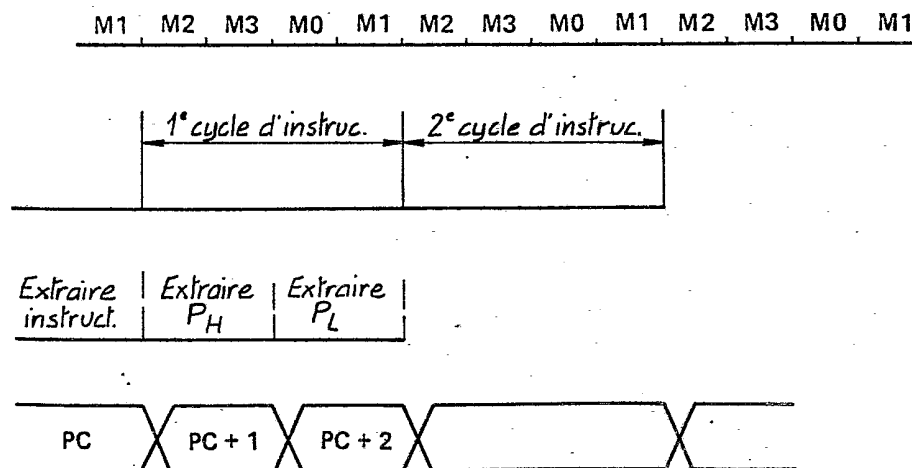


FIG.2(B)



3/4

FIG.3

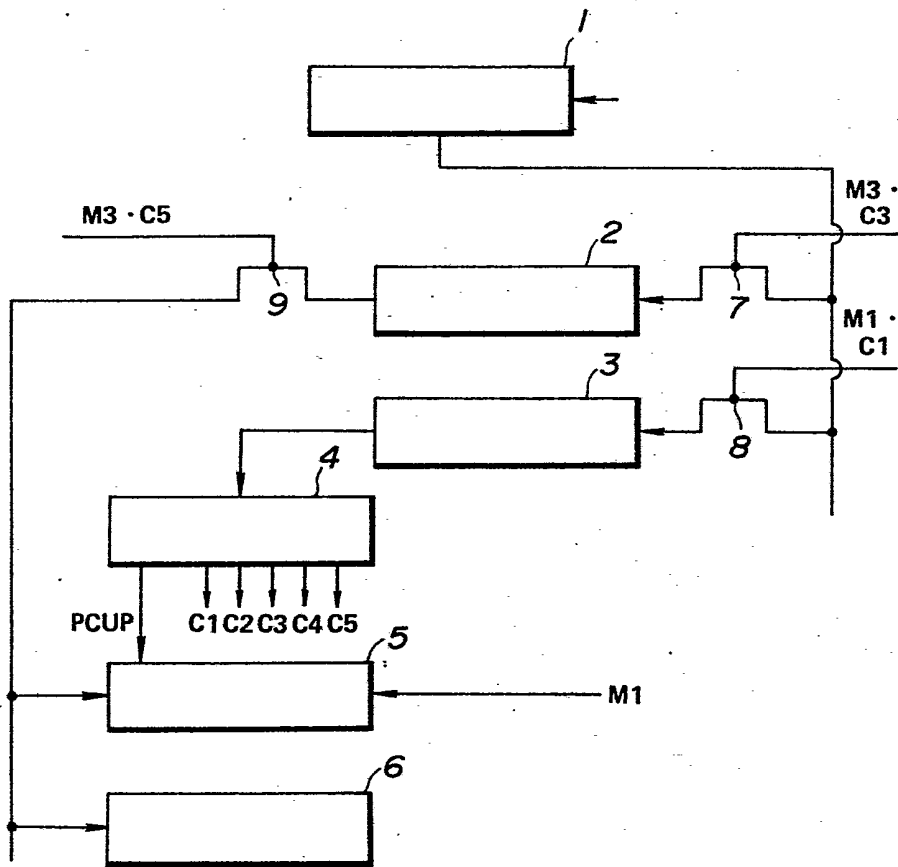
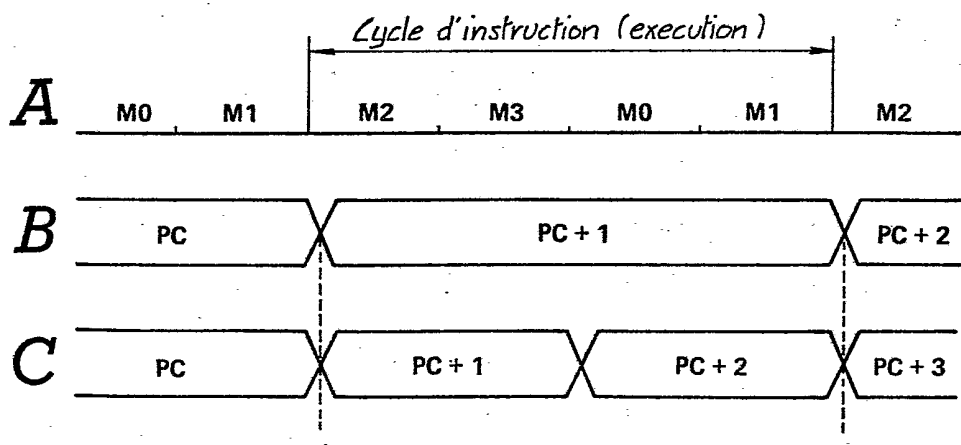


FIG.4



4/4

FIG. 5(A)

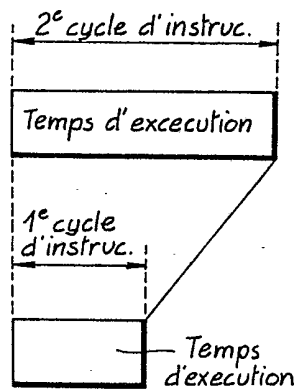


FIG. 5(B)

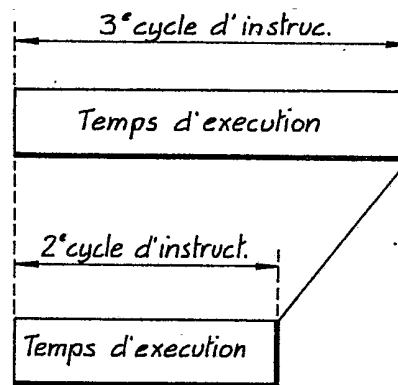


FIG. 6(A)

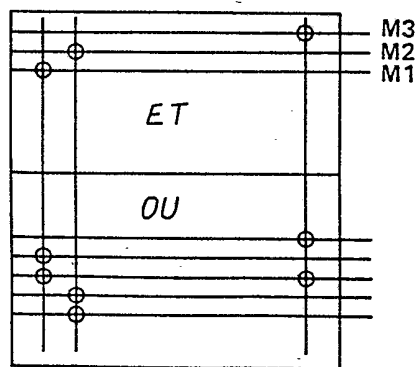


FIG. 6(B)

