(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 970 794**

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **11 50500**

(51) Int Cl[8] : ***G 06 F 13/16*** *(2012.01), G 06 F 15/173*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 21.01.11.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 27.07.12 Bulletin 12/30.

(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) :** *COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES* — FR.

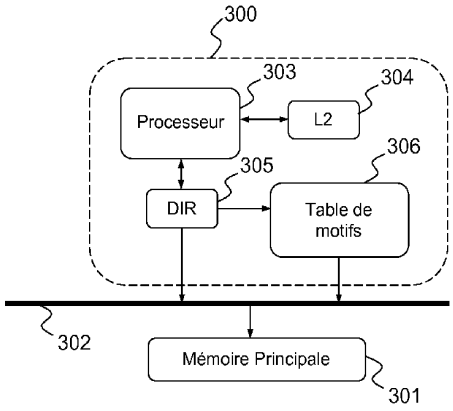
(72) **Inventeur(s) :** CUDENNEC LOIC, KOFUJI JUSSARA, ACQUAVIVA JEAN-THOMAS et CAMIER JEAN-SYLVAIN.

(73) **Titulaire(s) :** COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES.

(74) **Mandataire(s) :** MARKS & CLERK FRANCE.

(54) SYSTEME MULTI-COEURS ET PROCEDE DE COHERENCE DE DONNEES DANS UN TEL SYSTEME.

(57) L'invention a pour objet un système comportant une pluralité de coeurs et d'un bus de communication (302) permettant aux coeurs de communiquer entre eux, un coeur étant composé d'un processeur (303) et d'au moins une zone de mémoire cache (304). Au moins un coeur comporte une table de motifs (306) dans laquelle sont mémorisés un ensemble de motifs, un motif correspondant à une suite d'adresses mémoires associées à une donnée numérique composée de mots binaires mémorisés à ces adresses. Ce coeur comporte en outre des moyens pour faire correspondre l'une des adresses mémoires AdB d'une donnée numérique à un motif lui étant associé lorsque ledit coeur a besoin d'accéder à cette donnée ainsi que des moyens pour transmettre un message unique d'accès à une donnée numérique localisée dans la mémoire cache d'au moins un autre coeur du système, ledit message incluant les adresses mémoires composant le motif de la donnée recherchée.



FR 2 970 794 - A1

SYSTEME MULTI-CŒURS ET PROCEDE DE COHERENCE DE DONNEES DANS UN TEL SYSTEME

L'invention concerne un système multi-cœurs et un procédé de cohérence de données mémorisées dans un tel système.

Elle s'applique notamment au domaine des systèmes multi-cœurs et plus particulièrement du maintien de la cohérence de mémoire cache dans ces architectures.

Un système multi-cœurs est un système électronique comprenant une pluralité d'unités de calcul appelées cœurs. Les processeurs multi-cœurs en sont un exemple. Un tel processeur inclut ainsi plusieurs cœurs dans une même puce électronique et peut comprendre quelques dizaines à quelques centaines de cœurs.

Un système multi-cœurs permet d'effectuer plusieurs traitements en parallèle. Il est alors possible d'atteindre des puissances de calcul supérieures à celle d'un système mono-cœur tout en limitant le phénomène de dissipation thermique. Cependant, de tels systèmes doivent comprendre des moyens pour gérer intelligemment les ressources partagées par ces cœurs, comme par exemple des moyens pour gérer les accès à la mémoire cache de chaque cœur ainsi qu'à la mémoire externe aux cœurs, la mémoire externe étant aussi appelée mémoire centrale ou mémoire principale. Pour rappel, la mémoire cache est un type de mémoire placé habituellement à proximité de la ressource matérielle l'utilisant et dont l'objectif est de mémoriser des copies de données de manière à permettre un accès rapide à cette donnée par ladite ressource. Pour gérer la mémoire cache, des procédés permettant de maintenir de la cohérence des données mémorisées dans cette mémoire sont habituellement utilisés.

Un procédé de maintien de la cohérence de mémoire cache permet notamment la gestion de copies multiples d'une même donnée numérique, lesdites copies étant situées sur les mémoires caches associées à différents cœurs. Le maintien de la cohérence des copies de données permet notamment de simplifier la programmation des systèmes multi-cœurs.

Lorsqu'un défaut de cache est détecté, c'est-à-dire qu'une donnée recherchée par un cœur n'est pas disponible en mémoire cache, ledit cœur de calcul doit rechercher la donnée à l'extérieur, c'est-à-dire soit dans la mémoire cache d'un autre cœur, soit dans la mémoire principale. Pour cela, un cœur utilise habituellement une interface mémoire mettant en œuvre un protocole d'interface mémoire afin accéder aux données mémorisées à l'extérieur dudit cœur. Il résulte de l'exécution du protocole d'interface mémoire l'envoi de messages transitant par exemple sur un bus de communication et permettant aux cœurs de communiquer les uns avec les autres ainsi qu'avec la mémoire principale. Ces différentes opérations induisent pour un cœur la consommation de ressources de calcul pour exécuter des opérations associées au protocole d'interface mémoire. De plus, l'ensemble des messages envoyés peut occuper une partie significative de la bande passante du bus de communication et cela peut être préjudiciable pour un fonctionnement efficace du système multi-cœurs. En effet, si le bus de communication est engorgé, l'application fonctionnant sur le système multi-cœurs est ralentie. Un procédé de maintien de la cohérence de mémoire a notamment pour but de réduire le temps d'exécution lié au protocole d'interface mémoire ainsi que de limiter autant que possible le nombre de messages envoyés.

Un but de l'invention est notamment de pallier les inconvénients précités.

A cet effet l'invention a pour objet un système multi-cœurs comportant une pluralité de cœurs et d'un bus de communication permettant aux cœurs de communiquer entre eux, un cœur étant composé d'un processeur et d'au moins une zone de mémoire cache. Un cœur comporte une table de motifs dans laquelle sont mémorisés un ensemble de motifs, un motif correspondant à une suite d'adresses mémoires associées à une donnée numérique composée de mots binaires mémorisés à ces adresses. Ce cœur comporte en outre des moyens pour faire correspondre l'une des adresses mémoires AdB d'une donnée numérique à un motif lui étant associé lorsque ledit cœur a besoin d'accéder à cette donnée ainsi que des moyens pour transmettre un message unique d'accès à une donnée numérique localisée dans la mémoire cache d'au moins un autre cœur du

système, ledit message incluant les adresses mémoires composant le motif de la donnée recherchée.

Selon un aspect de l'invention, l'ensemble des cœurs du système comporte une table de motifs. La table de motifs d'un cœur est contrôlée par le processeur dudit cœur en collaboration avec un accélérateur matériel, ledit accélérateur ayant pour fonctions le stockage de la table de motifs ainsi que la mise en correspondance entre une adresse mémoire d'entrée AdB et l'un des motifs mémorisés.

Selon un autre aspect de l'invention, la suite d'adresses composant un motif est obtenue par une fonction de motif appliquée à l'adresse mémoire d'entrée AdB.

Les motifs utilisés sont définis, par exemple, de la manière suivante :

$$Desc = f(AdB, L, D)$$

où :

$f(\ )$ représente la fonction du motif, ladite fonction renvoyant les L adresses partant de l'adresse AdB, lesdites L adresses étant espacées de D adresses ;

$Desc$ représente le descripteur de motif, c'est-à-dire la suite d'adresses résultant de l'application de la fonction $f(\ )$ avec le jeu de paramètres *AdB, L et D* ;

AdB représente l'adresse de base, ladite adresse correspondant à la première adresse de la suite d'adresses composant le motif de la donnée recherchée ;

L est un nombre entier correspondant au nombre de mots binaires compris dans la donnée recherchée ;

D correspond au décalage permettant d'aller de l'adresse d'un mot binaire à l'adresse du mot binaire suivant compris dans le motif.

La table de motifs est, par exemple, chargée au démarrage du système. Cette table de motifs peut être une table de hachage.

Dans un mode de réalisation, lorsqu'un cœur comportant une table de motifs requiert un accès mémoire à une donnée associée à une adresse mémoire AbB et qu'un motif est trouvé, ce cœur envoie un message

à un autre cœur dit cœur gestionnaire de motif CGM en charge de la gestion de ce motif, ledit cœur CGM ayant pour fonction de trouver la donnée associée au motif dans sa mémoire cache et de l'envoyer au cœur demandeur.

Chaque cœur du système est, par exemple, identifié par un index, l'index d'un cœur CGM étant déterminé en utilisant l'expression :

CGM = (AdB/ TP) % N

dans laquelle :
CGMi représente l'index du cœur gestionnaire de motifs ;
TP représente la taille des pages mémoires du système ;
AdB représente l'adresse de base de la donnée recherchée ;
N représente le nombre total de cœur du système.

Une table de cohérence peut être mémorisée dans au moins l'un des cœurs du système, ladite table comprenant une indication de la présence ou de l'absence de données numériques dans la mémoire cache d'autres cœurs du système.

Selon un aspect de l'invention, un cœur est dit cœur gestionnaire classique CGC quand il comporte une donnée recherchée par un autre cœur dit cœur demandeur et comportant une table de cohérence, ladite donnée ayant été localisée dans le cœur CGC grâce à ladite table, un cœur CGC étant identifié par le cœur demandeur en utilisant l'expression suivante :

CGCi = (AdB /TL) % N

dans laquelle :
CGCi représente l'index du cœur gestionnaire classique ;
TL représente la taille d'une ligne de mémoire cache.

Les cœurs sont, par exemple, interconnectés par un réseau à topologie de type « mesh ».

L'invention a aussi pour objet un procédé de recherche de donnée pour accès mémoire par un cœur compris dans un système multi-cœurs tel que décrit précédemment, les étapes suivantes étant appliquées lorsqu'un

accès à une donnée mémorisée dans le système est demandé par ledit cœur :

- une étape de recherche de la donnée recherchée dans la mémoire cache dudit cœur ;
- si la recherche est infructueuse, une étape de vérification de la présence de la donnée recherchée dans la mémoire cache d'un autre cœur en utilisant une table de motifs ;
- si un motif associé à la donnée recherchée est trouvé, une étape d'envoi d'un unique message d'accès comportant les adresses du motif au cœur en charge dudit motif.

Le procédé comporte par exemple une étape de vérification de la présence de la donnée recherchée dans la mémoire cache d'un autre cœur en utilisant une table de cohérence.

Avantageusement, la mise en œuvre de l'invention permet une amélioration du temps d'exécution des applications mis en œuvre par des systèmes multi-cœurs, ainsi qu'une empreinte silicium limitée pour le support matériel mettant en œuvre le procédé de maintien de cohérence de cache.

D'autres caractéristiques et avantages de l'invention apparaîtront à l'aide de la description qui suit donnée à titre illustratif et non limitatif, faite en regard des dessins annexés parmi lesquels :

- la figure 1 donne un exemple de puce multi-cœurs ;
- les figures 2A, 2B, 2C et 2D donnent un exemple des différentes phases d'exécution d'une demande d'écriture par un cœur pour maintenir la cohérence de la mémoire cache ;
- la figure 3 présente un exemple d'architecture de cœur selon l'invention ;
- la figure 4 donne un exemple de table de motifs présente dans les cœurs d'un système multi-cœurs selon l'invention ;
- la figure 5 donne un exemple de succession d'opérations et d'envois de messages entre plusieurs cœurs appartenant à un système multi-cœurs mettant en œuvre l'invention ;
- la figure 6 donne un exemple de demande/requête d'écriture avec migration.

La figure 1 donne un exemple de puce multi-cœurs 100 composée de 49 cœurs de calcul 101. Une telle architecture peut comprendre en pratique un nombre quelconque de cœurs, par exemple 64. Elle est habituellement associée à une mémoire principale 102, ladite mémoire principale permettant de mémoriser de grandes quantités de données. Ce type de mémoire implique cependant un temps d'accès beaucoup plus important que pour les mémoires caches.

Les cœurs compris dans ce système multi-cœurs sont interconnectés par un réseau à topologie de type « mesh », c'est-à-dire que les communications sont établies de proches en proches, sans organisation centralisée.

Dans ce type d'architecture, un procédé de maintien de la cohérence des données mémorisées intervient notamment lorsqu'une donnée mémorisée dans la mémoire principale du système se retrouve répliquée dans les mémoires caches des différents cœurs au gré des lectures et des écritures.

Un cœur 103 est habituellement composé d'un processeur de calcul 104, d'une mémoire cache de niveau 1 dite L1 pour la mémorisation d'instructions 105 et de données 106, d'une mémoire cache de niveau 2 dite L2 107, d'un répertoire pour la cohérence de la mémoire cache de niveau 2 108, et de modules d'interfaces avec la mémoire principale 109 et les autres cœurs du système 110. La mémoire physique du système multi-cœurs est partagée et distribuée sur les cœurs avec un espace d'adressage unique.

Dans la suite de la description, le terme « granularité » désigne la taille du plus petit ensemble de bits pris en charge par le protocole de cohérence de cache, un tel ensemble étant appelé unité de cohérence.

L'utilisation d'une granularité fine permet de verrouiller un espace mémoire de petite taille sans affecter les accès aux autres données. Cette stratégie entraîne la gestion d'un volume important de métadonnées et peut avoir pour conséquence des problèmes de stockage et de temps de calcul.

A l'inverse, une granularité trop importante induit un problème de faux partage, là où le verrouillage d'une unité de cohérence interdit l'accès à des données non concernées par la mise-à-jour. Le choix de la granularité d'un protocole de cohérence reste donc avant tout dicté par l'infrastructure

matérielle sous-jacente. La granularité dépend de l'architecture du système multi-cœur.

La granularité peut être choisie, par exemple, égale à la longueur d'une ligne de mémoire cache. Des valeurs typiques de granularité sont 64 ou 128 octets selon l'architecture matérielle.

Dans la suite de la description, le terme « donnée » fait référence à un ensemble de mots binaires, lesdits mots binaires comportant un nombre de bits inférieur ou égal à l'unité de cohérence. Un mot binaire est associé à une mémoire ce qui implique que, si une donnée comporte plus d'un mot binaire, celle-ci est associée à une pluralité d'adresses.

Afin de gérer la cohérence des copies d'une donnée en mémoire cache, les procédés de maintien de la cohérence de cache s'appuient habituellement sur une représentation de l'état du système. Cette représentation est mise en œuvre, par exemple, à l'aide d'une table appelée table ou cache de cohérence 108 et est mémorisée dans chacun des cœurs du système. Pour un cœur donné, cette table contient pour chaque donnée des informations sur sa présence dans la mémoire cache d'autres cœurs du système ainsi que le mode d'accès courant à la donnée. Cette table est composée, par exemple, de lignes comprenant une adresse 112 et un ensemble de bits contenant des informations sur la cohérence 113 de ladite donnée mémorisée à cette adresse 112.

Un exemple de procédé de maintien cohérence de cache est mis en œuvre dans le cadre du protocole MESI, acronyme venant de l'expression anglo-saxonne « Modified Exclusive, Shared, Invalid ». Quand ce protocole est utilisé, le mode d'accès courant à une donnée peut être représenté par l'un des quatre états possibles. Ces quatre états sont définis ci-après :

- l'état « modifié », appelé aussi état M, indique que la donnée présente dans la mémoire cache n'est pas cohérente avec la donnée mémorisée dans la mémoire principale, ce qui implique que la mémoire principale doit être potentiellement mise à jour. Les copies de cette donnée localisées sur les autres cœurs ou en mémoire principale sont potentiellement différentes et ne peuvent être utilisées ;
- l'état « exclusif », appelé aussi état E, indique que la donnée est cohérente, c'est-à-dire qu'elle est identique à celle

présente dans la mémoire principale. De plus, elle n'est présente que dans la mémoire cache du cœur auquel est associé ce cache de cohérence. Dans ce cas, la donnée est en accès exclusif sur ce cœur ;

- l'état « partagé », appelé aussi état S, indique que la donnée est cohérente, en lecture seule sur ce cœur et que des copies dans le même état sont peut-être présentes sur d'autres cœurs ;
- l'état « invalide », appelé aussi état I, indique que la donnée présente sur ce cœur n'est pas à jour.

Les mots binaires d'information de cohérence 113 présents dans la table de cohérence 111 comprennent un champ indiquant quel est l'état de la donnée ainsi qu'un vecteur de B bits indiquant quels sont les cœurs dans lesquels une copie de ladite donnée est mémorisée. A titre d'exemple, la convention suivante peut être choisie : le i-ème bit dudit vecteur indique la présence ou l'absence d'une copie de cette donnée sur le i-ème cœur.

La représentation de l'état du système est gérée habituellement sur le principe d'un catalogue centralisé par un cœur particulier désigné habituellement par l'expression anglo-saxonne « Home-Node » et appelé dans la suite de la description cœur gestionnaire classique CGC. Lorsque le système multi-cœurs atteint une échelle importante, par exemple lorsque le système comporte plusieurs dizaines de cœurs, ce cœur gestionnaire est alors très sollicité et des problèmes de performances surviennent. Le catalogue peut alors être distribué en différentes vues locales présentes sur plusieurs voir sur la totalité des cœurs du système, chacun des cœurs étant responsable d'un sous-ensemble de données. Dans ce cas, chaque cœur du système peut être vu comme un cœur CGC, par exemple pour une demande d'accès en lecture ou écriture à une donnée particulière par un cœur demandeur.

Les figures 2A, 2B, 2C et 2D donnent un exemple des différentes phases exécutées lors d'une demande d'écriture par un cœur pour maintenir la cohérence de la mémoire cache.

Ces phases correspondent à la transmission de messages entre les différents cœurs et à l'exécution par les cœurs des processus associés. Dans cet exemple, un cœur appelé cœur demandeur CD 200 a un besoin d'écriture de données. A cette fin, il contacte un cœur gestionnaire classique CGC 201. Le cœur gestionnaire CGC contient des informations permettant de savoir quels sont les cœurs comportant une copie de la donnée recherchée, c'est-à-dire de la donnée à mettre à jour.

Dans un premier temps, comme illustré sur la figure 2A, le cœur demandeur 200 envoie un message 202 au cœur CGC 201. Ledit message comporte l'adresse de la donnée recherchée.

Dans un deuxième temps et comme illustré sur la figure 2B, des messages d'accès en mode exclusif 203, 204, 205, 206 sont envoyés aux cœurs 207, 208, 209, 210 ayant en leur possession une copie de la donnée à mettre à jour en leurs précisant l'adresse mémoire de ladite donnée.

Dans un troisième temps et comme illustré sur la figure 2C, les cœurs 207, 208, 209, 210 invalident leur copie et le notifient 211, 212, 213, 214 au cœur CGC 201 en retour.

Dans un quatrième temps et comme illustré sur la figure 2D, une autorisation d'accès en écriture 215 est transmise au demandeur, potentiellement avec une mise à jour de la donnée.

Il apparaît dans cet exemple qu'un nombre M significatif de messages est requis pour effectuer une demande d'accès en écriture. Certaines applications requièrent la lecture et/ou l'écriture de données composées de mots binaires mémorisés à des emplacements mémoire qui ne sont pas contigües, c'est-à-dire que les adresses associées à ces mots ne se suivent pas. C'est le cas par exemple de nombreuses applications de traitement d'image.Dans le cadre de certaines applications les adresses associées aux mots binaires composant une donnée sont prévisibles et régulières. Un exemple de telle application est l'application de calcul de l'image intégrale, comme décrite dans l'article de B. Kisacanin intitulé Integral Image Optimizations for Embedded Vision Applications, SSIAI 2008, IEEE Southwest Symposium on Image Analysis and Interpretation, pp.181-184, 24-26 March 2008. Le traitement consiste en la somme cumulative d'une matrice sur ses deux dimensions. Pour cela, l'algorithme doit accéder à des blocs rectangulaires de l'image à traiter.

Les procédés de maintien de cohérence de cache classiques opèrent individuellement sur chacun des mots binaires composant une donnée.

Dans le cadre d'applications de traitement d'image, l'image est mémorisée en plaçant les lignes L les unes à la suite des autres dans l'espace mémoire du système utilisé. Dans un système multi-cœurs, un nombre donné M de message est requis pour un accès mémoire. La lecture d'une colonne d'images dans une image composée de C colonnes et de L lignes de pixels entraîne alors l'émission de L×M messages, car la donnée recherchée comporte L mots binaires. Avantageusement, l'invention permet d'améliorer les performances des procédés classiques de maintien de la cohérence de cache en prenant en compte les caractéristiques prévisibles des accès en mémoire afin de réduire le nombre de messages émis. Il est notamment possible de passer de L×M messages émis à M messages émis pour une demande d'accès en écriture, en lecture ou une demande d'invalidation par exemple.

La figure 3 présente un exemple d'architecture de système multi-cœurs selon l'invention. Dans cet exemple, un seul cœur 300 est représenté. Celui-ci peut transmettre et/ou recevoir des données comprises dans la mémoire principale 301 à l'aide d'un bus de communication 302. Dans un but de clarté de l'exposé, un seul cœur est représenté sur cette figure, mais l'invention concerne un système multi-cœurs. Ainsi, une pluralité de cœurs peut accéder à la mémoire principale ainsi qu'à la mémoire cache des autres cœurs du système par l'intermédiaire dudit bus de communication 302.

Préférentiellement, le système multi-cœur selon l'invention se base sur un réseau de type « mesh » afin de permettre une gestion de la cohérence de cache performante, et ce même pour un nombre de cœurs important.

Le cœur 300 comprend processeur de calcul CPU 303, ledit CPU comprenant une mémoire cache privée de niveau L1, une mémoire cache partagée de niveau L2 304, une table de cohérence 305 et une autre table 306 appelée table de motifs 306. La table des motifs est une zone du cœur dans laquelle sont mémorisés des motifs. Un motif correspond à un ensemble d'adresses associées à une donnée. Pour obtenir cet ensemble

d'adresses, une fonction de motif est utilisée. Cette fonction permet de déterminer les adresses d'un motif à partir de paramètres comme par exemple une adresse de base AdB. Les motifs sont utilisés pour accéder rapidement à une donnée composée de plusieurs mots binaires. L'utilisation de ces motifs est particulièrement utile lorsque lesdits mots binaires sont mémorisés à des adresses qui ne se suivent pas. La table de motifs 306 est destinée à la mise en œuvre des accès mémoires. La table des motifs peut être gérée entièrement à partir du processeur du cœur 303 ou bien en collaboration avec un accélérateur matériel 306. Cet accélérateur matériel peut être mis en œuvre dans le cœur, ledit accélérateur ayant pour fonctions le stockage de la table de motifs ainsi que la mise en correspondance entre une adresse d'entrée et l'un des motifs mémorisés.

Un cœur comprend habituellement plusieurs zones de mémoire cache. Les données de cache de niveau L1 sont toujours présentes dans le cache de niveau L2.

Lorsque le cœur cherche à accéder à une donnée, un procédé comportant plusieurs étapes peut être appliqué. Un exemple est donné ci-après.

Le procédé comporte une première étape pendant laquelle il est vérifié si la donnée recherchée se trouve dans le cache de niveau L1.

Si la donnée y est trouvée, le procédé se termine.

Si la donnée est absente, un défaut de cache de niveau L1 est détecté.

Une deuxième étape a ensuite pour fonction de vérifier si la donnée recherchée se trouve dans le cache de niveau L2.

Si la donnée y est trouvée, le procédé se termine.

Si la donnée est absente, un défaut de cache de niveau L2 est détecté.

En cas de défaut de cache de niveau L2, une étape s'appuie sur l'analyse de la table de cohérence 305. Ladite table est, par exemple, analysée ligne par ligne.

Si l'adresse de la donnée est trouvée dans la table de cache, le processeur pour accéder à cette donnée et l'exécution du procédé se termine.

Si l'adresse de la donnée n'y est pas trouvée, un défaut de table de cache est détecté.

En cas de défaut de table de cache, une étape du procédé a pour fonction d'interroger le tableau de motifs 306 et si un motif est trouvé, d'envoyer un message permettant d'obtenir les données associées à l'application du motif et d'avoir accès ainsi à la donnée recherchée. L'utilisation des motifs et de la table des motifs est décrite ci-après.

Dans le cadre de l'invention, les motifs sont utilisés pour déterminer rapidement la localisation dans le système multi-cœurs des mots binaires composant une donnée. Différents formats de motifs peuvent être envisagés.

Préférentiellement, les motifs peuvent être déterminés en tenant compte de deux critères.

Le premier critère consiste à maximiser la concision de la fonction associée au motif. Plus la fonction associée au motif est concise, plus l'espace mémoire nécessaire à son stockage est réduit.

Le deuxième critère consiste à maximiser l'expressivité du format et les différentes classes de motifs pouvant être concernées. Des motifs complexes peuvent ainsi être mémorisés. Ces deux critères tendent malheureusement à être antagonistes.

A titre d'exemple, le motif simple suivant peut être utilisé :

$$Desc = f(AdB, L, D) \qquad (1)$$

dans lequel :

$f(\ )$ représente la fonction du motif. Cette fonction renvoie, par exemple, les L adresses partant de l'adresse AdB, lesdites L adresses étant espacées de D ;

$Desc$ représente le descripteur de motif, c'est-à-dire la suite d'adresses résultant de l'application de la fonction $f(\ )$ avec le jeu de paramètres AdB, *L et D* ;

AdB représente l'adresse de base, ladite adresse correspondant à la première adresse de la suite d'adresses composant le motif de la donnée recherchée ;

L est un nombre entier correspondant au nombre de mots binaires compris dans la donnée recherchée ;

D correspond au décalage permettant d'aller de l'adresse d'un mot binaire à l'adresse du mot binaire suivant compris dans le motif. En d'autres termes, D correspond à la distance entre deux adresses consécutives.

En utilisant le motif donné en exemple (1), il est par exemple possible d'accéder aux L pixels d'une colonne d'une image comportant C colonnes de pixels en indiquant que la longueur du motif est L et le décalage est C.

La sélection des motifs pour une application donnée peut être réalisée par analyse des accès mémoires lors d'exécutions de ladite application. Une fois que les motifs ont été sélectionnés, ces derniers peuvent être implémentés ou mémorisés dans une table de motifs.

Ces analyses peuvent être mises en œuvre en utilisant, par exemple, des techniques de profilage de l'application considérée.

La figure 4 donne un exemple de table de motifs. Comme explicité précédemment, l'ensemble des motifs associés à l'application mise en œuvre par le système multi-cœurs est mis à la disposition des cœurs de calcul. Pour cela, une table de motifs est présente dans chaque cœur. La table de motifs est, par exemple, chargée au démarrage du système. Une table de hachage peut être utilisée en tant que table de motifs dans la mémoire de l'accélérateur matériel chargé de la gestion de la table de motif lorsqu'un tel accélérateur est utilisé.

Une table de motifs ayant la structure d'une table de hachage possède la structure d'une mémoire cache avec une clé, ladite clé étant aussi désignée par le mot anglais « trigger ». La table de hachage permet de faire correspondre l'adresse à vérifier AdB 400 à l'un des motifs mémorisés 402. Pour cela, une fonction de hachage est utilisée. Elle est appliquée pour chaque clé de la table des motifs. Cette fonction utilise en entrée l'une des clés 401 et l'adresse recherchée AdB 400 comme paramètres d'entrées. Elle renvoie comme résultat un booléen indiquant si le motif associé à l'adresse

AdB est présent dans la table de motifs ou non. Si le motif est présent, l'adresse AdB est comparée à l'ensemble des adresses de base des motifs mémorisés AdB1, AdB2, AdB3, AdB4 afin de déterminer quel est le bon motif associé à l'adresse recherchée. L'utilisation d'une telle fonction de hachage permet de ne comparer l'adresse recherchée AdB aux adresses AdB1, AdB2, AdB3, AdB4 des motifs mémorisés que lorsqu'il est certain qu'un motif dont l'adresse de base est égale à AdB est mémorisé dans la table de motifs. Cela a pour avantage de réduire la complexité de calcul requise pour la recherche d'un motif mémorisé à partir de l'adresse recherchée AdB.

A titre d'exemple, si un cœur recherche la donnée présente à l'adresse @11 et que cette adresse correspond à l'une des adresses de base mémorisées 402 dans la table de motifs, un message est envoyé au processeur du cœur par un module chargé de la gestion de la table de motifs dudit cœur, ledit message indiquant qu'un motif a été trouvé et lui transmet le descripteur de ce motif.

En utilisant l'exemple de motif de l'expression (1), le descripteur correspondant à l'adresse de base @11 avec L=4 et D=2 sera égale à {@ 11, 4, 2}. Dans cet exemple, le descripteur de motif sera égal à :

$$f(@11,4,2) = \{@11, @13, @15, @17\} \quad (2)$$

Dans un mode de réalisation préféré, le procédé de maintien de la cohérence de mémoire cache selon l'invention est mis en œuvre dans un système multi-cœurs en s'appuyant d'une part sur une partie logicielle exécutée par les processeurs des cœurs et d'autre part sur un accélérateur matériel dédié à la gestion des motifs. La partie logicielle correspond à un protocole hybride prenant en compte les motifs d'accès mémoire. La partie logicielle comprend, par exemple, les moyens suivants :

- des moyens pour différencier les requêtes d'accès mémoire classiques des requêtes spéculatives ;
- des moyens pour émettre des requêtes dites spéculatives permettant de lire toutes les adresses d'un motif à partir d'une adresse de base ;

- les requêtes spéculatives envoient des messages par page (granularité de page).

La figure 5 donne un exemple de succession d'opérations et d'envois de messages entre plusieurs cœurs appartenant à un système multi-cœurs mettant en œuvre l'invention.

Dans cet exemple, un cœur appelé cœur demandeur CD 515 a besoin de lire une donnée numérique pour pouvoir exécuter un processus dont il a la charge.

Si cette donnée n'est pas dans sa mémoire cache de niveau L1, il va vérifier si ladite donnée n'est pas présente dans sa mémoire cache de niveau L2 500. Suite à cette vérification, soit la donnée est trouvée et le processeur du cœur demandeur 515 y accède 501, soit la donnée n'y est pas et un défaut de cache est détecté.

Lorsqu'un défaut de cache est détecté, le processeur du cœur demandeur 515 va tenter de trouver la donnée recherchée ailleurs que dans sa mémoire cache.

Les tables de cohérence et de motifs sont alors consultées 502.

Si suite à la consultation de la table de cohérence l'adresse correspondant à la donnée recherchée est trouvée, un message d'accès est envoyé 504 à un cœur gestionnaire classique CGC 514 associé à l'adresse recherchée AdB, un cœur gestionnaire classique ne prenant en compte la lecture de données qu'à une adresse unique. Le gestionnaire classique CGC 514 vérifie alors si la donnée n'est pas présente dans sa mémoire cache 509. Si elle l'est, un message de confirmation ACK est envoyé au cœur demandeur 510 avec la donnée recherchée, sinon un accès externe à la mémoire principale 511 permet d'acquérir la donnée et un message de confirmation ACK 512 est ensuite envoyé au cœur demandeur avec la donnée.

Si après consultation de la table de cohérence la donnée n'est pas trouvée, c'est au tour de la table de motifs d'être consultée par le cœur demandeur. Si suite à la consultation 502 de la table de motifs, l'adresse correspondant à la donnée recherchée est trouvée, un message appelé message spéculatif est envoyé au cœur gestionnaire du motif CGM 513. Ce

message contient le descripteur du motif correspondant à la donnée recherchée ainsi que son adresse de base, celui-ci étant normalement inclus dans ledit descripteur. Le cœur CGM 513 vérifie alors que la donnée est présente dans sa mémoire cache.

Si c'est le cas, un message ACK de confirmation 506 est envoyé au cœur demandeur simultanément avec la donnée requise dès que la donnée recherchée est retrouvée.

Au contraire, si la donnée n'est pas présente en mémoire cache, celle-ci est acquise 507 par un accès externe à la mémoire principale. Un message ACK de confirmation 508 est alors transmis au cœur demandeur avec la donnée recherchée. Sur le coeur CGM 513, le processus de recherche des données retourne en priorité la donnée demandée, puis itère sur la suite du motif. C'est pour cette raison que la requête est qualifiée de spéculative.

Il apparaît alors que l'introduction d'une table de motifs sur chaque cœur permet de mettre en place deux flux de messages pour les accès aux données. Un premier flux est généré par des accès à des données qui ne sont pas référencées dans la table des motifs.

Un second flux de messages est généré par les accès aux adresses présentes dans la table des motifs.

Dans une architecture multi-cœurs, les cœurs sont habituellement associés à des index. Ainsi, chaque cœur appartenant à un système multi-cœurs comportant N cœurs sera associé à un index allant de 1 à N par exemple.

A titre d'exemple, des fonctions judicieusement choisies permettent au cœur demandeur de retrouver l'index du cœur gestionnaire classique CGC et du cœur gestionnaire de motifs. Pour retrouver le cœur gestionnaire classique associé à une donnée ayant pour adresse de base AdB, le cœur demandeur peut utiliser l'expression suivante :

$$CGCi = (AdB / TL) \% N \qquad (3)$$

Cette expression découle de l'algorithme de round robin appliqué avec une granularité de la taille d'une ligne de cache TL. CGCi représente l'index du cœur gestionnaire classique.

Dans ce cas, la fonction pour retrouver le cœur gestionnaire de motifs CGM est :

$$CGMi = (AdB/TP) \% N \quad (4)$$

CGMi représente l'index du cœur gestionnaire de motifs. L'algorithme de round-robin est, cette fois-ci, appliqué avec une granularité de la taille TP des pages. La granularité des données, c'est-à-dire l'unité de cohérence, reste cependant égale à la taille d'une ligne de cache, le but étant d'éviter le problème du faux-partage, c'est-à-dire la saturation du système par un grand nombre de demandes de lecture ou d'écriture, par exemple quand tous les cœurs du système font une demande de lecture au même cœur.

La figure 6 donne un exemple de demande/requête d'écriture avec migration de données.

La migration de données permet de rapprocher les données mémorisées en mémoire cache des cœurs les utilisant pour que leur accès soit facilité.

Un cœur demandeur 600 émet une requête en écriture à l'adresse @a. Dans cet exemple, un défaut de cache et détecté. Le tableau de motifs du cœur 600 est consulté et une correspondance de motif est identifiée, le descripteur du motif étant {@a, @b, @c}. Ce descripteur est alors transmis au cœur gestionnaire de motif CGM approprié 601, c'est-à-dire au cœur associé au motif identifié par le cœur demandeur 600.

En lecture partagée, le cœur CGM 601 envoie un message au(x) cœur(s) 603 comportant le motif, ledit message comportant la donnée à écrire correspondant aux adresses du motif.

Afin d'accélérer l'accès aux données des adresses @a, @b et @c par d'autres cœurs demandeurs 604, 605, le cœur CGM 601 envoie également un message de migration au cœur contenant ces données afin que celui-ci puisse les copier dans la mémoire cache d'un autre cœur 606

plus proche des autres demandeurs. Les demandeurs proches de ce cœur pourront alors accéder rapidement à ces données. En d'autres termes, le cœur CGM 601 a anticipé la demande d'adresse spéculative @a, @b, @c, la conséquence étant la migration des données associées.

REVENDICATIONS

1- Système multi-cœurs comportant une pluralité de cœurs et d'un bus de communication (302) permettant aux cœurs de communiquer entre eux, un cœur étant composé d'un processeur (303) et d'au moins une zone de mémoire cache (304) caractérisé en ce que :

- au moins un cœur comporte une table de motifs (306) dans laquelle sont mémorisés un ensemble de motifs, un motif correspondant à une suite d'adresses mémoires associées à une donnée numérique composée de mots binaires mémorisés à ces adresses ;
- ce cœur comporte en outre des moyens pour faire correspondre l'une des adresses mémoires AdB d'une donnée numérique à un motif lui étant associé lorsque ledit cœur a besoin d'accéder à cette donnée ainsi que des moyens pour transmettre un message unique d'accès à une donnée numérique localisée dans la mémoire cache d'au moins un autre cœur du système, ledit message incluant les adresses mémoires composant le motif de la donnée recherchée.

2- Système selon la revendication 1 dans lequel l'ensemble des cœurs du système comporte une table de motifs (306).

3- Système selon l'une quelconque des revendications précédentes dans lequel la table de motifs d'un cœur (306) est contrôlée par le processeur dudit cœur (303) en collaboration avec un accélérateur matériel, ledit accélérateur ayant pour fonctions le stockage de la table de motifs ainsi que la mise en correspondance entre une adresse mémoire d'entrée AdB et l'un des motifs mémorisés.

4- Système selon l'une quelconque des revendications précédentes dans lequel la suite d'adresses composant un motif est obtenue par une fonction de motif appliquée à l'adresse mémoire d'entrée AdB.

5- Système selon l'une quelconque des revendications précédentes dans lequel les motifs utilisés sont définis de la manière suivante :

$$Desc = f(AdB, L, D)$$

où :
$f(\ )$ représente la fonction du motif, ladite fonction renvoyant les L adresses partant de l'adresse AdB, lesdites L adresses étant espacées de D adresses ;
$Desc$ représente le descripteur de motif, c'est-à-dire la suite d'adresses résultant de l'application de la fonction $f(\ )$ avec le jeu de paramètres *AdB, L et D* ;
AdB représente l'adresse de base, ladite adresse correspondant à la première adresse de la suite d'adresses composant le motif de la donnée recherchée ;
L est un nombre entier correspondant au nombre de mots binaires compris dans la donnée recherchée ;
D correspond au décalage permettant d'aller de l'adresse d'un mot binaire à l'adresse du mot binaire suivant compris dans le motif.

6- Système selon l'une quelconque des revendications précédentes dans lequel la table de motifs est chargée au démarrage du système.

7- Système selon l'une quelconque des revendications précédentes dans lequel la table de motifs est une table de hachage.

8- Système selon l'une quelconque des revendications précédentes dans lequel lorsqu'un cœur comportant une table de motifs requiert un accès mémoire à une donnée associée à une adresse mémoire AbB et qu'un motif est trouvé, ce cœur envoie un message à un autre cœur dit cœur gestionnaire de motif CGM en charge de la gestion de ce motif, ledit cœur CGM ayant pour fonction de trouver la donnée associée au motif dans sa mémoire cache et de l'envoyer au cœur demandeur.

9- Système selon la revendication 8 caractérisé en ce que chaque cœur du système est identifié par un index, l'index d'un cœur CGM étant déterminé en utilisant l'expression :

CGM = (AdB/ TP) % N

dans laquelle :
CGMi représente l'index du cœur gestionnaire de motifs ;
TP représente la taille des pages mémoires du système ;
AdB représente l'adresse de base de la donnée recherchée ;
N représente le nombre total de cœur du système.

10- Système selon l'un quelconque des revendications précédentes caractérisé en ce qu'une table de cohérence (305) est mémorisée dans au moins l'un des cœurs du système, ladite table comprenant une indication de la présence ou de l'absence de données numériques dans la mémoire cache d'autres cœurs du système.

11- Système selon la revendication 10 caractérisé en ce qu'un cœur est dit cœur gestionnaire classique CGC quand il comporte une donnée recherchée par un autre cœur dit cœur demandeur et comportant une table de cohérence, ladite donnée ayant été localisée dans le cœur CGC grâce à ladite table, un cœur CGC étant identifié par le cœur demandeur en utilisant l'expression suivante :

CGCi = (AdB /TL) % N

dans laquelle :
CGCi représente l'index du cœur gestionnaire classique ;
TL représente la taille d'une ligne de mémoire cache.

12- Système selon l'une quelconque des revendications précédentes caractérisé en ce que les cœurs sont interconnectés par un réseau à topologie de type « mesh ».

13- Procédé de recherche de donnée pour accès mémoire par un cœur compris dans un système multi-cœurs selon l'une quelconque des revendications précédentes, les étapes suivantes étant appliquées lorsqu'un accès à une donnée mémorisée dans le système est demandé par ledit cœur :

- une étape de recherche de la donnée recherchée dans la mémoire cache dudit cœur ;
- si la recherche est infructueuse, une étape de vérification de la présence de la donnée recherchée dans la mémoire cache d'un autre cœur en utilisant une table de motifs ;
- si un motif associé à la donnée recherchée est trouvé, une étape d'envoi d'un unique message d'accès comportant les adresses du motif au cœur en charge dudit motif.

14- Procédé selon la revendication 13 caractérisé en ce qu'il comporte une étape de vérification de la présence de la donnée recherchée dans la mémoire cache d'un autre cœur en utilisant une table de cohérence (305).

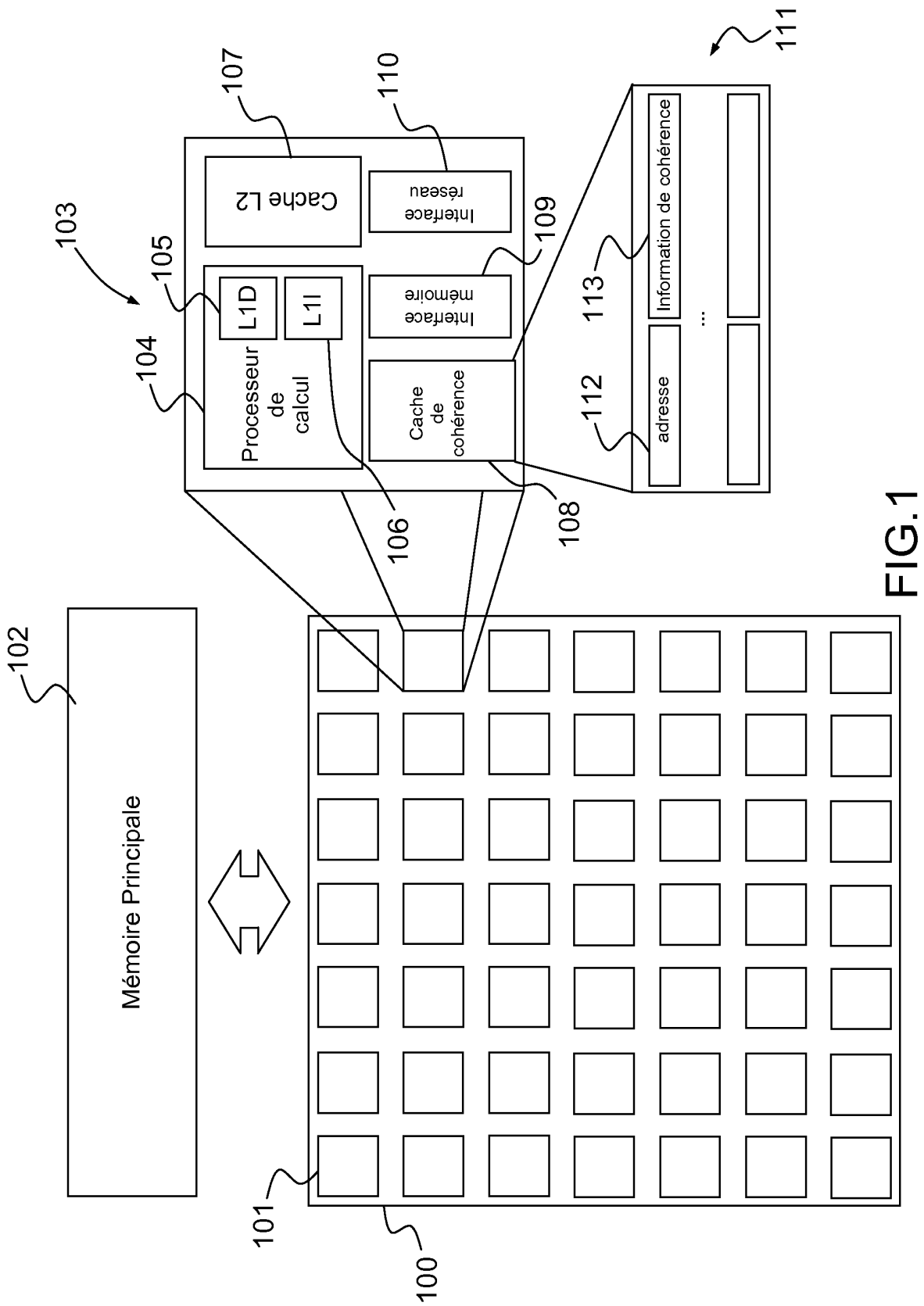

Mémoire Principale
102
101
100
103
104
105
Processeur de calcul
L1D
L1I
Cache L2
107
Interface réseau
110
Interface mémoire
109
Cache de cohérence
106
108
111
Information de cohérence
113
adresse
112

FIG.1

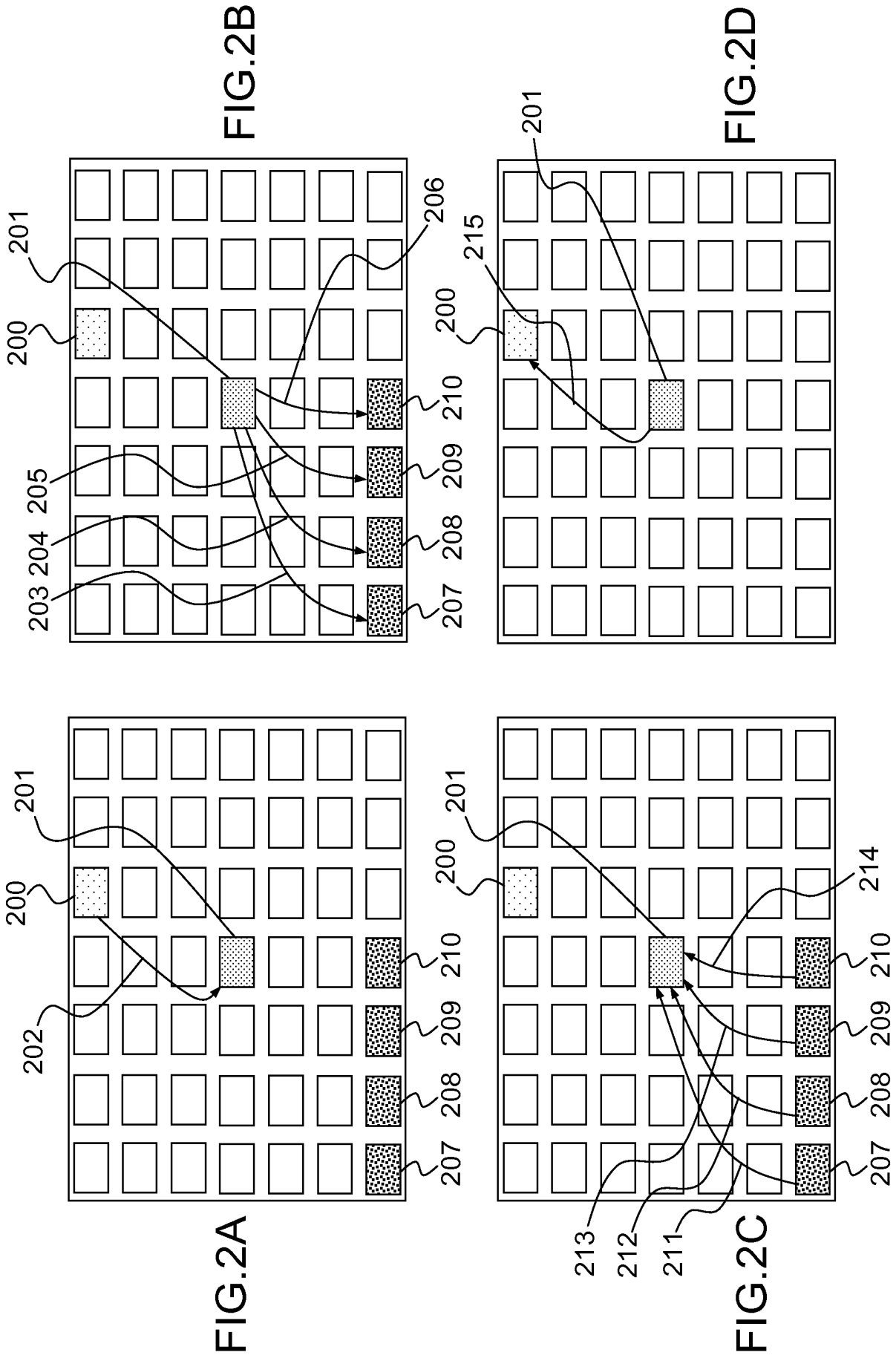


FIG.2B
201
200
206
205
204
203
210
209
208
207
FIG.2D
201
215
200
FIG.2A
201
200
202
210
209
208
207
FIG.2C
201
200
214
210
209
208
207
213
212
211

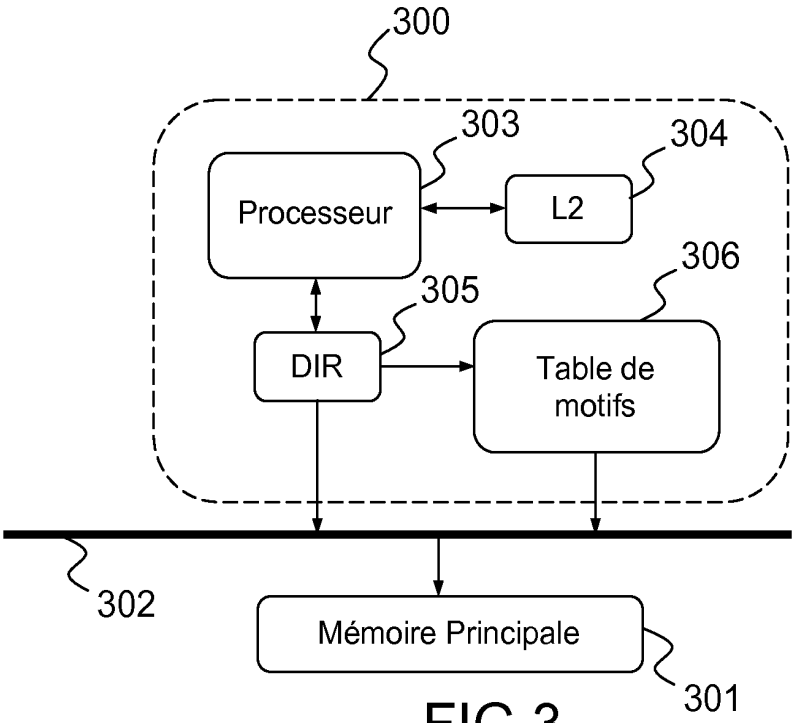


FIG.3

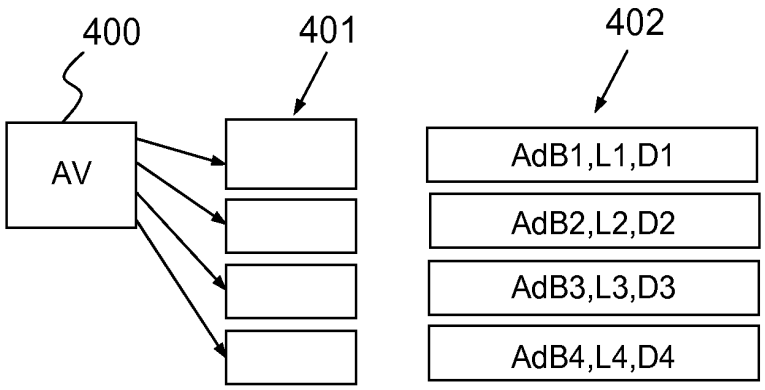


FIG.4

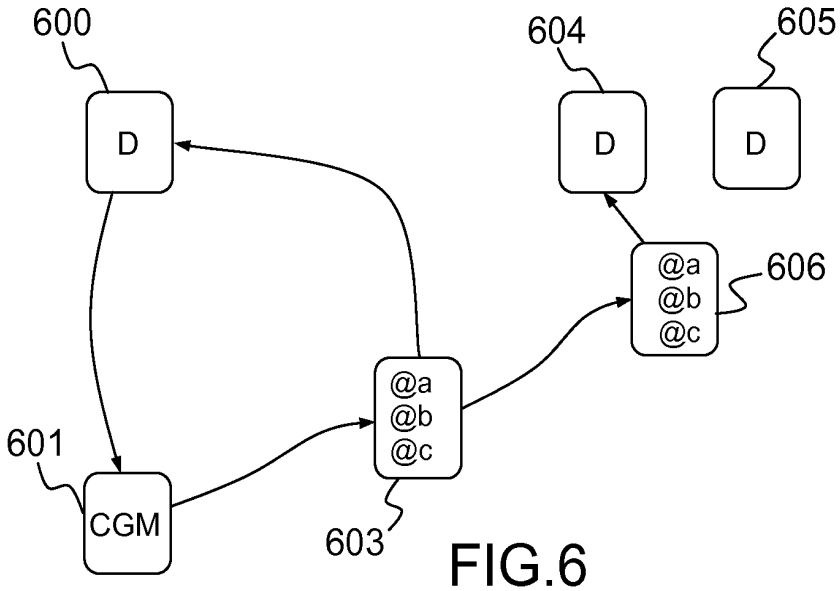


FIG.6

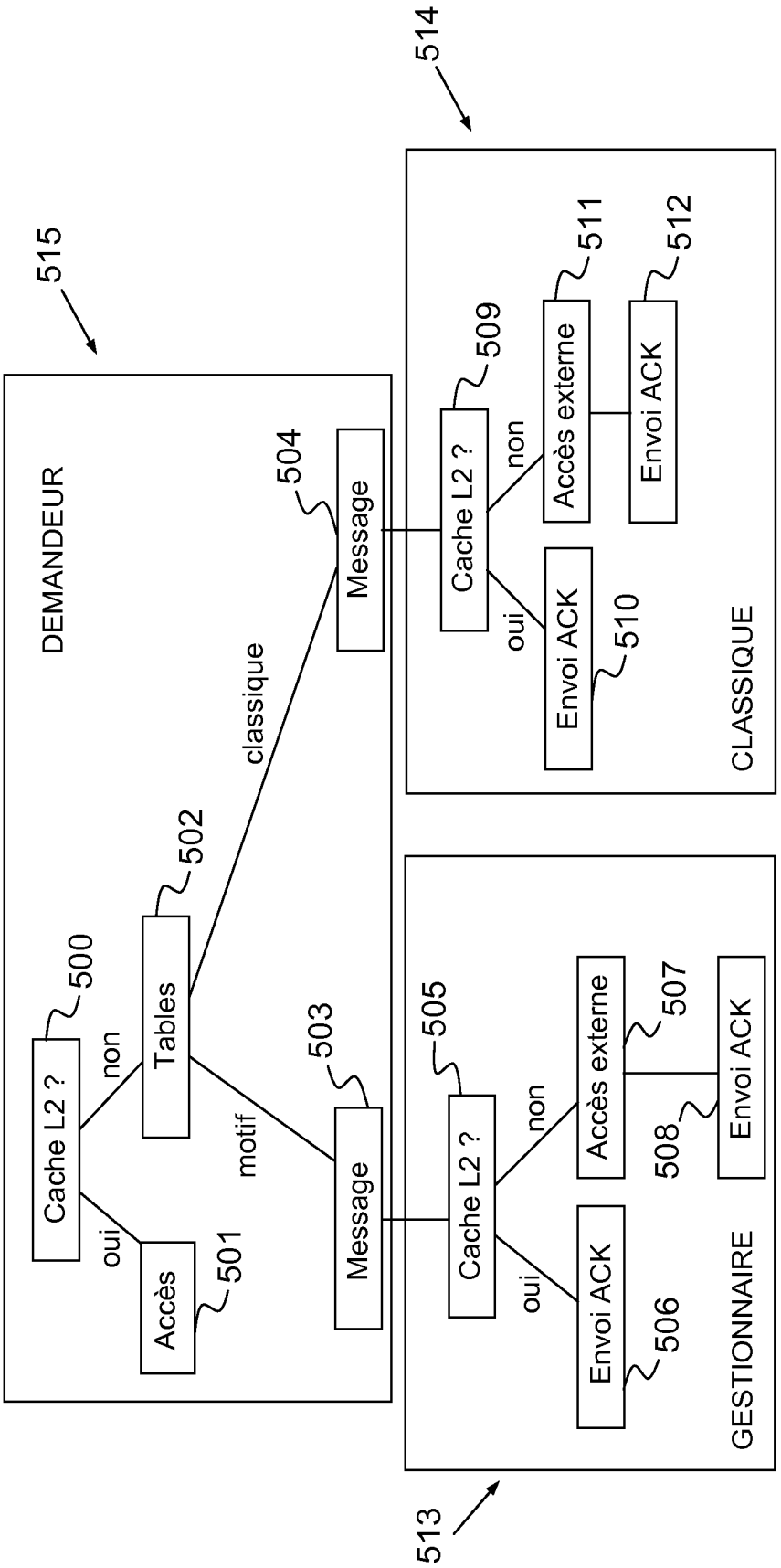

514
515
DEMANDEUR
Cache L2 ?
500
non
Tables
502
oui
Accès
501
classique
motif
504
Message
503
Message
Cache L2 ?
509
non
Accès externe
511
Envoi ACK
512
oui
Envoi ACK
510
CLASSIQUE
505
Cache L2 ?
non
Accès externe
507
508
Envoi ACK
oui
Envoi ACK
506
GESTIONNAIRE
513

FIG.5

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement national

FA 751038
FR 1150500

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 7 325 102 B1 (CYPHER ROBERT E [US]) 29 janvier 2008 (2008-01-29) * abrégé; figure 2 * * colonne 4, ligne 35 - colonne 6, ligne 41 * -----	1-14	G06F13/16 G06F15/173
A	US 7 363 435 B1 (STENSTROM PER O [SE]) 22 avril 2008 (2008-04-22) * le document en entier * -----	1-14	
A	US 5 829 017 A (OHTSUKA MASAOKI [JP]) 27 octobre 1998 (1998-10-27) * le document en entier * -----	1-14	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
			G06F

Date d'achèvement de la recherche	Examinateur
23 septembre 2011	Jardon, Stéphan

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

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 1150500 FA 751038

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 23-09-2011
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 7325102 B1	29-01-2008	AUCUN	
US 7363435 B1	22-04-2008	AUCUN	
US 5829017 A	27-10-1998	JP 3597247 B2	02-12-2004
		JP 8249862 A	27-09-1996
		US 6088765 A	11-07-2000

EPO FORM P0465