

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la
Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
20 juin 2019 (20.06.2019)



(10) Numéro de publication internationale
WO 2019/115902 A1

- (51) Classification internationale des brevets :
G06F 15/78 (2006.01)
- (21) Numéro de la demande internationale :
PCT/FR2018/052995
- (22) Date de dépôt international :
27 novembre 2018 (27.11.2018)
- (25) Langue de dépôt : français
- (26) Langue de publication : français
- (30) Données relatives à la priorité :
17 62068 13 décembre 2017 (13.12.2017) FR
- (71) Déposant : VSORA [FR/FR] ; 13-15 rue Jeanne Braconnier, 92360 MEUDON LA FORET (FR).
- (72) Inventeurs : MAALEJ, Khaled ; 217 rue du Faubourg Saint Honoré, 75008 PARIS (FR). NGUYEN, Trung Dung ; 7 rue Eric Tabarly, 91300 MASSY (FR). SCHMITT, Julien ; 4 rue de la Pérouse, 91300 MASSY (FR). BER-

NARD, Pierre-Emmanuel ; 1 rue Joseph Fouriaux, 92160 ANTONY (FR).

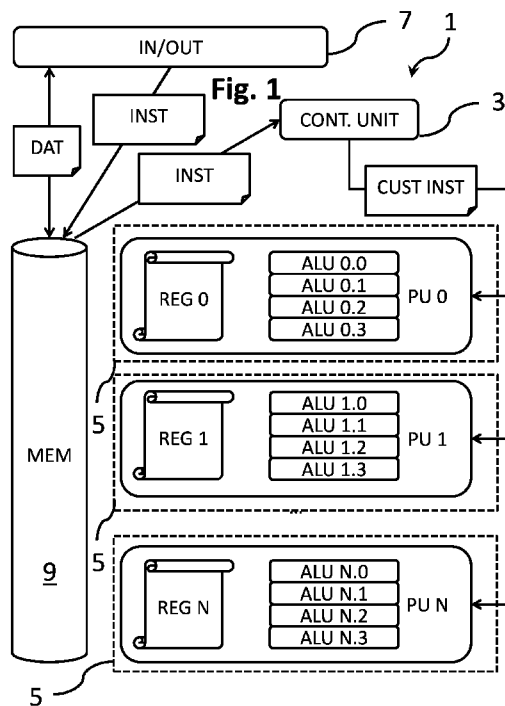
(74) Mandataire : CABINET PLASSERAUD ; 66 rue de la Chaussée d'Antin, 75440 PARIS CEDEX 09 (FR).

(81) États désignés (sauf indication contraire, pour tout titre de protection nationale disponible) : AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) États désignés (sauf indication contraire, pour tout titre de protection régionale disponible) : ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG,

(54) Title: PROCESSOR ARCHITECTURES

(54) Titre : ARCHITECTURES DE PROCESSEUR



(57) Abstract: A processor (1) comprising a control unit (3) and a plurality of processing units (5) interacting in accordance with an operating architecture, imposed dynamically by the control unit, from among at least two of the following architectures and combinations of the following architectures: - a single instruction stream multiple data stream architecture (SIMD), - a multiple instruction stream single data stream architecture (MISD), - a multiple instruction stream multiple data stream architecture (MIMD). The operating architecture is imposed dynamically by the control unit in accordance with: - configuration functions contained in a machine code and/or - data to be processed and current processing instructions received at the input of the processor.

(57) Abrégé : Un processeur (1) comprenant une unité de contrôle (3) et une pluralité d'unités de traitement (5) interagissant selon une architecture de fonctionnement imposée dynamiquement par l'unité de contrôle parmi au moins deux des architectures suivantes et des combinaisons d'architectures suivantes : - une architecture à unique flux d'instructions et multiples flux de données (SIMD), - une architecture à multiples flux d'instructions et unique flux de données (MISD), - une architecture à multiples flux d'instructions et multiples flux de données (MIMD). L'architecture de fonctionnement est imposée dynamiquement par l'unité de contrôle selon : - des fonctions de configuration incluses dans un code machine, et/ou - des données à traiter et d'instructions de traitement courantes reçues en entrée du processeur.



WO 2019/115902 A1

ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, RU, TJ, TM),
européen (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES,
FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK,
MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI
(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML,
MR, NE, SN, TD, TG).

Publiée:

— avec rapport de recherche internationale (Art. 21(3))

ARCHITECTURES DE PROCESSEUR

L'invention relève du domaine des processeurs, en particulier de leur fonctionnement intrinsèque.

5

Classiquement, les processeurs présentent une architecture définie lors de leur conception. L'architecture est au moins en partie définie par l'implémentation d'un jeu d'instructions machines que le processeur peut exécuter (ou ISA pour « *Instruction Set Architecture* »). Il est généralement admis que chaque structure connue peut être classée dans l'un des types (ou classes) suivants, définis selon la taxonomie de Flynn :

10

- Unique flux d'instructions et unique flux de données, ou SISD pour « *Single Instruction Single Data* » ;

- Unique flux d'instructions et multiples flux de données, ou SIMD pour « *Single Instruction Multiple Data* » ;

15

- Multiples flux d'instructions et unique flux de données, ou MISD pour « *Multiple Instructions Single Data* » ;

- Multiples flux d'instructions et multiples flux de données, ou MIMD pour « *Multiple Instructions Multiple Data* ».

20

Chacun de ces types d'architecture présente des avantages et des inconvénients connus. En fonction des applications prévues, le niveau de parallélisme entre les calculs diffère. Le choix d'une architecture rend généralement le processeur sur lequel elle est implémentée particulièrement efficace (rapide) pour certains calculs et lents pour d'autres. La sélection du type de processeur implique des compromis dans l'usage du processeur. Chaque type de

25

processeur tend à être spécialisé dans certains calculs au détriment d'autres calculs pour lesquels il est moins performant.

L'invention vient améliorer la situation.

30

Il est proposé un processeur comprenant une unité de contrôle et une pluralité d'unités de traitement. Les unités de traitement interagissant selon une architecture de fonctionnement imposée dynamiquement par l'unité de contrôle parmi au moins deux des architectures suivantes et des combinaisons d'architectures suivantes :

- une architecture à unique flux d'instructions et multiples flux de données (SIMD),

35

- une architecture à multiples flux d'instructions et unique flux de données (MISD),

- une architecture à multiples flux d'instructions et multiples flux de données (MIMD).

L'architecture de fonctionnement étant imposée dynamiquement par l'unité de contrôle selon :

- des fonctions de configuration incluses dans un code machine, et/ou

- des données à traiter et d'instructions de traitement courantes reçues en entrée du processeur.

5

Un tel processeur permet une adaptation dynamique et contextuelle de son fonctionnement interne. Lorsque les calculs à effectuer sont indépendants les uns des autres, ils peuvent être traités en parallèle, donc en même temps, par des unités de calculs distinctes les unes des autres.

Le traitement de l'ensemble des calculs est accéléré. Lorsque les calculs à effectuer sont dépendants les uns des autres, alors un traitement en parallèle n'est pas adapté. Les calculs récurifs sont un exemple de calculs peu adaptés au traitement en parallèle : pour effectuer un calcul, le résultat d'un calcul précédent est nécessaire. Une ou plusieurs unités de calcul doivent exécuter des calculs de manière séquentielle, un cycle après l'autre. Un tel processeur est polyvalent. Un tel processeur présente une architecture variante au cours de l'exécution des calculs en fonction des calculs eux-mêmes.

15

Selon un autre aspect, il est proposé un procédé de compilation d'un code source, mis en œuvre par des moyens informatiques, comprenant les étapes suivantes :

- recevoir un code source en entrée,

20 - rechercher dans le code source :

- des données de configuration imposant une architecture de fonctionnement d'un processeur,

- des premières parties du code source correspondant à des instructions de traitement de données incluant des séquences d'opérations élémentaires en cascade, et

25 - des secondes parties du code source correspondant à des instructions de traitement de données incluant des opérations élémentaires indépendantes les unes des autres.

Le procédé comprend en outre l'étape suivante :

- compiler le code source en un code machine.

Dans le cas où au moins une donnée de configuration imposant une architecture de fonctionnement d'un processeur a été identifiée et/ou dans le cas où au moins une première partie du code source et au moins une seconde partie du code source ont été identifiées, la compilation comprend l'inclusion dans le code machine de fonctions de configuration. Les fonctions de configuration sont agencées pour imposer dynamiquement à un processeur exécutant le code machine une architecture parmi au moins deux des architectures suivantes et des combinaisons d'architectures suivantes :

35

- une architecture à unique flux d'instructions et multiples flux de données (SIMD),
- une architecture à multiples flux d'instructions et unique flux de données (MISD),
- une architecture à multiples flux d'instructions et multiples flux de données (MIMD).

5 Optionnellement, le procédé de compilation d'un code source comprend en outre la vérification du respect d'un ensemble de règles préétablies dans les instructions de traitement en entrée, les fonctions de configuration incluses dans le code machine au cours de la compilation étant en outre sélectionnées en fonction du respect ou du non-respect desdites règles.

10 Selon un autre aspect, il est proposé un procédé de gestion d'architecture d'un processeur, mis en œuvre par des moyens informatiques, comprenant les étapes suivantes :

- recevoir des données à traiter et des instructions de traitement en entrée du processeur, et
- imposer dynamiquement à au moins une unité de traitement du processeur une architecture de fonctionnement en fonction desdites données à traiter et des instructions de traitement reçues,

15 les architectures de fonctionnement étant sélectionnées parmi au moins deux des architectures suivantes et combinaison des architectures suivantes :

- une architecture à unique flux d'instructions et multiples flux de données (SIMD),
- une architecture à multiples flux d'instructions et unique flux de données (MISD),
- une architecture à multiples flux d'instructions et multiples flux de données (MIMD).

20

Selon encore un autre aspect, il est proposé un support d'enregistrement non transitoire lisible par un ordinateur sur lequel est enregistré un programme de compilation comprenant des instructions pour la mise en œuvre du procédé ci-avant. Selon encore un autre aspect, il est proposé un programme informatique de compilation comportant des instructions pour la mise en œuvre du procédé de compilation, lorsque ce programme est exécuté par un processeur.

25

Selon encore un autre aspect, il est proposé un support d'enregistrement non transitoire lisible par une unité de contrôle d'un processeur sur lequel est enregistré un jeu d'instructions machines pour la mise en œuvre d'un procédé de gestion d'architecture tel que ceux définis dans les présentes. Selon encore un autre aspect, il est proposé un jeu d'instructions machine pour la mise en œuvre du procédé de gestion d'architecture, lorsque ce jeu d'instructions machine est exécuté par l'unité de contrôle d'un processeur.

30

D'autres caractéristiques, détails et avantages de l'invention apparaîtront à la lecture de la description détaillée ci-après, et à l'analyse des dessins annexés, sur lesquels :

35

- la figure 1 montre partiellement une architecture d'un processeur selon l'invention,
- la figure 2 montre un mode de fonctionnement d'un processeur selon l'invention,
- la figure 3 montre un mode de fonctionnement d'un processeur selon l'invention, et
- la figure 4 montre un détail de fonctionnement d'un processeur selon l'invention.

5

Les dessins et la description ci-après contiennent, pour l'essentiel, des éléments de caractère certain. Ils pourront donc non seulement servir à mieux faire comprendre la présente invention, mais aussi contribuer à sa définition, le cas échéant.

- 10 La figure 1 montre un processeur 1, parfois appelé unité centrale de traitement ou CPU pour « *Central Processing Unit* ». Le processeur 1 comprend :
- une unité de contrôle 3,
 - une partie de calcul incluant une pluralité d'unités de traitement 5,
 - une unité d'entrée-sortie 7, ou interface, et
- 15 - une unité mémoire 9.

- De manière connue en soit, le processeur 1 reçoit par l'intermédiaire de l'unité d'entrée-sortie 7 des données à traiter (« data ») et des instructions de traitement (« instructions »). Les données et instructions sont stockées dans l'unité de mémoire 9. L'unité de mémoire 9 peut être divisée
- 20 en plusieurs parties. Par exemple, l'unité de mémoire 9 comprend une partie données (ou « *Data pool* ») et une partie instructions (ou « *Instruction pool* »).

- Chaque unité de traitement 5 effectue les calculs sur des données et selon des instructions issues de celles stockées dans l'unité de mémoire 9. L'unité de contrôle 3 impose à chaque unité de
- 25 traitement 5 la manière d'effectuer les calculs élémentaires, notamment leur ordre, et attribut à chaque organe de calcul de l'unité de traitement 5 les opérations à exécuter.

- Dans les exemples décrits ici, chaque unité de traitement 5, ou PU pour « *Processing Unit* », comprend plusieurs organes de calcul : les unités arithmétiques et logiques, ou ALU pour
- 30 « *Arithmetic-Logic Unit* ». Chaque unité de traitement 5 comprend au moins une ALU et au moins un jeu de registres associé REG. Dans la suite et conformément aux figures, chaque unité de traitement 5 est numérotée de PU 0 à PU N. Chaque ALU est numérotée A.B, où « A » identifie l'unité de traitement PU A à laquelle l'ALU appartient et « B » est un identifiant de l'ALU parmi les autres ALU de l'unité de traitement PU A.

35

Le processeur 1 comprend au moins deux ALU réparties dans deux unités de traitement 5. Par exemple, chaque unité de traitement 5 comprend une unique ALU ou une pluralité d'ALU. Dans les exemples des figures 1 à 3, chaque unité de traitement 5 comprend quatre ALU numérotées 0 à 3. Lorsqu'une unité de traitement 5 comprend une pluralité d'ALU, l'unité de traitement 5 est dite multi-cœur.

Chaque ALU peut effectuer :

- des opérations arithmétiques sur des données entières (addition, soustraction, multiplication, division, etc.) ;
- 10 - des opérations arithmétiques sur des données flottantes (addition, soustraction, multiplication, division, inversion, racine carrée, logarithmes, trigonométrie, etc.) ;
- des opérations logiques (compléments à deux, « ET », « OU », « OU exclusif », etc.).

Les unités de traitement 5 et l'unité de mémoire 9 interagissent selon l'une et/ou l'autre des trois architectures suivantes :

- une architecture à unique flux d'instructions et multiples flux de données (SIMD),
 - une architecture à multiples flux d'instructions et unique flux de données (MISD),
 - une architecture à multiples flux d'instructions et multiples flux de données (MIMD).
- 20 Un exemple d'une architecture SIMD est représentée en figure 2. Dans cet exemple, les unités de traitement 5 interagissent selon l'architecture SIMD. Les données à traiter sont copiées (chargées) depuis l'unité de mémoire 9 sur chacun des jeux de registres REG 0, REG 1, ..., REG N de l'unité de traitement 5 correspondante. Les ALU effectuent les calculs. Les résultats sont écrits dans le jeu de registres REG 0, REG 1, ..., REG N. Puis, les résultats sont copiés
- 25 depuis les jeux de registres REG 0, REG 1, ..., REG N dans l'unité de mémoire 9.

Dans cet exemple, les unités de traitement 5 n'échangent pas directement de données les unes avec les autres. Les ALU de chaque unité de traitement 5 traitent les données et effectuent des calculs indépendamment d'une unité de traitement 5 à une autre. Le fonctionnement du

30 processeur 1 est parallélisé au niveau des unités de traitement 5.

Dans l'exemple de la figure 2, les interactions entre les ALU ne sont pas représentées. Ainsi, alors que les unités de traitement 5 interagissent entre elles selon une architecture SIMD, les ALU d'une même unité de traitement interagissent selon l'architecture SIMD, comme cela est

35 décrit ci-après relativement à la figure 4.

Un exemple de l'architecture MISD est représentée en figure 3. Dans cet exemple, les unités de traitement 5 interagissent selon l'architecture MISD. Les données sont copiées (chargées) depuis l'unité de mémoire 9 sur le jeu de registres d'une unique unité de traitement 5, ici le jeu de registres REG 0 de l'unité de traitement PU 0. Les ALU 0.0, 0.1, 0.2 et 0.3 effectuent des calculs. Les résultats sont écrits dans le jeu de registres REG 0. Puis, les résultats sont copiés depuis le jeu de registres REG 0 sur un jeu de registres d'une autre unité de traitement 5, ici sur le jeu de registres REG 1 de l'unité de traitement PU 1. À leur tour, les ALU 1.0, 1.1, 1.2 et 1.3 effectuent des calculs et les résultats sont écrits dans le jeu de registres REG 1. Ces opérations sont reproduites les unes à la suite des autres par chacune des unités de traitement 5 jusqu'à ce que les résultats soient écrits dans le jeu de registres de la dernière unité de traitement 5, soit ici le jeu de registres REG N de l'unité de traitement PU N. Puis, les résultats sont copiés depuis le dernier jeu de registres REG N dans l'unité de mémoire 9.

Dans cet exemple, les unités de traitement 5 échangent directement des données les unes avec les autres. Les ALU de chaque unité de traitement 5 effectuent des calculs sur des données qui sont elles-mêmes des résultats des calculs mis en œuvre par les autres unités de traitement 5. Le fonctionnement du processeur 1 n'est pas parallélisé au niveau des unités de traitement 5. Au contraire, les unités de traitement 5 présentent un fonctionnement en série, ou en cascade. Ce type de fonctionnement est par exemple adapté aux calculs dits récursifs. Les opérations mises en œuvre par les unités de traitement 5 peuvent être les mêmes mais sont appliquées à des données qui sont chaque fois différentes. En variante, les instructions pourraient aussi être différentes les unes des autres et les données être, elles-aussi, différentes les unes des autres.

Dans l'exemple de la figure 3, comme dans celui de la figure 2, les interactions entre les ALU ne sont pas représentées.

En figure 4, les fonctionnements de deux PU sont représentés.

L'unité de traitement PU X comprend quatre ALU. Les ALU de l'unité de traitement PU X interagissent entre elles. Les données sont chargées depuis le jeu de registres REG X sur chacune des ALU X.0, X.1, X.2, X.3 de l'unité de traitement PU X. Les ALU effectuent les calculs. Les résultats sont ensuite écrits dans le jeu de registres REG X.

Les ALU n'échangent pas directement de données les unes avec les autres.

Dans l'exemple de la figure 4, les architectures au niveau des unités de traitement 5 ne sont pas représentées. Ainsi, les unités de traitement 5 peuvent interagir selon l'une et/ou l'autre des architectures SIMD, MISD et MIMD, comme cela est décrit ci-avant relativement aux figures 2 et 3. Autrement dit, l'exemple de la PU X est compatible avec l'exemple de la figure 2 et avec l'exemple de la figure 3.

Les architectures de fonctionnement peuvent être imposées dynamiquement par l'unité de contrôle 3 en fonction des données à traiter et des instructions courantes reçues en entrée du processeur 1. Une telle adaptation dynamique des architectures peut être mise en œuvre dès le stade de la compilation, en adaptant les instructions machines générées par le compilateur en fonction du type de données à traiter et des instructions lorsque le type de données à traiter et les instructions peuvent être déduites du code source. Une telle adaptation peut aussi être mise en œuvre seulement au niveau du processeur lorsqu'il exécute un code machine classique lorsque ce processeur est programmé pour mettre en œuvre un ensemble d'instructions de configuration fonction des données à traiter et des instructions courantes reçues.

Ci-dessous est reproduit un exemple de code informatique des calculs à effectuer. Cet extrait de code correspond par exemple à des instructions d'un code source à mettre en œuvre par le processeur 1.

```
void processing()
{
    matrix A(4,4);
    25    matrix B(4,4);
        matrix C(4,4);
        C = A * B; // (1)
        C = C + B; // (2)
        matrix D (4,4);
    30    D = inv(A); // (3)
}
```

Dans l'exemple, le processeur comprend quatre unités de traitement PU, chaque unité de traitement PU comprend quatre unités arithmétiques et logiques ALU. Dans l'exemple, la multiplication matricielle est traitée en premier lieu tandis que l'addition matricielle est traitée

en second lieu. Le compilateur est capable de traiter des opérations matricielles en les décomposant en opérations élémentaires. Pour chaque opération matricielle, les opérations élémentaires qui la compose sont indépendantes les unes des autres. Autrement dit, le résultat d'une opération élémentaire est inutile pour mettre en œuvre une autre opération élémentaire.

- 5 Les opérations élémentaires peuvent donc être mises en œuvre en parallèle les unes des autres. L'addition de deux matrices de dimensions 4 par 4 nécessite 16 opérations élémentaires (addition). Cette addition matricielle, soit les 16 opérations élémentaires, peuvent être exécutées en un seul cycle. La multiplication des deux matrices de dimensions 4 par 4 nécessite 64 opérations élémentaires (multiplication + accumulation). Cette multiplication matricielle, soit
 10 les 64 opérations élémentaires, sont donc exécutées au minimum en quatre cycles.

- L'opération référencée « (3) » (inversion matricielle) comprend des opérations élémentaires dépendantes les unes des autres. Les résultats de certaines opérations élémentaires sont nécessaires à la mise en œuvre d'autres opérations élémentaires. Un traitement en parallèle n'est
 15 pas adapté. L'algorithme d'inversion doit donc être décomposé en N étapes de traitement. Chaque étape peut être effectuée sur une ALU en un cycle, mais avec des données distinctes les unes des autres et des traitements distincts les uns des autres.

étape_1 → étape_2 → étape_3 → étape_4 → étape_5 → → étape_N

20

Pour la mise en œuvre des opérations « (1) » et « (2) », le processeur 1 adopte une architecture SIMD. Pour la mise en œuvre de l'opération « (3) », le processeur 1 adopte une architecture MISD.

- 25 Par exemple, l'opération « (3) » comprend neuf étapes (N = 9). Trois itérations, ou trois cycles, sont donc nécessaires. Chaque cycle est mis en œuvre par une unité de traitement PU. L'affectation des N étapes d'un cycle aux ALU de l'unité de traitement PU se fait, par exemple, de la manière suivante :

Cycle 1 :

- 30 ALU 0 → étape_1
 ALU 1 → étape_2
 ALU 2 → étape_3
 ALU 3 → étape_4

Cycle 2 :

- 35 ALU 0 → étape_5

ALU 1 → étape_6

ALU 2 → étape_7

ALU 3 → étape_8

Cycle 3 :

5 ALU 0 → étape_9

Au cours du cycle 3, les ALU 1, ALU 2 et ALU 3 sont inutilisées.

Comme indiqué précédemment, les architectures de fonctionnement peuvent être imposées dynamiquement par l'unité de contrôle 3 en fonction des données et instructions courantes
10 reçues en entrée du processeur 1. Cela couvre deux cas.

Dans un premier cas, l'architecture et l'affectation des ressources sont fixées lors de la compilation. Autrement dit, un développeur d'un programme tiers (autre que ceux régissant le fonctionnement intrinsèque du processeur) peut inclure dans le code source des instructions de configuration spécifiques. Les instructions spécifiques sont transcrites en langage cible (lors de
15 la compilation) en instructions spécifiques (code machine) reconnues par l'unité de contrôle 3. À réception, sur le processeur 1, l'unité de contrôle 3 impose des architectures aux unités de traitement 5 de manière prédéfinie par les instructions. Dans ce cas, la responsabilité de l'optimisation du fonctionnement du processeur peut être laissée au créateur du programme
20 tiers. En fonction dudit programme, le programmeur est libre d'imposer ou non un fonctionnement particulier du processeur, c'est-à-dire ici une architecture choisie parmi SISD, MISD, MIMD ou une combinaison de celles-ci.

Dans un deuxième cas, l'architecture et l'affectation des ressources sont fixées de manière
25 préétablie en fonction d'un jeu d'instructions machines implémenté dans le processeur. Un tel jeu d'instructions est généralement implémenté avant commercialisation et utilisation du processeur. Le jeu d'instructions machine n'est pas destiné à être modifié par les utilisateurs du processeur. Dans ce deuxième cas, à réception des instructions sur le processeur 1, l'unité de contrôle 3 met en œuvre un procédé de gestion d'architecture, ou de configuration des
30 architectures, préalablement à la mise en œuvre des instructions reçues en entrée. Par exemple, l'unité de contrôle 3 transmet au préalable à chaque unité de traitement PU des données de configuration propres à chaque unité de traitement PU. Chacun des jeux de données de configuration est stocké dans un registre de configuration accessible à l'unité de traitement correspondante. Ensuite, les unités de traitement PU reçoivent des instructions de traitement
35 génériques (communes à toutes les PU) d'un bus d'instruction. Chaque PU met en œuvre les

instructions de traitement d'une manière qui varie en fonction des données de configuration préalablement reçues et stockées dans le registre de configuration. Autrement dit, chaque PU interprète les instructions de traitement génériques au moyen des données de configuration pour adapter le traitement à mettre en œuvre. Le deuxième cas permet d'introduire une flexibilité
5 dans le contrôle. Des instructions génériques peuvent être transmises à toutes les PU, quelle que soit l'architecture à adopter (SIMD, MISD, MIMD). La transmission au préalable des données de configuration permet de sélectionner l'architecture adoptée effectivement par les PU à réception d'instructions génériques.

10 Dans les deux cas, les architectures peuvent être dynamiques, c'est-à-dire évoluer au fil des étapes d'exécution des instructions reçues, notamment en fonction de la nature des calculs à effectuer. Par exemple, une architecture SIMD peut être imposée par défaut et une architecture MISD être imposée pour les calculs récursifs. Les deux cas sont compatibles : le processeur 1
15 peut être agencé pour mettre en œuvre un procédé de gestion des architectures en fonction d'instructions de configuration spécifiques reçues (contenues dans le code machine issu de la compilation) et pour mettre en œuvre un procédé de gestion des architectures selon un jeu d'instructions machines en l'absence ou en complément d'instructions de configuration spécifiques parmi les instructions reçues en entrée.

20 Selon un autre point de vue, il peut être considéré qu'en mettant en œuvre des instructions de configuration destinées à adapter à la volée l'architecture du processeur 1, l'unité de contrôle 3 transforme les instructions de traitement reçues de manière usuelle en entrée en instructions adaptées ou « macro-instructions » (ou « custom instructions »). Les instructions adaptées contiennent à la fois des instructions de traitement et des instructions de configuration. Lorsque
25 les instructions adaptées transmises à chacune des unités de traitement 5 sont toutes identiques, le processeur fonctionne en mode SIMD. Les unités de traitement 5 effectuent toutes les mêmes opérations sur des données à traiter différentes. Lorsque les instructions adaptées transmises à chacune des unités de traitement 5 sont différentes les unes des autres, le processeur fonctionne en mode MISD ou MIMD. Les unités de traitement 5 effectuent des opérations distinctes les
30 unes des autres sur des données à traiter identiques (MISD) ou différentes (MIMD).

Ci-dessous est reproduit un deuxième exemple de code informatique de calculs à effectuer :

```
void processing()  
{  
35 matrix A(4,4) [2];
```

```

matrix D (4,4) [2];
D = inv(A); // (3)
}

```

5 Dans l'exemple est défini un tableau de matrices, chaque matrice étant de dimension 4 par 4, le tableau étant de taille 2 (comprenant deux matrices). La fonction notée « *inv* » sur un tableau de matrices consiste à inverser chaque élément du tableau, c'est-à-dire inverser chacune des deux matrices de dimension 4 par 4. Comme dans l'exemple précédent, le processeur comprend quatre ALU.

10

L'affectation des N étapes aux ALU se fait, par exemple, de la manière suivante :

cycle 1 : étapes 1 à 4 pour la matrice A[0] ;

cycle 2 : étapes 1 à 4 pour la matrice A[1] ;

cycle 3 : étapes 5 à 8 pour la matrice A[0] ;

15 cycle 4 : étapes 5 à 8 pour la matrice A[1] ;

cycle 5 : étape 9 pour la matrice A[0] ;

cycle 6 : étape 9 pour la matrice A[1].

20 On peut également optimiser l'exécution de la fonction « *inv* » en regroupant les cycles 5 et 6 en un cycle commun. Ainsi, une unique PU est nécessaire pour mettre en œuvre l'étape 9 sur les deux matrices. En variante, l'ALU 0 peut être affectée aux calculs relatifs à la matrice A[0] tandis que l'ALU 1 est affectée aux calculs relatifs à la matrice A[1].

25 L'attribution de chaque ALU à des opérations peut être prévue dès la compilation si au moins une partie des données est connue à ce stade, notamment la taille des matrices et la taille du tableau. L'attribution peut être réalisée de manière dynamique. Par exemple, l'attribution peut être imposée par l'unité de contrôle 3 en fonction d'un jeu d'instructions machines implémenté sur le processeur 1. Dans ce cas, le jeu d'instructions machines, est enregistré sur un support d'enregistrement non transitoire (par exemple une partie de l'unité mémoire 9) lisible par l'unité
30 de contrôle 3 pour mettre en œuvre un procédé de gestion de l'architecture du processeur 1.

Par exemple, l'unité de contrôle 3 est agencée pour imposer à une unité de traitement 5 la mise en œuvre d'un premier ensemble d'opérations par l'ensemble des ALU, puis le premier ensemble d'opérations est réitéré sur chacun des éléments du tableau (chacune des matrices
35 dans l'exemple précédent). Ensuite, le nombre d'opérations réalisables en parallèle (non

interdépendantes) est estimé. Par exemple, le nombre de ressources (le nombre d'ALU) est divisé par le nombre d'opérations à mettre en œuvre. Enfin, l'attribution des opérations à chaque ALU est effectuée de sorte que certaines au moins des opérations soient réalisées en parallèle les unes des autres par des ALU distinctes d'unités de traitement 5 distinctes.

5

Ainsi, l'architecture des unités de traitement 5 peut varier au cours du temps. Au cours d'un traitement, l'architecture des unités de traitement 5 peut alterner entre SIMD, MISD et MIMD.

L'invention ne se limite pas aux exemples de processeurs décrits ci-avant, seulement à titre
10 d'exemple, mais elle englobe toutes les variantes que pourra envisager l'homme de l'art dans le cadre de la protection recherchée. L'invention concerne aussi un jeu d'instructions machines implémentable dans un processeur pour obtenir un tel processeur, l'implémentation d'un tel jeu d'instructions machines sur un processeur, le procédé de gestion d'architecture de processeur mis en œuvre par le processeur, le programme informatique comprenant le jeu d'instructions
15 machines correspondant, ainsi que le support d'enregistrement sur lequel est enregistré informatiquement un tel jeu d'instructions machines.

Revendications

1. Processeur (1) comprenant une unité de contrôle (3) et une pluralité d'unités de traitement (5), les unités de traitement (5) interagissant selon une architecture de fonctionnement imposée dynamiquement par l'unité de contrôle parmi au moins deux des architectures suivantes et des combinaisons d'architectures suivantes :
- une architecture à unique flux d'instructions et multiples flux de données (SIMD),
 - une architecture à multiples flux d'instructions et unique flux de données (MISD),
 - une architecture à multiples flux d'instructions et multiples flux de données (MIMD),
- 10 l'architecture de fonctionnement étant imposée dynamiquement par l'unité de contrôle selon :
- des fonctions de configuration incluses dans un code machine, et/ou
 - des données à traiter et d'instructions de traitement courantes reçues en entrée du processeur.
2. Procédé de compilation d'un code source, mis en œuvre par des moyens informatiques, comprenant les étapes suivantes :
- recevoir un code source en entrée,
 - rechercher dans le code source :
 - des données de configuration imposant une architecture de fonctionnement d'un processeur,
 - des premières parties du code source correspondant à des instructions de traitement de données incluant des séquences d'opérations élémentaires en cascade, et
 - des secondes parties du code source correspondant à des instructions de traitement de données incluant des opérations élémentaires indépendantes les unes des autres,
- le procédé comprenant en outre l'étape suivante :
- 25 - compiler le code source en un code machine,
- dans le cas où au moins une donnée de configuration imposant une architecture de fonctionnement d'un processeur a été identifiée et/ou dans le cas où au moins une première partie du code source et au moins une seconde partie du code source ont été identifiées, la compilation comprend l'inclusion dans le code machine de fonctions de configuration, lesdites
- 30 fonctions de configuration étant agencées pour imposer dynamiquement à un processeur exécutant le code machine une architecture parmi au moins deux des architectures suivantes et des combinaisons d'architectures suivantes :
- une architecture à unique flux d'instructions et multiples flux de données (SIMD),
 - une architecture à multiples flux d'instructions et unique flux de données (MISD),
 - 35 - une architecture à multiples flux d'instructions et multiples flux de données (MIMD).

3. Procédé selon la revendication 2, comprenant en outre la vérification du respect d'un ensemble de règles préétablies dans les instructions de traitement en entrée, les fonctions de configuration incluses dans le code machine au cours de la compilation étant en outre
5 sélectionnées en fonction du respect ou du non-respect desdites règles.

4. Support d'enregistrement non transitoire lisible par un ordinateur sur lequel est enregistré un programme de compilation comprenant des instructions pour la mise en œuvre du procédé selon l'une des revendications 2 et 3.

10

5. Programme informatique de compilation comportant des instructions pour la mise en œuvre du procédé selon l'une des revendications 2 et 3, lorsque ce programme est exécuté par un processeur.

15 6. Procédé de gestion d'architecture d'un processeur (1), mis en œuvre par des moyens informatiques, comprenant les étapes suivantes :

- recevoir des données à traiter et des instructions de traitement en entrée du processeur, et
- imposer dynamiquement à au moins une unité de traitement (5) du processeur une architecture de fonctionnement en fonction desdites données à traiter et des instructions de traitement reçues,

20 les architectures de fonctionnement étant sélectionnées parmi au moins deux des architectures suivantes et combinaison des architectures suivantes :

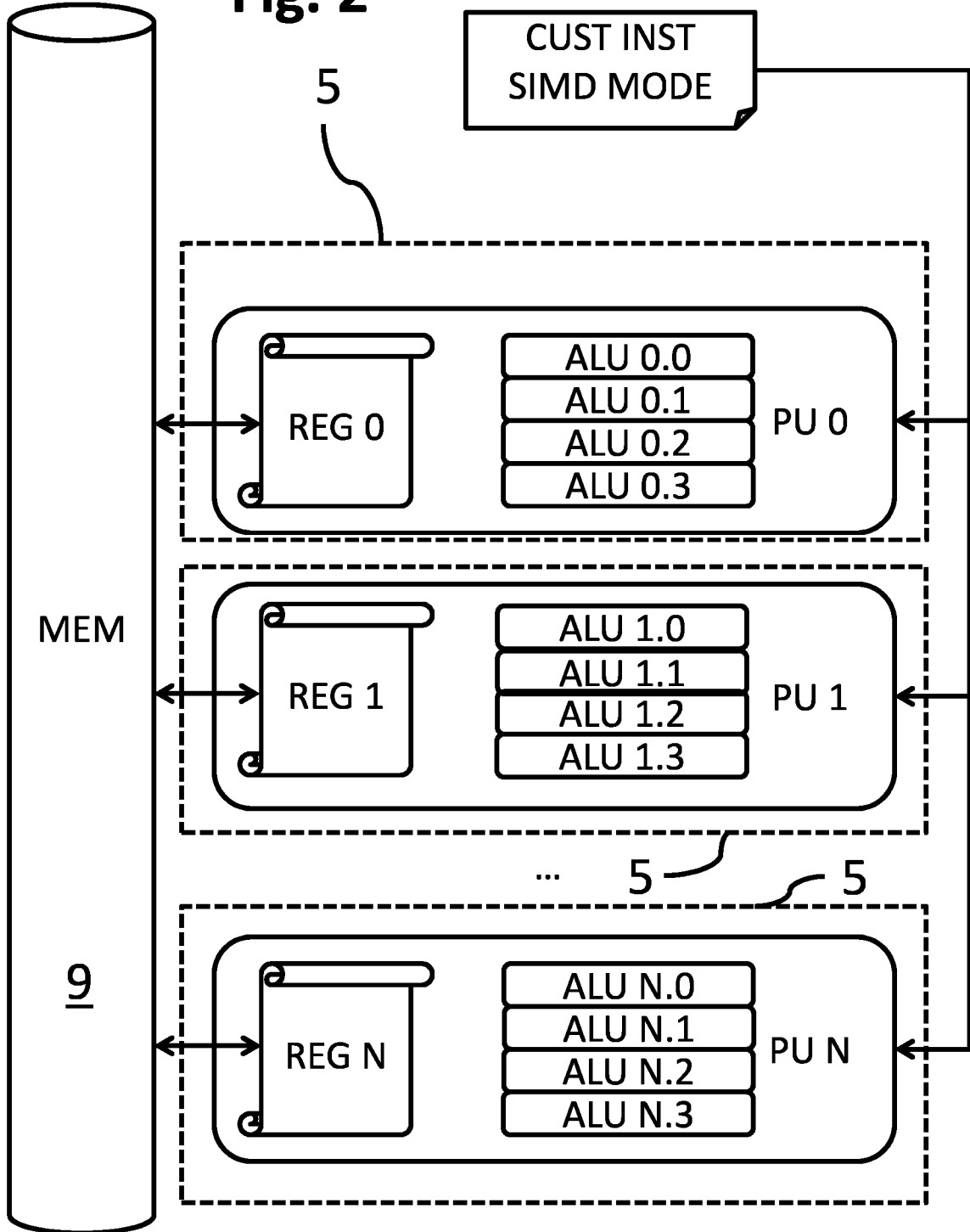
- une architecture à unique flux d'instructions et multiples flux de données (SIMD),
- une architecture à multiples flux d'instructions et unique flux de données (MISD),
- une architecture à multiples flux d'instructions et multiples flux de données (MIMD).

25

7. Support d'enregistrement non transitoire lisible par une unité de contrôle (3) d'un processeur (1) sur lequel est enregistré un jeu d'instructions machine pour la mise en œuvre du procédé selon la revendication 6.

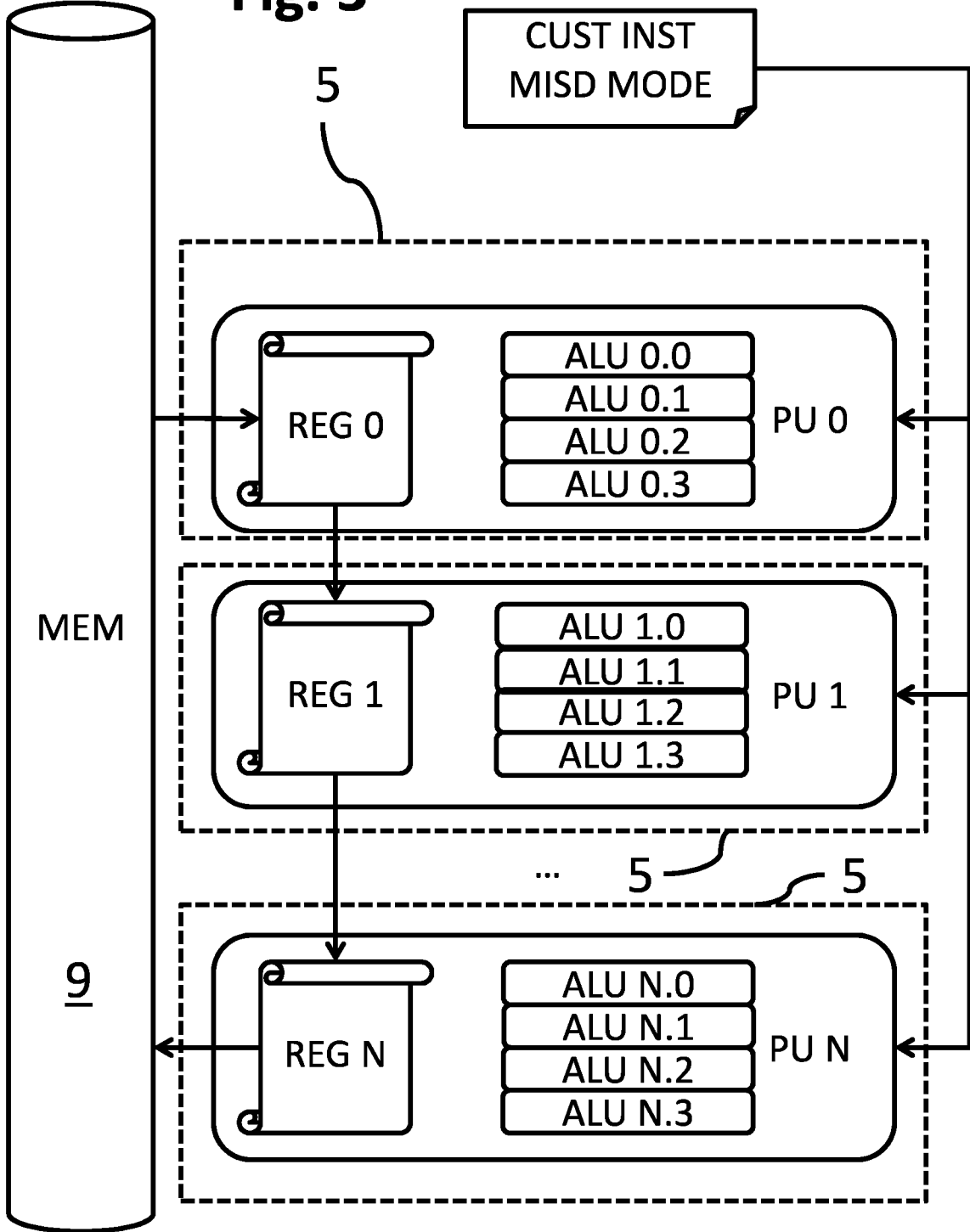
30 8. Jeu d'instructions machine pour la mise en œuvre du procédé selon la revendication 6, lorsque ce jeu d'instructions machine est exécuté par l'unité de contrôle d'un processeur.

Fig. 2



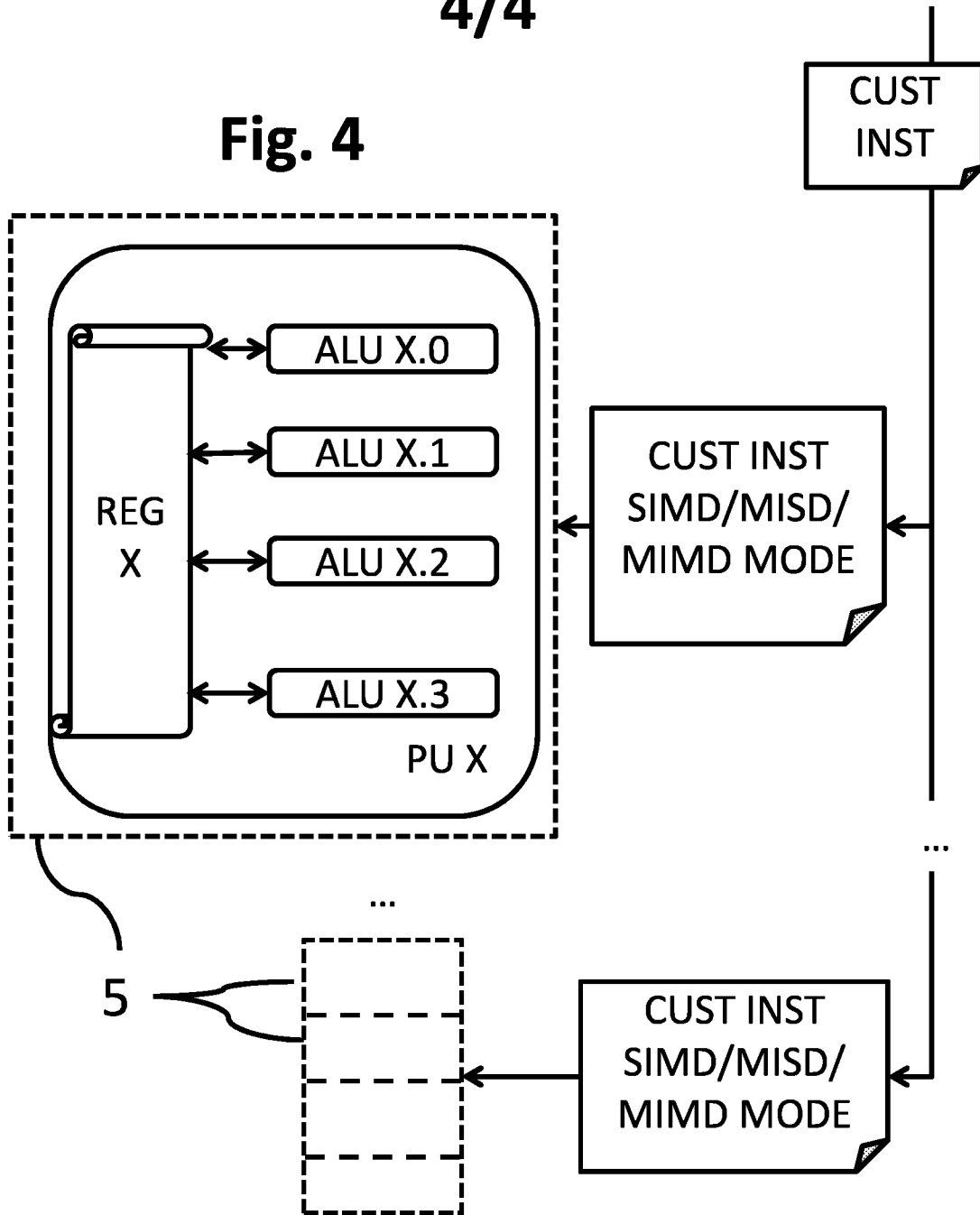
3/4

Fig. 3



4/4

Fig. 4



INTERNATIONAL SEARCH REPORT

International application No
PCT/FR2018/052995

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F15/78
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 903 771 A (SGRO JOSEPH A [US] ET AL) 11 May 1999 (1999-05-11) figure 1 column 2, line 35 - line 42 column 6, line 62 - line 65 -----	1-8
X	US 5 475 856 A (KOGGE PETER M [US]) 12 December 1995 (1995-12-12) column 3, line 8 - line 11 column 6, line 13 - line 20 column 12, line 33 - line 43 -----	1-8
X	US 2009/282223 A1 (LYUH CHUN-GI [KR] ET AL) 12 November 2009 (2009-11-12) the whole document -----	1-8

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search 8 February 2019	Date of mailing of the international search report 18/02/2019
--	--

Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer De Poy, Iker
--	--

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/FR2018/052995

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
US 5903771	A	11-05-1999	AU 725592 B2	12-10-2000
			CA 2243547 A1	24-07-1997
			EP 0875027 A1	04-11-1998
			US 5903771 A	11-05-1999
			WO 9726593 A1	24-07-1997

US 5475856	A	12-12-1995	AT 177547 T	15-03-1999
			CA 2073516 A1	28-05-1993
			CN 1072788 A	02-06-1993
			DE 69228586 D1	15-04-1999
			DE 69228586 T2	14-10-1999
			EP 0544127 A2	02-06-1993
			JP 2647315 B2	27-08-1997
			JP H05233569 A	10-09-1993
			KR 970008529 B1	24-05-1997
			US 5475856 A	12-12-1995

US 2009282223	A1	12-11-2009	KR 20090116511 A	11-11-2009
			US 2009282223 A1	12-11-2009

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°

PCT/FR2018/052995

A. CLASSEMENT DE L'OBJET DE LA DEMANDE INV. G06F15/78 ADD.		
Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB		
B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE		
Documentation minimale consultée (système de classification suivi des symboles de classement) G06F		
Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche		
Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés) EPO-Internal		
C. DOCUMENTS CONSIDERES COMME PERTINENTS		
Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	US 5 903 771 A (SGRO JOSEPH A [US] ET AL) 11 mai 1999 (1999-05-11) figure 1 colonne 2, ligne 35 - ligne 42 colonne 6, ligne 62 - ligne 65 -----	1-8
X	US 5 475 856 A (KOGGE PETER M [US]) 12 décembre 1995 (1995-12-12) colonne 3, ligne 8 - ligne 11 colonne 6, ligne 13 - ligne 20 colonne 12, ligne 33 - ligne 43 -----	1-8
X	US 2009/282223 A1 (LYUH CHUN-GI [KR] ET AL) 12 novembre 2009 (2009-11-12) le document en entier -----	1-8
<input type="checkbox"/> Voir la suite du cadre C pour la fin de la liste des documents <input checked="" type="checkbox"/> Les documents de familles de brevets sont indiqués en annexe		
* Catégories spéciales de documents cités:		
"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent "E" document antérieur, mais publié à la date de dépôt international ou après cette date "L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée) "O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens "P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée	"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention "X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément "Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier "&" document qui fait partie de la même famille de brevets	
Date à laquelle la recherche internationale a été effectivement achevée 8 février 2019		Date d'expédition du présent rapport de recherche internationale 18/02/2019
Nom et adresse postale de l'administration chargée de la recherche internationale Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Fonctionnaire autorisé De Poy, Iker

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale n°

PCT/FR2018/052995

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 5903771	A	11-05-1999	AU 725592 B2	12-10-2000
			CA 2243547 A1	24-07-1997
			EP 0875027 A1	04-11-1998
			US 5903771 A	11-05-1999
			WO 9726593 A1	24-07-1997

US 5475856	A	12-12-1995	AT 177547 T	15-03-1999
			CA 2073516 A1	28-05-1993
			CN 1072788 A	02-06-1993
			DE 69228586 D1	15-04-1999
			DE 69228586 T2	14-10-1999
			EP 0544127 A2	02-06-1993
			JP 2647315 B2	27-08-1997
			JP H05233569 A	10-09-1993
			KR 970008529 B1	24-05-1997
			US 5475856 A	12-12-1995

US 2009282223	A1	12-11-2009	KR 20090116511 A	11-11-2009
			US 2009282223 A1	12-11-2009
