

①②

DEMANDE DE BREVET D'INVENTION

A1

②② Date de dépôt : 26.02.03.

③③ Priorité : 01.03.02 US 10086494.

④③ Date de mise à la disposition du public de la
demande : 05.09.03 Bulletin 03/36.

⑤⑥ Liste des documents cités dans le rapport de
recherche préliminaire : *Ce dernier n'a pas été
établi à la date de publication de la demande.*

⑥① Références à d'autres documents nationaux
apparentés :

⑦① Demandeur(s) : HEWLETT PACKARD COMPANY —
US.

⑦② Inventeur(s) : SOLTIS DONALD CHARLES JR.

⑦③ Titulaire(s) :

⑦④ Mandataire(s) : REGIMBEAU.

⑤④ DISPOSITIF ET PROCEDES POUR INTERFACER UNE MEMOIRE CACHE.

⑤⑦ Un système de traitement (100) comprend un proces-
seur (106), une mémoire principale (130), un cache (108) et
une interface crossbar (120) entre le processeur et le cache.
Dans un système à plusieurs processeurs (400), une plura-
lité de plages (138) d'adresses de la mémoire principale
peuvent être cartographiées à une pluralité de caches et
une pluralité de caches peuvent être cartographiés à une
pluralité de processeurs. Par conséquent, un degré impor-
tant de souplesse est fourni pour configurer un système de
traitement.

FR 2 836 732 - A1



DISPOSITIF ET PROCEDES POUR INTERFACER UNE MEMOIRE
CACHE

CHAMP DE L'INVENTION

La présente invention se rapporte de manière générale à des systèmes de traitement et, de manière plus particulière, à la manière d'interfacer une
5 mémoire cache dans des systèmes de traitement.

ARRIERE PLAN DE L'INVENTION

Dans des systèmes à plusieurs processeurs, une pluralité de processeurs et d'autres agents du système tels que des caches et des mémoires peuvent être
10 fabriqués sur une seule puce. Cependant, il peut être difficile de configurer la puce d'une manière qui convient à une variété d'applications cibles. Il pourrait être souhaitable, pour un système donné par exemple, de varier les tailles et les structures de
15 cache conformément à la fréquence selon laquelle des données seraient demandées auprès des adresses de la mémoire principale pendant des applications cibles. Un processeur donné pourrait tirer avantage d'un grand cache, tandis qu'un autre processeur en général
20 pourrait être ralenti par un grand cache. Bien qu'un

accroissement de la taille du cache puisse améliorer la performance d'un processeur, une diminution des retours peut s'installer à mesure que la latence du cache augmente.

5 Il serait souhaitable d'avoir de la souplesse pour configurer la mémoire cache sur une puce à plusieurs processeurs, de sorte qu'une seule configuration de puces puisse à la fois convenir à des applications intensives au point de vue de l'utilisation du cache et
10 à celles faisant une utilisation relativement faible du cache. En outre, il serait souhaitable de pouvoir augmenter la mémoire cache disponible pour un processeur sur une telle puce sans augmenter la latence du cache d'une manière excessive.

15

RESUME DE L'INVENTION

Selon un mode de réalisation, l'invention propose un système de traitement comprenant un processeur, une mémoire principale et un cache configuré pour recevoir des données d'une adresse de la mémoire principale sur
20 une requête pour la donnée faite par le processeur, le système de traitement comprenant une interface à matrice de croisements ("crossbar" ci-dessous) entre le processeur et le cache.

Dans une variante de mode de réalisation,
25 l'invention propose que dans ledit système de traitement la mémoire principale soit commandée par un contrôleur de mémoire, ladite interface crossbar étant alors configurée pour relier le contrôleur de mémoire, le processeur et le cache.

30 Dans une autre variante de mode de réalisation, l'invention propose que dans ledit système de traitement ladite interface crossbar comprenne une pluralité de ports par l'intermédiaire desquels le cache et le processeur sont liés sur la base de

l'adresse en mémoire principale.

Dans une autre variante de mode de réalisation, l'invention propose que dans ledit système de traitement le processeur soit configuré pour associer
5 au moins une plage d'adresses de la mémoire principale au cache.

Dans une autre variante de mode de réalisation, l'invention propose que ledit système comprenne en outre une pluralité de caches, le processeur comprenant
10 une table de plages d'adresses, chaque plage d'adresses étant associée à un cache.

Dans cette variante particulière, l'invention propose aussi que dans ledit système de traitement la table de plages d'adresses soit programmable pour
15 changer au moins une des plages d'adresses et un cache associé au processeur .

Dans une variante de mode de réalisation encore différente, l'invention propose que dans ledit système de traitement, l'interface crossbar soit configurée
20 pour renvoyer la donnée demandée par le processeur au cache et au processeur en parallèle.

Dans une variante de mode de réalisation encore différente, l'invention propose que ledit système de traitement comprenne en outre une pluralité de
25 processeurs reliés au cache par l'intermédiaire de l'interface crossbar.

Dans une variante de mode de réalisation encore différente, l'invention propose que dans ledit système de traitement, l'interface crossbar soit configurée
30 pour fournir une synchronisation du signal pour une transaction asynchrone entre le cache et le processeur.

L'invention propose en outre un procédé pour configurer un système de traitement à plusieurs processeurs, ledit procédé comprenant les étapes
35 consistant à cartographier une pluralité de plages

d'adresses de la mémoire principale à une pluralité de caches (418) ; à cartographier les caches à une pluralité de processeurs (414) ; et à relier les processeurs et les caches en utilisant une interface
5 crossbar (426).

BREVE DESCRIPTION DES DESSINS

La présente invention pourra être comprise de manière plus complète à partir de la description détaillée et des dessins d'accompagnement, dans
10 lesquels :

La figure 1 est un diagramme d'un système de traitement de la technique ancienne ;

La figure 2 est un diagramme d'un système de traitement conformément à un mode de
15 réalisation de la présente invention ;

La figure 3 est un diagramme d'une conceptualisation d'un schéma pour cartographier une mémoire cache conformément à un mode de réalisation ;

20 La figure 4 est un diagramme d'une conceptualisation d'une transaction de requête envoyée par un processeur conformément un mode de réalisation ;

La figure 5 est une conceptualisation d'une transaction de retour envoyée par un
25 cache conformément à un mode de réalisation ;

La figure 6 est un diagramme d'une conceptualisation des transactions de retour envoyées par
30 un contrôleur de mémoire conformément à un mode de réalisation ; et

la figure 7 est un diagramme d'un mode de réalisation d'un système à plusieurs processeurs.

DESCRIPTION DETAILLEE DES MODES DE REALISATION

PREFERENTIELS

La description qui fait suite des modes de réalisation préférentiels est uniquement exemplaire par nature et elle n'est en aucune manière destinée à limiter l'invention, son application ou ses utilisations.

Un système à plusieurs processeurs de la technique ancienne, donné à titre d'exemple, est indiqué d'une manière générale par une référence numérique 10 à la figure 1. Le système 10 comprend une pluralité de processeurs 14, chaque processeur ayant un cache 18 pour tenir des données utilisées par le processeur 14. Chaque cache 18 est configuré pour recevoir des lignes de données demandées par le processeur associé 14 à depuis une mémoire principale 22. Un "crossbar" 26 relie les caches 18 à deux contrôleurs de mémoire 30 par l'intermédiaire de ports 32. Chaque contrôleur 30 commande la lecture des données depuis, et l'écriture des données vers, une moitié de la mémoire principale 22. De manière spécifique, le contrôleur de mémoire 30a commande des adresses dans une moitié 22a et le contrôleur de mémoire 30b commande des adresses dans l'autre moitié 22b, de la mémoire principale 22.

Les adresses pour stocker des données dans les deux moitiés dans la mémoire principale 22 sont cartographiées sur chacun des caches 18. Par exemple, lorsque le processeur 14a demande une donnée à la mémoire principale 22, la demande est dirigée vers le cache associé 18a. Un arrangement d'étiquettes 34 du cache 18a fait l'objet d'une recherche pour déterminer si la donnée est déjà stockée dans un arrangement de données de cache 38 du cache 18a. Si l'opération de cache est une réussite, c'est-à-dire si l'on détermine que la donnée est déjà dans le cache 18a de

l'arrangement de données 38, la donnée est renvoyée depuis le cache 18a vers le processeur 14a.

Si l'opération de cache n'est pas une réussite, c'est-à-dire si l'on détermine que la donnée n'est pas
5 dans le cache 18a de l'arrangement de données 38, la donnée est récupérée auprès de la mémoire principale 22 par l'intermédiaire du contrôleur de mémoire 30 qui commande l'adresse de la donnée demandée. La donnée
récupérée est transmise via le crossbar 26 à
10 l'arrangement de données 38 du cache 18a. La donnée est ensuite transférée du cache 18a vers le processeur 14a.

Des données issues d'une adresse de la mémoire principale 22 peuvent être stockées dans un cache 18 et être par la suite modifiées par le processeur associé
15 14. De manière typique, un schéma de cohérence est utilisé pour conserver une cohérence des données, par exemple, dans le cas où le processeur met à jour son cache associé 18 avec la donnée modifiée. De tels schémas sont conçus pour assurer que la donnée la plus
20 récente soit écrite vers la mémoire principale 22 et/ ou d'autres caches 18. De manière typique, l'information utilisée pour maintenir la cohérence du cache est stockée dans un arrangement d'étiquettes 34 de cache 18. Une telle information peut être mise à
25 jour, par exemple, lorsque le cache 18 reçoit une donnée issue de la mémoire principale 22 et/ ou du processeur associé 14.

Le crossbar 26 rend possible, pour un contrôleur de mémoire 30, de mettre à jour deux caches 18 avec la
30 même donnée en même temps, c'est-à-dire dans le même cycle d'horloge 10 du système. Cependant, chaque processeur 14 obtient la donnée mise à jour de manière indirecte, c'est-à-dire, depuis son cache associé 18 après que le cache associé 18 ait été mis à jour par un
35 contrôleur de mémoire 30.

Accroître la taille d'un cache donné 18 permet au cache 18 de contenir, à tout instant, un nombre de lignes de données issues du stockage principal 22 plus grand qu'avant l'accroissement. Par conséquent, d'une
5 manière générale, la performance d'un processeur 14 peut s'améliorer lorsque le cache associé 18 est agrandi. Cependant, à mesure que la taille du cache 18 s'accroît la latence, c'est-à-dire le temps nécessaire pour renvoyer la donnée vers le processeur associé 14 à
10 partir du cache 18, augmente aussi. La latence augmente au moins en partie parce qu'un processeur 14 est configuré de manière typique, pour attendre pendant une durée temporelle fixée, une requête de données en cours. A mesure que l'on augmente la taille du cache
15 18, ce temps d'attente fixe du processeur est aussi augmenté d'une manière typique pour permettre de rechercher de données dans la zone de mémoire cache agrandie. De manière spécifique, le processeur 14 est typiquement reconfiguré au plan matériel pour augmenter
20 le temps d'attente. Par conséquent la performance et la souplesse du processeur 14 peuvent être limitées par la performance du cache 18, ce qui peut aussi affecter la performance globale du système 10. Ce cas peut se présenter en particulier là où le système 10 réside sur
25 une seule puce.

Un système de traitement conforme à un mode de réalisation de la présente invention est indiqué d'une manière générale par une référence numérique 100 à la figure 2. Le système 100 comprend une pluralité
30 d'agents ou de modules 102 du système, interconnectés pour effectuer des fonctions du système. D'une manière générale, les modules 102 communiquent l'un avec l'autre en (a) émettant des requêtes pour des données et/ ou (b) émettant des données en réponse à de telles
35 requêtes. Comme cela sera décrit plus loin de manière

détaillée, chaque module 102 est identifié dans le système 100 par un identificateur unique du module (ID de module), utilisé pour acheminer des communications ou des transactions entre un expéditeur et un destinataire.

Le système 100 est configuré sur une seule puce 104. Des modules 102 du système 100 comprennent une pluralité de processeurs 106 et une pluralité de mémoires cache ou caches 108. On envisage cependant que d'autres modes de réalisation puissent inclure aussi peu qu'un seul processeur 106 et/ ou un seul cache 108 et que d'autres modes de réalisation puissent être configurés sur plus d'une puce. Chaque cache 108 comprend un arrangement d'étiquettes 110 et un arrangement de données 112.

Une interface crossbar 120 relie des agents du système 102, par exemple les processeurs 106 et les caches 108, par l'intermédiaire d'une pluralité de ports 122. De manière spécifique, les caches 108a, 108b, 108c et 108d accèdent au crossbar 120 par l'intermédiaire de ports 122c, 122d, 122e et 122f respectivement, et les processeurs 106a et 106b accèdent au crossbar 120 par l'intermédiaire des ports 122a et 122b respectivement. Lorsqu'une pluralité de modules communique les uns avec les autres par l'intermédiaire de transactions au travers du crossbar 120, des ID des modules expéditeurs et destinataires sont inclus dans chaque transaction. Les ID de modules sont vérifiés par rapport à une table de routage (non montrée) pour identifier un port de crossbar 122 pour chacun des modules qui communiquent 102. La transaction est ensuite acheminée au travers du crossbar 120 entre les ports 122 convenables. Il est possible de transmettre plus d'une transaction à un instant donné par l'intermédiaire du crossbar 120, et un module 102

peut envoyer des transactions à plus d'un module récepteur 102 au cours du même cycle d'horloge 100. Dans le cas où un module 102 envoie une transaction de manière asynchrone au crossbar 120, le crossbar 120
5 fournit une synchronisation pour une telle transaction.

Une mémoire principale 130 est reliée au crossbar 120 par l'intermédiaire d'un contrôleur de mémoire 132 en un port 122g. Comme cela sera décrit plus loin de manière plus détaillée, des plages d'adresses A, B, C
10 et D de la mémoire principale 130 sont cartographiées sur les caches 108. Dans le présent mode de réalisation donné à titre d'exemple, l'ensemble des plages d'adresses A, B, C et D sont cartographiées sur chacun des caches 108. D'autres cartographies différentes sont
15 toutefois possibles. Toutes les plages ou, en variante, un nombre inférieur à toutes les plages de la mémoire 130 peuvent être cartographiées, par exemple, sur moins que l'ensemble des caches 108. Un cache donné 108 est configuré pour recevoir des données issues des adresses
20 de la mémoire principale 130 cartographiée à ce cache particulier, lorsqu'un processeur 106 demande des données.

D'une manière générale, un processeur donné 106 peut être associé à un ou une pluralité de caches 108
25 et un cache donné 108 peut être associé à un ou une pluralité de processeurs 106, comme nous allons le décrire maintenant. Chacun des processeurs 106 comprend une table programmable 134 des plages d'adresses 138 adressables par le processeur donné 106. Pour chaque
30 plage d'adresses 138, la table 134 comprend un ID de modules 142 identifiant un cache 108 auquel la plage d'adresses 138 est cartographiée.

Par exemple, comme nous allons le décrire de manière plus détaillée plus loin, le processeur 106a
35 obtient une donnée de cache en correspondance à des

plages d'adresses A et B de la mémoire principale par l'intermédiaire du port 122c, qui assure la liaison au cache 108a. Le processeur 106a obtient une donnée de cache en correspondance aux plages d'adresses C et D
5 par l'intermédiaire du port 122d, qui assure la liaison au cache 108b. Le processeur 106b obtient une donnée de cache pour les plages A à D à partir des caches 108a à 108d respectivement, par l'intermédiaire des ports du crossbar 122c à 122f respectivement.

10 Une association des caches 108 au processeur 106, telle que décrite en référence à la figure 2, est illustrée plus en détail à la figure 3, dans laquelle la cartographie de la mémoire principale 130 sur des caches 108 par des processeurs 106 est indiquée d'une
15 manière générale par une référence numérique 200. Comme cela a été décrit plus haut, la table 138 du processeur 106a fait une association 204 des plages A et B de la mémoire avec le cache 108a et des plages C et D avec le cache 108b. La table 138 du processeur 106b fait une
20 association 208 de la plage A de la mémoire avec le cache 108a, de la plage B de la mémoire avec le cache 108c, de la plage C de la mémoire avec le cache 108b et de la plage D de la mémoire avec le cache 108d. (Il est évident que les associations 204 et 208 et les plages
25 A-D de la mémoire sont dessinées dans la figure 3 de manière à conceptualiser leur relation mutuelle en connexion avec la cartographie 200. Par conséquent leurs étendues relativement à la mémoire principale 130 est relativement les unes aux autres ne sont que des
30 approximations à la figure 3.)

Lorsque le processeur 106a demande une donnée stockée dans une adresse à l'intérieur de la plage A d'adresses de la mémoire principale, une transaction de requête, par exemple une requête indiquée d'une manière
35 générale par une référence numérique 300 à la figure 4,

est envoyée au cache 108a. La requête 300 comprend une ID de module 304 identifiant le processeur expéditeur 106a. Une ID de module 308 identifiant le cache destinataire 108a est obtenue à partir de la table 134
5 des plages d'adresses (montrée à la figure 2) et incluse dans la requête 300. La requête 300 comprend aussi l'adresse 312 de la mémoire principale à partir de laquelle la donnée fait l'objet de la requête. Bien sûr, d'autres données peuvent être incluses dans la
10 requête 300, par exemple, pour distinguer la requête 300 de tout autre requête qui peut être en cours d'exécution entre les deux modules 106a et 108a. On comprendra que les figures 4 à 6 représentent des
15 conceptualisations, et que beaucoup d'éléments transactionnels, de données et de formats de commande et de protocoles de transaction sont possibles. La table de routage (non montrée) est utilisée pour faire correspondre les ID des modules 304 et 308 aux ports 122a et 122c respectivement, et le crossbar 120 relie
20 le processeur 106a au cache 108a par l'intermédiaire des ports 122a et 122c.

Une recherche d'étiquette est effectuée dans l'arrangement d'étiquettes 110 du cache 108a, comme cela est connu dans la technique, pour déterminer si la
25 donnée demandée est dans le cache 108a. Si l'opération de cache est une réussite, la donnée demandée est renvoyée par l'intermédiaire du crossbar 120 au processeur 106a dans une transaction de retour de données, par exemple une transaction de retour indiquée
30 d'une manière générale par la référence numérique 320 à la figure 5. La transaction de retour 320 comprend les ID des modules 324 et 328 identifiant les modules expéditeurs et destinataires 108a et 106a, respectivement, ainsi que la donnée 332 demandée par le
35 processeur 106a. Les ID des modules sont vérifiés par

rapport à la table de routage, comme cela a été décrit plus haut, et la transaction de retour 320 est routée par l'intermédiaire des ports 122c et 122a du crossbar 120 vers le processeur 106a.

5 Si l'opération de cache n'est pas une réussite, la requête 300 est expédiée au contrôleur de mémoire 132 qui obtient la donnée demandée à partir de la plage A de la mémoire principale 130 (montrée à la figure 2). Le contrôleur de mémoire 132 renvoie la donnée demandée
10 par l'intermédiaire du crossbar 120 dans deux transactions parallèles, par exemple les transactions indiquées par les références numériques 340 et 344 à la figure 6. La transaction 340 est envoyée au cache 108a et comprend une ID de module 348 identifiant le
15 contrôleur de mémoire expéditeur 132, une ID de module 352 identifiant le cache destinataire 108a et la donnée demandée 356. La transaction 344 est envoyée au processeur 106a et comprend le ID de module 348 du
20 contrôleur de mémoire, la donnée demandée 356 et un ID de module 360 identifiant le processeur destinataire 106a. Le cache 108a met à jour son arrangement de données 112 à l'aide de la nouvelle donnée et met à jour son arrangement d'étiquettes 110 à l'aide de la nouvelle information d'étiquette. La cohérence du cache
25 peut être conservée en utilisant des schémas de cohérence tels que celui qui a été décrit précédemment en connexion avec le système 10 de la technique ancienne. Par exemple, le contrôleur de mémoire 132 peut mettre à jour l'arrangement de données 112 et
30 l'arrangement d'étiquettes 110, de tout autre cache 108 qui avait précédemment demandé la donnée à la même plage A d'adresses de la mémoire.

Lorsque l'on souhaite augmenter la mémoire principale 130 pour une configuration particulière du
35 système de traitement, une ou plusieurs tailles de l'un

ou d'une pluralité de caches 108 peuvent être modifiées de sorte que la mémoire supplémentaire puisse être cartographiée sur le ou les caches 108 sans changer l'interface crossbar 120. Des caches 108 peuvent aussi
5 être ajoutés ou ôtés du système de traitement 100, par exemple, pour accommoder des changements des plages de mémoire qui sont en cours de cartographie vers les caches 108. La table 134 d'un processeur donné 106 est programmable pour augmenter ou réduire le nombre de
10 caches 108 associés au processeur 106 et/ ou pour changer les plages 138 de la mémoire principale qui sont cartographiées sur des caches 108.

Un système à plusieurs processeurs conforme à un autre mode de réalisation de la présente invention est
15 indiqué d'une manière générale par une référence numérique 400 à la figure 7. Le système 400 comprend une pluralité de processeurs 414 reliés à une pluralité de caches 418 par l'intermédiaire d'une pluralité de crossbars 424 joints pour former une interface 424. Une
20 mémoire principale 430 est cartographiée sur les caches 418 et est aussi reliée à l'interface du crossbar 424 par l'intermédiaire de deux contrôleurs de mémoire 434. Des agents supplémentaires du système 400 sont liés à l'interface 424 et comprennent, par exemple, un système
25 d'entrée/ sortie 438.

Les modes de réalisation décrits plus haut rendent possible de modifier facilement une conception particulière de puce, pour accorder les besoins en cache de processeurs et d'applications cibles
30 particuliers. A l'intérieur d'un système à plusieurs processeurs donné, chaque processeur peut être cartographié avec uniquement autant de caches que cela peut être avantageux (qui peuvent différer selon les différents processeurs du système). En outre, un
35 processeur peut être cartographié pour utiliser

différents caches pour des plages différentes de la mémoire principale. Par conséquent, la latence peut être minimisée.

L'interface crossbar décrite plus haut fournit une
5 liaison à vitesse élevée entre des processeurs et des caches. L'interface crossbar rend aussi possible de fournir une communication asynchrone entre un processeur et un cache. Une recherche dans un cache, et une récupération des données dans un cache, peuvent
10 être effectuées de manière plus rapide qu'avec des structures conventionnelles de cache. Les modes de réalisation précédents rendent possibles de mettre à jour une mémoire cache et le processeur associé en parallèle, au lieu d'avoir à déplacer les données vers
15 le cache et de déplacer ensuite les données du cache vers le processeur. Parce que les mémoires caches précédentes peuvent être facilement modifiées en taille pour une configuration particulière à plusieurs processeurs, un processeur peut facilement être
20 configuré avec une taille de cache convenable pour une utilisation particulière. En outre, il est possible de changer le nombre de caches, par exemple d'augmenter le nombre pour une configuration donnée sans augmenter la latence.

25 La capacité décrite plus haut des processeurs à partager des caches (et/ ou des parties de ceux-ci) rend possible une grande variété de cartographies, de caches sur des processeurs et de la mémoire principale sur des caches. Par conséquent, il est possible de
30 configurer une grande variété des caractéristiques d'un système de traitement sans devoir changer l'interface crossbar. Une configuration particulière de puce peut donc être utilisée pour une variété d'applications plus grande que cela ne serait possible avec des
35 configurations ayant des processeurs et des caches

intégrés de manière conventionnelle.

La description de l'invention est purement
exemplaire par nature, et, par conséquent, des
variantes qui ne s'éloignent pas de l'esprit de
5 l'invention sont destinées à être dans la portée de
l'invention. De telles variantes ne devront pas être
considérées comme s'éloignant de l'esprit et de la
portée de l'invention.

REVENDICATIONS

1. Système de traitement (100) comprenant un processeur (106), une mémoire principale (130) et un cache (108) configuré pour recevoir des données d'une adresse de la mémoire principale sur une requête pour
5 la donnée faite par le processeur, le système de traitement comprenant une interface crossbar (120) entre le processeur et le cache.
2. Système de traitement (100) de la revendication 1
10 dans lequel la mémoire principale (130) est commandée par un contrôleur de mémoire (132), l'interface crossbar (120) étant configurée pour relier le contrôleur de mémoire, le processeur (106) et le cache (108).
- 15 3. Système de traitement (100) de la revendication 1 dans lequel l'interface crossbar (120) comprend une pluralité de ports (122) par l'intermédiaire desquels le cache (108) et le processeur (106) sont liés sur la
20 base de l'adresse en mémoire principale.
4. Système de traitement (100) de la revendication 1 dans lequel le processeur (106) est configuré pour associer au moins une plage d'adresses de la mémoire
25 principale (130) au cache (108).
5. Système de traitement (100) de la revendication 1 comprenant en outre une pluralité de caches (108), le processeur (106) comprenant une table de plages
30 d'adresses (134), chaque plage d'adresses (138) étant associée à un cache (108).

6. Système de traitement (100) de la revendication 5 dans lequel la table de plages d'adresses (134) est programmable pour changer au moins une des plages d'adresses (138) et un cache (108) associé au processeur (106).

7. Système de traitement (100) de la revendication 1 dans lequel l'interface crossbar (120) est configurée pour renvoyer la donnée demandée par le processeur (106) au cache (108) et au processeur en parallèle.

8. Système de traitement (100) de la revendication 1 comprenant en outre une pluralité de processeurs (106) reliés au cache (108) par l'intermédiaire de l'interface crossbar (120).

9. Système de traitement (100) de la revendication 1 dans lequel l'interface crossbar (120) est configurée pour fournir une synchronisation du signal pour une transaction asynchrone entre le cache (108) et le processeur (106).

10. Procédé pour configurer un système de traitement à plusieurs processeurs (400) comprenant les étapes consistant à :

cartographier une pluralité de plages (138) d'adresses de la mémoire principale à une pluralité de caches (418) ;

30 cartographier les caches à une pluralité de processeurs (414) ; et

relier les processeurs et les caches en utilisant une interface crossbar (426).

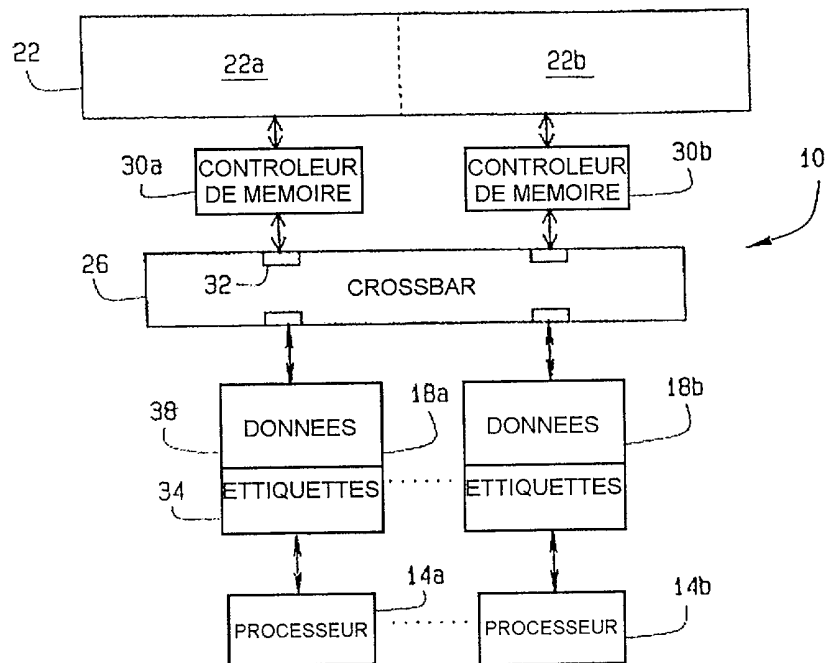
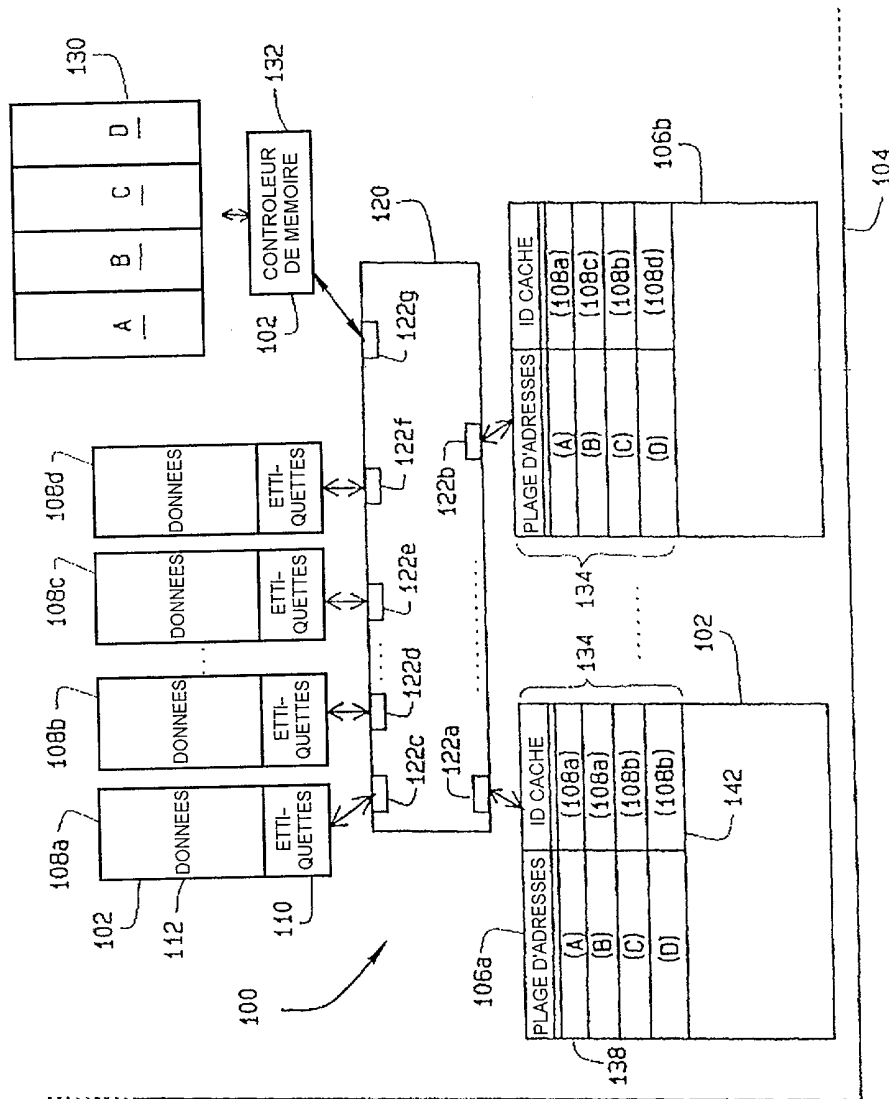


FIG. 1

TECHNIQUE ANCIENNE



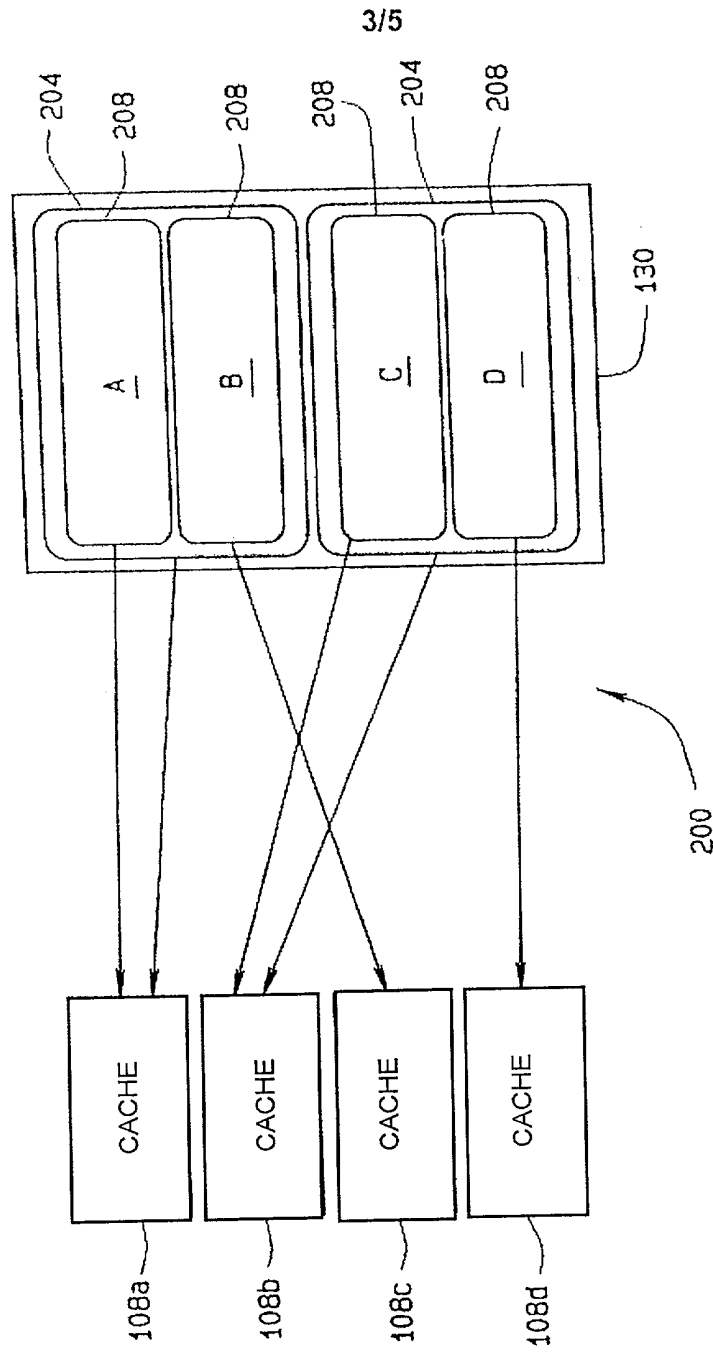


FIG. 3

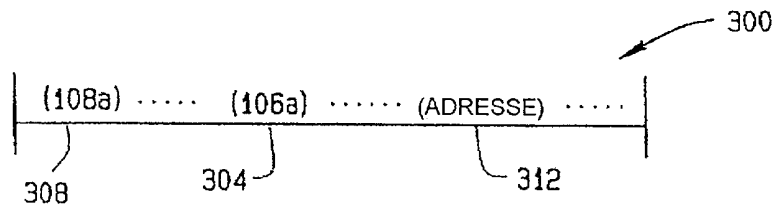


FIG. 4

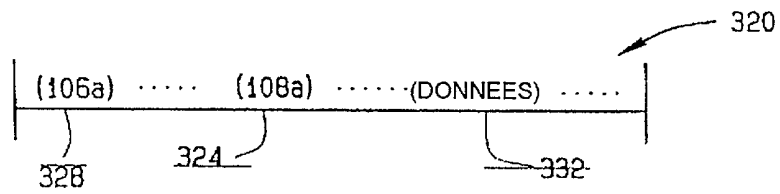


FIG. 5

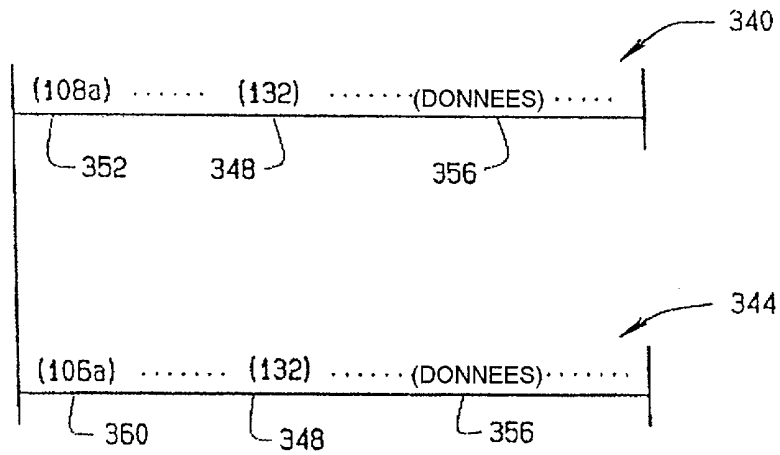


FIG. 6

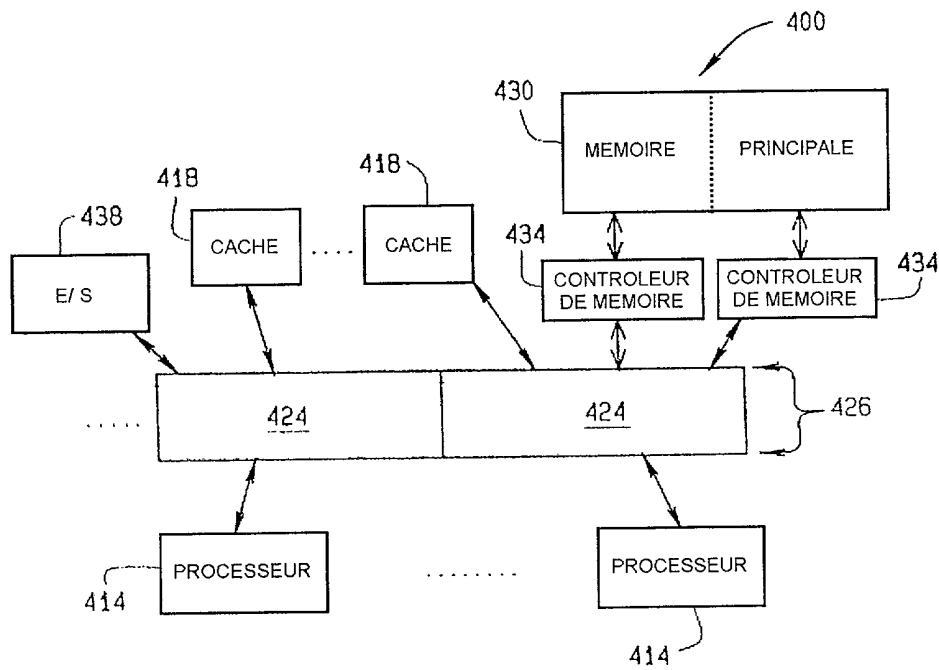


FIG. 7