

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 19.03.10.

30 Priorité :

43 Date de mise à la disposition du public de la demande : 23.09.11 Bulletin 11/38.

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) : DELORD RAYMOND — FR.

72 Inventeur(s) : DELORD RAYMOND.

73 Titulaire(s) : DELORD RAYMOND.

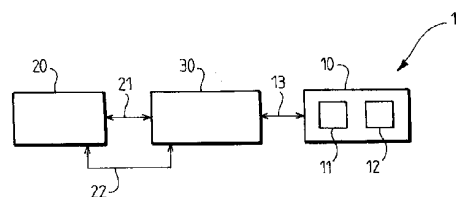
74 Mandataire(s) : LOYER ET ABELLO.

54 SYSTEME DE TRAITEMENT DE DONNEES.

57 Un dispositif d'interfaçage (30) entre un microprocesseur graphique (20) et une mémoire centrale (10) est apte à :
pré-interpréter une séquence d'instructions d'un programme exécutable stocké dans la mémoire centrale,
charger dans la mémoire cache des instructions ou données,

maintenir un annuaire d'adresses associant les adresses de la mémoire cache correspondant aux instructions ou données chargées dans la mémoire cache à des adresses virtuelles utilisables par les interfaces mémoire du microprocesseur graphique, et

transférer ladite séquence d'instructions pré-interprétée au microprocesseur graphique en substituant lesdites adresses virtuelles auxdites adresses de la mémoire centrale, de sorte que le microprocesseur graphique exécute ladite séquence d'instructions en produisant des requêtes d'accès mémoire désignant les adresses virtuelles.



L'invention se rapporte au domaine du traitement des données numériques, en particulier aux systèmes de traitement massivement parallèles tels que les microprocesseurs graphiques.

5

Les ordinateurs actuels comportent généralement une carte graphique qui est principalement constituée d'un microprocesseur graphique, de mémoire graphique et d'une interface avec le système d'affichage. Les microprocesseurs graphiques ont été initialement conçus pour exécuter des calculs sur des images, 10 pour les afficher et pour gérer la mémoire graphique. Une spécificité des microprocesseurs graphiques est leur capacité à exécuter des instructions à données parallèles. Cela signifie que le microprocesseur graphique peut exécuter une instruction sur une multitude de données en même temps. Cette caractéristique est aussi connue sous l'acronyme anglophone SIMD (single instruction multiple data). 15 Certains modèles comportent aussi des unités de traitement de type MIMD (multiple instruction multiple data). Les microprocesseurs graphiques sont donc très parallélisés pour effectuer des opérations sur un gros volume de données, qui correspondent typiquement aux couleurs de chaque pixel de l'écran. Au-delà de ces fonctions premières, les microprocesseurs graphiques ont évolué vers une plus 20 grande capacité de programmation permettant d'envisager d'autres applications demandant d'énormes quantités de calcul en parallèle, comme décrit par exemple dans « GPU Computing » in Proceedings of the IEEE, Vol. 96, No. 5, pp. 879-899, 2008.

A la différence d'un microprocesseur usuel, un microprocesseur graphique 25 est un coprocesseur qui a besoin d'un microprocesseur maître. En effet, les interactions d'un microprocesseur graphique avec l'extérieur consistent principalement en :

- une interaction avec la mémoire graphique servant à lire et écrire des instructions et des données dans la mémoire graphique,

- une interaction via le bus principal de l'ordinateur avec le microprocesseur maître. Cette interaction via le bus principal, par exemple de type PCI Express, est définie exactement par le constructeur du microprocesseur graphique. Elle peut servir à modifier le registre adresse ou tout autre registre du microprocesseur graphique, à spécifier les données sur lesquelles ce dernier opère, et/ou effacer ou invalider le cache du microprocesseur graphique. Cette interaction permet notamment au microprocesseur maître d'ordonner au microprocesseur graphique de débiter une exécution ou de lui communiquer une adresse mémoire où se trouvent des instructions à exécuter.
- 10 - une interaction via le bus principal de l'ordinateur avec la mémoire centrale, servant à accéder aux instructions indiquées par le microprocesseur maître, et
 - une interaction avec l'écran via un convertisseur analogique numérique.

Pour qu'un traitement soit exécuté par le microprocesseur graphique, un ordinateur conventionnel met en œuvre les étapes suivantes :

- 15 - un programme principal chargé dans la mémoire centrale de l'ordinateur comporte un sous-programme préparé spécifiquement pour être exécuté par le microprocesseur graphique ;
 - lorsque le microprocesseur maître qui exécute le programme principal atteint une instruction appelant le sous-programme, il envoie une requête en direction du périphérique carte graphique ;
 - 20 - avec cette requête, le microprocesseur maître envoie au microprocesseur graphique l'adresse dans la mémoire centrale où se trouve le code que la carte graphique doit exécuter ;
 - à réception de la requête, le microprocesseur graphique effectue une requête d'accès à la mémoire centrale pour copier vers la mémoire graphique le segment de code indiqué par le microprocesseur maître ;
 - 25 - une fois que le segment de code est copié en mémoire graphique, le microprocesseur graphique exécute ces instructions. Pour cela, les instructions sont chargées depuis la mémoire graphique, typiquement des bancs de mémoire SDRAM, pour être exécutées dans une multitude d'unités de
 - 30

traitement parallèles. Pour autant que le parallélisme soit correctement exploité, ce traitement est beaucoup plus rapide que celui du microprocesseur maître. En effet, le microprocesseur graphique est optimisé pour effectuer des calculs identiques sur plusieurs groupes de données simultanément.

5 Par rapport à un microprocesseur conventionnel, un microprocesseur graphique comporte une mémoire cache de plus faible capacité qui n'est pas adaptée à des calculs généraux.

10 Selon un mode de réalisation, l'invention fournit un système de traitement de données comportant :

une mémoire centrale apte à stocker un programme exécutable comprenant des instructions, par exemple des instructions à données parallèles, et des données utilisables par le programme,

15 un microprocesseur graphique comprenant des unités de traitement programmables, de préférence des unités de traitement massivement parallèles, convenant pour exécuter les instructions et des interfaces mémoire à bande passante élevée convenant pour la communication avec des mémoires rapides, et un dispositif d'interfaçage comprenant :

20 un module de contrôle de la mémoire centrale pour lire et écrire dans la mémoire centrale,

une mémoire cache apte à stocker des portions du programme exécutable et des données,

un module de communication relié aux interfaces mémoire à bande passante élevée

25 du microprocesseur graphique, et

un module de commande apte à :

pré-interpréter une séquence d'instructions du programme exécutable pour déterminer des adresses de la mémoire centrale devant être appelées lors d'une exécution de la séquence d'instructions,

30 charger dans la mémoire cache des instructions ou données stockées auxdites

- adresses de la mémoire centrale,
maintenir un annuaire d'adresses associant les adresses de la mémoire cache correspondant aux instructions ou données chargées dans la mémoire cache à des adresses virtuelles utilisables par les interfaces mémoire du microprocesseur
- 5 graphique, et
- transférer la séquence d'instructions pré-interprétée au microprocesseur graphique en substituant les adresses virtuelles aux adresses de la mémoire centrale, de sorte que le microprocesseur graphique exécute la séquence d'instructions en produisant des requêtes d'accès mémoire désignant lesdites adresses virtuelles, et
- 10 le module de communication étant apte à recevoir les requêtes d'accès mémoire et à traiter les requêtes d'accès mémoire en lisant ou en réécrivant des instructions ou données dans les positions de la mémoire cache qui sont associées aux adresses virtuelles selon l'annuaire d'adresses.

Selon des modes de réalisation avantageux, le système peut présenter une

15 ou plusieurs des caractéristiques suivantes :

- le module de commande est apte à maintenir un deuxième annuaire d'adresses associant les adresses de la mémoire cache correspondant aux instructions ou données chargées dans la mémoire cache aux adresses de la mémoire centrale depuis lesquelles les instructions ou données ont été chargées, et

20 à produire des requêtes d'accès à la mémoire centrale pour transférer des instructions ou données écrites ou réécrites dans la mémoire cache à la suite des requêtes d'accès mémoire produites par le microprocesseur graphique vers des positions de la mémoire centrale qui sont associées aux adresses de la mémoire cache selon le deuxième annuaire d'adresses.

- 25 - le module de commande est apte à décoder des instructions de contrôle de flux contenues dans ladite séquence d'instructions et, lorsqu'une instruction de branchement est rencontrée, poursuivre la pré-interprétation des instructions en suivant plusieurs branches possibles de la séquence d'instructions jusqu'à ce que le branchement soit finalement déterminé par le microprocesseur graphique.

- la séquence d'instructions comporte des microcodes adaptés au microprocesseur graphique, par exemple dans un langage assembleur du processeur graphique.

- le module de commande est apte à décoder une macro-instruction dans ladite séquence d'instructions pour produire des microcodes adaptés au microprocesseur graphique et à transférer les microcodes au microprocesseur graphique à la place de la macro-instruction. Par exemple, il peut être prévu d'utiliser un jeu d'instructions plus complexe au niveau du dispositif d'interfaçage que dans le microprocesseur graphique, notamment afin d'augmenter le niveau d'abstraction et la compacité du modèle de programmation et donc d'augmenter le débit d'instructions avec la mémoire centrale.

- une pluralité de microprocesseurs graphiques sont prévus pour exécuter des séquences d'instructions respectives, le dispositif d'interfaçage comportant une pluralité de modules de commande et une pluralité de modules de communication attribués respectivement à chacun des microprocesseurs graphiques.

- la mémoire cache comporte plusieurs éléments de mémoire présentant des niveaux de latence différents, le module de commande étant apte à gérer ladite mémoire cache de manière à stocker les instructions et données susceptibles d'être appelées le plus souvent dans un élément de mémoire du niveau de latence le plus bas. Une telle structuration de la mémoire cache du dispositif d'interfaçage peut être effectuée de manière similaire aux couches de mémoire cache des microprocesseurs conventionnels, connues par exemple sous le nom de L2 cache et L3 cache.

- le module de contrôle de la mémoire centrale est apte à gérer une pile des requêtes d'accès dans la mémoire centrale pour effectuer lesdits accès par rafales. L'accès en rafales est une propriété de certaines technologies de SDRAM, typiquement utilisées dans les mémoires graphiques. Un tel accès par rafale est particulièrement adapté à un mode de réalisation dans lequel la mémoire centrale est constitué d'éléments de mémoire rapides de type GDDR (de l'anglais Graphics Double Data Rate).

- un bus principal, par exemple de type PCI Express, est relié à un port correspondant du microprocesseur graphique et à un port correspondant du dispositif d'interfaçage pour leur permettre de communiquer entre eux et/ou avec d'autres composants ou périphériques du système.

5 Selon des modes de réalisations, le dispositif d'interfaçage peut aussi être relié à un microprocesseur maître ou à des périphériques, tels qu'un dispositif d'accès réseau, un disque dur ou autre. De telles liaisons peuvent être réalisées via des modules d'entrée sortie intégrés au dispositif d'interfaçage ou via des composants séparés, par exemple un chipset southbridge ou un chipset northbridge.

10 L'invention fournit également un procédé d'interfaçage entre un microprocesseur graphique et une mémoire centrale, ledit procédé comprenant : pré-interpréter une séquence d'instructions d'un programme exécutable stocké dans ladite mémoire centrale pour déterminer des adresses de la mémoire centrale devant être appelées lors d'une exécution de ladite séquence d'instructions,

15 charger dans une mémoire cache des instructions ou données stockées auxdites adresses de la mémoire centrale,

maintenir un annuaire d'adresses associant les adresses de la mémoire cache correspondant auxdites instructions ou données chargées dans la mémoire cache à des adresses virtuelles utilisables par des interfaces mémoire du microprocesseur

20 graphique,

transférer ladite séquence d'instructions pré-interprétée au microprocesseur graphique en substituant lesdites adresses virtuelles auxdites adresses de la mémoire centrale, de sorte que le microprocesseur graphique exécute ladite séquence d'instructions en produisant des requêtes d'accès mémoire désignant

25 lesdites adresses virtuelles, et

traiter lesdites requêtes d'accès mémoire en lisant ou en écrivant des instructions ou données dans les positions de la mémoire cache qui sont associées auxdites adresses virtuelles selon ledit annuaire d'adresses.

Certains aspects de l'invention partent du constat que, dans une
30 architecture de carte graphique conventionnelle, le transfert global du code devant

être exécuté par le microprocesseur graphique et des données correspondantes depuis la mémoire centrale vers la mémoire graphique représente généralement un facteur de ralentissement du traitement en raison de la latence de la mémoire centrale. Certains aspects de l'invention sont basés sur l'idée d'utiliser les canaux

5 mémoire à bande passante très élevée d'un microprocesseur graphique pour accéder à une mémoire cache qui est alimentée à la volée avec les instructions et données nécessaires au traitement du microprocesseur graphique au cours de l'exécution de ce traitement, notamment afin de supprimer la nécessité d'un transfert global préalable de ces instructions et données. Certains aspects de

10 l'invention partent de l'idée de réaliser un système de traitement de données programmable en combinant un microprocesseur graphique utilisé essentiellement comme une unité arithmétique et logique extrêmement puissante avec un dispositif complémentaire regroupant les autres fonctions essentielles d'un microprocesseur usuel, notamment le contrôle des relations avec une mémoire centrale et avec des

15 équipements périphériques. Certains aspects de l'invention sont fondés sur l'idée de réaliser une telle combinaison en émulant l'environnement conventionnel d'un microprocesseur graphique actuel, notamment la relation du microprocesseur graphique avec une mémoire graphique dédiée contenant l'ensemble des données et instructions nécessaires à l'exécution d'un traitement, sans qu'une telle mémoire

20 graphique ne soit en relation directe avec le microprocesseur graphique, voire sans qu'elle ne soit physiquement présente.

L'invention sera mieux comprise, et d'autres buts, détails, caractéristiques et

25 avantages de celle-ci apparaîtront plus clairement au cours de la description suivante de plusieurs modes de réalisation particuliers de l'invention, donnés uniquement à titre illustratif et non limitatif, en référence aux dessins annexés. Sur ces dessins :

La figure 1 est une représentation schématique fonctionnelle d'un système

30 de traitement de données selon un mode de réalisation,

La figure 2 est une représentation schématique fonctionnelle d'un mode de réalisation du dispositif d'interfaçage pouvant être utilisé dans le système de la figure 1,

La figure 3 est un diagramme d'étapes représentant un mode de réalisation d'un traitement pouvant être mis en œuvre par le dispositif d'interfaçage de la figure 2,

La figure 4 est une représentation schématique fonctionnelle d'un système de traitement de données selon un autre mode de réalisation.

10 En référence à la figure 1, un calculateur massivement parallèle 1 comporte une mémoire vive centrale 10, au moins un microprocesseur graphique 20 et un circuit intégré de traitement intermédiaire 30 assurant l'interfaçage entre la mémoire centrale 10 et le ou les microprocesseurs graphiques 20.

Le microprocesseur graphique 20 est par exemple du type HD5870 du fabricant AMD qui possède 1600 unités de traitement permettant d'effectuer 1600 calculs en parallèle. Dans le calculateur 1, le microprocesseur graphique 20 est essentiellement utilisé en tant qu'une unité arithmétique et logique pour effectuer des quantités de calculs importantes en parallèle.

Le circuit intégré 30 remplit plusieurs fonctions: il supplée le microprocesseur graphique 20 pour les tâches dans lesquelles le microprocesseur graphique 20 ne peut pas fonctionner de manière autonome, notamment accéder à la mémoire centrale 10. Il pilote le microprocesseur graphique 20 de manière à exécuter des instructions présentes en mémoire vive en lui fournissant les instructions à la volée depuis la mémoire centrale 10 au cours de l'exécution. Il simule l'environnement habituel du microprocesseur graphique 20, en envoyant les instructions et données via le canal mémoire du microprocesseur graphique 20 et en simulant le comportement du bus principal d'un ordinateur par le canal bus du microprocesseur graphique. Pour cela, le circuit intégré 30 est relié à la mémoire centrale 10 par un bus mémoire 13, à un ou des microprocesseurs graphiques 20 via leurs ports mémoires 21 et via leurs ports bus principal 22, par exemple de type Bus

PCI Express, et éventuellement à des périphériques tels que contrôleur réseau et disque dur et/ou un microprocesseur programmable non représentés.

La mémoire centrale 10 permet de stocker temporairement un programme exécutable 11 et des données 12. Ces instructions et données sont répertoriées sous
5 certaines adresses. Afin d'accéder aux instructions et aux données présentes en mémoire 10, le bus mémoire 13 comporte un canal adresse et un canal données. Lorsqu'une requête d'accès est effectuée sur le canal adresse, après un certain temps de latence, la réponse est renvoyée via le canal de données. En spécifiant le mode lecture ou écriture selon la technique connue, le circuit intégré 30 peut donc stocker
10 et lire des données et instructions dans la mémoire 10. De préférence, la mémoire 10 est conçue pour être rapide, par exemple à partir de bancs de mémoire GDDR ou d'un grand nombre de barrettes de mémoire standard. La mémoire 10 peut être partagée avec d'autres composants ou périphériques du calculateur 1.

Le programme 11 comporte typiquement trois types d'instructions: des
15 instructions de lecture et d'écriture permettant un accès aux données du programme, des instructions d'opérations servant à effectuer toutes sortes d'opérations arithmétiques ou booléennes telles que additions, soustractions, multiplications, divisions, opérations booléennes, ou comparaisons, notamment sur des données parallèles, et des instructions de contrôle de flux qui modifient l'ordre
20 d'exécution des instructions, telles que les boucles, les tests, ou autres ruptures de séquence. Ces instructions sont par exemple initialement codées dans un langage assembleur du microprocesseur graphique 20 ou seront transcodées dans un tel langage par le circuit 30 au cours de l'exécution.

L'exécution du programme 11 est effectuée pour l'essentiel par le
25 microprocesseur graphique 20, notamment les instructions d'opérations à données parallèles. Le circuit 30 traite essentiellement les instructions de lecture et d'écriture et les instructions de contrôle de flux du programme 11, de manière à préparer les instructions et les opérandes pour le microprocesseur graphique 20 en avance temporelle sur le traitement du microprocesseur graphique 20. Pour cela, le circuit
30 30 comporte certains des composants habituels d'un microprocesseur, notamment

une horloge pour gérer la fréquence de chaque partie, un contrôleur mémoire pour gérer les entrées-sorties mémoires, un compteur ordinal pour contenir en permanence l'adresse de la prochaine instruction, un décodeur pour pré interpréter les instructions venant de la mémoire, une mémoire cache pour permettre au
5 microprocesseur graphique 20 d'accéder rapidement aux instructions et données utilisées, et un annuaire de type TLB (acronyme de l'anglais : Translation Lookaside Buffer) pour associer les adresses utilisées dans la mémoire centrale 10 à des adresses virtuelles.

Le circuit 30 peut aussi comporter une unité arithmétique et logique pour
10 prendre en charge certaines opérations élémentaires, comme calculer un saut d'adresse, et un séquenceur pour décider de l'ordre d'exécution des instructions par le microprocesseur graphique 20. En outre, le décodeur peut comprendre un jeu d'instruction spécifique différent du jeu d'instructions du microprocesseur graphique 20, notamment des instructions complexes regroupant plusieurs instructions du
15 microprocesseur graphique dans le but d'augmenter le débit effectif d'instructions. Dans ce cas, le décodeur remplit aussi la fonction de transcoder ces instructions complexes en instructions compréhensibles par le microprocesseur graphique 20.

Le calculateur 1 permet d'exécuter les instructions du programme 11 de la manière suivante. Les instructions initialement stockées dans la mémoire 10 sont
20 préchargées par le circuit 30 dans sa mémoire cache. Le circuit 30 pré-interprète les instructions dans l'ordre de déroulement du programme 11, pour autant que cet ordre soit déterminable a priori. Le circuit 30 détecte notamment les instructions de branchement, qui lui permettent de précharger les instructions suivantes. Le microprocesseur graphique 20 reçoit du circuit 30 un flot d'instructions
25 préinterprétées via son canal mémoire 21 et des commandes principales via son canal bus principal 22.

Ce fonctionnement va être maintenant illustré par un exemple. Le calcul à faire par le microprocesseur graphique 20 est d'ajouter la valeur en adresse 1 et la valeur en adresse 2 de la mémoire centrale 10. Les instructions, la valeur en adresse
30 1 et la valeur en adresse 2 ont été mises en mémoire cache par le circuit 30. Le

circuit 30 communique au microprocesseur graphique 20 une adresse fictive X et lui demande de commencer l'exécution. L'adresse fictive X est comprise par le microprocesseur graphique 20 comme une adresse dans la mémoire graphique habituellement prévue, mais qui fait ici défaut. Le microprocesseur graphique 20
5 envoie une requête à travers son canal mémoire 21 pour connaître l'instruction. Il croit communiquer avec une mémoire graphique dédiée et effectue donc une requête standard pour connaître l'instruction en l'adresse virtuelle X.

Le circuit 30 reçoit cette requête et répond immédiatement car l'instruction se situe déjà en mémoire cache. De préférence, le circuit 30 définit en même temps
10 les deux valeurs d'opérandes dans la mémoire cache du microprocesseur graphique 20 pour faire l'économie d'une requête supplémentaire. Le microprocesseur graphique 20 exécute le calcul et renvoie le résultat dans la mémoire cache du circuit 30 en effectuant une requête vers une adresse virtuelle censée correspondre à une position dans la mémoire graphique dédiée conventionnelle. Le circuit 30
15 récupère ainsi le résultat qu'il peut transférer dans la mémoire centrale 10. En procédant de cette façon, le calculateur 1 peut exploiter le microprocesseur graphique 20 comme une unité de calcul massivement parallèle exécutant un programme directement depuis la mémoire centrale 10.

Dans un mode de réalisation comportant plusieurs microprocesseurs
20 graphiques, le circuit 30 peut aussi traiter l'ordonnancement des instructions à la manière d'un système « out of order » pour répartir les tâches sur les divers microprocesseurs graphiques de manière à optimiser la durée du traitement.

La pré-interprétation des instructions peut comporter des calculs d'adresses, par exemple dans le cas d'un saut d'adresse. Un tel calcul peut être
25 exécuté par le circuit 30. Dans une variante de réalisation, le circuit 30 confie au microprocesseur graphique 20 les calculs d'adresses rencontrés au cours de la préinterprétation des instructions. Ces calculs sont alors traités comme des tâches parallèles et indépendantes du traitement principal du flux d'instructions par le microprocesseur graphique 20. En effet, les résultats de ces calculs d'adresses
30 doivent être disponibles dans la phase de préinterprétation du flux d'instructions,

c'est-à-dire en avance temporelle sur l'exécution de ce flux d'instructions par le microprocesseur graphique 20. Les deux variantes peuvent aussi être combinées. De préférence dans ce cas, le circuit 30 comporte un ordonnanceur pour aiguiller les calculs d'adresse, soit vers une unité arithmétique et logique du circuit 30, soit vers
5 le microprocesseur graphique 20.

En référence à la figure 2, on décrit maintenant un autre mode de réalisation d'un dispositif d'interfaçage entre une mémoire vive et des microprocesseurs graphiques. Les mêmes chiffres de référence désignent des éléments identiques ou similaires à ceux de la figure 1.

10 Le dispositif d'interfaçage 130 comporte des modules de mémoire cache hiérarchisés, à savoir un cache partagé 131 et un cache rapide 132. Le dispositif d'interfaçage 130 comporte des modules d'entrée-sortie, à savoir un module de contrôle mémoire 133 pour accéder à une mémoire centrale 10, un module de communication 134 pour échanger des données avec un microprocesseur graphique
15 20 à travers des canaux mémoires 135 et un canal bus principal 136, un module de contrôle réseau 137, par exemple de type Ethernet ou Infiniband, pour accéder à un réseau de communication 140 et un module de contrôle de stockage 138 pour échanger des données avec une mémoire de masse 150, par exemple un disque dur.

Le dispositif d'interfaçage 130 fonctionne sous la commande d'un module
20 de commande 145. Le fonctionnement du dispositif d'interfaçage 130 va être décrit en référence à la figure 3.

Le module de commande 145 reçoit l'ordre de commencer l'exécution d'un programme stocké dans la mémoire centrale. Cet ordre comporte l'adresse d'entrée dans le programme.

25 A l'étape 161, le module de commande 145 commence à charger le programme dans la mémoire cache 131 à partir de son adresse d'entrée et à pré-interpréter les instructions dans l'ordre d'exécution du programme. Cette pré-interprétation consiste à déterminer les adresses mémoire des instructions et données devant être lues pour exécuter le programme. Les données préchargées
30 sont typiquement des portions successives du programme, et non le programme

dans son entier. La granularité de ces portions dépend des débits entre les différents sous-systèmes du dispositif d'interfaçage 130, elle peut être de quelques instructions.

5 A l'étape 162, les instructions et données sont chargées depuis la mémoire centrale à mesure que leurs adresses sont déterminées et sont stockées dans les mémoires caches 131 et 132 pour être tenues à disposition du microprocesseur graphique 20.

10 A l'étape 163, le module de commande 145 génère et maintient un annuaire d'adresses 139, du type Translation Lookaside Buffer, pour associer les adresses des mémoires caches aux adresses de la mémoire centrale.

A l'étape 164, le module de commande 145 génère et maintient un autre annuaire d'adresses 146, du type Translation Lookaside Buffer, pour associer les adresses des mémoires caches à des adresses virtuelles exploitables par les canaux mémoire du microprocesseur graphique 20.

15 A l'étape 165, le module de communication 134 transfère les instructions, à mesure qu'elles sont pré interprétées, au microprocesseur graphique 20 en substituant aux adresses mémoires les adresses virtuelles qui leurs sont associées par les annuaires 139 et 146. Parallèlement, un module de gestion 147 prédit les requêtes du microprocesseur graphique 20 pour placer dans le cache rapide 132
20 attribué à ce dernier les données et instructions qui sont le plus susceptibles d'être appelées pendant l'exécution du flux d'instructions par le microprocesseur graphique 20. De préférence, le cache rapide 132 est capable de répondre au microprocesseur graphique 20 avec une latence d'un cycle ou deux, alors que la latence atteint près de neuf cycles dans les mémoires usuelles GDDR5.

25 A l'étape 166, le module de communication 134 traite les requêtes d'accès mémoire que le microprocesseur graphique 20 produit pour appeler les adresses virtuelles qui lui ont été communiquées. A l'aide de l'annuaire 146, le module de communication 134 peut traiter ces requêtes à partir des mémoires caches, en particulier à partir du cache rapide 132 lorsque la prédiction effectuée par le module
30 de gestion 147 est efficace. Ces requêtes sont générées et traitées à un très haut

débit accepté par les canaux mémoire du microprocesseur graphique 20, par exemple de l'ordre de 200 Go/s.

Le module de communication 134 est un module esclave qui fonctionne en fait sous la commande du microprocesseur graphique 20 et à la fréquence imposée
5 par celui-ci, comme une mémoire graphique. Il remplit la fonction d'une zone tampon qui permet d'adapter le flux de données à la fréquence d'horloge du microprocesseur graphique. Le module de communication 134 peut donc fonctionner à une fréquence différente du reste du dispositif 130.

A l'étape 167, à mesure que les instructions sont exécutées par le
10 microprocesseur graphique 20, les résultats du traitement réécrits par ce dernier dans les mémoires cache du dispositif 130 sont réécrits dans la mémoire centrale 10 aux adresses associées par les annuaires 139 et 146.

Bien que la figure 3 représente des étapes séquentielles dans un but didactique, il est possible de réaliser les fonctions du dispositif 130 en exécutant
15 certaines étapes dans un autre ordre, ou simultanément ou en fusionnant plusieurs des étapes représentées.

Pour certaines technologies de mémoire à bande passante élevée, par exemple les mémoires synchrones de type SDRAM, il est préférable de traiter les accès à la mémoire 10 sous la forme de rafales d'un volume donné. Pour cela, les
20 requêtes d'accès à la mémoire centrale peuvent être stockées temporairement dans une pile des requêtes d'accès 149.

Le dispositif 130 peut piloter plusieurs microprocesseurs graphiques de la même manière. Pour cela, comme représenté en traits interrompus, il peut être prévu des modules 145, 146, 132 et 134 attribués à chacun des microprocesseurs
25 graphiques. Ceux-ci peuvent avoir des fréquences d'horloges différentes.

Dans l'étape de pré-interprétation 161, le module 145 effectue la prédiction des branchements pour pouvoir mettre en cache les données qui seront appelées ultérieurement, notamment des branchements conditionnels. Certains registres de mémoire cache servent à simuler les différentes options de branchements. Le retard
30 d'exécution du microprocesseur graphique 20 sur le module 145 est compensé en

mettant en cache toutes les possibilités de branchement. Pour cela, le module 145 fonctionne essentiellement comme l'unité de contrôle d'un microprocesseur pour faire les opérations relatives aux adresses et la reconnaissance des instructions de contrôle de flux qui effectuent des branchements. Il fait typiquement le traitement

5 suivant: par défaut, il incrémente un compteur ordinal. Il peut aussi affecter ou additionner un offset. En cas de rencontre de plusieurs branchements conditionnels, le module 145 suit les différents parcours possibles du programme tant que les branchements ne sont pas finalement déterminés par le microprocesseur graphique

20.

10 Ce fonctionnement va être maintenant illustré par un exemple. Dans cet exemple, l'état initial du microprocesseur graphique 20 comporte deux variables A et B (par exemple A=10 et B=30) qui ont été obtenues après des calculs dans les registres du microprocesseur et qui ne sont pas encore retournées au dispositif 130. Parallèlement, le module 145 interprète le segment de code suivant:

15 (Adresse0) C=A+B;
 (Adresse1) Si (C>10)
 (Adresse2) alors A=B-A;
 (Adresse3) sinon A=A*B;

Le module 145 accède à la première instruction qu'il envoie directement à

20 son cache, pour qu'elle soit appelée ultérieurement par le microprocesseur graphique 20. Le module 145 suit la séquence d'instructions en termes d'adresse. Ici, il incrémente et passe à l'adresse 1. Il reconnaît alors un branchement conditionnel : si $(A+B) > 10$ aller adresse 2, sinon aller adresse 3. Il charge alors l'instruction à l'adresse 2 et l'instruction à l'adresse 3 dans son cache, sans savoir si la condition est

25 vérifiée afin que les données soient rapidement à la disposition du microprocesseur graphique 20 quelle que soit la branche finalement suivie par l'exécution.

Le fonctionnement des annuaires 139 et 146 va maintenant être illustré par un exemple. Lors de l'import depuis la mémoire 10 d'une donnée, une adresse de la mémoire cache est affectée à la donnée par le module de contrôle mémoire 133.

30 L'association entre ces adresses est enregistrée dans l'annuaire 139. Lorsque la

même donnée est communiquée au microprocesseur graphique 20, le module de commande 145 lui donne une autre adresse virtuelle connue du microprocesseur graphique 20.

Par exemple, dans le cas de la suite d'instructions plus haut, notons 1000
5 l'adresse de A dans la mémoire 10. Le module de contrôle mémoire 133 attribue une nouvelle adresse à A dans la mémoire cache, l'adresse 1. L'annuaire 139 enregistre la correspondance $A \leftrightarrow 1000 \leftrightarrow 1$.

De l'autre côté, pour que le calcul soit effectué par le microprocesseur graphique 20, le module de commande 145 lui transmet les instructions en
10 référant A par une adresse virtuelle que le microprocesseur graphique 20 peut utiliser sur son interface mémoire, par exemple l'adresse 333. L'annuaire 146 enregistre la correspondance $A \leftrightarrow 1 \leftrightarrow 333$. Après exécution de son calcul, le microprocesseur graphique 20 renvoie une demande de stockage de la nouvelle valeur de A ($A=20$) à l'adresse virtuelle 333. A l'aide des annuaires, le module de
15 commande 145 peut alors stocker la valeur de A dans la mémoire cache et la réécrire au point d'adresse 1000 dans la mémoire centrale 10.

Une unité de prévision 148 a pour fonction de différencier les données présentes dans les caches du microprocesseur graphique 20 de celles qui en sont absentes, par exemple en exploitant un protocole de mise en cache et un protocole
20 d'ordonnement du microprocesseur graphique 20. Cette différenciation peut être par exemple indiquée dans l'annuaire 146 et exploitée par le module de commande 145 pour ordonner les tâches confiées au microprocesseur graphique 20 d'une manière qui minimise les requêtes d'accès mémoire de ce dernier. Dans le cas où plusieurs microprocesseurs graphiques sont présents, le contenu de leur cache peut
25 aussi être pris en compte pour déterminer les tâches devant être confiées à chacun.

Pour illustrer le point précédent, si le microprocesseur 20 possède dans ses caches les variables A et B, mais non les variables C et D, il est plus efficace de lancer ou poursuivre un calcul impliquant A et B avant de lancer une tâche impliquant C et D. On notera que dans des calculs parallèles, les variables peuvent être des vecteurs
30 ou des tableaux volumineux.

Le dispositif 130 transfère et/ou génère des instructions à la volée dont le microcode est par exemple de l'assembleur GPU. Il peut aussi comporter un module de gestion non représenté pour commander le microprocesseur graphique 20 via le port bus principal, par exemple de type PCI Express. Les commandes principales, par exemple pour débiter l'exécution du programme, peuvent provenir de l'extérieur via les modules d'entrée-sortie du dispositif 130.

En référence à la figure 4, on décrit maintenant un autre mode de réalisation d'un dispositif d'interfaçage 230 entre une mémoire vive 210 et au moins un microprocesseur graphique 220. Les éléments identiques ou similaires à ceux de la figure 1 portent les mêmes chiffres de référence augmentés de 200. Ici, le dispositif 230 comporte une interface reliée à un concentrateur 80, par exemple un chipset Southbridge, pour accéder à divers périphériques, par exemple un disque dur 250, un modem Ethernet 240 et/ou d'autres appareils 90.

Dans les modes de réalisation ci-dessus, la mise en mémoire cache est un moyen intermédiaire pour donner au microprocesseur graphique un accès indirect à très large bande passante au contenu d'une mémoire centrale, ce qui évite d'avoir à charger l'ensemble des programmes et données dans une mémoire graphique dédiée. La mémoire centrale peut être réalisée avec des technologies similaires aux mémoires graphiques conventionnelles.

Les modes de réalisation ci-dessus font référence à des microprocesseurs graphiques. Toutefois, on peut utiliser de la même manière d'autres types de coprocesseurs programmables, de préférence massivement parallèles, qui présentent des propriétés similaires à un microprocesseur graphique. Par propriétés similaires, on entend ici la subordination à un dispositif maître, notamment un microprocesseur conventionnel, apte à donner des ordres au coprocesseur, la capacité du coprocesseur d'effectuer complètement un calcul indépendamment du dispositif maître à la suite d'un ordre, la capacité du coprocesseur d'interagir avec des périphériques qui lui sont dédiés, par exemple la mémoire graphique conventionnelle ou un autre périphérique, et la capacité de traitement élevée du coprocesseur. Dans des modes de réalisation, cette capacité de traitement élevée

peut aussi résulter du fait qu'un nombre élevé de coprocesseurs sont reliés au dispositif d'interfaçage.

Certains des éléments représentés, notamment les différents modules, peuvent être réalisés sous différentes formes, de manière unitaire ou distribuée, au
5 moyen de composants matériels et/ou logiciels. Des composants matériels utilisables sont les circuits intégrés spécifiques ASIC, les réseaux logiques programmables FPGA ou les microprocesseurs programmables. Des composants logiciels peuvent être écrits dans différents langages de programmation, par exemple C, C++, C#, Sh, GLSL, HLSL, Brook. Cette liste n'est pas exhaustive.

10 Bien que l'invention ait été décrite en liaison avec plusieurs modes de réalisation particuliers, il est bien évident qu'elle n'y est nullement limitée et qu'elle comprend tous les équivalents techniques des moyens décrits ainsi que leurs combinaisons si celles-ci entrent dans le cadre de l'invention.

L'usage du verbe « comporter », « comprendre » ou « inclure » et de ses
15 formes conjuguées n'exclut pas la présence d'autres éléments ou d'autres étapes que ceux énoncés dans une revendication. L'usage de l'article indéfini « un » ou « une » pour un élément ou une étape n'exclut pas, sauf mention contraire, la présence d'une pluralité de tels éléments ou étapes. Plusieurs moyens ou modules peuvent être représentés par un même élément matériel.

20 Dans les revendications, tout signe de référence entre parenthèses ne saurait être interprété comme une limitation de la revendication.

REVENDICATIONS

1. Système de traitement de données (1) comportant :
 - une mémoire centrale (10) apte à stocker un programme exécutable comprenant des instructions et des données utilisables par ledit programme,
 - 5 un microprocesseur graphique (20) comprenant des unités de traitement programmables convenant pour exécuter lesdites instructions et des interfaces mémoire à bande passante élevée (21, 135) convenant pour la communication avec des mémoires rapides,
 - et un dispositif d'interfaçage (30, 130) comprenant :
 - 10 un module de contrôle de la mémoire centrale (133) pour lire et écrire dans la mémoire centrale,
 - une mémoire cache (131, 132) apte à stocker des portions dudit programme exécutable et desdites données,
 - un module de commande (145) apte à :
 - 15 pré-interpréter (161) une séquence d'instructions dudit programme exécutable pour déterminer des adresses de la mémoire centrale devant être appelées lors d'une exécution de ladite séquence d'instructions,
 - charger (162) dans la mémoire cache des instructions ou données stockées auxdites adresses de la mémoire centrale,
 - 20 maintenir (164) un annuaire d'adresses associant les adresses de la mémoire cache correspondant auxdites instructions ou données chargées dans la mémoire cache à des adresses virtuelles utilisables par les interfaces mémoire du microprocesseur graphique, et
 - transférer (165) ladite séquence d'instructions pré-interprétée au microprocesseur
 - 25 graphique en substituant lesdites adresses virtuelles auxdites adresses de la mémoire centrale, de sorte que le microprocesseur graphique exécute ladite séquence d'instructions en produisant des requêtes d'accès mémoire désignant lesdites adresses virtuelles, et
 - un module de communication (134) relié auxdites interfaces mémoire à bande
 - 30 passante élevée du microprocesseur graphique pour recevoir lesdites requêtes

d'accès mémoire et apte à traiter lesdites requêtes d'accès mémoire en lisant ou en réécrivant des instructions ou données dans les positions de la mémoire cache qui sont associées auxdites adresses virtuelles selon ledit annuaire d'adresses.

2. Système selon la revendication 1, caractérisé par le fait que ledit module de commande est apte à :

maintenir (163) un deuxième annuaire d'adresses associant les adresses de la mémoire cache correspondant auxdites instructions ou données chargées dans la mémoire cache aux adresses de la mémoire centrale depuis lesquelles lesdites instructions ou données ont été chargées, et

produire (166) des requêtes d'accès à la mémoire centrale pour transférer des instructions ou données écrites dans la mémoire cache à la suite des requêtes d'accès mémoire produites par le microprocesseur graphique vers des positions de la mémoire centrale qui sont associées aux adresses de la mémoire cache selon ledit deuxième annuaire d'adresses.

3. Système selon la revendication 1, caractérisé par le fait que ledit module de commande (145) est apte à :

décoder des instructions de contrôle de flux contenues dans ladite séquence d'instructions, et

lorsqu'une instruction de branchement est rencontrée, poursuivre la pré-interprétation des instructions en suivant plusieurs branches possibles de la séquence d'instructions jusqu'à ce que le branchement soit finalement déterminé par le microprocesseur graphique.

4. Système selon la revendication 1, caractérisé par le fait que ladite séquence d'instructions comporte des microcodes adaptés au microprocesseur graphique.

5. Système selon la revendication 1, caractérisé par le fait que ledit module de commande est apte à décoder une macro-instruction dans ladite séquence d'instructions pour produire des microcodes adaptés au microprocesseur graphique et à transférer lesdits microcodes au microprocesseur graphique à la place de ladite macro-instruction.

6. Système selon la revendication 1, caractérisé par le fait qu'il comporte une pluralité de microprocesseurs graphiques apte à exécuter des séquences d'instructions respectives, ledit dispositif d'interfaçage comportant une pluralité de modules de commande et une pluralité de modules de communication
5 attribués respectivement à chacun desdits microprocesseurs graphiques.

7. Système selon la revendication 1, caractérisé par le fait que ladite mémoire cache comporte plusieurs éléments de mémoire (131, 132) présentant des niveaux de latence différents, ledit dispositif de commande étant apte à gérer ladite mémoire cache de manière à stocker les instructions et données susceptibles d'être
10 appelées le plus souvent dans un élément de mémoire du niveau de latence le plus bas.

8. Système selon la revendication 1, caractérisé par le fait que ledit module de contrôle de la mémoire centrale est apte à gérer une pile des requêtes d'accès (149) dans la mémoire centrale pour effectuer lesdits accès par rafales.

9. Système selon la revendication 1, caractérisé par le fait qu'il
15 comporte un bus PCI Express relié à un port correspondant (22) du microprocesseur graphique et à un port correspondant du dispositif d'interfaçage.

10. Procédé d'interfaçage entre un microprocesseur graphique et une mémoire centrale, ledit procédé comprenant :

20 pré-interpréter (161) une séquence d'instructions d'un programme exécutable stocké dans ladite mémoire centrale pour déterminer des adresses de la mémoire centrale devant être appelées lors d'une exécution de ladite séquence d'instructions,

charger (162) dans une mémoire cache (131, 132) des instructions ou données
25 stockées auxdites adresses de la mémoire centrale,

maintenir (164) un annuaire d'adresses associant les adresses de la mémoire cache correspondant auxdites instructions ou données chargées dans la mémoire cache à des adresses virtuelles utilisables par des interfaces mémoire du microprocesseur graphique,

30 transférer (165) ladite séquence d'instructions pré-interprétée au microprocesseur

graphique en substituant lesdites adresses virtuelles auxdites adresses de la mémoire centrale, de sorte que le microprocesseur graphique exécute ladite séquence d'instructions en produisant des requêtes d'accès mémoire désignant lesdites adresses virtuelles, et

- 5 traiter (166) lesdites requêtes d'accès mémoire en lisant ou en écrivant des instructions ou données dans les positions de la mémoire cache qui sont associées auxdites adresses virtuelles selon ledit annuaire d'adresses.

1/2

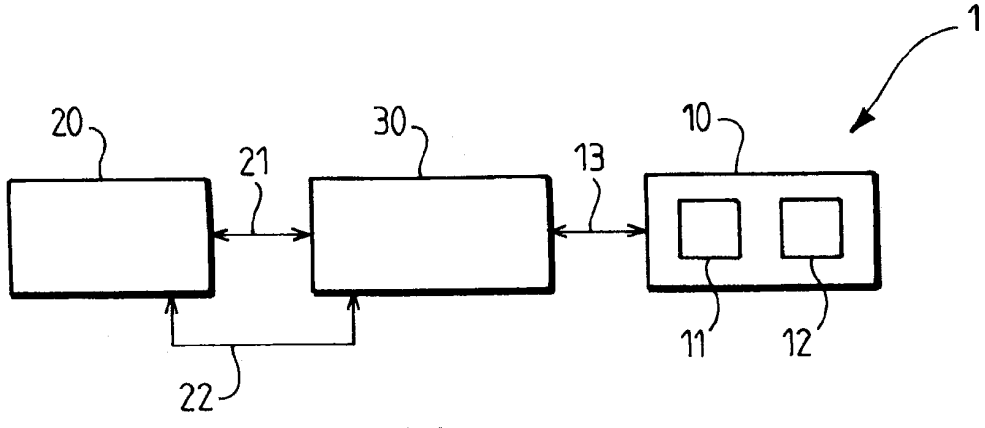


FIG. 1

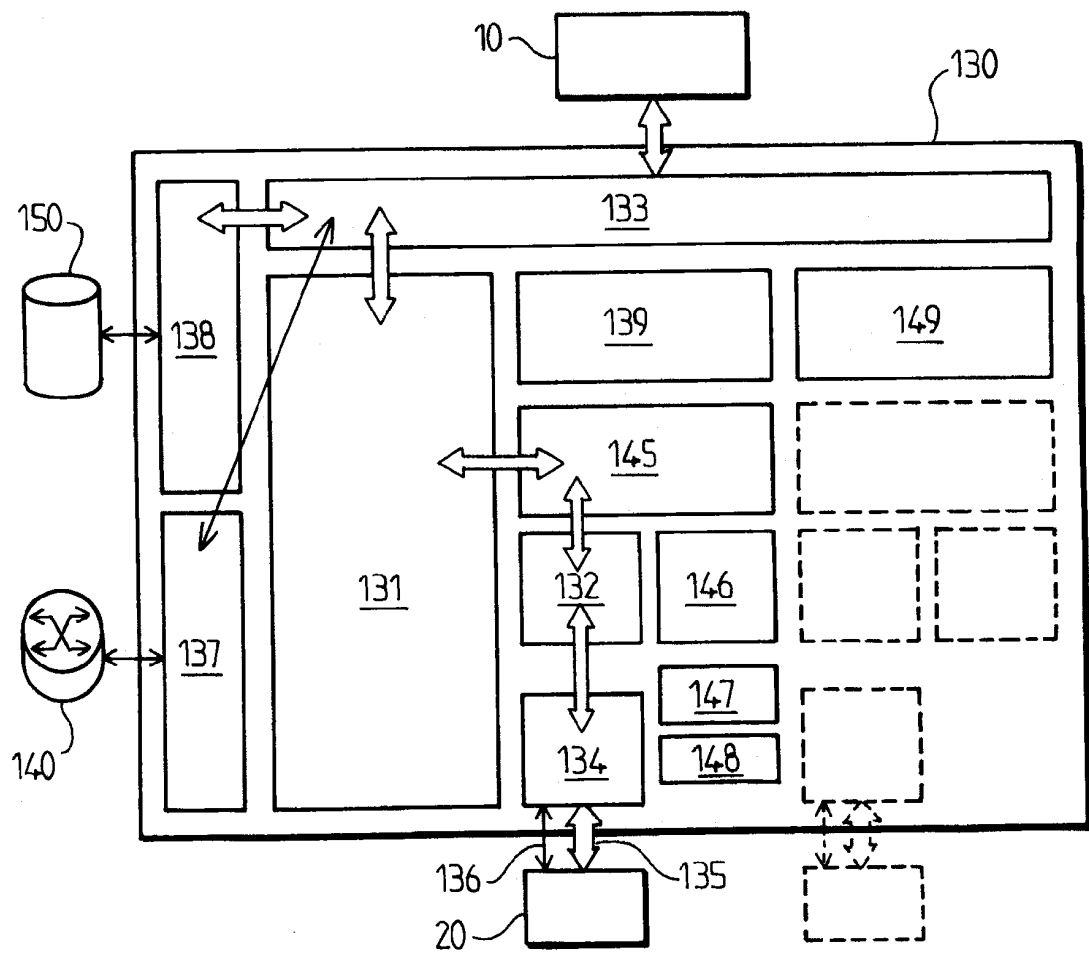


FIG. 2

2/2

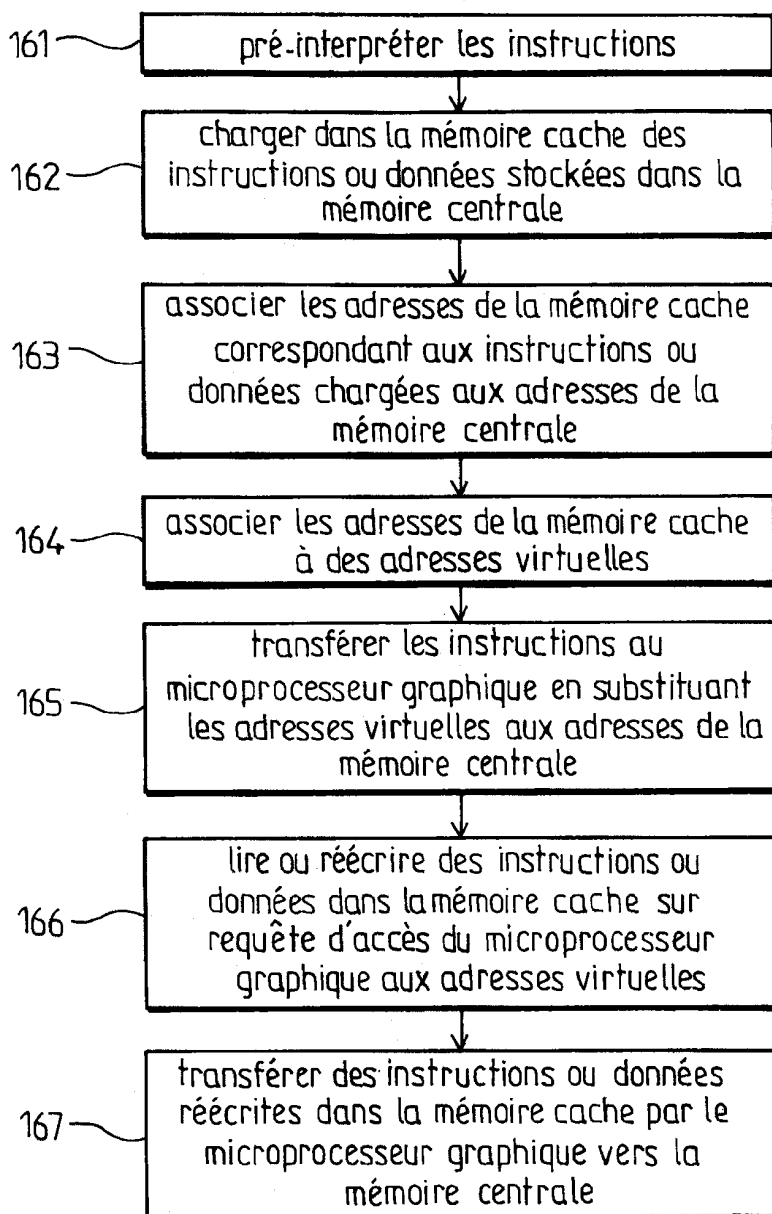


FIG.3

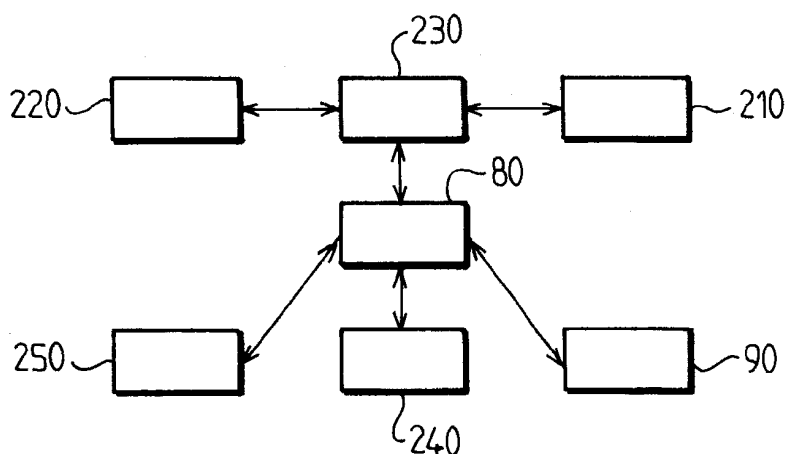


FIG.4



**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

N° d'enregistrement
national

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

FA 736154
FR 1051986

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	US 2002/174300 A1 (KOTANI ATSUSHI [JP] ET AL) 21 novembre 2002 (2002-11-21) * alinéa [0045] - alinéa [0066] * -----	1,10	G06F9/06 G06F9/38 G06F13/10 G06F13/14
A	EP 0 389 175 A2 (FUJITSU LTD [JP]; FUJITSU MICROCOMPUTER SYSTEMS [JP]) 26 septembre 1990 (1990-09-26) * colonne 7 - colonne 16 * -----	1,10	
A	US 5 949 436 A (HORAN RONALD T [US] ET AL) 7 septembre 1999 (1999-09-07) * abrégé * -----	1,10	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
			G06F
Date d'achèvement de la recherche		Examineur	
27 octobre 2010		Klocke, Lynn	
CATÉGORIE DES DOCUMENTS CITÉS			
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		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	

**ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE
RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 1051986 FA 736154**

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 **27-10-2010**

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
US 2002174300	A1	21-11-2002	JP 3676257 B2	27-07-2005
			JP 2002342160 A	29-11-2002

EP 0389175	A2	26-09-1990	AUCUN	

US 5949436	A	07-09-1999	AUCUN	
