

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 30.06.00.

30 Priorité :

43 Date de mise à la disposition du public de la
demande : 04.01.02 Bulletin 02/01.

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) : GEMPLUS Société en commandite
par actions — FR.

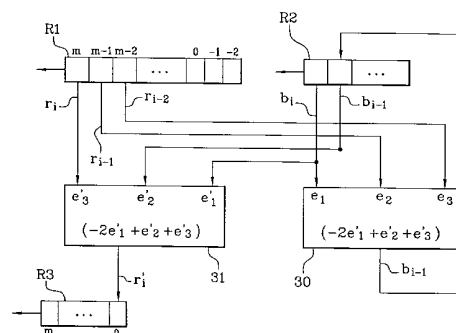
72 Inventeur(s) : JOYE MARC et YEN SUNG MING.

73 Titulaire(s) :

74 Mandataire(s) : CABINET BALLOT.

54 PROCÉDE DE CONVERSION DE LA REPRÉSENTATION BINAIRE D'UN NOMBRE DANS UNE
REPRÉSENTATION BINAIRE SIGNÉE.

57 Un procédé de conversion dans une représentation
binaire signée ($r'_m, \dots, r'_0$) d'un nombre r est basé sur un traite-
ment gauche vers droite des bits de la représentation bi-
naire ($r_{m-1}, \dots, r_0$) et permet d'obtenir une représentation
équivalente de la représentation dite de Reitwiesner. L'utili-
sation d'un tel procédé de conversion avec des traitements
arithmétiques de type gauche droite permettent d'améliorer
les performances de calcul et facilite leur réalisation maté-
rielle.



PROCEDE DE CONVERSION DE LA REPRESENTATION BINAIRE D'UN
NOMBRE DANS UNE REPRESENTATION BINAIRE SIGNEE

La présente invention concerne un procédé d'obtention d'une représentation binaire signée d'un nombre à partir de sa représentation binaire.

Les techniques de calcul arithmétique rapide utilisent
5 depuis longtemps la représentation binaire signée, parce qu'elle permet de calculer de manière très efficace des additions, multiplications ou divisions.

De nombreux circuits intégrés comprennent un dispositif de conversion d'un nombre en code binaire signé à
10 partir de sa représentation binaire et des dispositifs de calcul arithmétique pour effectuer des opérations avec cet opérande en représentation binaire signée. En général, un dispositif de conversion inverse est prévu, pour fournir le résultat sous la forme binaire.

15 On rappelle que dans la représentation binaire signée, un nombre est exprimé sur m chiffres de valeurs "0", "1" ou "-1". Plusieurs représentations binaires signées d'un même nombre existent. Par exemple, le nombre 5 peut s'écrire comme $(1\ 1\ -1)$, soit $1 \times 2^2 + 1 \times 2^1 - 1 \times 2^0$.

20 Mais il peut aussi s'écrire $(1\ 0\ 1)$.

Parmi les représentations binaires signées possibles, il en existe une de remarquable, appelée représentation de Reitwiesner. Cette représentation a comme caractéristiques particulièrement intéressantes, d'être
25 unique, creuse, et de présenter le plus petit poids de Hamming.

Si on note $r = (r_{m-1}, \dots, r_0)$ la représentation binaire du nombre r , la représentation binaire signée de

Reitwiesner, que l'on note $r=(r'_m, r'_{m-1}, \dots, r'_2, r'_1, r'_0)$, comprend 1 chiffre supplémentaire.

Dire que cette représentation est creuse signifie qu'elle vérifie que la multiplication de deux chiffres

5 adjacents est nulle, ce qui s'écrit :

$$r'_i \cdot r'_{i-1} = 0, \forall i \in \{0, \dots, m\}.$$

Le poids de Hamming exprime quant à lui le nombre de chiffres non nuls dans la représentation considérée.

Avoir un poids de Hamming minimum signifie donc un
10 nombre optimum, maximum, de chiffres à "0". Or, moins il y a de chiffres "1" ou "-1" dans la représentation, moins il y a de calculs à réellement effectuer, d'où une amélioration en termes de rapidité de calculs.

C'est donc une propriété particulièrement intéressante
15 de la représentation de Reitwiesner en matière de calcul arithmétique rapide.

L'algorithme qui permet d'obtenir cette représentation binaire signée remarquable à partir de la représentation binaire $(r_{m-1}, \dots, r_0)$ d'un nombre r , peut

20 être décrit comme suit :

-On utilise une variable de calcul c , en binaire.

-On initialise à "0" le bit c_0 de rang 0 de cette variable c .

-On ajoute à la représentation binaire $(r_{m-1}, \dots, r_0)$
25 du nombre r , un bit de rang $m+1$ et un bit de rang m , tous deux initialisés à "0" : $r_{m+1}=0$; $r_m=0$

-puis pour i égal 0 jusqu'à i égal m , on calcule :

$$c_{i+1} = \lfloor (c_i + r_i + r_{i+1}) / 2 \rfloor$$

$$r'_i = c_i + r_i - 2c_{i+1}$$

30 où $\lfloor t \rfloor$ signifie la valeur par défaut, c'est à dire, la valeur entière immédiatement inférieure au nombre t . En pratique, comme la somme arithmétique des bits $c_i + r_i + r_{i+1}$ se code sur deux bits au maximum en binaire, le bit c_{i+1}

est obtenu en décalant d'une position vers la droite de cette somme.

On peut voir que dans cet algorithme de Reitwiesner, les bits r_i de la représentation binaire du nombre r sont traités de droite à gauche, pour obtenir chacun des chiffres r'_i de la représentation binaire signée, depuis i égal 0 jusqu'à i égal m .

Différents travaux ont pu montrer qu'il n'existait pas d'algorithme dans lequel les bits seraient traités de gauche à droite et par lequel on obtiendrait la représentation binaire signée de Reitwiesner.

Or, dans nombre d'applications, les algorithmes de calcul arithmétique rapide appliqués à un opérande r , traitent les chiffres r'_i de la représentation binaire signée de cet opérande de gauche à droite. Ces algorithmes se révèlent en effet, dans certains cas, plus efficaces tant sur le plan du temps de calcul que sur le plan matériel (moins de registres nécessaires).

Si on prend l'exemple du calcul de l'exponentiation, un algorithme d'exponentiation modulaire d'un nombre α par un exposant r basé sur le traitement droite vers gauche (ARL) peut être comme suit :

-On applique en entrées de l'algorithme un nombre α , et la représentation binaire signée ($r'_m, r'_{m-1}, \dots, r'_2, r'_1, r'_0$).

-On initialise à 1 l'accumulateur qui contient le résultat M , soit $M=1$.

-On initialise un registre temporaire qui contient une variable de calcul S avec le nombre α auquel est appliqué l'exponentiation.

-On réalise pour i égal 0 jusqu'à i égal $m-1$ la boucle suivante :

-si $r'_i=1$, on multiplie le contenu M de l'accumulateur par la variable S , et on charge le

résultat dans l'accumulateur, ce qui s'écrit :
 $M=M.S$

- 5 -si $r'_i=-1$, on multiplie le contenu M de l'accumulateur par l'inverse de la variable S , et on charge le résultat dans l'accumulateur, ce qui s'écrit $M=M.S^{-1}$.

Un algorithme d'exponentiation modulaire du nombre α par l'exposant r basé sur un traitement gauche vers droite (ALR) de l'exposant est le suivant :

- 10 -On applique en entrées le nombre α , et la représentation binaire signée $(r'_m, r'_{m-1}, \dots, r'_2, r'_1, r'_0)$ de l'exposant r .

-On initialise à 1 le contenu M de l'accumulateur : $M=1$.

- 15 -Pour i égal m à i égal 0 :

-On effectue la multiplication du contenu de l'accumulateur par lui-même et on charge l'accumulateur avec le résultat, ce qui s'écrit : $M=M^2$ puis,

- 20 -si $r'_i=1$, on multiplie le contenu de l'accumulateur par α et on charge le résultat dans l'accumulateur : $M=M.\alpha$ ou
 -si $r'_i=-1$, on multiplie le contenu de l'accumulateur par l'inverse de α et on charge
 25 le résultat dans l'accumulateur : $M=M.\alpha^{-1}$.

- L'algorithme droite vers gauche ARL nécessite donc au moins un registre supplémentaire, pour la variable S , que l'algorithme gauche vers droite ALR. En outre, S étant variable, il faut calculer S^{-1} à chaque fois que
 30 la condition associée est réalisée dans l'algorithme droite vers gauche. Dans l'algorithme gauche vers droite, α étant déterminé (fixe), son inverse α^{-1} peut n'être calculée qu'une seule fois et mémorisé.

Dans l'invention, on a cherché à cumuler les avantages du traitement gauche vers droite à la fois dans le procédé de conversion d'un opérande et dans le traitement arithmétique utilisant cet opérande.

- 5 Ainsi, dans l'invention, on a recherché une représentation binaire signée équivalente de la représentation de Reitwiesner, c'est à dire une représentation qui présente le même poids de Hamming pour une même longueur m (nombre de chiffres dans la
- 10 représentation).

L'invention a donc pour objet un procédé de conversion d'une représentation binaire d'un nombre dans une représentation binaire signée comprenant un nombre optimum, c'est à dire maximum, de zéro (poids de

15 Hamming minimal), et dans lequel les bits sont traités de la gauche vers la droite.

L'invention a aussi pour objet un dispositif de conversion permettant l'obtention d'une représentation binaire signée correspondante.

- 20 Telle que caractérisée, l'invention concerne donc un procédé d'obtention d'une représentation binaire signée sur $m+1$ chiffres $(r'_m, \dots, r'_0)$ d'un nombre à partir de sa représentation binaire sur m bits $(r_{m-1}, \dots, r_0)$. Selon l'invention, ce procédé consiste :

- 25 -A ajouter dans la représentation binaire du nombre r à convertir un bit de rang m , un bit de rang -1 et un bit de rang -2 et à initialiser lesdits bits à la valeur 0,

- A prendre au moins une variable de calcul et à
- 30 initialiser à 0 un bit de rang m de cette variable, et

-A effectuer $m+1$ itérations d'une boucle de conversion, pour i égal m jusqu'à i égal 0, la boucle de conversion comprenant les opérations suivantes :

- 5 -Appliquer en entrées de premiers moyens de calcul le bit de rang i de la variable de calcul b , et les bits de rangs $i-1$ et $i-2$ de la représentation binaire du nombre r à convertir, lesdits premiers moyens fournissant en sortie le bit de rang $i-1$ de la variable de calcul, en effectuant une opération arithmétique équivalente à la valeur par défaut de la somme arithmétique divisée par deux des bits appliqués en entrées,
- 10 -Appliquer en entrées de deuxièmes moyens de calcul, les bits de rang i et $i-1$ de la variable de calcul b , et le bit de rang i de la représentation binaire du nombre à convertir r , lesdits moyens de calcul effectuant une opération
- 15 arithmétique équivalente à la multiplication par moins deux du bit de rang i de la variable de calcul b , et la somme arithmétique du résultat précédent avec les deux autres bits appliqués en entrée, pour fournir en sortie le chiffre de rang
- 20 i de la représentation binaire signée.
- D'autres caractéristiques et avantages de l'invention sont détaillés dans la description suivante, faite à titre indicatif et non limitatif de l'invention et en référence aux dessins annexés dans lesquels :
- 25 -la figure 1 représente un circuit intégré comprenant un tel dispositif de conversion et un dispositif de calcul arithmétique d'une exponentiation modulaire.
- la figure 2 représente un schéma-bloc d'un dispositif de conversion selon l'invention,
- 30 -la figure 3 représente une table de correspondance utilisable pour effectuer une conversion selon le procédé de l'invention;
- la figure 4 représente un exemple de réalisation d'un dispositif de conversion utilisant une telle table; et

-la figure 5 représente un autre exemple de réalisation d'un dispositif de conversion selon l'invention, utilisant des opérateurs booléens.

Sur la figure 1, on a représenté un circuit intégré du type à microprocesseur μP et comprenant une mémoire MEM et un générateur de nombre aléatoire GEN.

Ce circuit intégré comprend en outre un dispositif 1 de calcul d'exponentiation d'un nombre α par un exposant r . Dans l'exemple, le nombre α est un nombre fixe contenu en mémoire MEM et l'exposant r est un nombre fourni en binaire $(r_{m-1}, \dots, r_0)$ par le générateur de nombre aléatoire.

Un dispositif de conversion 2 en binaire signé est prévu dans le circuit intégré, pour convertir le nombre r appliqué comme exposant dans le dispositif 1 de calcul.

On obtient en sortie de ce dispositif de conversion, la représentation binaire signée $(r'_m, \dots, r'_0)$ selon l'invention, appliquée en entrée du dispositif 1 de calcul. Cette représentation binaire signée est telle qu'elle présente un poids de Hamming optimum, à l'instar de la représentation dite de Reitwiesner.

Le dispositif 1 de calcul représenté sur la figure 1 permet d'appliquer l'algorithme d'exponentiation modulaire gauche vers droite ALR précédemment décrit. Dans cet exemple, le dispositif de calcul comprend un accumulateur 10 pour les résultats intermédiaires et le résultat final de l'exponentiation et trois registres : un registre 11 pour l'exposant, un registre 12 pour le nombre α , et un registre 13 pour son inverse α^{-1} . En réalité, le registre 11 pour l'exposant est un double registre : car chaque chiffre "0", "1" "-1" en binaire signé est habituellement codée sur au moins deux bits.

Dans un exemple, ce code comprend deux bits : un bit pour le signe et un bit pour la valeur absolue du chiffre. Dans cet exemple, on a alors : "0" = (X,0) ; "1"=(0,1) ; "-1"=(1,1) où X signifie que le bit peut
5 valoir indifféremment 0 ou 1. Dans l'exemple de la figure 1, X vaut 0.

Le dispositif de calcul 1 comprend encore deux multiplexeurs 20 et 21, un multiplieur 22. Un circuit de commande 23 et un commutateur 24 fournissent les
10 signaux de commande des multiplexeurs.

Le circuit de commande 23 reçoit le bit de valeur absolue r'_{il} du chiffre r'_i de rang i de l'exposant r (et un signal de séquençement C_k). Si ce bit vaut "1", il active (com1) le commutateur 24, pour appliquer
15 le bit r'_{ih} de signe du chiffre r'_i de l'exposant en entrée de commande du multiplexeur 20. Selon que le bit r'_{ih} vaut "0" ou "1", le multiplexeur 20 transmet alors α ou α^{-1} en entrée du deuxième multiplexeur 21.

Ce deuxième multiplexeur 21 reçoit sur une autre
20 entrée, le contenu M de l'accumulateur 10. Il est commandé par une sortie Com2 du circuit de commande 23, pour appliquer sur une entrée du multiplieur 22, soit la sortie du premier multiplexeur 20, c'est à dire α ou α^{-1} , soit le contenu M de l'accumulateur.

25 Le résultat en sortie du multiplieur est chargé dans l'accumulateur.

Le circuit de commande 23 est tel que dans chaque itération de la boucle de calcul d'un bit de résultat de l'exponentiation modulaire, il commande d'abord la
30 commutation de l'entrée M en sortie du multiplexeur 21, pour calculer M.M dans le multiplieur, puis, si la valeur absolue r'_{il} du chiffre r'_i de l'exposant vaut 1, pour commander la commutation de la sortie du premier multiplexeur 20 (α ou α^{-1}) en sortie du multiplexeur 21.

Le dispositif de calcul permet ainsi le calcul de l'exponentiation de α par r , par un traitement gauche vers droite.

L'association d'un dispositif de conversion gauche droite et d'un dispositif de traitement arithmétique traitant de gauche à droite le résultat de la conversion, permet d'effectuer tous ces traitements en temps réel. En effet, dès que le chiffre r_i est calculé et disponible en sortie du dispositif 2 de conversion, il est immédiatement utilisé dans le dispositif 1 de traitement arithmétique.

Avec un dispositif de conversion traitant les bits de droite à gauche (Reitwiesner), on ne peut utiliser la représentation binaire signée qu'une fois qu'elle est entièrement calculée, puisque le dernier chiffre converti r_m est le premier utilisé dans le dispositif de traitement arithmétique traitant les chiffres de gauche à droite.

L'homogénéité des traitements de gauche à droite des bits dans le dispositif de conversion selon l'invention et des chiffres dans le dispositif de calcul, permet donc de simplifier la réalisation matérielle de ces dispositifs, en économisant notamment des registres et d'améliorer les performances en termes de vitesse de calcul.

Le dispositif de conversion ne s'applique pas qu'à un dispositif de calcul d'une exponentiation. Il s'applique aussi bien au calcul de la multiplication d'un nombre α par un nombre r en représentation binaire signée, ou à la multiplication scalaire d'un point par le nombre r en représentation binaire signée. Dans ce dernier cas, une application particulièrement intéressante de l'invention concerne le cas où le point est pris sur une courbe elliptique.

La figure 2 représente un schéma bloc d'un dispositif de conversion 2 illustrant un procédé de conversion selon l'invention.

Le procédé d'obtention d'une représentation binaire signée selon l'invention, utilisant un traitement gauche vers droite est le suivant :

-On ajoute dans un registre R1 contenant la représentation binaire du nombre r, un bit r_m de rang m, un bit r_{-1} de rang -1 et un bit r_{-2} de rang -2 et on initialise ces trois bits à la valeur 0.

-On prend une variable de calcul b dans un registre R2, et on initialise à 0 le bit b_m de rang m de cette variable.

-On effectue m+1 itérations d'une boucle de conversion, depuis i égal m jusqu'à i égal 0, la boucle de conversion comprenant les opérations suivantes :

-Application en entrées e1, e2, e3 de premiers moyens de calcul 30 du bit b_i et des bits r_{i-1} , r_{i-2} , pour calculer la somme arithmétique divisée par deux de ces bits et fournir comme bit de sortie b_{i-1} , le bit correspondant à la valeur par défaut du résultat, ce qui s'écrit :

$$b_{i-1} = \lfloor (b_i + r_{i-1} + r_{i-2}) / 2 \rfloor.$$

-Application en entrées e1', e2', e3' de deuxièmes moyens de calcul 31, des bits b_i , b_{i-1} de la variable de calcul b, et du bit r_i de rang i de la représentation binaire du nombre r, pour calculer la somme arithmétique de moins deux fois le bit b_i avec les deux autres bits appliqués en entrées, et fournir en sortie le chiffre r'_i de la représentation binaire signée, ce qui s'écrit :

$$r'_i = -2.b_i + b_{i-1} + r_i.$$

Les premiers et deuxièmes moyens de calcul peuvent effectuer l'opération arithmétique indiquée, ou une opération arithmétique équivalente.

Ces calculs sont effectués au moyen d'une variable b .

- 5 Mais il est possible de mettre en œuvre l'invention en utilisant plusieurs variables.

Comme représenté sur la figure 2, un dispositif de conversion permettant de mettre en œuvre ce procédé comprend le registre R1 pour contenir la représentation
10 binaire signée du nombre r , y compris les bits r_m , r_{-1} , r_{-2} ajoutés à cette représentation. Ce registre est du type à décalage vers la gauche, correspondant au sens de traitement des bits de gauche ($i=m$), à droite ($i=0$). Les sorties de rang m , $m-1$ et $m-2$ du registre R1 sont
15 ainsi respectivement appliquées sur l'entrée e_3 des deuxièmes moyens de calcul 31, et sur les entrées e_2 et e_3 des premiers moyens de calcul 30. A la fin de chaque itération, le registre R1 est décalé d'une position à gauche, en sorte que les sorties de rang m , $m-1$ et $m-2$
20 fournissent à chaque itération les bits de rang i , $i-1$ et $i-2$ correspondants.

Le dispositif de conversion comprend un autre registre R2 pour la variable de calcul b , également du type à décalage vers la gauche. Dans l'exemple, ce registre
25 comprend pour chaque itération, le bit b_i et le bit b_{i-1} . Comme le bit b_{i-1} est établi à partir du bit b_i , les premiers moyens de calcul sont activés en premier, pour charger dans le registre R2 la nouvelle valeur b_{i-1} , à utiliser dans l'itération en cours, dans les deuxièmes
30 moyens de calcul.

A la fin de l'itération, le registre R2 est décalé d'une position vers la gauche, pour l'itération suivante.

- Dans l'invention, on a pu établir une table de correspondance correspondant au procédé de conversion qui vient d'être exposé. Cette table de correspondance représentée sur la figure 3 permet de simplifier la
- 5 conversion selon l'invention. En effet, elle donne directement le résultat en sortie, en fonction des entrées. En outre le registre R2 de la figure 2 peut alors être réalisé sous forme d'un simple latch, pour contenir, pour l'itération i en cours, le bit b_i .
- 10 Pour obtenir cette table, on a déterminé la valeur du bit b_{i-1} pour chaque triplet possible $(e1, e2, e3) = (b_i, r_{i-1}, r_{i-2})$, et la valeur du chiffre r'_i pour chaque triplet possible en entrées $(e1', e2', e3') = (b_i, r_i, b_{i-1})$, à partir de l'algorithme de conversion selon
- 15 l'invention exposé précédemment en relation avec la figure 2.
- On a ainsi pu montrer que l'on pouvait déterminer la valeur du bit b_{i-1} et du chiffre r'_i , à partir du quadruplet $(b_i, r_i, r_{i-1}, r_{i-2})$, pour chacune des
- 20 valeurs possibles de ce dernier.
- Avec cette table de conversion, seul le bit b_i de la variable b est utilisé comme entrée pour l'itération en cours. Le bit b_{i-1} fourni en sortie sert pour l'itération suivante. Ainsi, avec une telle table de correspondance
- 25 le registre R2 peut se réduire à un simple latch, dont la nouvelle valeur est chargée à la fin de chaque itération, par le bit b_{i-1} fourni en sortie par la table. Cette nouvelle valeur devient celle du bit b_i dans l'itération suivante.
- 30 Un exemple de réalisation pratique d'un dispositif de conversion selon l'invention utilisant une telle table de conversion est représenté sur la figure 4. Il comprend une mémoire adressable par son contenu, notée CAM. La table de correspondance de la figure 3 est

programmée dans cette mémoire en sorte qu'en appliquant sur les bits d'adresse A0, A1, A2, A3 de cette mémoire, les bits b_i , r_i , r_{i-1} et r_{i-2} , elle fournit en sortie le bit b_{i-1} et le chiffre r'_i correspondants.

- 5 A la fin de chaque itération, le bit b_{i-1} est chargé dans le registre R2. Ce bit devient le bit b_i pour l'itération suivante de la boucle.

Le chiffre r'_i est chargé dans un registre R3, qui comprend en pratique un premier registre R3H, pour
10 contenir le bit de signe r'_{iH} de chaque chiffre de la représentation binaire signée et un deuxième registre R3L, pour contenir sa valeur absolue r'_{iL} .

Un autre exemple de réalisation d'un dispositif de conversion selon l'invention est représenté sur la
15 figure 5. Dans cet exemple, le dispositif de conversion est basé sur des opérateurs booléens. Là encore, le registre R2 peut être un simple latch rechargé par le bit b_{i-1} déterminé dans l'itération courante.

La définition des opérateurs booléens est issue d'un
20 travail de manipulation logique et de minimalisation de l'algorithme correspondant au procédé de conversion et à la table de correspondance selon l'invention.

Dans l'invention, on a pu définir ces opérateurs à partir des bits b_i , r_i , r_{i-1} et r_{i-2} , pour obtenir
25 directement en sortie le bit b_{i-1} et le chiffre r'_i .

En utilisant le codage sur deux bits représentant le signe et la valeur absolue, les opérations booléennes suivantes ont pu être définies :

$$\begin{aligned} b_{i-1} &= /b_i . r_{i-1} . r_{i-2} + b_i . r_{i-1} + b_i . r_{i-2} ; \\ 30 \quad r'_{iH} &= b_i ; \\ r'_{iL} &= /b_i . r_{i-1} . r_{i-2} + /b_i . r_i + .b_i . /r_i + b_i . /r_{i-1} . /r_{i-2} . \end{aligned}$$

On notera que lorsque le chiffre r'_i est nul, soit si $r'_{iL}=0$, le bit de signe r'_{iH} peut être pris indifféremment égal à 0 ou 1. Faire $r'_{iH}=b_i$, couvre ainsi tous les cas,

si on accepte une valeur quelconque 0 ou 1 pour le bit de signe dans le cas où le chiffre vaut 0. Mais on peut prévoir que ce bit de signe soit fixé, par exemple fixé à 0, lorsque le chiffre vaut 0, comme dans l'exemple

5 donné à la figure 1.

Un exemple de réalisation d'un dispositif de conversion comprenant des opérateurs booléens correspondante est représenté sur la figure 5. Dans cet exemple, le dispositif de conversion comprend, en plus des

10 registres R1 et R2 :

-Une première porte P1 de type ET, comprenant une entrée inverseuse pour recevoir le bit r_i , et une entrée non inverseuse pour recevoir le bit b_i ;

15 -Une deuxième porte P2 de type ET avec les entrées inverses de la première;

-Une troisième porte P3 de type ET, comprenant une entrée inverseuse pour recevoir le bit r_{i-1} , une autre entrée inverseuse pour recevoir le bit r_{i-2} et une entrée non inverseuse pour recevoir le bit b_i ;

20 -Une quatrième porte P4 de type ET, recevant les entrées inverses de la troisième porte;

-Une cinquième porte P5 de type ET comprenant une entrée non inverseuse pour recevoir le bit r_{i-1} , et une autre entrée non inverseuse pour recevoir le bit b_i ;

25 -Une sixième porte P6 de type ET, comprenant une entrée non inverseuse pour recevoir le bit r_{i-2} , et une autre entrée non inverseuse pour recevoir le bit b_i ;

-Une septième porte P7 de type OU, recevant en entrées, les sorties des première, deuxième, troisième et

30 quatrième portes, et fournissant en sortie la valeur absolue r'_{iL} (0 ou 1) du chiffre r'_i de rang i de la représentation binaire signée, le bit de signe r'_{iH} étant fourni par le bit b_i ; et

-Une huitième porte P8 de type OU, recevant en entrées, la sortie des quatrième, cinquième et sixième portes, pour fournir en sortie le bit b_{i-1} .

Les registres R1, R2 et R3 qui contiennent
5 respectivement la représentation binaire de r , la variable de calcul et le résultat de la conversion en binaire signé sont commandés comme dans la figure 4.

L'invention qui vient d'être décrite est utilisable dans beaucoup d'applications, et notamment celles
10 utilisant des procédés cryptographiques. Elle est applicable pour fournir la représentation binaire signée d'un opérande utilisé dans un traitement arithmétique quelconque, traitant de préférence les bits de gauche à droite pour en retirer le plus grand
15 bénéfice en terme de simplicité de réalisation et de vitesse de traitement. On notera que l'on a donné un exemple d'application de l'invention à l'exponentiation d'un nombre fixe par un nombre r variable (aléatoire) qui utilise les techniques de calcul par carrés et
20 multiplications (*square and multiply*). L'invention s'applique aussi bien à l'exponentiation d'un nombre aléatoire par un nombre r fixe, qui utilise les techniques des chaînes additives. Plus généralement, l'invention s'applique à des dispositifs de calcul
25 arithmétique pour la multiplication d'un nombre α par le nombre r en représentation binaire signée, ou à la multiplication scalaire d'un point par le nombre r . Dans ce dernier cas, une application particulièrement intéressante de l'invention concerne le cas où le point
30 est pris sur une courbe elliptique. La liste des applications possibles n'est pas exhaustive.

Dans toutes ces applications le nombre d'opérations est réduit, comme avec la représentation de Reitwiesner, du fait du poids de Hamming minimum, avec les avantages

supplémentaires lié au traitement gauche vers droite
(simplicité matérielle, réduction du nombre de
registre, traitement en temps réel).

REVENDICATIONS

1. Procédé de conversion dans une représentation binaire signée sur $m+1$ chiffres ($r'_m, \dots, r'_0$) d'un nombre (r) à partir de sa représentation binaire sur m bits ($r_{m-1}, \dots, r_0$), caractérisé en ce qu'il
- 5 consiste :
- a) à ajouter dans la représentation binaire du nombre à convertir (r) un bit (r_m) de rang m , un bit (r_{-1}) de rang -1 et un bit (r_{-2}) de rang -2 et à initialiser lesdits bits à la valeur 0,
- 10 b) à utiliser au moins une variable de calcul (b), et
- c) à initialiser à la valeur 0 un bit de rang m (b_m) de ladite variable,
- d) à effectuer pour i égal m jusqu'à i égal 0 une
- 15 boucle de conversion, comprenant les opérations suivantes :
- (1) appliquer en entrées ($e1, e2, e3$) de premiers moyens de calcul (30) le bit de rang i (b_i) de la variable de calcul (b), et les
- 20 bits de rangs $i-1$ et $i-2$ (r_{i-1}, r_{i-2}) de la représentation binaire du nombre à convertir, lesdits premiers moyens fournissant en sortie le bit de rang $i-1$ (b_{i-1}) de la variable de calcul (b), en effectuant une opération arithmétique équivalente à la valeur par
- 25 défaut de la somme arithmétique divisée par deux desdits bits appliqués en entrées ($e1, e2, e3$),
- (2) appliquer en entrées ($e1', e2', e3'$) de deuxièmes moyens de calcul (31), les bits (b_i, b_{i-1}) de rang i et $i-1$ de la variable de calcul
- 30

(b), et le bit (r_i) de rang i de la représentation binaire du nombre à convertir (r), lesdits moyens de calcul effectuant une opération arithmétique équivalente à la multiplication par moins deux du bit de rang i de la variable de calcul (b), et la somme arithmétique du résultat précédent avec les deux autres bits appliqués en entrée, pour fournir en sortie le chiffre de rang i (r'_i) de la représentation binaire signée.

2. Procédé selon la revendication 1, caractérisé en ce qu'il consiste à appliquer la représentation binaire signée en entrées d'un dispositif (1) de calcul arithmétique dont ledit nombre (r) est un opérande, ledit dispositif de calcul traitant les chiffres dudit opérande du rang le plus élevé au rang le moins élevé.

3. Procédé de conversion selon l'une quelconque des revendications précédentes, caractérisé en ce qu'il comprend l'utilisation d'une table de conversion à quatre entrées comme premiers et deuxièmes moyens de calcul, lesdites entrées correspondant au bit (b_i) de rang i de la variable de calcul (b), et aux bits (r_i , r_{i-1} , r_{i-2}) de rang i, i-1 et i-2 de la représentation binaire du nombre à convertir (r), pour fournir directement en sortie la valeur du bit (b_{i-1}) de rang i-1 de la variable de calcul (b) et la valeur du chiffre (r'_i) de rang i de la représentation binaire signée.

4. Procédé de conversion selon l'une des revendications 1 à 3, caractérisé en ce que chaque chiffre (r'_i) de la représentation binaire signée est fourni en sortie sous la forme d'un code à au moins deux bits.
- 5
5. Procédé de conversion selon la revendication 4, caractérisé en ce que chaque chiffre est codé sur deux bits, un premier bit correspondant à la valeur absolue (r'_{iL}) et un deuxième bit correspondant au bit de signe (r'_{iH}), les premiers bits des chiffres de la
- 10 représentation binaire signée étant mémorisés dans un premier registre de sortie (R3L) et les deuxième bits de ces chiffres étant mémorisés dans un deuxième registre de sortie (R3H), lesdits registres de sortie
- 15 étant du type à décalage vers la gauche.
6. Dispositif de conversion en binaire signé pour la mise en œuvre d'un procédé selon l'une des revendications 3 à 5, caractérisé en ce qu'il
- 20 comprend une mémoire adressable par son contenu (CAM), le bit (b_i) de rang i de la variable de calcul (b), et les bits (r_i , r_{i-1} , r_{i-2}) de rang i , $i-1$ et $i-2$ de la représentation binaire du nombre à convertir (r) étant appliqués à la dite mémoire comme bits
- 25 d'adresse (A0, A1, A2, A3).
7. Dispositif de conversion mettant en œuvre le procédé de conversion selon l'une quelconque des revendications 1, 2, 4 ou 5, caractérisé en ce que
- 30 les premiers et deuxième moyens de calcul comprennent des opérateurs booléens.

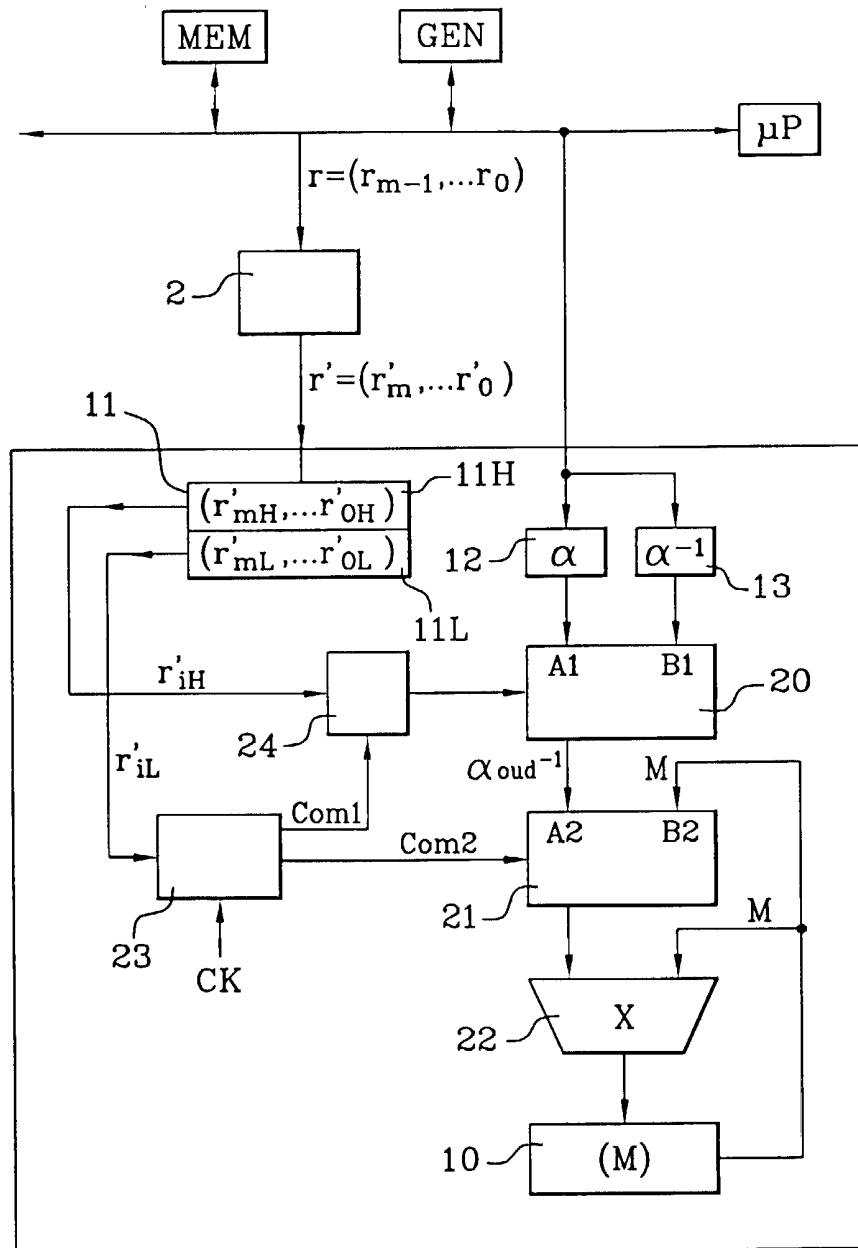
8. Dispositif de conversion selon la revendication 7, caractérisé en ce que lesdits opérateurs booléens reçoivent en entrées le bit (de rang i de la variable de calcul (b), noté b_i , et les bits de rang i , $i-1$ et $i-2$ de la représentation binaire du nombre à convertir (r), respectivement notés r_i , r_{i-1} et r_{i-2} , pour fournir en sortie le bit de rang $i-1$ de la variable de calcul (b), noté b_{i-1} et le chiffre de rang i , noté r'_i de la représentation binaire signée.
9. Dispositif de conversion selon la revendication 8 dans lequel chaque chiffre de la représentation binaire signée est fourni sous forme d'un code à deux bits selon la revendication 5, caractérisé en ce que lesdits opérateurs booléens comprennent un ensemble de portes logiques aptes à appliquer sur les bits d'entrées b_i , r_i , r_{i-1} et r_{i-2} une première opération booléenne équivalente à $/b_i.r_{i-1}.r_{i-2} + b_i.r_{i-1} + b_i.r_{i-2}$, pour fournir en sortie le bit b_{i-1} et une deuxième opération booléenne équivalente à $/b_i.r_{i-1}.r_{i-2} + /b_i.r_i + .b_i./r_i + b_i./r_{i-1}./r_{i-2}$, pour fournir en sortie le bit représentant la valeur absolue (r_{iH}) du chiffre de rang i (r'_i) de la représentation binaire signée, le bit (r_{iH}) représentant le signe dudit chiffre étant égal au bit de rang i (b_i) de la variable de calcul ou égal à 0 ou 1 si ledit chiffre est nul.
10. Dispositif de conversion selon la revendication précédente, caractérisé en ce que lesdits opérateurs booléens comprennent:
- a) Une première porte ($P1$) de type ET, comprenant une entrée inverseuse pour recevoir le bit (r_i) de rang

- i de la représentation binaire du nombre à convertir (r), et une entrée non inverseuse pour recevoir le bit (b_i) de rang i de la variable de calcul;
- 5 b) Une deuxième porte (P2) de type ET avec les entrées inverses de la première;
- c) Une troisième porte (P3) de type ET, comprenant une entrée inverseuse pour recevoir le bit (r_{i-1}) de rang i-1 de la représentation binaire du nombre à convertir (r), une autre entrée inverseuse pour recevoir le bit (r_{i-2}) de rang i-2 de la représentation binaire du nombre à convertir (r), et une entrée non inverseuse pour recevoir le bit (b_i) de la variable de calcul (b);
- 10 d) Une quatrième porte (P4) de type ET, recevant les entrées inverses de la troisième porte;
- e) Une cinquième porte (P5) de type ET comprenant une entrée non inverseuse pour recevoir le bit (r_{i-1}) de rang i-1 de la représentation binaire du nombre à convertir (r), et une autre entrée non inverseuse pour recevoir le bit (b_i) de rang i de la variable de calcul (b);
- 15 f) Une sixième porte de type ET, comprenant une entrée non inverseuse pour recevoir le bit (r_{i-2}) de rang i-2 de la représentation binaire du nombre à convertir (r), et une autre entrée non inverseuse pour recevoir le bit (b_i) de rang i de la variable de calcul (b);
- 20 g) Une septième porte (P7) de type OU, recevant en entrées, les sorties des première, deuxième, troisième et quatrième portes, et fournissant en sortie le bit (r'_{il}) représentant la valeur absolue
- 25 h) Une huitième porte (P8) de type OU, recevant en entrées, les sorties des première, deuxième, troisième et quatrième portes, et fournissant en sortie le bit (r'_{il}) représentant la valeur absolue
- 30 i) Une neuvième porte (P9) de type OU, recevant en entrées, les sorties des première, deuxième, troisième et quatrième portes, et fournissant en sortie le bit (r'_{il}) représentant la valeur absolue

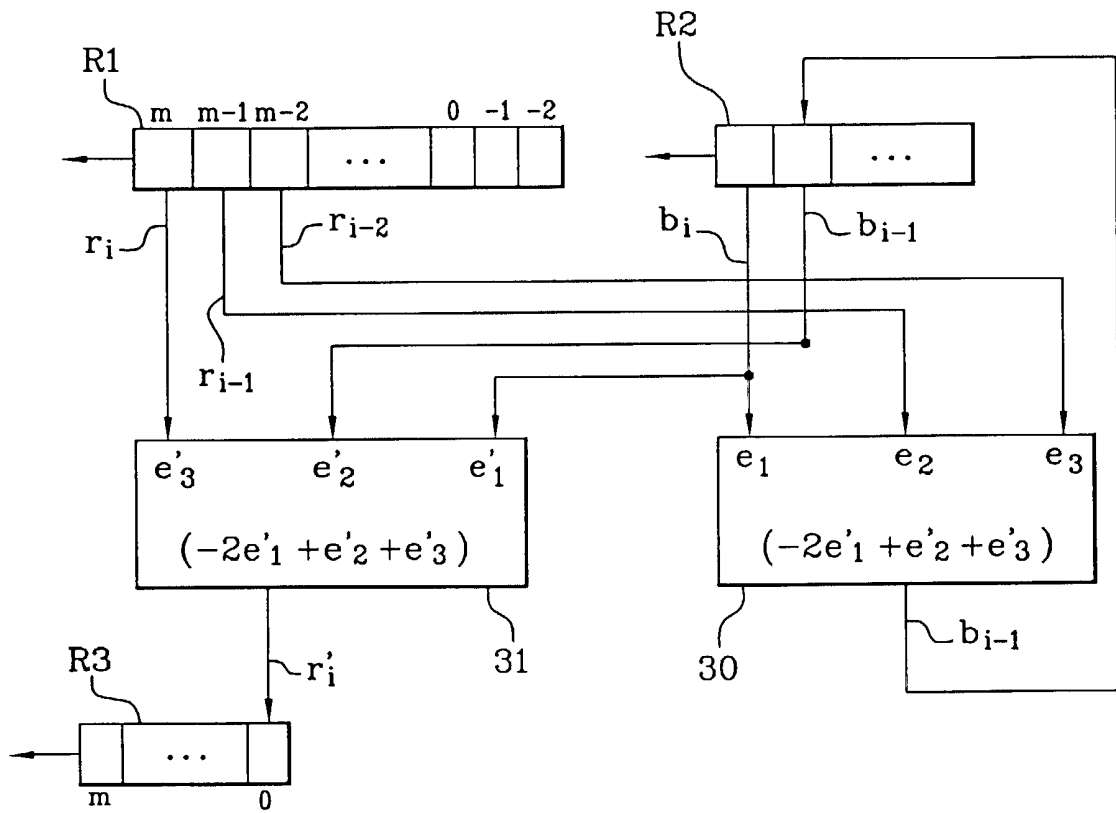
- (0 ou 1) du chiffre (r'_i) de rang i de la représentation binaire signée, le bit (r'_{1H}) représentant le signe étant fourni par le bit (b_i) de rang i de la variable de calcul (b); et
- 5 h) Une huitième porte (P8) de type OU, recevant en entrées, la sortie des quatrième, cinquième et sixième portes, pour fournir en sortie le bit (b_{i-1}) de rang $i-1$ de la variable de calcul (b).
- 10 11. Dispositif de conversion selon l'une quelconque des revendications 6 à 10, caractérisé en ce qu'il comprend un premier registre d'entrée (R1) à décalage vers la gauche pour contenir la représentation binaire du nombre à convertir (r) et un deuxième
- 15 registre (R2) à décalage vers la gauche, pour contenir la variable de calcul (b).
- 20 12. Dispositif selon la revendication 11, caractérisé en ce que ledit deuxième registre (R2) est un latch, qui contient le bit de rang i correspondant à l'itération courante, et en ce que le bit de rang $i-1$ délivré en sortie dans cette itération est chargée dans le latch pour fournir le bit de rang i dans l'itération suivante.
- 25 13. Circuit intégré comprenant un dispositif de conversion selon l'une quelconque des revendications 6 à 12, caractérisé en ce qu'il comprend en outre un dispositif (1) de calcul arithmétique selon un
- 30 traitement du chiffre de rang le plus élevé au chiffre de rang le plus faible de la représentation binaire signée.

14. Circuit intégré selon la revendication 13,
caractérisé en ce que ledit dispositif (1) de calcul
arithmétique est un dispositif d'exponentiation d'un
5 nombre (α) par la représentation binaire signée du
nombre (r) appliqué au dispositif de conversion.
15. Circuit intégré selon la revendication 13,
caractérisé en ce que ledit dispositif (1) de calcul
10 arithmétique est un dispositif de multiplication d'un
nombre (α) par la représentation binaire signée du
nombre (r) appliqué au dispositif de conversion.
16. Circuit intégré selon la revendication 13,
15 caractérisé en ce que ledit dispositif (1) de calcul
arithmétique est un dispositif de multiplication
scalaire d'un point (α) par la représentation binaire
signée du nombre (r) appliqué au dispositif de
conversion.
20
17. Circuit intégré selon la revendication 16,
caractérisé en ce que ledit point est pris sur une
courbe elliptique.

1/4

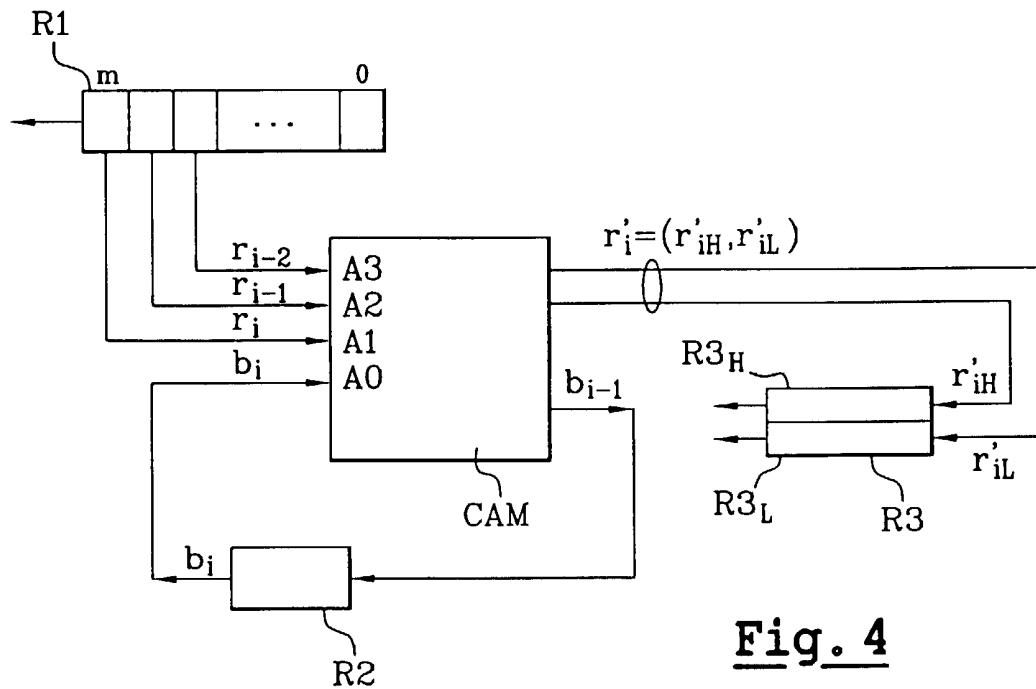
**Fig. 1**

2/4

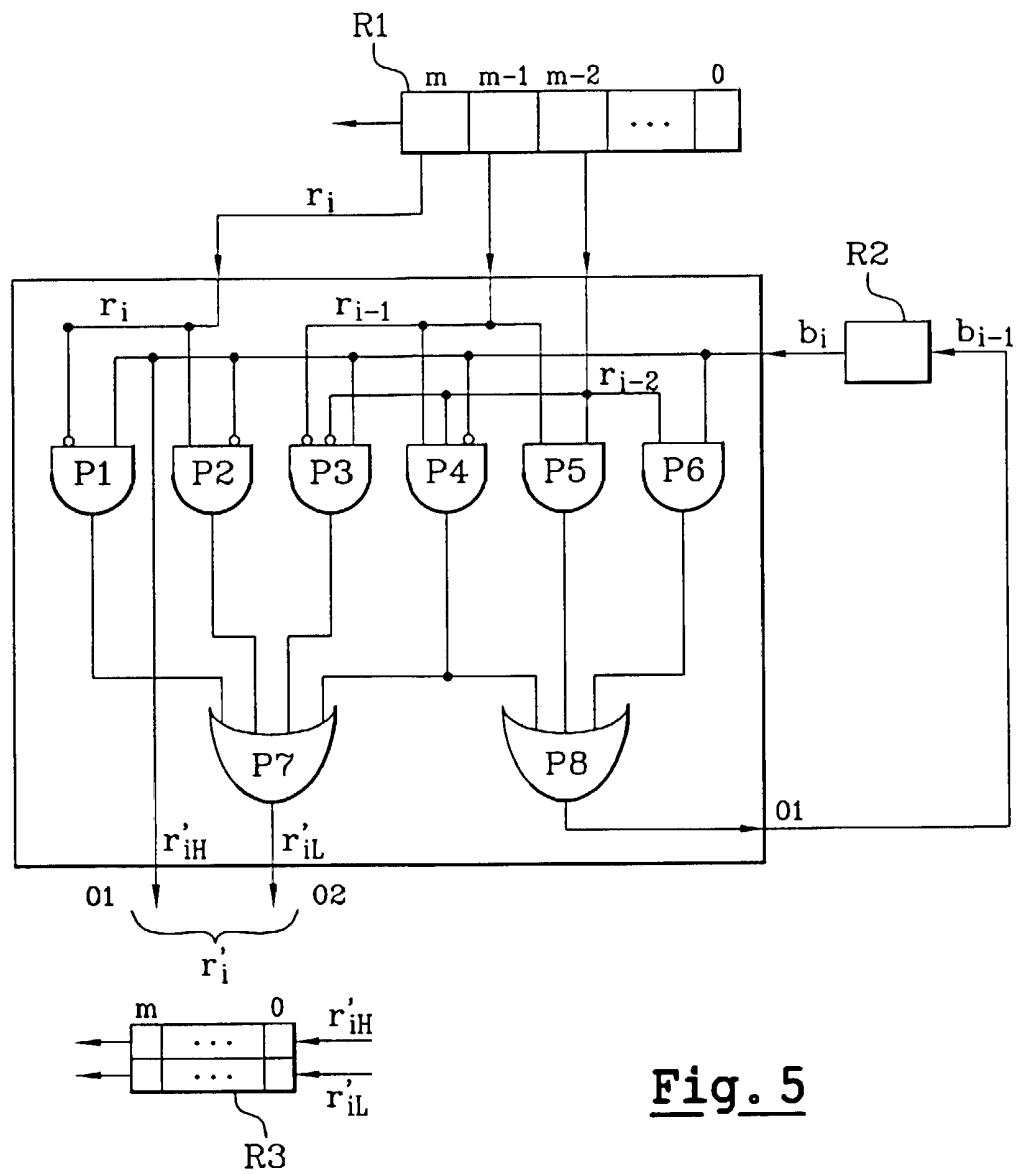
Fig. 2

3/4

b_i	r_i	r_{i-1}	r_{i-2}	b_{i-1}	$r'_i = (r'_{iH}, r'_{iL})$
0	0	0	X	0	$0 = (X, 0)$
0	0	1	0	0	$0 = (X, 0)$
0	0	1	1	1	$1 = (0, 1)$
0	1	0	X	0	$1 = (0, 1)$
1	0	1	X	1	$-1 = (1, 1)$
1	1	0	0	0	$-1 = (1, 1)$
1	1	0	1	1	$0 = (X, 0)$
1	1	1	X	1	$0 = (X, 0)$

Fig. 3Fig. 4

4 / 4



RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

2811168

N° d'enregistrement
national

FA 588922
FR 0008547

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		
X	GOTO MUNEHIRO: "NONBINARY CYCLIC AN CODES" ELECTRON COMMUN JAP MAY 1973, vol. 56, no. 5, mai 1973 (1973-05), pages 31-37, XP000990544	1-5	H03M7/12 G06F7/48 H03K19/173
Y	* le document en entier *	6-17	
Y	DE 43 08 112 A (HERRFELD ANDREAS) 13 octobre 1994 (1994-10-13) * le document en entier *	6-17	
A	US 4 215 417 A (NISHITANI TAKAO) 29 juillet 1980 (1980-07-29) * colonne 4, ligne 19 - ligne 55 *	1-17	
A	CHIANG A C L ET AL: "ARITHMETIC NORMS AND BOUNDS OF THE ARITHMETIC AN CODES" IEEE TRANSACTIONS ON INFORMATION THEORY, US, IEEE INC. NEW YORK, vol. IT-16, no. 4, juillet 1970 (1970-07), pages 470-476, XP000760887 ISSN: 0018-9448 * le document en entier *	1-17	DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7)
A	CHAO-LIANG CHEN ET AL: "A simplified signed powers-of-two conversion for multiplierless adaptive filters" 1996 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS. CIRCUITS AND SYSTEMS CONNECTING THE WORLD, ISCAS 96 (CAT. NO.96CH35876), 1996 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS. CIRCUITS AND SYSTEMS CONNECTING THE WORLD. ISCAS 96, ATLANT, pages 364-367 vol.2, XP002165296 1996, New York, NY, USA, IEEE, USA ISBN: 0-7803-3073-0 * le document en entier *	1-17	H03M G06F

Date d'achèvement de la recherche		Examineur
12 avril 2001		Barel-Faucheux, C

CATÉGORIE DES DOCUMENTS CITÉS		T : théorie ou principe à la base de l'invention
X : particulièrement pertinent à lui seul	Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie	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.
A : arrière-plan technologique	O : divulgation non-écrite	D : cité dans la demande
P : document intercalaire		L : cité pour d'autres raisons
		& : membre de la même famille, document correspondant