

①2

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 29.06.90.

③0 Priorité : 15.09.89 US 407928.

④3 Date de la mise à disposition du public de la demande : 22.03.91 Bulletin 91/12.

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

⑥0 Références à d'autres documents nationaux apparentés :

⑦1 Demandeur(s) : SUN MICROSYSTEMS, INC — US.

⑦2 Inventeur(s) : Priem Curtis et Malachowsky Chris.

⑦3 Titulaire(s) :

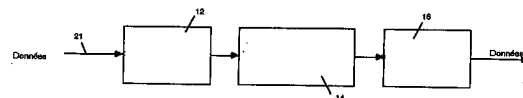
⑦4 Mandataire : Cabinet Regimbeau Martin Schrimpf Warcoïn Ahner.

⑤4 Dispositif de conversion de nombres.

⑤7 L'invention concerne un dispositif de conversion de nombres.

Ce circuit comporte un premier circuit de conversion de nombres incluant des moyens (12) pour convertir des nombres présentés dans un premier format de nombres en des nombres exprimés dans un format de nombre modulo 256 à points d'accès multiples, et des moyens (14) pour manipuler des nombres basés sur le format de nombre modulo 256 à points d'accès multiples, en utilisant des opérations classiques de transformation matricielle, et un second circuit de conversion de nombres (16).

Application notamment aux stations de travail informatiques à système d'affichage graphique.



La présente invention concerne des systèmes informatiques et plus particulièrement des dispositifs servant à convertir des nombres entre des formats en vue de leur utilisation dans une section arithmétique matricielle d'un accélérateur graphique à fonctionnement
5 extrêmement rapide.

Lors de la conception de systèmes informatiques, on cherche en permanence à accroître la vitesse de fonctionnement de tels systèmes et à les rendre aptes
10 à traiter des quantités plus importantes d'informations. Ces deux objectifs se traduisent directement par l'exécution d'une plus grande quantité de travail. La capacité des ordinateurs à exécuter une plus grande quantité de travail est également accrue par leur aptitude à
15 effectuer des types de travaux plus différents. Par exemple, on utilise des ordinateurs dans de nombreuses activités nouvelles, dans le cadre de l'arrivée de dispositifs d'affichage graphiques d'ordinateur, qui permettent de présenter non seulement des nombres et des
20 textes, mais des images représentant les significations de ces nombres et permettant de mieux comprendre les textes. L'aptitude à faire pivoter, translater et régler l'échelle de ces images a conduit à une utilisation extensive de l'ordinateur dans le domaine du dessin industriel et de la conception industrielle. Un grand nombre
25 de personnes dans le domaine de l'industrie de l'informatique se sont progressivement convaincues au fait qu'une représentation graphique doit être présentée dans la plupart des systèmes informatiques.

30 Malheureusement, la présentation d'un affichage de graphiques par un système informatique requiert une puissance de traitement conséquente de ce système. Par exemple, la présentation d'une seule image de graphiques sur le dispositif d'affichage de l'ordinateur d'un poste
35 de travail de taille standard satisfaisante requiert la

mémorisation d'une information incluant environ un millier de pixels dans une direction horizontale et environ un millier de pixels dans une direction verticale. C'est pourquoi il faut mémoriser une information qui concerne
5 environ un million de pixels devant être affiché pour chaque image. Dans un système qui peut présenter un certain nombre de couleurs différentes au niveau affichage, chacun de ces pixels peut contenir huit bits d'une information numérique concernant le pixel particulier. Par
10 conséquent, il faut traiter et mémoriser environ huit millions d'octets d'informations pour chaque trame ou image devant être présentée sur l'affichage de sortie.

On comprendra que, étant donné que les trames sont mises à jour trente fois par seconde dans l'affichage de sortie de manière à créer un déplacement sans
15 pillotement, la quantité totale d'informations devant être présentées dans le dispositif d'affichage de sortie représente un nombre très élevé. Le simple fait de traiter une telle quantité d'informations pour obtenir une
20 représentation de graphiques requiert un intervalle de temps important disponible pour une unité centrale de traitement (CPU) et peut ralentir sensiblement le fonctionnement même du plus rapide de tels processeurs. C'est pour cette raison que les systèmes informatiques incluent
25 maintenant de façon usuelle des accélérateurs graphiques aptes à assister l'unité centrale de traitement dans ses opérations en prenant en charge une partie de la fonction de traitement des données concernant l'affichage des graphiques sur le dispositif d'affichage de sortie de
30 l'ordinateur. Le déchargement de l'unité centrale de traitement d'une certaine partie des fonctions de traitement de graphiques à un accélérateur de graphiques permet d'accroître de façon substantielle la vitesse avec laquelle n'importe quel système d'ordinateur particulier
35 est à même de traiter des informations graphiques.

Par conséquent, des tentatives ont été faites pour concevoir des accélérateurs graphiques à fonctionnement très rapide. L'une des principales fonctions, que l'on peut obtenir avec un accélérateur graphique est le traitement du système arithmétique matriciel nécessaire pour déplacer les images de graphiques sur l'affichage de sortie d'un ordinateur. De telles opérations matricielles sont nécessaires pour manipuler à la fois des figures graphiques bidimensionnelles et tridimensionnelles de manière à faire tourner, translater, régler l'échelle et manipuler d'une autre manière les figures graphiques particulières devant être affichées sur l'affichage de sortie de l'ordinateur.

Un accélérateur graphique peut être très utile pour exécuter ces opérations étant donné qu'il libère l'unité centrale de traitement de la nécessité de recalculer en série différents sommets des figures devant être manipulés lors de chaque manipulation de la figure devant être affichée. Un accélérateur de graphiques peut accomplir les nombreuses opérations nécessaires à l'aide d'un traitement matériel des données et à accroître fortement le fonctionnement du système informatique.

Cependant il subsiste encore un problème principal pour l'obtention d'opérations extrêmement rapides. Ce problème est basé sur le fait qu'il est nécessaire d'utiliser un accélérateur graphique pour manipuler des données dans une pluralité de formats de nombres différents. Par exemple, l'information traitée par l'unité centrale de traitement apparaît normalement selon un format de nombres entiers qui doit apparaître sous ce format, lorsqu'elle est utilisée par un dispositif d'affichage de sortie étant donné qu'un dispositif d'affichage ne traite pas des fractions de pixels. D'autre part, de nombreuses manipulations exécutées avec des nombres très élevés utilisés pour le traitement scientifique requiè-

rent l'utilisation d'un format à virgule flottante. De tels nombres doivent pouvoir être représentés dans des graphiques associés à de tels projets scientifiques. Il est évident que de tels nombres à virgule flottante doivent être finalement convertis sous le format de nombres entiers pour une présentation sur un dispositif d'affichage de sortie de l'ordinateur.

En outre, le format décrit sous le sigle FRACT est particulièrement utile pour manipuler un type particulier d'affichage de graphiques dans un procédé désigné sous le terme de manipulation de formes. L'utilisation d'un tel système graphique avec un ordinateur est décrit dans la demande de brevet US N° de série 07/252 589 ayant pour titre "Procédé et dispositif pour manipuler des images" déposé aux noms de Rocchetti et Donato le 3 Octobre 1988. Le système des nombres FRACT utilise un format entièrement différent de celui des systèmes utilisant le format des nombres entiers et le format des nombres à virgule flottante.

Dans des systèmes informatiques de l'art antérieur, on faisait appel à l'unité centrale de traitement (CPU) pour exécuter la plupart des conversions de nombres. Ainsi, bien qu'un nombre à virgule flottante puisse être traité moyennant l'utilisation d'un coprocesseur à virgule flottante, il doit être finalement converti pour un processeur du système sous un format de nombres entiers de sorte qu'on peut l'utiliser pour afficher une sortie graphique particulière sur un dispositif d'affichage de sortie d'un ordinateur. La conversion de nombres entre des formats de nombres différents par un processeur est traitée en série et retarde sensiblement le fonctionnement du système. Par exemple, pour convertir un nombre à virgule flottante en un nombre entier en utilisant l'unité centrale de traitement, il est nécessaire d'indiquer à l'unité centrale de traitement le format, avec lequel

l'information est représentée, et d'indiquer à l'unité centrale de traitement le format, dans lequel la sortie est désirée, puis d'amener l'unité centrale de traitement à se référer à un processus auxiliaire pour convertir le
5 nombre, pour obtenir la sortie du traitement auxiliaire et finalement d'utiliser une sortie dans un système basé sur de nouveaux nombres. Il est évident aux spécialistes de la technique que cette opération ralentit de façon sensible le fonctionnement de n'importe quel système in-
10 formatique.

C'est pourquoi un but de la présente invention est d'accélérer le fonctionnement d'un système informatique en prévoyant des dispositions permettant de traiter rapidement la conversion de nombres entre des formats de
15 nombres différents dans un accélérateur graphique utilisé avec un tel système informatique.

Cet objectif et d'autres objectifs de la présente invention sont atteints à l'aide d'un accélérateur graphique qui comporte un système de transformation
20 matricielle incluant des premiers et seconds circuits de conversion de nombres, un premier de ces circuits incluant des moyens pour convertir des nombres présents selon les formats des nombres entiers, de nombres à virgule flottante et de nombres FRACT en des nombres exprimés
25 dans le format de nombres modulo 256 à points d'accès multiples, tandis que le second circuit inclut des moyens pour convertir des nombres basés selon le format de nombres modulo 256 à points d'accès multiples en les formats nombres entiers, de nombres à virgule flottante
30 et de nombres FRACT, en utilisant des opérations classiques de transformation matricielle.

D'autres caractéristiques et avantages de la présente invention ressortiront de la description donnée ci-après prise en référence aux dessins annexés, sur les-
35 quels les éléments identiques sont désignés par les mêmes

chiffres de référence et parmi lesquels :

- la figure 1 montre une illustration de trois formats de nombres qui peuvent être utilisés par l'accélérateur graphique associé à la présente invention;

5 - la figure 2 représente le format modulo 256 à points d'accès multiples utilisé de façon interne par le circuit de transformation matricielle conforme à la présente invention;

10 - la figure 3 représente un schéma-bloc illustrant le circuit du système de transformation matricielle conforme à la présente invention;

- la figure 4 représente un schéma-bloc illustrant le circuit servant à réaliser le processus de conversion d'entrée dans la présente invention; et

15 - la figure 5 représente un schéma-bloc montrant le circuit servant à réaliser le processus de conversion de sortie dans la présente invention.

On va maintenant certaines parties des descriptions détaillées sous la forme d'algorithmes et des représentations symboliques d'opérations effectuées sur des bits de données dans une mémoire d'ordinateur. Ces descriptions et représentations sous forme d'algorithmes sont les moyens utilisés par les spécialistes de la technique de traitement des données pour faire comprendre de la manière la plus efficace la nature de leur travail à d'autres spécialistes de la technique.

Un algorithme est ici, et d'une manière générale, conçu comme étant une séquence autocohérente d'étapes aboutissant à un résultat désiré. Les étapes sont celles nécessitées par des manipulations physiques effectuées sur des grandeurs physiques. Habituellement, mais non nécessairement, ces grandeurs possèdent la forme de signaux électriques ou magnétiques aptes à être mémorisés, transférés, combinés, comparés et manipulés d'une autre manière. Il s'est avéré approprié parfois, princi-

palement en raison de l'usage commun, de se référer à ces signaux sous le terme de bits, de valeurs, d'éléments, de symboles ou de caractères, de termes, de nombres ou analogues. Cependant, il faut se rappeler que tous ces termes et d'autres termes similaires peuvent être associés aux grandeurs physiques appropriées et sont simplement des appellations commodes appliquées à ces grandeurs.

En outre, les manipulations exécutées sont souvent désignées sous la forme de termes tels que additions ou comparaisons, qui sont habituellement associés à des opérations mentales effectuées par un opérateur humain. Dans la plupart des cas, aucune capacité de ce type d'un opérateur humain n'est nécessaire ou souhaitable dans n'importe laquelle des opérations décrites ici, qui font partie de la présente invention; les opérations sont des opérations machine. Des machines utiles pour exécuter les opérations conformes à la présente invention incluent des ordinateurs à usage général ou d'autres dispositifs semblables. Dans tous les cas, on conservera en mémoire la distinction existant entre les opérations relatives à un procédé pour le fonctionnement d'un ordinateur et le procédé de calcul lui-même. La présente invention concerne un dispositif et des étapes opératoires pour faire fonctionner un ordinateur pour qu'il traite des signaux électriques ou d'autres signaux physiques (c'est-à-dire mécaniques, chimiques) pour produire d'autres signaux physiques désirés.

En se référant maintenant à la figure 1, on y voit représentés trois formats de nombres, que l'on peut utiliser dans un système informatique comportant un accélérateur graphique agencé conformément à la présente invention. Le premier format de nombre représenté est le format de nombres entiers, qui inclut trente-deux bits d'informations dans le type de complément à deux. La

virgule décimale (en réalité binaire) est supposée apparaître à droite du bit situé le plus à droite. Le format de nombres entiers permet de traiter des nombres allant de -2^{31} à $2^{31}-1$ dans un format à trente-deux bits. Les spécialistes de la technique comprendront que le format de nombres entiers permet de traiter uniquement des nombres entiers et ne permettent pas de traiter des fractions ni des nombres décimaux.

Le second format représenté sur la figure 1 est ce qu'on appelle le format FRACT. Dans ce format, trente-deux bits de mémoire sont prévus dans ce format de complément à deux. Les seize bits les plus à gauche dans le format FRACT représentent des nombres entiers binaires, tandis que les seize bits les plus à droite représentent les parties fractionnelles de ces nombres. Bien que les bits de la partie entière représentent des puissances positives de deux commençant par 2^0 et augmentant vers la gauche à partir de la virgule binaire, les bits de la partie fractionnaire représentent des puissances négatives de deux, commençant par 2^{-1} , la puissance négative augmentant vers la droite à partir de la virgule. Lorsqu'on utilise ce format, on peut aisément représenter le nombre entier et une partie fractionnaire. Le format FRACT est particulièrement utile pour le traitement d'objets graphiques en vue de leurs représentations sur un dispositif d'affichage de sortie d'un ordinateur, comme on le comprendra à partir de la description fournie dans la demande de brevet copendante mentionnée précédemment.

Le troisième format de nombre représenté sur la figure 1 est le format à virgule flottante à précision simple du Institut of Electrical and Electronics Engineers (IEEE), dont les trente-deux positions binaires incluent un bit de signe placé complètement à gauche, huit bits qui sont utilisés pour représenter un exposant et 23

bits, qui sont utilisés pour représenter la mantisse, sans signe, d'un nombre à virgule flottante. Dans la représentation IEEE, un bit supplémentaire, qui représente un zéro de tête et est supposé être présent tout à fait à gauche de la mantisse à vingt-trois bits de sorte que la mantisse inclut en réalité vingt-quatre bits quand on utilise cette représentation. Le format à virgule flottante à précision simple IEEE permet de représenter des nombres compris entre environ -2^{128} et 2^{127} .

10 Les spécialistes de la technique constateront qu'un processus permettant de traiter des nombres dans le format de nombres entiers ne peut probablement pas traiter des nombres ni dans le format FRACT, ni dans le format à précision simple IEEE. De façon analogue, un processus permettant de traiter des nombres dans le format FRACT est probablement incapable de traiter des nombres dans le format de nombres entiers et dans le format à précision simple IEEE. Enfin, un processus permettant de traiter des nombres dans le format à précision simple IEEE est probablement incapable de traiter des nombres dans le format de nombres entiers et dans le format FRACT. C'est pour cette raison qu'on utilisait, d'une manière générale, dans des systèmes informatiques de l'art antérieur, l'unité centrale de traitement ou un coprocesseur à virgule flottante pour convertir des nombres d'un format en un autre avant qu'il soit traité par l'accélérateur de graphiques associé à ce système. Cependant, comme cela a été indiqué précédemment, une telle manipulation ralentit sensiblement le fonctionnement du système informatique pour la présentation d'affichage de sortie de graphiques.

L'invention supprime l'obligation d'utiliser l'unité centrale de traitement pour réaliser des conversions de nombres et de ce fait accélère notablement le fonctionnement de l'accélérateur de graphiques. Il

exécute ces tâches en fournissant un nouveau format de nombres, en lequel chacun des trois formats de nombres représenté sur la figure 1 est converti avant sa manipulation par l'accélérateur de graphiques. Ce format de nombres est utilisé de façon interne par le système de transformation de l'accélérateur de graphiques pour exécuter toutes les opérations arithmétiques matricielles. Des nombres délivrés sur la base de l'utilisation de ce nouveau format de nombres sont convertis de façon appropriée en chacun des formats incluant le format de nombres entiers, le format FRACT et le format à virgule flottante à précision simple IEEE en vue d'être utilisé ensuite par le système une fois que les manipulations du système de transformation ont été effectuées. Le système est également capable de recevoir l'information en provenance du circuit extérieur et de la délivrer à ce circuit dans le format interne.

La figure 2 représente ce nouveau format de nombres, qui est désigné sous le terme de format de nombre modulo 256 à points d'accès multiples (désigné ci-après par "modulo 256"), utilisé conformément à la présente invention. Ce format comprend trente-six positions binaires individuelles incluant les quatre bits les plus à gauche utilisés pour représenter un exposant, les huit bits suivants utilisés pour représenter un élément d'un nombre entier pourvu d'un signe, et les vingt-quatre bits les plus à droite utilisés pour représenter un composant fractionnaire comme dans le format de nombres FRACT. Ce format de nombres est utilisé d'une manière légèrement différente que n'importe lequel des autres formats de nombres.

Cependant il est aisément adaptable pour la conversion à partir de chacun des formats comme cela va être décrit. Bien que seuls quatre bits soient prévus pour les exposants, tout accroissement d'une unité de

l'exposant est considéré comme une multiplication de la mantisse par 256 (un décalage de huit bits). En outre, le nombre de bits disponibles pour la mantisse est nettement plus élevé que dans le cas du format à simple précision IEEE. Par conséquent l'espace suffisant pour loger la majeure partie des nombres rencontrés est fourni par le nouveau format.

Pour lire un nombre représenté dans le format modulo 256, la mantisse pourvue d'un signe multiplié par 256 est élevé à la puissance indiquée par les quatre bits de l'exposant moins huit (le bit de rang élevé de l'exposant). Le bit de rang élevé de l'exposant est un pour tous les exposants positifs. Le un présent dans le bit de rang élevé du format modulo 256 permet la conversion aisée en direction et à partir du format IEEE. La nécessité de la présence d'un dans la position binaire élevée requiert cependant de soustraire un huit décimal pour obtenir la valeur correcte de l'exposant.

On va considérer un nombre binaire présent dans la mantisse, la virgule binaire étant positionnée comme représenté sur la figure 2. Si les quatre bits de l'exposant sont 1000, la virgule binaire reste dans sa position initiale comme représenté sur la figure 2, huit bits étant présents à droite du bit de poids le plus faible de l'exposant. Si les bits de l'exposant sont 1001, ce qui indique un exposant un, la virgule binaire est effectivement décalée de huit bits vers la droite de la position représentée sur la figure 2. Si l'exposant est 1010, ce qui indique un exposant égal à deux, la virgule binaire est décalée effectivement de seize bits vers la droite de la position représentée sur la figure 2.

Il est très facile de passer des différents formats classiques de nombres au format modulo 256 et inversement. Par exemple, la conversion à partir du format de nombres entiers requiert simplement que le nombre en-

tier soit placé avec son bit de poids le plus faible dans l'espace du bit le plus à droite de l'espace de la mantisse modulo 256 et que l'exposant du format modulo 256 soit fixé à 1011 (ce qui représente un exposant correspondant aux trois décimales. Cette valeur de l'exposant 5 provoque un décalage effectif de la virgule binaire séparant le nombre entier et la partie fractionnaire depuis sa position normale de trente-quatre bits à gauche du bit le plus à droite de la mantisse jusqu'à la position de 10 droite finale de la mantisse, ce qui a pour effet que le nombre mémorisé représente un nombre entier. Ce nombre est ensuite normalisé de manière que les bits s'insèrent mieux dans l'espace fourni par le format modulo 256, moyennant le déplacement de la virgule binaire aussi loin 15 que possible vers la gauche selon des échelons de huit bits, et au moyen de la réduction d'une unité de l'exposant pour chaque échelon. Ainsi, si le nombre entier pourvu d'un signe nécessitait globalement huit bits ou moins, il serait déplacé de trente-quatre bits vers la 20 gauche de manière à être situé dans l'espace disponible, et l'exposant serait réduit de trois et deviendrait 1000, de sorte que la virgule binaire serait ramenée dans sa position normale dans le format modulo 256. D'autre part, si le nombre entier nécessitait neuf bits, mais moins de 25 dix-sept bits au total, il serait déplacé sur seulement seize bits vers la gauche pour s'insérer dans l'espace disponible, les exposants seraient réduits de deux et deviendraient 1001, de sorte que la virgule binaire serait située à huit bits vers la droite de sa position normale 30 dans le format modulo 256.

Pour réaliser la conversion depuis le format de nombres FRACT en le format de nombres modulo 256, il suffit que le nombre FRACT soit placé dans sa position normale son bit de poids le plus faible étant situé dans 35 l'espace du bit le plus à droite de l'espace de la man-

tisse modulo 256 et que l'exposant modulo 256 soit réglé à 1001 (augmenté d'une unité) étant donné que ceci provoque le déplacement effectif de la virgule binaire de huit bits vers la droite comme cela est décrit dans la discussion relative à la figure 2. Ensuite, le nombre FRACT est normalisé de la même manière que les nombres entiers par déplacement de la partie entière aussi loin que possible vers la gauche, par incrément de huit bits, et par réduction d'une unité de l'exposant pour chaque tel échelon. Par conséquent, si la partie formant nombre entier pourvu d'un signe nécessitait au total huit bits ou moins, elle déplacerait huit bits vers la gauche de manière à les insérer dans l'espace disponible, et l'exposant serait réduit de un en passant à 1000 de sorte que la virgule binaire serait ramenée dans sa position normale dans le format modulo 256. Ceci est particulièrement avantageux dans le format FRACT étant donné qu'il permet d'obtenir la précision maximale avec l'espace disponible.

Enfin, la conversion d'un nombre exprimé dans le format de précision simple IEEE en le format modulo 256 requiert que la mantisse non pourvue d'un signe soit convertie en un nombre pourvu d'un signe et que le nombre soit décalé de manière à régler les différents bits de base de l'exposant. Une forme différente de normalisation est nécessaire pour des nombres indiqués dans le format à virgule flottante à simple position IEEE. De tels nombres incluent une mantisse et un exposant; cependant l'exposant dans le format IEEE est une puissance de deux. Par conséquent il est nécessaire de compenser la différence de la valeur attribuée à l'exposant dans le format à virgule flottante et dans le format modulo 256. Etant donné que chaque accroissement de l'unité dans l'exposant du nombre à virgule flottante est un accroissement d'une puissance simple de deux, alors que chaque accroissement

d'une unité dans l'exposant du format modulo 256 représente 256 à une puissance additionnelle, il est nécessaire de compenser cette différence en déplaçant la virgule binaire dans la mantisse. Par exemple, le nombre
5 $0.1011110X2^{18}$ dans le format à virgule flottante est représenté par le nombre $10.11110X256^2$ dans le format modulo 256. D'autre part, le nombre $0,1011110X2^6$ dans le format à virgule flottante peut être représenté par le nombre $101111.0X 256^0$ dans le format modulo 256. Dans
10 chacun de ces cas, la conversion requiert que, lorsque l'exposant représentant une puissance de deux est changé en une puissance de 256, la virgule binaire dans la mantisse soit déplacée. Par conséquent, il est souvent nécessaire à la fois de modifier l'exposant pour représen-
15 ter une puissance correcte de 256 et de décaler la virgule binaire dans la mantisse, d'un nombre sélectionné de positions binaires individuelles, de manière à normaliser les nombres lors de la conversion depuis le format à virgule flottante à simple précision IEEE dans le format mo-
20 dulo 256.

On comprendra que, bien que seuls quatre bits soient prévus pour les exposants dans le format modulo 256, alors que huit bits sont prévus dans le format à simple précision IEEE, huit bits additionnels sont prévus
25 pour la mantisse du nombre dans le format modulo 256. Par conséquent, le nombre exprimé fournit presque la gamme complète des nombres dans le format à simple précision IEEE. Plus particulièrement, les nombres, qui peuvent être exprimés dans le format modulo 256 pour représenter
30 les nombres du format à précision simple IEEE, vont de moins 2^{64} à $2^{63}-1$. Bien que ceci n'inclut pas une puissance entière de deux à chaque extrémité de la gamme et permette d'exprimer moins que ce que peut exprimer le format à simple précision IEEE, il s'est avéré être en-
35 tièrement suffisant pour exprimer des nombres utilisés

dans des dispositifs d'affichage de sortie graphiques. En outre un nombre peut être conservé d'une manière plus précise lors de l'exécution d'opérations dans le format modulo 256 en raison du nombre supérieur de bits disponibles dans la mantisse.

La figure 3 représente, sous la forme d'un schéma-bloc, l'agencement de base du circuit conforme à la présente invention.

Comme cela est représenté sur la figure 3, des signaux d'entrée représentant des nombres entiers, des nombres FRACT et des nombres à format à virgule flottante, sont reçus à partir d'un bus de transmission de données dans une unité de conversion d'entrée 12. Dans le système dans lequel la présente invention est utilisée, les nombres entiers sont envoyés à une adresse de sorte qu'ils peuvent être détectés par l'unité de conversion d'entrée 12, cette unité sachant qu'ils sont présents sous ce format. De façon analogue, des nombres apparaissant respectivement dans le format d'entrée FRACT et dans le format d'entrée à virgule flottante, sont envoyés à des adresses de sorte qu'ils sont identifiés par l'unité de conversion d'entrée 12 comme apparaissant dans ces formats, qui doivent être convertis et normalisés de la manière décrite précédemment pour chacun de ces formats.

Les nombres présents dans l'unité de conversion d'entrée sont convertis dans le format modulo 256 et sont transférés au moteur ou système de transformation 14 de l'accélérateur graphique. La figure 4 représente sous la forme d'un schéma-bloc le circuit de l'unité de conversion d'entrée 12 servant à exécuter cette conversion d'entrée. Le circuit 12 représenté sur la figure 4 inclut un bus de transmission de données 21, dans lequel des données d'entrée sont reçues selon le format de nombres entiers, le format FRACT ou le format IEEE. Les données à virgule flottante sont transférées à un convertisseur 200 trai-

tant des nombres complémentaires à deux et sans signe, et un calculateur 201 de l'exposant à virgule flottante. Le calculateur 201 de l'exposant à virgule flottante calcule la modification des exposants conformément à la description donnée précédemment et commande un dispositif de décalage à tambour 202 servant à décaler la mantisse sur un nombre correct de bits, selon des incréments de un bit, de sorte que les nombres présents dans les deux formats sont équivalents. Le calculateur à virgule flottante délivre un exposant destiné à être utilisé dans le format modulo 256 tandis que le dispositif de décalage à tambour 202 fournit une mantisse pourvue d'un signe. Ces bits sont transmis à un multiplexeur 220 pour être utilisés par le système de transformation 14 représenté sur la figure 3.

Les données formées de nombres entiers et les données FRACT apparaissant dans le bus de transmission de données 21 sont envoyées à un calculateur 210 d'exposants de nombres entiers/nombres FRACT, qui calcule l'exposant approprié pour le nombre converti conformément aux détails décrits précédemment. Le calculateur 210 envoie un exposant approprié au multiplexeur 220 pour le format modulo 256 et commande un dispositif de décalage à tambour à huit bits 211 pour qu'il envoie au multiplexeur 220 une mantisse décalée du nombre approprié de positions binaires. En fonction du format converti, le multiplexeur 25 est activé de manière à transférer les bits appropriés et l'exposant au système de transformation 14.

Le système de transformation 14 traite les nombres conformément à des formules matricielles standard bien connues dans la technique et adaptées cependant au format unique de nombres conformes à la présente invention et envoie des signaux de sortie à l'unité de conversion de sortie 16. Les particularités du circuit permettant de réaliser la transformation de nombres sont

décrites dans la demande de brevet copendante ayant pour titre "Dispositif pour manipuler des nombres dans un ordinateur", déposé au nom de Priem et Malachowski, et possédant la même date de priorité de la présente demande.

5 L'unité de conversion 16 convertit des nombres depuis le format modulo 256 en chacun des formats de nombres entiers, de nombres FRACT et de nombres à virgule flottante, en vue de leur utilisation ultérieure par le système. Par exemple, le format de nombres entiers est le

10 format, dans lequel des nombres doivent être utilisés pour la mémorisation dans le tampon d'images de sortie. D'autre part, le format FRACT, le format à virgule flottante et le format interne modulo 2546 peuvent être utilisés par le système informatique pour d'autres opérations

15 effectuées par ce système, qui ne concerne pas la mémorisation dans la mémoire tampon de frame.

La figure 5 représente un circuit utilisé pour réaliser la conversion du format de nombres modulo 256 au format de nombres entiers, au format FRACT et au format à

20 virgule flottante. L'unité de conversion de sortie 16 représentée sur la figure 5 reçoit des données dans le format modulo 256. Si les données doivent être utilisées dans ce format, elles sont transférées directement à un premier multiplexeur 303 et, à partir de là, à un second

25 multiplexeur 320 en vue de leur utilisation par d'autres parties du système. Le multiplexeur 303 est nécessaire pour diviser les trente-six bits du format modulo 256 en dix-huit positions binaires, qui peuvent être utilisées par le système.

30 Si le signal de sortie désiré doit être présent selon le format IEEE, les trente-deux bits de la mantisse pourvue d'un signe sont transférés à un convertisseur 300 de demande de nombres complémentaires 2 et en des nombres non pourvus d'un signe. Le bit de signe est transféré di-

35 rectement au multiplexeur 320, tandis que les autres bits

sont transférés à un dispositif de décalage à tambour 302, qui peut décaler la mantisse du nombre modulo 256, d'un nombre sélectionné de positions de bits individuels. Le nombre de bits dont la mantisse doit être décalée, est commandé par

5 un circuit transformé de calcul d'exposants flottants, qui fonctionne conformément à la description indiquée précédemment pour normaliser les exposants des deux formats et commander le décalage de la mantisse moyennant l'utilisation du dispositif de décalage à tambour 302.

10 L'exposant situé dans le format IEEE est transféré depuis le calculateur 301 au multiplexeur 320, tandis que la mantisse, décalée de façon appropriée, est transférée du dispositif de décalage 302 au multiplexeur 320.

Si la donnée doit être convertie soit le format

15 FRACT, soit dans le format de nombres entiers, l'exposant est transféré à un dispositif 310 de commande d'exposants de nombres entiers/de nombres FRACT, qui calcule l'exposant approprié dans le format désiré de la manière décrite précédemment, et commande le dispositif de déca-

20 lage 311 pour qu'il décale la mantisse du nombre approprié d'incrémentes de huit bits de manière à la positionner correctement. l'exposant est transféré par le dispositif de commande 310 au multiplexeur 320, et la mantisse, décalée de façon appropriée, est transférée de

25 puis le dispositif de décalage 311 au multiplexeur 320.

Bien que la présente invention ait été décrite dans une forme de réalisation préférée, les spécialistes de la technique noteront que l'on peut y apporter de nombreux changements et modifications sans sortir du cadre

30 de l'invention.

REVENDEICATIONS

1. Circuit arithmétique matriciel, caractérisé en ce qu'il comporte un premier circuit de conversion de nombres incluant des moyens (12) pour convertir des nombres présentés dans un premier format de nombre en des nombres exprimés dans un format de nombre modulo 256 à points d'accès multiples, et des moyens (14) pour manipuler des nombres basés sur le format de nombre modulo 256 à points d'accès multiples en utilisant des opérations classiques de transformation matricielle.

2. Circuit arithmétique matriciel selon la revendication 1, caractérisé en outre en ce qu'il comporte un second circuit (16) de conversion de nombres incluant des moyens pour convertir des nombres exprimés dans le format de nombre modulo 256 à points d'accès multiples en des nombres présentés dans le premier format de nombre.

3. Circuit arithmétique matriciel selon la revendication 1, caractérisé en ce que le premier format de nombre est le format des nombres entiers.

4. Circuit arithmétique matriciel selon la revendication 1, caractérisé en ce que le premier format de nombre est le format des nombres à virgule flottante.

5. Circuit arithmétique matriciel selon la revendication 1, caractérisé en ce que le premier format de nombre est le format des nombres dits "FRACT".

6. Circuit arithmétique matriciel selon la revendication 1, caractérisé en ce que le premier circuit (12) de conversion de nombres inclut en outre des moyens (210) pour convertir des nombres présentés dans un second format de nombres en des nombres exprimés avec le format de nombre modulo 256 à points d'accès multiples.

7. Circuit arithmétique matriciel selon la revendication 1, caractérisé en ce que le premier circuit de conversion de nombres (12) comporte en outre des moyens pour convertir des nombres présentés dans un second

format de nombres en des nombres exprimés . dans le format de nombres modulo 256 à points d'accès multiples, et que le second circuit (16) de conversion de nombres inclut des moyens pour convertir des nombres exprimés dans le format de nombres modulo 256 à points d'accès multiples en des nombres présentés dans le premier format de nombres, et des moyens pour convertir des nombres exprimés dans le format de nombre modulo 256 à points d'accès multiples en des nombres présentés dans le second format de nombres.

10 8. Circuit arithmétique matriciel selon la revendication 6, caractérisé en ce que le premier format de nombre est le format des nombres entiers et que le second format de nombre est un format des nombres à virgule flottante.

15 9. Circuit arithmétique matriciel selon la revendication 7, caractérisé en ce que le premier format de nombre est le format des nombres entiers et que le second format de nombre est un format de nombres à virgule flottante.

20 10. Circuit arithmétique matriciel selon la revendication 6, caractérisé en ce que le premier format de nombre est le format de nombres entiers et que le second format de nombre est le format dit "FRACT".

25 11. Circuit arithmétique matriciel selon la revendication 7, caractérisé en ce que le premier format de nombre est le format de nombres entiers et que le second format de nombre est le format dit "FRACT".

30 12. Circuit arithmétique matriciel selon la revendication 6, caractérisé en ce que le premier format de nombre est un format de nombres à virgule flottante et que le second format de nombre est le format dit "FRACT".

35 13. Circuit arithmétique matriciel selon la revendication 7, caractérisé en ce que le premier format de nombre est un format de nombres à virgule flottante et

que le second format de nombre est le format dit "FRACT".

14. Circuit arithmétique matriciel selon la revendication 6, caractérisé en ce que le premier circuit
5 de conversion de nombres comprend en outre des moyens (12) pour convertir des nombres présentés dans un troisième format de nombre en des nombres exprimés dans le format de nombres modulo 256 à points d'accès multiples.

15. Circuit arithmétique matriciel selon la revendication 6, caractérisé en ce que le premier circuit
10 de conversion de nombres comprend en outre des moyens (12) pour convertir des nombres présentés dans un troisième format de nombre en des nombres exprimés dans le format de nombres modulo 256 à points d'accès multiples, et que
15 le second circuit (16) de conversion de nombres comporte en outre des moyens pour convertir des nombres exprimés, dans le format de nombres modulo 256 à points d'accès multiples en des nombres présentés dans le troisième format de nombre .

20 16. Circuit arithmétique matriciel selon la revendication 14, caractérisé en ce que le premier format de nombre est le format des nombres entiers, le second format de nombre est un format de nombres à virgule flottante et le troisième format de nombre est le format
25 dit "FRACT".

17. Circuit arithmétique matriciel selon la revendication 15, caractérisé en ce que le premier format de nombre est le format des nombres entiers, le second format de nombre est un format de nombres à virgule
30 flottante et le troisième format de nombre est le format dit "FRACT".

18. Circuit arithmétique matriciel selon la revendication 1, caractérisé en ce que les moyens servant à convertir des nombres présentés dans le premier format de
35 nombres en des nombres exprimés dans le format de nombres

modulo 256 à points d'accès multiples comprennent des moyens (211;311) pour décaler la virgule binaire d'un nombre présenté dans un tel premier format de nombre selon des incréments de huit bits et conserver des bits indiquant la valeur du décalage selon des valeurs de 256 élevé à une puissance.

19. Circuit arithmétique matriciel, caractérisé en ce qu'il comporte un premier circuit (12) de conversion de nombres incluant des moyens pour convertir des nombre présentés dans un premier format de nombre en des nombre exprimés dans le format de nombre modulo 256 à points d'accès multiples, des moyens pour manipuler des nombres basés dans le format de nombre modulo 256 à points d'accès multiples en utilisant des opérations classiques de transformation matricielle, et un second circuit (16) de conversion de nombres incluant des moyens pour convertir des nombres exprimés dans le format de nombre modulo 256 à points d'accès multiples en des nombres présentés dans le premier format de nombre, les conversions entre formats de nombres et les manipulations de nombres étant telles que la précision des nombres est conservée lors de chaque opération.

20. Circuit arithmétique matriciel selon la revendication 19, caractérisé en ce que les moyens pour convertir des nombres présentés dans un premier format de nombre en des nombres exprimés dans un format de nombres modulo 256 à points d'accès multiples comprennent des moyens (211;311) pour décaler la virgule binaire d'un nombre situé dans un tel premier format de nombre selon des incréments de huit bits et à conserver des bits indiquant la valeur du décalage sous la forme de valeurs de 256 élevées à une puissance.

21. Procédé pour traiter des nombres, caractérisé en ce qu'il inclut les étapes consistant à convertir des nombres présentés dans un premier format de nombre en

des nombres exprimés dans un format de nombre modulo 256 à points d'accès multiples, manipuler des nombres basés dans le format de nombre modulo 256 à points d'accès multiples en utilisant des opérations classiques de transformation matricielle, et convertir des nombres exprimés dans le format de nombre modulo 256 à points d'accès multiples en des nombres présentés dans le premier format de nombre .

22. Procédé pour traiter des nombres selon la revendication 21, caractérisé en ce que le premier format de nombre est le format des nombres entiers.

23. Procédé pour manipuler des nombres selon la revendication 21, caractérisé en ce que le premier format de nombre est un format de nombres à virgule flottante.

24. Procédé pour manipuler des nombres selon la revendication 21, caractérisé en ce que le premier format de nombre est un format de nombres dit "FRACT".

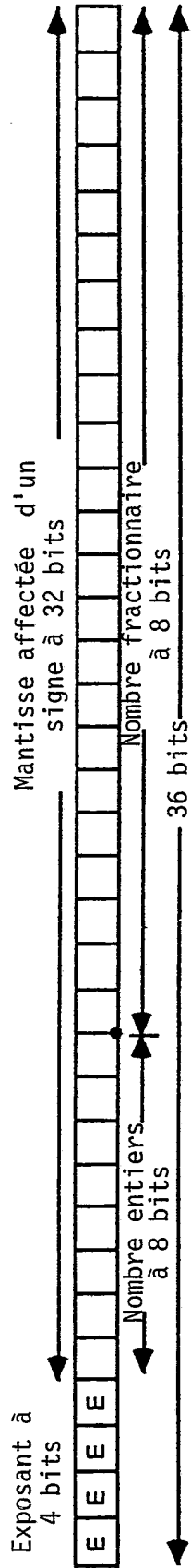


FIGURE 2

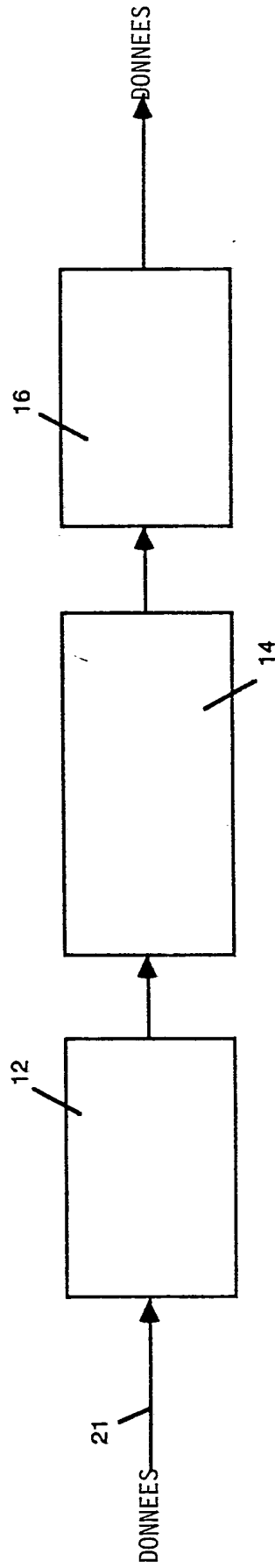


FIGURE 3

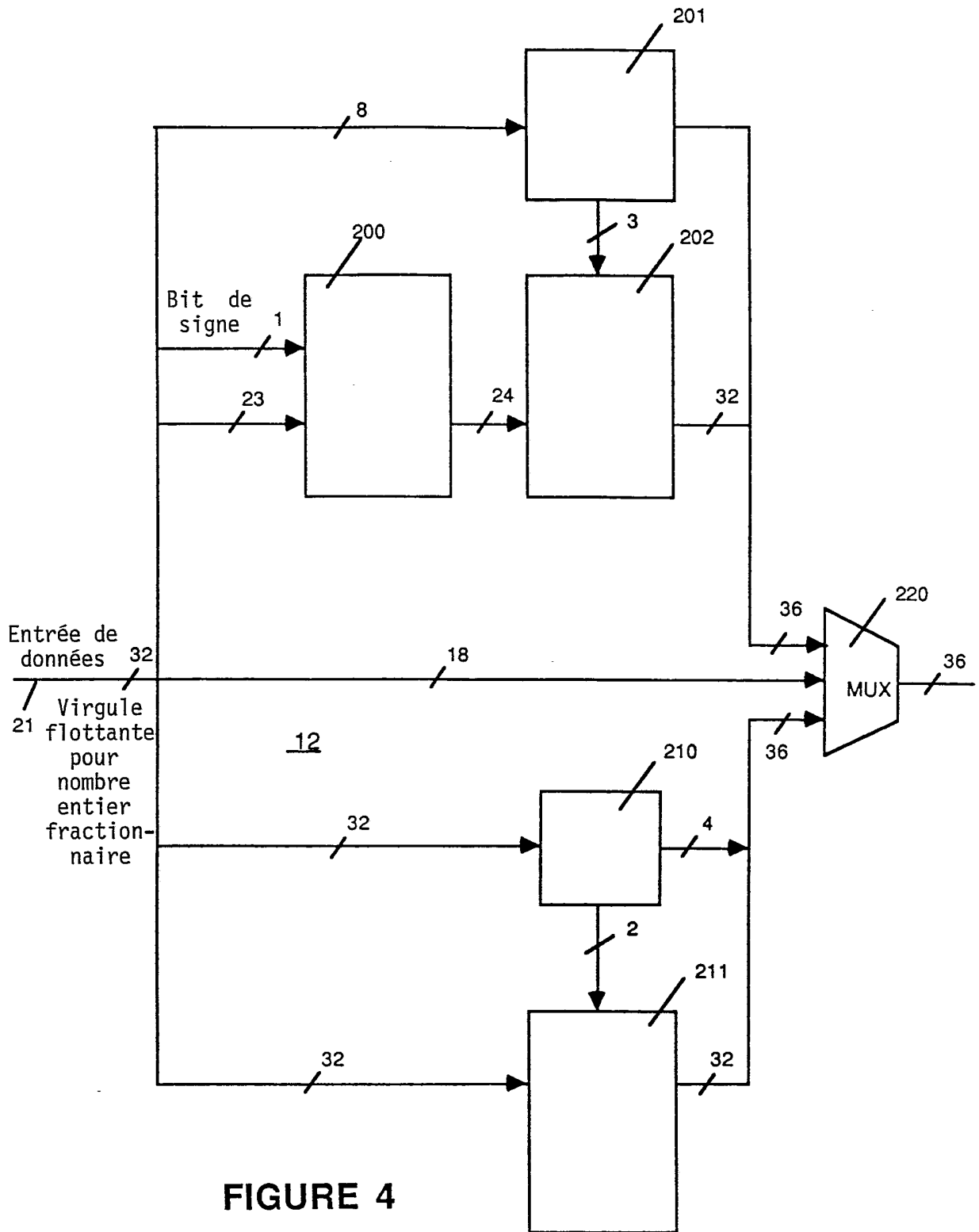


FIGURE 4

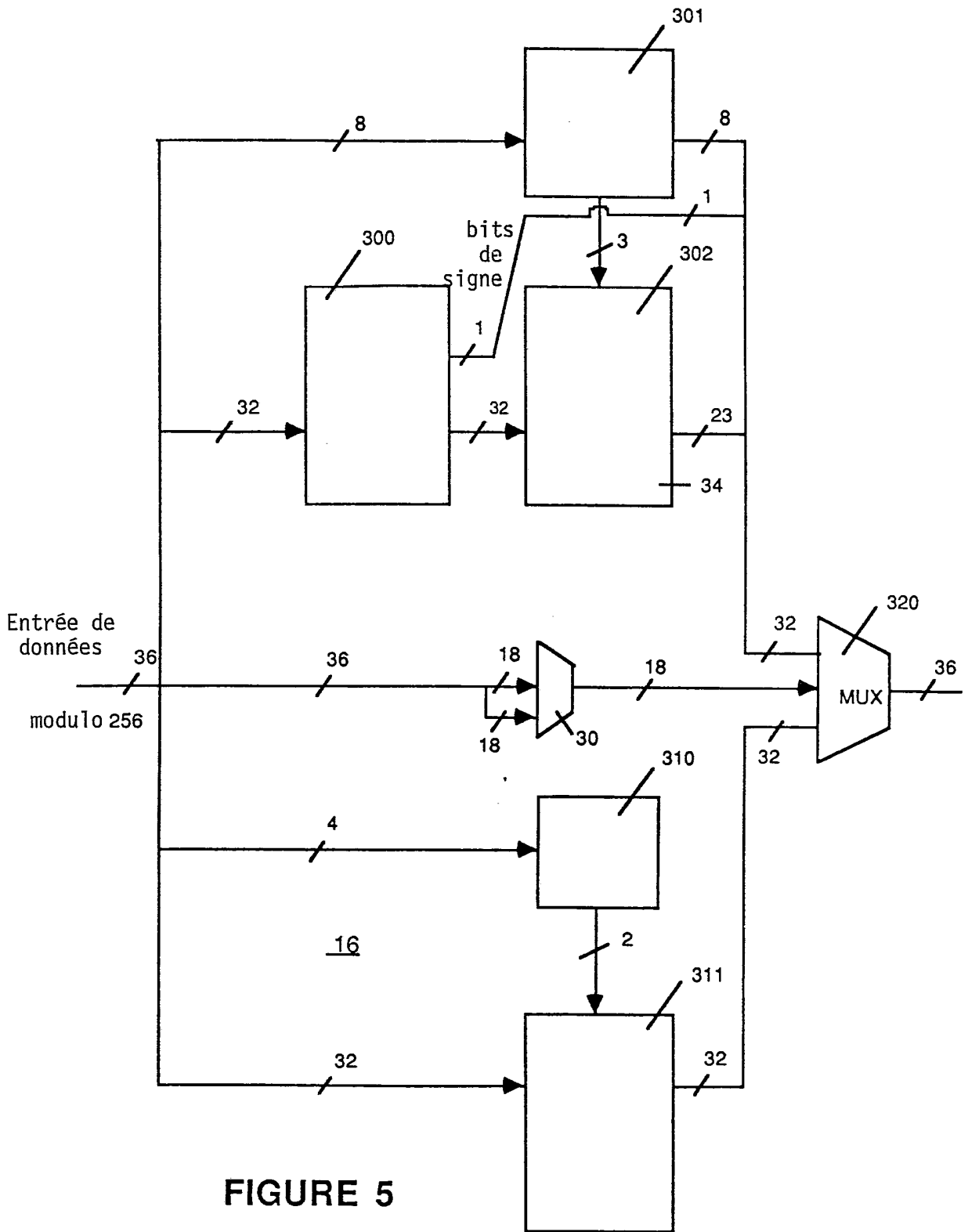


FIGURE 5