

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 07.02.02.

30 Priorité : 09.02.01 US 09779965.

43 Date de mise à la disposition du public de la
demande : 16.08.02 Bulletin 02/33.

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

60 Références à d'autres documents nationaux
apparentés :

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

72 Inventeur(s) : FERRERI RICHARD A et WANG LAN-
ZHONG.

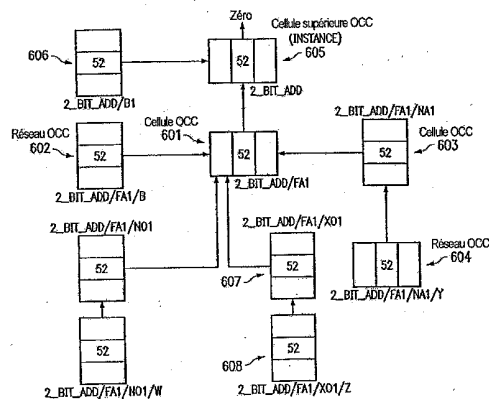
73 Titulaire(s) :

74 Mandataire(s) : REGIMBEAU.

54 PROCÉDE ET DISPOSITIF POUR SUIVRE UNE CONNECTIVITE DE RESEAU AU MOYEN D'UNE HIERARCHIE
DE CONCEPTION.

57 Le procédé de transit entre des événements de réseau de niveau supérieur de niveau inférieur (602, 604) dans un modèle hiérarchique d'un système, qui comprend des première et seconde cellules (601, 603), auxquelles sont associés les événements de réseau consiste à définir un événement de réseau actuel, récupérer un premier pointeur descriptif de l'événement de réseau actuel pour modéliser un objet, récupérer un premier pointeur de propriétaire (52) pour un événement de réseau actuel, désignant la cellule de propriétaire pour cet événement, et utiliser l'objet du modèle de réseau actuel et la cellule de propriétaire pour parcourir le modèle hiérarchique.

Application à la conception assistée par ordinateur de systèmes à très haute densité d'intégration.



1

La présente demande a trait d'une manière générale à la conception VLSI CAO, c'est-à-dire la conception assistée par ordinateur de circuits à très haute densité d'intégration, et de façon spécifique un dispositif
5 et un procédé pour exécuter une conception utilisant un modèle d'événement léger.

Des systèmes antérieurs de conception assistée par ordinateurs (CAO) représentent des conceptions sous la forme d'une connectivité hiérarchique qui fournit une
10 information de conception à des concepteurs de systèmes, avec différents niveaux d'abstraction. Un exemple d'une telle configuration schématique 100 est représenté sur la figure 1, annexée à la présente demande, qui illustre les relations entre une cellule 101, un port 102, un réseau
15 103, une instance 104 et une instance de port 105. Une telle configuration est connue comme étant un modèle de connectivité replié. Le bloc de cellule 101 décrit un dispositif et une structure du système, par exemple un additionneur complet. Une cellule contient des ensembles
20 d'instances d'autres cellules, des réseaux (qui sont des fils) et des interfaces externes de liaison avec la cellule (ports). Le bloc de réseau 103 décrit les fils qui constituent des connexions internes à l'intérieur du bloc de cellule. Le bloc du port 102 décrit une interface de
25 liaison avec la cellule et fournit les points de connexion pour les réseaux (fils) pour le transfert ("Co") de signaux dans et hors de la logique du bloc de cellule.

Comme indiqué précédemment, un bloc de cellule fournit la définition d'un dispositif ou d'une structure.
30 Lorsqu'une cellule a été définie, elle peut être instanciée (auquel cas un bloc d'instance 104 de cette cellule est créé) de sorte qu'elle peut être utilisée dans d'autres définitions de cellules. De cette manière, on peut créer une hiérarchie de conception. Le bloc d'instance décrit les
35 dispositifs ou structures utilisés pour former la

fonctionnalité d'une cellule, par exemple pour un additionneur à deux bits : deux instances de cellules d'additionneur complet sont créées. Précisément lorsqu'une instance enregistre l'instanciation (ou l'utilisation) d'un bloc de cellule, le bloc d'instance de port 105 enregistre l'instanciation des ports dans la cellule. Les instances de ports nous permettent d'enregistrer les réseaux spécifiques qui sont connectés à une instance donnée.

La nature hiérarchique de l'information mémorisée dans le modèle de connectivité replié est représentée à titre d'exemple uniquement sur les figures 2A à 2C annexées à la présente demande. La figure 2A représente le niveau le plus élevé de la hiérarchie, c'est-à-dire le bloc de cellule 200, qui est un additionneur à deux bits. Le bloc de l'additionneur à deux bits comporte 8 ports (les entrées A1, B1, A0, B0, Cin; et les sorties Cout, S1 et S0). Il contient les réseaux qui sont connectés à ces ports (A1,B1,A0,B0,Cin,Cout,S1 et S0) en plus d'un réseau interne (Co) qui n'est pas connecté à un port au niveau de la limite de la cellule, mais est néanmoins un réseau contenu dans la définition de la cellule de l'additionneur à 2 bits. Enfin, l'additionneur à 2 bits contient deux instances, le bloc FA0 202 et le bloc FA1 203, qui sont toutes deux des instances d'un additionneur complet (ou d'un additionneur à un bit).

La figure 2B représente le niveau inférieur en second de la hiérarchie du système en montrant la définition de cellule pour l'additionneur complet. On notera que les instances FA0 (bloc 202) et FA1 (bloc 203) au niveau supérieur sont décrites par une cellule au niveau inférieur suivant de la hiérarchie. Le bloc 202 de l'additionneur complet comporte 5 ports (les ports d'entrée A, B, et Cin; les ports de sortie S et Co). Il contient également 11 réseaux (réseaux pourvus de ports A, B, Cin, Co, et S; réseaux internes Co sig_1, sig_2, sig_3, sig_4 et

sig_5) et 8 instances (2 instances d'un inverseur, I1 et I2; 2 instances d'une porte NON-OU à 2 entrées, NO1, NO2; 2 instances d'une porte OU-EXCLUSIF à 2 entrées, XO1 et XO2; et 2 instances d'une porte NON-ET à 2 entrées, NA1 et NA2).

5 Enfin, la figure 2C représente l'une des cellules au niveau le plus bas de la hiérarchie, la porte NON-ET à 2 entrées. On notera qu'il existe 3 autres cellules à ce même niveau de hiérarchie, qui ne sont pas représentées (à savoir l'inverseur, la porte NON-OU à 2 entrées et la porte OU-EXCLUSIF à 2 entrées). Des niveaux supplémentaires de

10 hiérarchie peuvent exister, par exemple un ou plusieurs niveaux plus élevés que sur la figure 2A.

Un modèle de connectivité replié fournit une représentation plus efficace de données de conception VLSI

15 de source, telles qu'elles sont visibles par le concepteur de systèmes VLSI. Cependant une limitation fondamentale du modèle de connectivité replié est son aptitude à représenter un objet adressable réellement unique pour chaque objet créé parmi les nombreux niveaux de la

20 hiérarchie de conception. Bien que ceci ne soit pas un problème pour de nombreux outils de conception CAO existants, c'est de plus en plus un problème pour l'analyse de production suivante et pour les outils de conception, qui doivent analyser des entités de conception qui

25 englobent la hiérarchie. On peut suivre la conception des figures 2A à 2C pour illustrer la limitation fondamentale d'un modèle de connectivité replié. Sur ces figures, la cellule de niveau supérieur, figure 2A, contient deux instances de la cellule "full_adder", FA0 et FA1. La

30 cellule "full_adder" contient deux instances du bloc "nand_2", NA1 et NA2. Cette information est enregistrée dans le modèle de connectivité replié tel que représenté sur la figure 4 annexée à la présente demande. On notera que sur ce schéma, il existe une seule cellule 406 pour les

35 instances NA1 et NA2 (respectivement les blocs 415 et 405).

Mais lorsqu'on observe la même conception sous la forme représentée sur la figure 3, annexée à la présente demande il existe en réalité deux événements différents de l'instance NA1 (FA0/NA1 et FA1/NA1). Ceci est valable pour
5 l'instance NA2 (et par conséquent les réseaux contenus dans les cellules de description pour chacune de ces instances). Le modèle de connectivité replié enregistre uniquement le fait qu'une seule instance de cellule "nand_2" désignée par NA1 est un élément de la cellule "full_adder".

10 Une technique usuelle employée pour éviter ce problème consiste à exécuter un processus "d'aplatissement". Ce processus d'aplatissement d'une conception hiérarchique supprime tous les niveaux intermédiaires de la hiérarchie de sorte que seuls des éléments primitifs
15 existent. Il existe deux problèmes principaux liés à l'aplatissement. Tout d'abord l'aplatissement utilise une grande quantité de mémoire d'ordinateur. Avec des conceptions actuelles de microprocesseurs, il est fréquemment impossible d'aplatir l'ensemble de la
20 hiérarchie. Le second problème est que l'aplatissement est un processus exécuté dans un seul sens. Une fois l'aplatissement réalisé, il est possible d'associer en retour des éléments de circuit aplatis à une vue hiérarchique.

25 Pour ces raisons, la représentation du modèle d'événement (ou non replié) devient une représentation plus importante pour de nombreux outils de conception CAO actuels. Dans un modèle d'événement typique, chaque cellule est mémorisée y compris les cellules qui sont dupliquées,
30 tout en conservant la notion de la hiérarchie de conception d'origine. L'avantage principal d'un modèle d'événement est qu'il permet à ces outils d'obtenir les avantages d'un aplatissement (possibilité de voir une vue aplatie de la conception et des réseaux d'interconnexion qui englobent la
35 hiérarchie) sans perte d'information hiérarchique. En

outre, l'utilisation d'un modèle d'événement fournit une certaine souplesse de manoeuvre au développeur des outils, de sorte que ces derniers n'ont pas à construire un modèle pour représenter l'ensemble de la conception, mais
5 uniquement pour les éléments actuellement évalués. A titre d'exemple, chaque additionneur 202 et 203 est mémorisé séparément dans le modèle, de même que chaque second circuit NON-ET (N2) 205, 208 ou chaque additionneur et chaque transistor N2 207, 209 du circuit N2, comme
10 représenté sur la figure 3. La figure 3 représente la vue à niveaux multiples de la cellule de la figure 1 avec les différents niveaux de l'exemple des figures 2A-2C. (On notera qu'à titre de simplification les autres éléments du circuit ainsi que les sous-éléments, par exemple I1, I2,
15 X01, X02, N01 et N02 ne sont pas représentés). Cependant des circuits intégrés modernes, par exemple des processeurs, comprennent des millions d'instances. Par conséquent la taille du modèle devient rapidement importante étant donné que les niveaux inférieurs sont
20 ajoutés au modèle. Les systèmes informatiques actuels ne comportent aucune mémoire adéquate pour mémoriser le modèle d'événement complet. Cependant, les niveaux inférieurs deviennent plus importants pour les concepteurs. La présence des niveaux inférieurs dans le modèle permet une
25 analyse plus détaillée du système, par exemple pour des pertes parasites à partir des connexions du réseau, qui améliore de ce fait la vitesse et l'efficacité du système.

La figure 4 représente un exemple partiel de la représentation en mémoire du modèle de connectivité replié pour la conception schématique représentée sur la figure 2.
30 Dans ce modèle, une seule copie d'instances de niveau inférieur est conservée. Des pointeurs 409-414 sont utilisés pour associer une cellule supérieure à des instances de niveau inférieur de cette cellule. La figure 4
35 modélise le même exemple comme représenté sur les figures

2A à 2C et 3. Ce modèle ne mémorise aucune instance spécifique ni aucune information d'événement de réseau. Au lieu de cela, chaque instance d'une cellule désigne une cellule générique de niveau supérieur connue comme étant la
5 cellule de descripteur). Par exemple la cellule d'additionneur à deux bits 400 inclut deux instances d'additionneurs complets 400, 403. Chacune de ces instances 402, 403 désigne, au moyen de pointeurs 410, 409, la même cellule de descripteur d'additionneur complet 404. Des
10 additionneurs complets autres que les instances 402, 403 désignent également la cellule d'additionneur complet 404. Par conséquent une quantité moins importante d'informations est mémorisée pour le modèle. Cette relation de descripteurs entre des instances et des cellules utilisant
15 un pointeur de cellule de description est établie à travers les différents niveaux de hiérarchie de conception, et par exemple des cellules NON-ET 405, 415 désignent toutes deux une cellule NON-ET 406, et des transistors 407, 416 désignent tous deux la cellule P 408. Par conséquent le
20 modèle replié possède uniquement une cellule pour chaque type d'instance, tandis que le modèle d'événement possède une cellule différente pour chaque événement de chaque instance. Par exemple le modèle replié possède une seule cellule pour le transistor P, à savoir la cellule 408,
25 tandis que le modèle d'événement, tel que représenté sur la figure 3, comporte quatre cellules (ou des événements uniques de la cellule), l'une pour chacun de P1 et P2 de FA0, et P1 et P2 de FA1 (on notera que des cellules P supplémentaires peuvent être présentes dans le modèle
30 d'événement, étant donné que X01, X02, N01, N02, I1 et I2 sont également susceptibles d'utiliser un transistor P en tant que sous-élément, mais ne sont pas représentés sur la figure 3. Par conséquent le modèle replié est nettement plus petit que le modèle d'apparition. On notera que le
35 modèle replié inclut une information de port, qui n'a pas

été représentée sur la figure 4 à titre de simplification. Le modèle de la figure 4 ne représente pas toutes les cellules, par exemple N01 ou N02 à titre de simplification.

Cependant, le modèle replié ne mémorise pas
5 toutes les informations spécifiques d'événement. Cela signifie qu'une analyse exécutée sur le modèle ne reflète pas de façon précise certains des aspects physiques de la conception actuelle. Par exemple l'analyse d'un retard ou d'effets parasites n'est pas précise étant donné que le
10 contexte de chaque instance n'est pas mémorisé dans la conception. Etant donné que seule une cellule générique est mémorisée pour chaque type d'instance, alors la même information est utilisée dans l'analyse de chaque instance lors de la conception. En réalité chaque instance est
15 différente. Par exemple, seul un ensemble d'informations pour le réseau Y est mémorisé dans le modèle replié, mais chaque événement du réseau Y est différent, par exemple chaque événement possède des retards différents et/ou des effets parasites différents étant donné par exemple le
20 positionnement différent pour chaque réseau Y. Cependant le modèle replié ne mémorise pas ces différences et par conséquent ne fournit pas une analyse précise de la conception.

En outre, l'analyse du modèle replié est
25 difficile étant donné qu'un transit ou un déplacement montant et descendant des signaux du réseau dans la hiérarchie est gêné en raison de l'absence d'informations. En utilisant le 2-BIT_ADD à titre d'exemple : il existe un réseau B au niveau médian de la hiérarchie du système
30 (additionneur complet, figure 2B). Le réseau B connecte trois instances N01, X01 et NA1. On suppose qu'un concepteur veut suivre une information de retard de cadencement et par conséquent a besoin de passer au niveau inférieur suivant, par exemple le niveau Y dans NA1 qui est
35 raccordé au réseau B. Cependant dans le modèle replié, il

est très difficile de trouver quels réseaux au niveau inférieur (ou au niveau supérieur) sont raccordés au réseau actuel étant donné que différents réseaux font partie de différentes cellules et qu'il n'existe dans le modèle replié aucune information qui décrit les connexions de niveau inférieur du réseau actuel B pour l'instance de cellule de niveau inférieur NO1, XO1 et NA1. (Les connexions du réseau de niveau supérieur pour 2_BIT-ADD). En utilisant un port et une instance de port, on peut trouver le réseau correspondant au niveau suivant comme par exemple Y dans NA. Cependant, des concepteurs ne peuvent pas indiquer les différences entre le réseau Y 211 et le réseau Y 212 étant donné qu'il existe un seul Y 420 dans le bloc de cellule NA 406 et que le réseau Y 211 et que le réseau Y 212 font partie de l'instance NA1 415. Cette difficulté s'applique également au niveau supérieur dans la hiérarchie. C'est pourquoi, un suivi efficace de signaux du réseau montant et descendant dans la hiérarchie est un challenge important pour les outils de conception de l'art antérieur.

Les concepteurs peuvent utiliser des tables de contrôle pour mémoriser une certaine information d'instance unique, qui n'est pas mémorisée dans le modèle replié, comme par exemple les données de retard de cadencement pour le réseau Y 211 et le réseau Y 212. On notera que de telles tables ne font pas partie du modèle replié et sont mémorisées séparément du modèle. Les tables sont créées par les concepteurs et incluent une information indiquant que le concepteur est intéressé par l'utilisation dans l'analyse de la conception. Cependant des modèles, qui utilisent ces tables, ne présentent aucun rendement de mémoire et tendent à fonctionner lentement en raison de mauvaises conceptions "maison". De même les tables sont transitoires et sont conservées uniquement pour l'analyse à laquelle on s'intéresse, puis sont éliminées. C'est

pourquoi chaque dispositif requiert sa propre table de contrôle. C'est pourquoi les tables de contrôle "maison" ne sont pas portables, c'est-à-dire que les tables de contrôle ne peuvent pas être réutilisées par différents concepteurs travaillant sur le système plus étendu. Chaque concepteur crée ses propres tables. Les tables de contrôle permettent aux concepteurs d'écrire une certaine information du réseau dans des fichiers. Lors du suivi d'un pas à des instances de ports pour obtenir l'instance de niveau désirée, le fichier correspondant est ouvert pour récupérer l'information nécessaire. Cependant ceci ne présente aucun rendement de mémoire et est lent et n'est pas réutilisable par différentes applications.

L'art antérieur ne fournit pas aux concepteurs un mécanisme pour parcourir des réseaux dans le sens descendant et montant dans la hiérarchie, ce qui est critique pour n'importe quel réseau global basé sur une application d'analyse comme par exemple une analyse de cadencement.

RÉSUMÉ DE L'INVENTION

La présente invention concerne un système et un procédé qui parcourent des événements de réseaux d'un modèle d'événement léger.

Le modèle d'événement léger utilise de préférence l'arrangement du modèle replié, mais inclut des noeuds d'événement qui sont associés au modèle replié. Il en résulte que le modèle d'événement léger augmente l'agencement du modèle replié de manière à inclure les noeuds d'événement (le contexte manquant), puis délivre un API riche permettant à des programmeurs de parcourir les noeuds d'événement. Il est préférable d'avoir au moins un noeud d'événement pour chaque instance de cellule ou réseau dans le modèle replié. Chaque noeud d'événement inclut des données spécifiques d'événement ou un pointeur désignant de telles données, un pointeur désignant un noeud d'événement

de propriétaire et un pointeur désignant une instance de modèle replié. Par conséquent la majeure partie de l'information, qui est présente dans un modèle d'événement complet, peut être incluse dans le modèle d'événement léger
5 selon l'invention.

Pour conserver le poids léger du modèle d'événement, une information, qui peut être obtenue à partir du modèle replié (comme par exemple la relation entre les ports, les instances de ports et les réseaux)
10 n'est de préférence pas dupliquée dans la représentation en mémoire. L'invention permet à un utilisateur de se déplacer dans le sens montant ou dans le sens descendant à travers la hiérarchie du réseau (à un niveau à un instant) sans nécessiter la connaissance des détails de la mise en oeuvre
15 ou des détails des relations avec le modèle de connectivité replié. Des événements de ports ne sont pas conservés étant donné que cette information peut être obtenue à partir du modèle replié et que l'inclusion de ces événements dans le modèle d'événement augmenterait l'utilisation de la mémoire
20 d'environ deux ordres de grandeur. C'est pourquoi, la possibilité de se déplacer aisément en montant et en descendant à l'intérieur de la hiérarchie du réseau sans avoir à connaître les détails de mise en oeuvre est essentiel pour former un modèle d'événement utilisable.
25 Ceci est obtenu en prévoyant deux dispositifs d'itération pour des réseaux d'événements qui fournissent l'illusion que le modèle mémorise la totalité des réseaux d'événements connectés à n'importe quel réseau d'événement situé un niveau au-dessus ou au-dessous dans la hiérarchie, ce qui
30 fournit une interface très simple et abstraite pour l'utilisateur en dépit du fait qu'un calcul très compliqué est effectué dans ces dispositifs d'itération.

L'invention commence le parcours en initialisant le dispositif itératif selon l'invention avec une
35 information concernant le réseau d'événement actuel à

partir du noeud d'événement selon l'invention, qui décrit le possesseur et le descriptif de modèle replié du réseau d'événement. Lors du parcours descendant, un itérateur de port est utilisé de préférence, alors que dans le parcours
5 montant, un itérateur d'instance de port est de préférence utile. Ceci est dû à la convention de conception à partir de l'extérieur selon lequel à partir de l'extérieur d'un objet une connexion est une instance de port, alors qu'à partir de l'intérieur, la connexion est un port.

10 Lors du parcours dans le sens montant, l'itérateur trouve le port suivant qui connecte le réseau du modèle replié indiqué par le descripteur utilisé pendant l'initialisation.

Lors du parcours dans le sens montant, l'itérateur trouve le port suivant qui connecte le réseau de
15 modèle replié indiqué par le descripteur utilisé pendant l'initialisation. L'itérateur localise alors l'instance dans le modèle replié désigné par le descripteur du possesseur de l'objet du réseau. Ensuite, en utilisant
20 l'instance et le port suivant, l'itérateur localise l'instance de port correspondante. Etant donné que chaque instance de port peut être connectée uniquement à un réseau, l'itérateur peut récupérer le réseau qui se raccorde à l'instance de port, qui est le réseau de modèles
25 repliés qui est situé, dans la hiérarchie, à un niveau plus élevé que l'objet de réseau d'événement d'origine. L'itérateur utilise ensuite le pointeur du possesseur d'objet du réseau en tant que code de recherche pour rechercher un conteneur de carte du réseau de modèles repliés récupérés.
30 Cette recherche détermine l'apparition d'un objet du réseau qui est associé au réseau de modèle replié et se situe, dans la hiérarchie, à un niveau plus élevé que l'objet de réseaux d'événements d'origine. S'il n'existe aucune adaptation à la recherche, l'itérateur répète l'opération
35 pour le port suivant, qui est connecté au modèle replié. Si

pendant le fonctionnement l'itérateur rencontre des zéros, alors il arrête l'opération.

Lors du parcours dans le sens descendant, l'itérateur trouve l'instance de port suivante qui réalise
5 le raccordement au réseau de modèle replié indiqué par le descripteur utilisé pendant l'initialisation. L'itérateur récupère ensuite le port qui décrit l'instance de port. Etant donné que chaque port peut être connecté uniquement à un réseau, l'itérateur récupère ensuite le réseau qui est
10 connecté au port dans le modèle replié. L'itérateur utilise le possesseur de l'objet du réseau en tant que code de recherche pour rechercher un conteneur de carte du possesseur de l'instance de port, qui est une instance. Cette recherche fournit la cellule d'événement qui possède
15 l'objet de réseau qui est situé, dans la hiérarchie, à un niveau inférieur à celui de l'objet du réseau d'événement original. S'il n'existe aucune concordance dans cette recherche, l'itérateur répète l'opération pour l'instance de port suivant, qui est connectée au modèle replié. Cette
20 recherche peut être répétée pour trouver des objets de réseaux d'événements supplémentaires, qui sont inférieurs d'un niveau à l'objet de réseau d'origine. Si pendant le fonctionnement, l'itérateur rencontre des zéros, l'itérateur arrête alors le fonctionnement.

25 La présente invention concerne un système et un procédé, qui facilitent le parcours dans un modèle d'événement replié léger, qui inclut des aspects à la fois du modèle replié et du modèle d'événement.

Le modèle d'événement léger selon l'invention
30 utilise l'arrangement du modèle replié, mais inclut des noeuds d'apparition qui sont associés au modèle replié. Il existe au moins un noeud d'événement pour chaque instance d'un type d'objet dans le modèle. Chaque noeud d'événement inclut des données spécifiques d'événement ou un pointeur
35 désignant de telles données, un pointeur désignant un noeud

d'événement parent, et un pointeur désignant une instance de modèle replié. Par conséquent la majeure partie de l'information, qui est présente dans un modèle d'événement complet, peut être incluse dans le modèle d'événement léger selon l'invention. Etant donné que le modèle d'événement selon l'invention est plus petit que le modèle d'événement complet, l'ensemble de conception de circuits complexes, par exemple de microprocesseurs, peut être représenté par le modèle d'événement léger selon l'invention. C'est pourquoi on peut examiner des caractéristiques à faible niveau de la conception, par exemple des parasites.

Les pointeurs ou d'autres interfaces abstraites permettent de modifier l'information des noeuds du modèle replié pour qu'elle inclut les données uniques des instances du modèle d'événement, y compris son emplacement hiérarchique dans le modèle. C'est pourquoi, l'ensemble du modèle d'événement complet peut être représenté par le modèle d'événement léger selon l'invention, en fonction de la quantité de données spécifiques associées au noeud (soit mémorisées dans le noeud, soit désignées par le noeud). On notera que les noms de noeuds d'événement n'ont pas à être mémorisés dans les noeuds. S'il est nécessaire pour l'analyse et n'est pas mémorisé, le nom du noeud peut être constitué d'une information dans le modèle d'événement léger selon l'invention. On notera que les noeuds d'événement ne mémorisent pas l'information qui est déjà présente dans le modèle replié, par exemple une information concernant des noeuds enfants et/ou une information concernant des noeuds de ports, de cellules et de réseaux. Par conséquent, la mémoire requise pour le modèle d'événement léger selon l'invention est nettement inférieure à celle requise pour le modèle d'événement complet. Par exemple un noeud selon l'invention, qui mémorise seulement trois pointeurs, peut nécessiter 12 octets, alors qu'un noeud d'événement complet peut

nécessiter 40 octets ou un plus grand nombre d'octets, ce qui signifie que le modèle d'événement léger selon l'invention requiert seulement environ 30 % de la mémoire requise pour le modèle d'événement complet.

5 Un autre aspect de la présente invention réside dans le fait que le modèle d'événement léger peut être affiché à un utilisateur de telle sorte que seul un niveau désiré de hiérarchie pour une partie désirée du modèle est présenté à l'utilisateur. Dans l'art antérieur, le modèle
10 d'événement complet impose le déploiement du modèle replié au niveau le plus bas du modèle. On notera que ceci permet que le modèle comporte une plus grande quantité d'informations détaillées (niveau inférieur) devant être mémorisée pour une partie particulière, tout en comportant seulement
15 une information moins détaillée (niveau plus élevé) mémorisée dans le reste du modèle. Ceci permet d'utiliser un arbre partiel, dont certaines branches comportent une information plus détaillée que d'autres branches.

 Comme indiqué précédemment, il n'est pas
20 nécessaire que les noms de noeuds d'événement soient mémorisés dans les noeuds. Cependant, le modèle requiert un travail avec des outils qui requièrent des noms de noeuds pour effectuer une analyse sur les conceptions qui utilisent le modèle. De façon typique l'analyse est une
25 analyse à base hiérarchique et par conséquent des noms hiérarchiques sont requis. Cependant, dans le cadre de l'invention, les noms des noeuds peuvent être constitués par une information contenue dans le modèle d'événement léger selon l'invention, par exemple les pointeurs de
30 noeuds parents, les pointeurs d'instances repliées, etc. C'est l'un des moyens selon lesquels le modèle d'événement léger selon l'invention apparaît à l'utilisateur comme étant un modèle d'événement complet.

 Le modèle d'événement léger selon l'invention
35 maintient l'information spécifique d'instance d'une manière

plus permanente et organisée que celle de l'art antérieur. Les tables de contrôle utilisées dans l'art antérieur sont situées à l'extérieur du modèle. Les tables de contrôle sont également écrites par un concepteur pour sa propre
5 utilisation et sont jetées lorsque l'utilisation est terminée. L'information nécessaire à un instant ultérieur et/ou par un concepteur différent est réécrite dans une nouvelle table de contrôle. Par conséquent l'utilisation de tables de contrôle conduit à une analyse sujette à erreur
10 et inconsistante. C'est pourquoi il suffit que l'information spécifique pour le modèle selon l'invention soit écrite une fois et maintenue avec le modèle selon l'invention.

Dans ce qui précède on a mis en relief d'une
15 manière assez détaillée les caractéristiques et les avantages techniques de la présente invention afin que la description détaillée de l'invention, qui va suivre, puisse être mieux comprise. Les spécialistes de la technique noteront que la conception et la forme de réalisation
20 spécifique décrites peuvent être aisément utilisées en tant que base pour modifier ou concevoir d'autres structures destinées à fonctionner pour assumer les mêmes objectifs que la présente invention. Les nouvelles caractéristiques, dont on estime qu'elles sont caractéristiques de
25 l'invention, à la fois du point de vue organisation et procédé de fonctionnement, ainsi que d'autres buts et avantages seront mieux compris à la lecture de la description qui va suivre prise en référence aux dessins annexés. On comprendra cependant expressément que chacune
30 des figures est fournie uniquement à titre d'illustration et de description et n'a aucun caractère limitatif pour l'invention.

De façon plus précise l'invention concerne un
procédé de parcours entre un événement de réseau de niveau
35 supérieur et un événement de réseau de niveau inférieur

dans un modèle hiérarchique d'un type de système, selon lequel le modèle inclut une première cellule qui décrit un premier dispositif dans le système et une seconde cellule qui décrit un second dispositif dans le système,

5 l'événement de réseau de niveau supérieur étant associé à la première cellule et l'événement de réseau de niveau inférieur étant associé à la seconde cellule, et la première cellule ayant, dans le modèle, au moins un niveau plus élevé que la seconde cellule, caractérisé en ce que le

10 procédé comprend les étapes consistant à :

- définir un événement de réseau actuel, qui est l'un de l'événement de réseau de niveau supérieur et de l'événement de réseau de niveau inférieur,
- recevoir un premier pointeur descriptif pour

15 l'événement de réseau actuel, le premier pointeur descriptif désignant un objet modèle qui est utilisé pour décrire l'événement de réseau actuel,- recupérer un premier pointeur de propriétaire pour l'événement de réseau actuel, le pointeur de

20 propriétaire désignant la cellule de propriétaire de l'événement de réseau actuel; et- utiliser l'objet du modèle du réseau actuel et la cellule de propriétaire pour parcourir le modèle hiérarchique.

25 Selon une autre caractéristique de l'invention, l'événement de réseau actuel est l'événement de réseau de niveau inférieur, la cellule de propriétaire est la seconde cellule et le modèle est parcouru depuis l'événement de réseau de niveau inférieur jusqu'à l'événement de réseau de

30 niveau supérieur, et l'étape d'utilisation comprend la localisation d'un port dans le modèle, qui est associé à l'objet de modèle de réseau actuel.

- Selon une autre caractéristique de l'invention, l'étape d'utilisation comprend en outre :

35 la récupération d'un second pointeur descriptif

pour la cellule de propriétaire, le second pointeur descriptif désignant une instance de modèle qui est utilisée pour décrire la cellule de propriétaire, et

la localisation de l'instance de cellule de propriétaire dans le modèle désigné par le second pointeur descriptif.

Selon une autre caractéristique de l'invention, l'étape d'utilisation comprend en outre la localisation d'une instance de port dans le modèle utilisant l'instance de cellule de propriétaire et le port.

Selon une autre caractéristique de l'invention, l'étape d'utilisation comprend en outre la localisation d'un objet de modèle de réseau de niveau supérieur, qui est associé à l'instant de port.

Selon une autre caractéristique de l'invention, l'étape d'utilisation comprend en outre :

la récupération d'un second pointeur de propriétaire pour la cellule de propriétaire, le second pointeur de propriétaire désignant la première cellule, et la recherche d'un conteneur associé à l'objet d'un modèle de réseau de niveau supérieur utilisant le second pointeur de propriétaire en tant que code de recherche,

le conteneur incluant une liste d'événements de réseau associés à l'objet de modèle de réseau de niveau supérieur.

Selon une autre caractéristique de l'invention, l'événement de réseau actuel est un événement de réseau de niveau supérieur, la cellule de propriétaire est la première cellule et le modèle est parcouru depuis l'événement de réseau de niveau supérieur jusqu'à l'événement de réseau de niveau inférieur, et que l'étape d'utilisation comprend la localisation d'une instance de port dans le modèle, qui est associé à l'objet de modèle de réseau actuel.

Selon une autre caractéristique de l'invention, l'étape d'utilisation comprend en outre :

la récupération d'un second pointeur descriptif pour l'instance de port, le second pointeur descriptif désignant un port qui est utilisé pour décrire l'instance de port, et

la localisation du port dans le modèle désigné par le second pointeur descriptif.

Selon une autre caractéristique de l'invention, l'étape d'utilisation comprend en outre la localisation d'un objet de modèle de réseau de niveau inférieur, qui est associé au port.

Selon une autre caractéristique de l'invention, l'étape d'utilisation comprend en outre :

la récupération d'un second pointeur de propriétaire pour l'instance de port, le second pointeur de propriétaire désignant une instance de modèle qui est utilisée pour décrire la seconde cellule,

la recherche d'un conteneur associé à la seconde instance de modèle de cellule avec utilisation du second pointeur de propriétaire en tant que code de recherche,

le conteneur incluant une liste de cellules associées à la seconde instance de modèle de cellule.

D'autres caractéristiques et avantages de la présente invention ressortiront de la description donnée ci-après prise en référence aux dessins annexés, sur lesquels :

- la figure 1, dont il a déjà été fait mention, représente une illustration schématique d'un modèle d'événement complet de l'art antérieur;

- les figures 2A à 2C, dont il a déjà été fait mention, montrent une représentation graphique de l'information hiérarchique dans un modèle d'événement complet de l'art antérieur;

- la figure 3, dont il a déjà été fait mention,

montre une représentation graphique du modèle d'événement complet de l'exemple des figures 2A à 2C;

- la figure 4, dont il a déjà été fait mention, représente une illustration schématique d'un modèle de connectivité replié de l'art antérieur utilisant l'exemple
5 des figures 2A à 2C;

- la figure 5 représente une illustration schématique d'un noeud d'événement selon l'invention;

- la figure 6 représente une partie de l'agence-
10 ment de la figure 3, qui utilise le noeud selon l'invention de la figure 5;

- la figure 7 représente un organigramme du fonctionnement de l'invention pour remonter un niveau de réseau dans la hiérarchie;

- la figure 8 représente un organigramme du
15 fonctionnement de l'invention pour descendre d'un niveau de réseau dans la hiérarchie; et

- la figure 9 représente un schéma-bloc d'un système informatique qui est adapté pour la mise en oeuvre
20 de la présente invention.

Comme indiqué précédemment, une limitation fondamentale du modèle de connectivité replié de l'art antérieur est son impossibilité à représenter un objet adressable réellement unique pour chaque objet créé dans la
25 hiérarchie de conception. Il en résulte que la mémorisation, efficace du point de vue espace de mémoire, du modèle replié ne conserve pas le contexte dans lequel une instance ou un réseau apparaît, de sorte que le modèle replié ne conserve pas un contexte suffisant pour mémoriser
30 une information spécifique à un événement. La capacité à maintenir ce contexte est notamment importante pour des outils d'analyse et de modélisation de circuits (comme par exemple des moteurs d'analyse de cadencement et de bruit et des outils de compréhension des circuits) qui requièrent
35 une modélisation des effets détaillés du réseau lorsqu'il

est représenté sous la forme aplatie (c'est-à-dire lorsqu'il est éventuellement formé sur du silicium). Par exemple on considère le réseau B 421 sur la figure 2B. Dans la représentation de connectivité repliée (figure 4), il
5 existe seulement un élément de mémorisation en mémoire pour représenter le réseau B 421 dans la cellule FA 404.

Pour trouver les réseaux qui sont connectés au réseau B421 qui est un niveau plus élevé dans la hiérarchie, les liaisons de descripteurs 409, 410 suivent
10 les instances dans la cellule 2_BIT_ADD, qui sont décrites par la cellule FA 404. Dans ce cas les instances FA0 et FA1 sont trouvées. Pour trouver les réseaux auxquels le réseau B est connecté, nous devons alors trouver le ou les réseaux dans la cellule 2_BIT_ADD, qui sont connectés à l'instance
15 de port B rattachée à chaque instance, et il faut trouver FA0 et FA1. Dans ce cas, l'instance de port B de l'instance FA0 est connectée au réseau B0 dans la cellule 2_BIT_ADD et l'instance de port B de l'instance FA1 est connectée au réseau B1. Cependant, à partir du réseau unique B421 dans
20 le modèle replié, il n'existe pas une information suffisante pour déterminer quelle apparition du réseau B est analysée (c'est-à-dire s'il est connecté à B0 ou B1). L'absence de contexte devient plus problématique étant donné qu'un plus grand nombre de niveaux de hiérarchie sont
25 ajoutés (par exemple on considère l'ambiguïté supplémentaire qui est créée si la cellule 2_BIT_ADD devait être instanciée plusieurs fois d'un niveau dans le sens croissant dans la hiérarchie). Ceci peut être mieux illustré dans l'exemple suivant le réseau B abaissé d'un
30 niveau dans la hiérarchie. Sur la figure 2B, ce réseau est connecté à trois instance NO1, XO1 et NA1 dans le modèle replié. Etant donné que la cellule FA est instanciée 2 fois dans la cellule 2_BIT_ADD (instances FA0 à FA1), il existe 2 événements du réseau B qui est connecté à 2 événements de
35 chaque instance (FA0/NO1, FA0/ZO1, FA0/NA1, FA1/NO1,

FA1/Z01, FA1/NA1). En exécutant de nombreux types d'analyses du réseau B, il faut connaître l'événement analysé de manière à obtenir des résultats précis.

En outre, dans le réseau suivant B dans la
5 cellule NA1, on peut voir que le réseau B est connecté au réseau Y 206 à l'intérieur de la cellule NA (à partir de la figure 2C). Dans le modèle replié, ce réseau est représenté comme Y 420, que possède la cellule NA 406. Cependant les concepteurs ne peuvent pas distinguer lequel des quatre
10 événements du réseau Y ils comportent (FA0/NA1/Y 211, FA0/NA2/Y 212, FA1/NA1/Y 207 ou FA1/NA2/Y 209).

La conception du modèle de connectivité replié est un artefact du mode avec lequel des concepteurs de circuits introduisent des conceptions schématiques. Le
15 concept de hiérarchie est utile pour le concepteur de circuits étant donné qu'il lui permet de fonctionner à des niveaux plus élevés d'abstraction et de réduire la complexité. Cependant, lorsque la hiérarchie est aplatie (éliminée), le modèle de connectivité replié est
20 insuffisant pour maintenir le contexte hiérarchique nécessaire pour analyser et mémoriser une information spécifique d'événement (comme par exemple le positionnement actuel, l'orientation et la taille de chaque événement). Il est important de noter que le processus d'aplatissement de
25 la hiérarchie est une étape nécessaire au cours de l'apparition de la conception d'une microplaquette du point de vue fonctionnel abstrait du concepteur pour la production réelle du circuit sur du silicium. Il en résulte qu'il est important de disposer d'un grand nombre des
30 outils actuels pour modéliser les effets de signaux, comme on les voit dans la forme aplatie.

Avant le modèle d'événement léger selon l'invention, les concepteurs possédaient un mécanisme non réutilisable pour assister l'analyse d'événements de
35 réseaux à travers la hiérarchie de conception. Comme

indiqué précédemment, l'aplatissement est devenu impossible pour la majorité des circuits créés dans des microprocesseurs actuels (et dans des processus actuels de fabrication de circuits intégrés de l'état de la technique). Il en résulte que de nombreuses applications étaient basées sur des conventions de dénominations et des structures de données (telles que des tables de contrôle ou des cartes), conservées à l'extérieur du modèle de connectivité replié de manière à réaliser une mémorisation pour conserver le contexte nécessaire pour identifier de façon unique des événements de réseaux et fournir un emplacement pour la mémorisation spécifique à l'événement. Ces structures externes présentent les inconvénients de ne pas être réutilisables et/ou de ne pas comporter une intégration plus serrée dans le cadre de la connectivité repliée (ce qui fournit un parcours plus facile entre deux vues différentes de conception).

La figure 5 décrit un exemple d'un mode d'événement 50, qui fournit l'information spécifique à l'événement et est associé à un objet correspondant dans le modèle replié. Il existe au moins un noeud d'événement pour des instances correspondantes de réseaux, de cellules et de port d'un modèle replié. Par exemple, en utilisant le modèle de l'art antérieur, figure 4, il est prévu un noeud d'événement 50 pour chaque instance 402, 403 de l'additionneur complet de la cellule d'additionneur à deux bits 400 ainsi que pour les réseaux, ports et instances de ports de la cellule 400.

Chaque noeud inclut des données spécifiques d'événement 51. Ces données peuvent être l'information qui est utilisée pour rendre unique l'instance, c'est-à-dire par exemple pour rendre l'instance sous la forme de valeur unique de retard de cadencement du réseau, de résistance, de capacité, de grille de puissance, de test de simulation. Par exemple la figure 3 représente deux réseaux Y 211 et

212, l'un dans FA0 et l'autre dans FA1. Cependant les réseaux B0 et B1 ont des longueurs différentes, ce qui a pour effet que les caractéristiques de cadencement pour chaque réseau Y sont différentes.

5 Chaque noeud 50 inclut un pointeur 52 de noeuds de propriétaires. On notera que le propriétaire est une instance d'événement et que chaque réseau possède un propriétaire différent. Ce pointeur 52 dispose les noeuds d'événement sous un format hiérarchique par désignation
10 d'une instance d'événement précédente dans la hiérarchie de noeuds. Etant donné que ceci est unique pour n'importe quel mode d'apparition et signifie que deux noeuds d'événement possédant un même descripteur (instance, réseau, instance de bord) ne peuvent pas se partager le même pointeur 52, le
15 pointeur peut être utilisé pour un code de recherche pour une récupération rapide d'un quelconque noeud d'événement dans le conteneur STL à partir du modèle replié. Par exemple l'épaisseur du réseau Y 211 et le
20 2_bit_adder/FA0/NA1 et le possesseur pour le réseau Y 212 est le 2_bit_adder/FA1/NA1.

 Chaque noeud 50 inclut également un pointeur 53 de description de modèles repliés. Ce pointeur 53 désigne l'instance ou le réseau ou l'instance de port, auquel le noeud est associé dans le modèle replié. Par exemple à la
25 fois 211 et 212 comportent le pointeur 53 désignant le net_Y 420 (figure 4). Ce partage d'informations permet une économie importante sur l'espace de mémoire du modèle. Le pointeur 53 permet à des noeuds d'événement possédant le même descripteur de se partager la même information dans le
30 modèle replié. Ce pointeur et le pointeur 52 jouent un rôle très important dans l'algorithme de parcours selon l'invention.

 Tous les noeuds d'événement 50 sont mémorisés physiquement dans une instance correspondante, un réseau
35 correspondant et une instance de port correspondante dans

le modèle replié. Par exemple des noeuds de réseaux d'événements 2_BIT_ADD/FA0/B et 2_BIT_ADD/FA1/B sont mémorisés dans le réseau B au niveau du bloc de cellule FA 404.

5 Comme indiqué précédemment, le parcours d'un réseau passant par les différents niveaux de hiérarchie est très difficile étant donné l'absence de pointeurs entre les niveaux. Ceci peut être résolu par mémorisation d'une information additionnelle, par exemple des pointeurs dans
10 le modèle. Cependant ceci augmente fortement la taille du modèle et rend plus difficile la mémorisation et l'analyse du modèle. C'est pourquoi il est préférable d'utiliser le procédé et le dispositif selon l'invention pour effectuer le parcours entre des niveaux du réseau. La présente
15 invention inclut deux algorithmes : le parcours montant pour récupérer le réseau d'événement parent au niveau plus élevé de la hiérarchie, qui est connecté au réseau d'événement actuel; et le parcours descendant pour récupérer le ou les réseaux d'événement enfant au niveau
20 inférieur de la hiérarchie, qui est raccordé au réseau d'événement actuel.

La figure 6 représente un ensemble du modèle d'événement léger selon l'invention 600, représenté à un niveau élevé, comportant un réseau de niveau supérieur 602
25 qui est associé à une instance de cellules de niveau supérieur 601, et un réseau de niveau inférieur 604 qui est associé à une instance de cellule de niveau inférieur 603. Le procédé et le dispositif de parcours selon l'invention permettent à un concepteur de passer du réseau 602 au
30 réseau 604 ou de passer du réseau 604 au réseau 602. Les relations entre ces réseaux sont désignées comme étant les relations parent (supérieure) et enfant (inférieure). Ces relations de cette figure montrent l'exemple illustré sur les figures 2-4, dans lesquelles le réseau de niveau
35 supérieur B1 est raccordé au réseau de niveau inférieur B,

qui à son tour est connecté aux réseaux de niveau les plus bas Y, Z, W. On notera qu'un enfant peut avoir au maximum un seul parent, bien qu'un parent puisse avoir plusieurs enfants. Par conséquent n'importe quel réseau particulier
5 peut comporter un parent ou ne comporter aucun parent (c'est-à-dire est le signal le plus intense) et à tout réseau particulier peut posséder un ou plusieurs enfants ou n'en posséder aucun (c'est-à-dire que c'est le réseau de signal le plus faible). De même on notera qu'il est
10 possible que le réseau ne soit connecté à aucun réseau d'événement enfant et ne soit pas encore le réseau du signal de niveau le plus faible. Ceci peut se produire étant donné que le réseau fait partie d'un arbre partiel, dans lequel un ou plusieurs enfants n'ont pas été
15 introduits dans l'arbre.

Le procédé et le dispositif, ou itérateur, de parcours selon l'invention possèdent deux étages. Le premier étage initialise l'itérateur et le second étage trouve le réseau au niveau désiré, s'il existe. On notera
20 que l'itérateur s'élève ou descend d'un niveau à la fois. Le procédé, selon lequel l'itérateur fonctionne, dépend du fait que l'utilisateur désire monter d'un niveau pour aboutir à un noeud parent ou descendre d'un niveau pour aboutir à un noeud enfant. La figure 7 représente les
25 étapes d'initialisation et de localisation pour l'itérateur de parcours parent, qui remonte d'un niveau. La figure 8 illustre les étapes d'initialisation et de localisation pour l'itérateur de parcours parent, qui remonte d'un niveau. La figure 8 représente les étapes d'initialisation
30 et de localisation pour l'itérateur de parcours enfant, qui descend d'un niveau. Chaque itérateur englobe une recherche complète qui chemine à travers le modèle replié et le modèle d'événement, mais masque la complexité à l'utilisa-
teur.

35 La figure 7 représente le parcours visant à

récupérer le réseau d'événement parent à partir du niveau
situé au-dessus. On comprendra mieux cette opération en la
décrivant sous la forme d'un exemple. Dans l'exemple, un
concepteur désire passer de l'objet de réseau d'événement
5 602 à l'objet de réseau d'événement 606 de la figure 6, qui
représente également le passage du réseau B de 203 au
réseau B1 de 200 sur la figure 3. Sur la figure 7, l'itéra-
teur 700 est initialisé par l'affectation de l'objet de
réseau d'événement actuel à l'itérateur. Ceci affecte un
10 réseau d'événement de départ - le réseau d'événement actuel
à l'itérateur 701 (par exemple le bloc 602 2_BIT_ADD/FA1/B
de la figure 6). L'itérateur récupère ensuite en 703 le
possesseur de l'objet de réseau actuel (par l'intermédiaire
du pointeur 52 du réseau d'événement actuel) qui est la
15 cellule d'événement à laquelle le possesseur de l'objet de
réseau est associé (par le pointeur 52 du réseau
d'événement actuel), qui est la cellule d'événement à
laquelle est associé l'objet de réseau, par exemple le bloc
601, 2_BIT_ADD/FA1. L'itérateur récupère également en 702
20 le descripteur de l'objet de réseau actuel, qui est l'objet
de réseau dans le modèle replié (par l'intermédiaire du
pointeur 53 du réseau d'événement pour obtenir le réseau B
421). L'itérateur affecte ensuite, en 704, le descripteur à
l'itérateur de port, qui est dans le modèle replié pour
25 achever l'initialisation.

L'itérateur appelle ensuite le processus suivant
next() 705 pour commencer à trouver l'objet d'événement de
réseau parent, qui est le bloc 602. L'itérateur utilise en
706 le descripteur situé dans la partie itérateur de port
30 pour essayer de récupérer le premier port qui est connecté
à l'objet de réseau, qui, dans l'exemple, revient au port B
422. On notera qu'un port peut connecter uniquement un
réseau, bien qu'un réseau puisse être connecté à de
nombreux ports. En outre on notera que des ports sont des
35 points de connexion et à partir de l'intérieur d'un bloc,

la connexion est un port, et à partir de l'extérieur, la connexion est une instance de port. Par exemple, comme cela est représenté sur la figure 4, le cadre de port B@FA0 424 et le cadre de port B@FA1 423 sont des instances de ports
5 dans le modèle replié comme en 105, et ces instances de ports sont connectées respectivement au réseau B0 218 et au réseau B1 219, bien qu'à l'intérieur des instances 203 et 202 (de la figure 2A). Cependant à partir de l'une ou l'autre des instances 203 ou 202, qui sont le full_bit_add
10 de la figure 2B, le port B 422 est connecté au réseau B 421. Ainsi les réseaux B0 209 et B1 208 sont connectés au réseau B 421.

L'itérateur effectue alors une vérification pour savoir si le port actuel est zéro en 707 ou si le
15 possesseur d'objets du réseau est un zéro en 708. Si l'un ou l'autre de ces éléments est vrai, alors la procédure s'arrête en 715 étant donné que l'objet du réseau est le signal le plus intense des deux et n'est pas connecté à un réseau parent, ou bien l'itérateur a déjà parcouru tous les
20 réseaux dans les objets de niveau supérieur, qui sont connectés au réseau d'événement actuel. Ceci peut également indiquer un arbre partiel, et par exemple sig_4 de la figure 2B n'a pas de parent.

Si les deux vérifications fournissent un résultat
25 négatif, comme pour le parcours montant du réseau d'événement 2_BIT_ADD/FA1/B 602, l'itérateur utilise en 709 le possesseur de réseau d'événement actuel 601, qui est un objet d'instance d'événement 2_BIT_ADD/FA1, pour récupérer son descripteur FA1 403 dans le modèle replié. On notera
30 que chaque instance possède au moins une ou plusieurs instances de ports qui sont connectées à cette instance.

L'itérateur utilise une fonction "GetPort Instance" de l'instance dans le modèle replié avec l'information du port 422, qu'il a reçue lors du pas 701,
35 et récupère l'instance de port correspondante Port_B@FA1

423 lors du pas 710. Lors du pas 711, étant donné que chaque instance de port peut uniquement connecter un réseau, alors l'itérateur peut aisément trouver le réseau B1 219 (419 de la figure 4) dans le modèle replié qui est
5 connecté à l'instance de port 423, à savoir le réseau B1 du bloc 400.

L'itérateur utilise alors le pointeur du possesseur (noeud d'instance d'événement 2_BIT_ADD/FA1 601), qui est le noeud d'instance d'événement 2_BIT_ADD
10 605, en tant que code de recherche pour rechercher un ensemble de noeuds du réseau d'événement contenus dans un conteneur de carte dans B1 419 (ou 219) du modèle replié lors du pas 712. L'itérateur effectue alors une vérification pour savoir s'il existe une concordance en
15 713. S'il existe une concordance, alors il existe un réseau d'événement qui est connecté au réseau d'événement actuel, au niveau plus élevé, dans cet exemple il s'agit du réseau d'événement sur 2_BIT_ADD/B1 606. C'est une opération de performance $O(\log N)$, qui est une recherche binaire qui
20 rétrécit la base de données de moitié lors de chaque incrément. Dans l'exemple, ce pas est le 606. L'objet de réseau d'événement est renvoyé et l'itérateur est prêt pour le processus suivant pour le pas 704.

Si l'itérateur ne trouve pas le pointeur, alors
25 il revient au bloc 706 et effectue une nouvelle exécution en récupérant la partie suivante qui est raccordée à l'objet de réseau.

La figure 8 illustre le parcours dans le sens descendant pour récupérer un ou des réseaux d'événements au
30 niveau inférieur, qui se raccordent au réseau d'événement actuel. A nouveau cette opération sera mieux comprise si on la décrit sur la base d'un exemple. Dans l'exemple un concepteur désire passer de l'objet du réseau d'événement 602 à l'objet du réseau d'événement 604 sur la figure 6,
35 qui est également le réseau B de 203 vers le réseau Y 202

sur la figure 3. Sur la figure 8, l'itérateur 800 est initialisé par l'affectation de l'objet de réseau d'événement actuel à l'itérateur. Lors du pas 801, ceci affecte un réseau d'événement de départ, le réseau d'événement actuel, à l'itérateur, par exemple bloc 602, 2_BIT_ADD/FA1/B de la figure 6. L'itérateur récupère alors en 803 le possesseur de l'objet de réseau actuel (au moyen du pointeur 52 du réseau d'événement actuel affecté), qui est la cellule d'événement à laquelle est associé l'objet de réseau, par exemple le bloc 601, 2_BIT_ADD/FA1. L'itérateur récupère également en 802 le descripteur de l'objet de réseau actuel, qui est l'objet de réseau dans le modèle replié (au moyen du pointeur 53 du réseau d'événement actuel aboutissant au réseau B 421). L'itérateur affecte ensuite en 804 le descripteur du réseau d'événement actuel, le réseau B 421 à un itérateur d'instance de port qui se situe dans le modèle replié pour achever l'initialisation. Le réseau B 421 inclut une liste de toutes les instances de ports, qui sont connectées à ce réseau.

L'itérateur appelle alors le processus next() en 805 pour commencer à trouver les objets d'événement du réseau enfant. L'objet désiré de l'exemple étant le bloc 604 (2_BIT_ADD/FA1/NA2/Y). L'itérateur d'instance de port est appelé en 806 pour localiser l'instance de port suivante de l'objet de réseau, qui dans cet exemple est port_Y en NA1 425.

L'itérateur effectue ensuite une vérification pour savoir si l'instance de port récupérée est nulle en 807. Si elle est vraie, alors le processus s'achève en 815, étant donné que l'objet de réseau possède le niveau le plus bas et ne se raccorde pas à un réseau enfant, ou bien l'itérateur a déjà parcouru tous les réseaux situés au niveau inférieur, qui se raccordent au réseau actuel. Si la vérification fournit un résultat négatif, comme par exemple dans cet exemple, alors l'itérateur récupère le descripteur

de l'instance de port, qui est le port correspondant, lors du pas 808. C'est le port de modèle replié Y 429 au-dessous du bloc 406. L'itérateur appelle alors la fonction de l'objet de port Y GetNet() pour localiser le réseau
5 connecté au port lors du pas 809. On notera que chaque port est connecté uniquement à un réseau. Dans cet exemple, le port Y 429 est connecté au réseau Y 420.

Lors du pas 811, l'itérateur utilise ensuite le pOwner (601) en tant que code de recherche pour rechercher
10 un conteneur de carte de l'instance du propriétaire de l'instance de port 425 INA1 415. L'opération de recherche est une opération de recherche O (LogN) dans un ensemble de noeuds de réseaux d'événements contenus dans le conteneur de carte. Lors du pas 810, le résultat de la recherche est
15 renvoyé, à savoir 2_BIT_ADD/FA1/NA1 603.

Lors du pas 812, l'itérateur utilise la cellule occ revenant du pas 810, à savoir 603, lorsque le code de recherche pour rechercher un conteneur de carte d'un réseau localisé au moyen du pas 809, à savoir Y 420. Le conteneur
20 du réseau comprend une liste de réseaux d'événements (réseaux occ) qui utilisent le réseau Y 420 du modèle replié, en tant que descripteur de ces ensembles. L'itérateur vérifie s'il existe des concordances en 813. Si une concordance est trouvée, alors l'itérateur revient à
25 l'objet de réseau d'événement, et dans ce cas il s'agit du réseau d'événement 2_BIT_ADD/FA1/NA1/Y 604, et est prêt pour l'opération suivante 814. Si une concordance n'est pas trouvée, alors l'itérateur revient au cadre 806 pour recommencer la séquence, si cela est souhaité. Par exemple
30 lors de la répétition de cette opération, l'itérateur revient au réseau d'événement 2_BIT_ADD/FA1/X01/Z, puis au réseau d'événement 2_BIT_ADD/FA1/NO1/W lors d'une opération suivante désirée. Etant donné que le réseau d'événement actuel connecte uniquement 3 réseaux d'événements au niveau
35 inférieur de la hiérarchie, l'opération dernier itérateur->

Next() revient à ZÉRO lorsqu'il atteint l'extrémité 815.

Lorsqu'ils sont mis en oeuvre en logiciel, les éléments de la présente invention sont essentiellement les segments de code pour effectuer les tâches nécessaires. Les
5 segments de programme ou de code peuvent être mémorisés dans un support lisible par le processeur ou être transmis par un signal de données d'ordinateur incorporé dans une onde porteuse, ou un signal modulé par une porteuse, par l'intermédiaire d'un milieu de transmission. Le "milieu
10 lisible par un processeur" peut inclure n'importe quel milieu qui peut mémoriser ou transférer une information. Des exemples du milieu lisible par un processeur incluent un circuit électronique, un dispositif de mémoire à semiconducteurs, une mémoire ROM, une mémoire flash, une
15 mémoire ROM effaçable (EROM), une disquette, un disque compact CD ROM, un disque optique, un disque dur, un support à fibres optiques, une liaison à fréquences radio (RF), etc. Le signal de données d'ordinateur peut inclure n'importe quel signal qui peut se propager dans un milieu
20 de transmission tels que des canaux de réseaux électroniques, des fibres optiques, de l'air, une transmission électromagnétique, des liaisons RF, etc. Les segments de code peuvent être téléchargés par l'intermédiaire de réseaux informatiques comme par exemple
25 le réseau dit Internet, le réseau dit Intranet, etc.

La figure 9 représente un système informatique 900 adapté pour utiliser la présente invention. L'unité centrale de traitement (CPU) 901 est couplée à un bus 902 du système. L'unité CPU 901 peut être n'importe quelle
30 unité CPU à usage général, comme par exemple un processeur HP P8500 ou un processeur Intel Pentium. Cependant la présente invention n'est pas limitée par l'architecture de l'unité 901 tant que l'unité CPU 901 supporte les opérations selon l'invention comme cela est décrit ici. Le
35 bus 902 est couplé à une mémoire à accès direct (RAM) 203,

qui peut être une mémoire SRAM, une mémoire DRAM ou une mémoire SDRAM. La mémoire ROM 904 est également couplée au bus 902, cette mémoire pouvant être une mémoire PROM, EPROM ou EEPROM. La mémoire ROM 903 et la mémoire ROM 904
5 conservent les données d'utilisateurs et les données du système et des programme comme cela est connu dans la technique.

Le bus 902 est également couplé à une carte 905 de contrôleur d'entrée/sortie (I/O), à une carte 911
10 d'adaptateurs de communications, à une carte d'interface d'utilisateur 908 et à une carte d'affichage 909. La carte I/O 905 est connectée des dispositifs de mémorisation 906, comme par exemple une ou plusieurs des unités suivantes : unité de disque dur, unité de disque compact, unité de
15 disquette, unité de bande, au système informatique. La carte de communications 911 est adaptée pour coupler le système informatique 900 à un réseau 912, qui peut être formé par un ou plusieurs des systèmes suivants : réseau téléphonique, réseau local (LAN) et/ou large comme un
20 réseau (WAN), un réseau Ethernet et/ou le réseau Internet. La carte 908 d'interface d'utilisateur accouple des dispositifs d'entrée d'utilisateurs, tels qu'un clavier 913 et un dispositif de pointage 907, au système informatique 900. La carte d'affichage 909 est activée par l'unité CPU
25 901 pour commander l'affichage sur un dispositif d'affichage 910.

Bien que l'on ait décrit de façon détaillée la présente invention et ses avantages, on comprendra que différents changements, substitutions et modifications
30 peuvent être exécutés ici sans sortir du cadre de l'invention. En outre la portée de la présente invention n'est pas censée être limitée aux formes de réalisation particulières de processus, machines, fabrication, compositions de matières, moyens, procédés et pas d'exécution décrits ici.

REVENDICATIONS

1. Procédé de parcours entre un événement de réseau de niveau supérieur (602) et un événement de réseau de niveau inférieur (604) dans un modèle hiérarchique d'un type de système, selon lequel le modèle inclut une première cellule (601) qui décrit un premier dispositif dans le système et une seconde cellule (603) qui décrit un second dispositif dans le système, l'événement de réseau de niveau supérieur étant associé à la première cellule et l'événement de réseau de niveau inférieur étant associé à la seconde cellule, et la première cellule ayant, dans le modèle, au moins un niveau plus élevé que la seconde cellule, caractérisé en ce que le procédé comprend les étapes consistant à :

15 définir un événement de réseau actuel, qui est l'un de l'événement de réseau de niveau supérieur et de l'événement de réseau de niveau inférieur,

recevoir (702,802) un premier pointeur descriptif (53) pour l'événement de réseau actuel, le premier pointeur descriptif désignant un objet modèle qui est utilisé pour décrire l'événement de réseau actuel,

20 récupérer (703,803) un premier pointeur de propriétaire (52) pour l'événement de réseau actuel, le pointeur de propriétaire désignant la cellule de propriétaire de l'événement de réseau actuel, et

25 utiliser l'objet du modèle du réseau actuel et la cellule de propriétaire pour parcourir le modèle hiérarchique.

2. Procédé selon la revendication 1, caractérisé en ce que l'événement de réseau actuel est l'événement de réseau de niveau inférieur, la cellule de propriétaire est la seconde cellule et le modèle est parcouru depuis l'événement de réseau de niveau inférieur jusqu'à l'événement de réseau de niveau supérieur, et que l'étape d'utilisation comprend la localisation (706) d'un port dans

le modèle, qui est associé à l'objet de modèle de réseau actuel.

3. Procédé selon la revendication 2, caractérisé en ce que l'étape d'utilisation comprend en outre :

5 la récupération (709) d'un second pointeur descriptif pour la cellule de propriétaire, le second pointeur descriptif désignant une instance de modèle qui est utilisée pour décrire la cellule de propriétaire, et

10 la localisation de l'instance de cellule de propriétaire dans le modèle désigné par le second pointeur descriptif.

4. Procédé selon la revendication 3, caractérisé en ce que l'étape d'utilisation comprend en outre la localisation (710) d'une instance de port dans le modèle
15 utilisant l'instance de cellule de propriétaire et le port.

5. Procédé selon la revendication 4, caractérisé en ce que l'étape d'utilisation comprend en outre la localisation (711) d'un objet de modèle de réseau de niveau supérieur, qui est associé à l'instant de port.

20 6. Procédé selon la revendication 5, caractérisé en ce que l'étape d'utilisation comprend en outre :

 la récupération d'un second pointeur de propriétaire pour la cellule de propriétaire, le second pointeur de propriétaire désignant la première cellule, et

25 la recherche (712) d'un conteneur associé à l'objet d'un modèle de réseau de niveau supérieur utilisant le second pointeur de propriétaire en tant que code de recherche,

 le conteneur incluant une liste d'événements de
30 réseau associés à l'objet de modèle de réseau de niveau supérieur.

7. Procédé selon la revendication 1, caractérisé en ce que l'événement de réseau actuel est un événement de réseau de niveau supérieur, la cellule de propriétaire est
35 la première cellule et le modèle est parcouru depuis

l'événement de réseau de niveau supérieur jusqu'à l'événement de réseau de niveau inférieur, et que l'étape d'utilisation comprend la localisation (806) d'une instance de port dans le modèle, qui est associé à l'objet de modèle de réseau actuel.

5
8. Procédé selon la revendication 7, caractérisé en ce que l'étape d'utilisation comprend en outre :

la récupération (808) d'un second pointeur descriptif pour l'instance de port, le second pointeur descriptif désignant un port qui est utilisé pour décrire l'instance de port, et

10
la localisation du port dans le modèle désigné par le second pointeur descriptif.

9. Procédé selon la revendication 8, caractérisé en ce que l'étape d'utilisation comprend en outre la localisation (809) d'un objet de modèle de réseau de niveau inférieur, qui est associé au port.

10. Procédé selon la revendication 9, caractérisé en ce que l'étape d'utilisation comprend en outre :

20
la récupération d'un second pointeur de propriétaire pour l'instance de port, le second pointeur de propriétaire désignant une instance de modèle qui est utilisée pour décrire la seconde cellule,

la recherche (811) d'un conteneur associé à la seconde instance de modèle de cellule avec utilisation du second pointeur de propriétaire en tant que code de recherche,

25
le conteneur incluant une liste de cellules associées à la seconde instance de modèle de cellule.

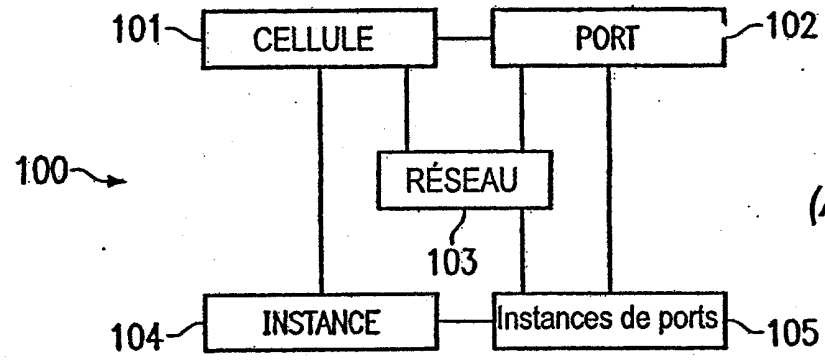


FIG. 1
(Art antérieur)

200
FIG. 2A
(Art antérieur)

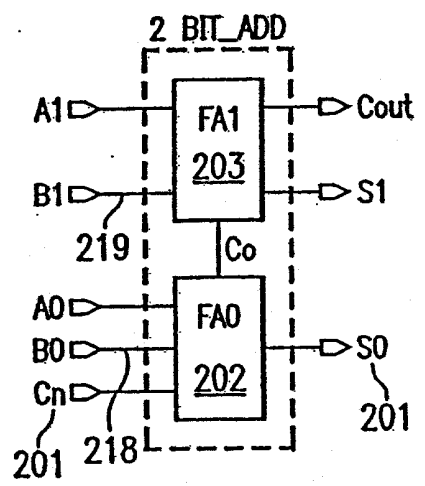


FIG. 2C
(Art antérieur)

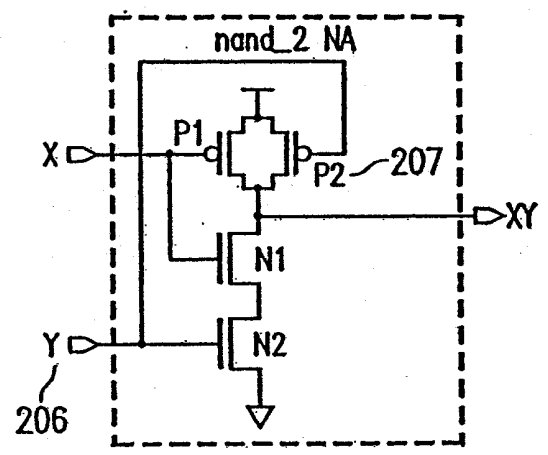
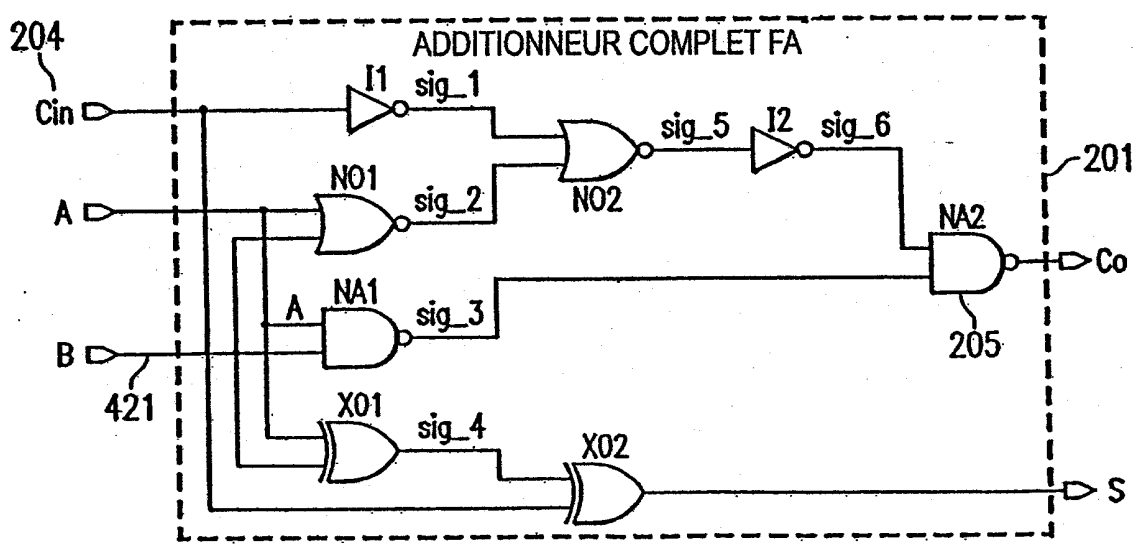


FIG. 2B
(Art antérieur)



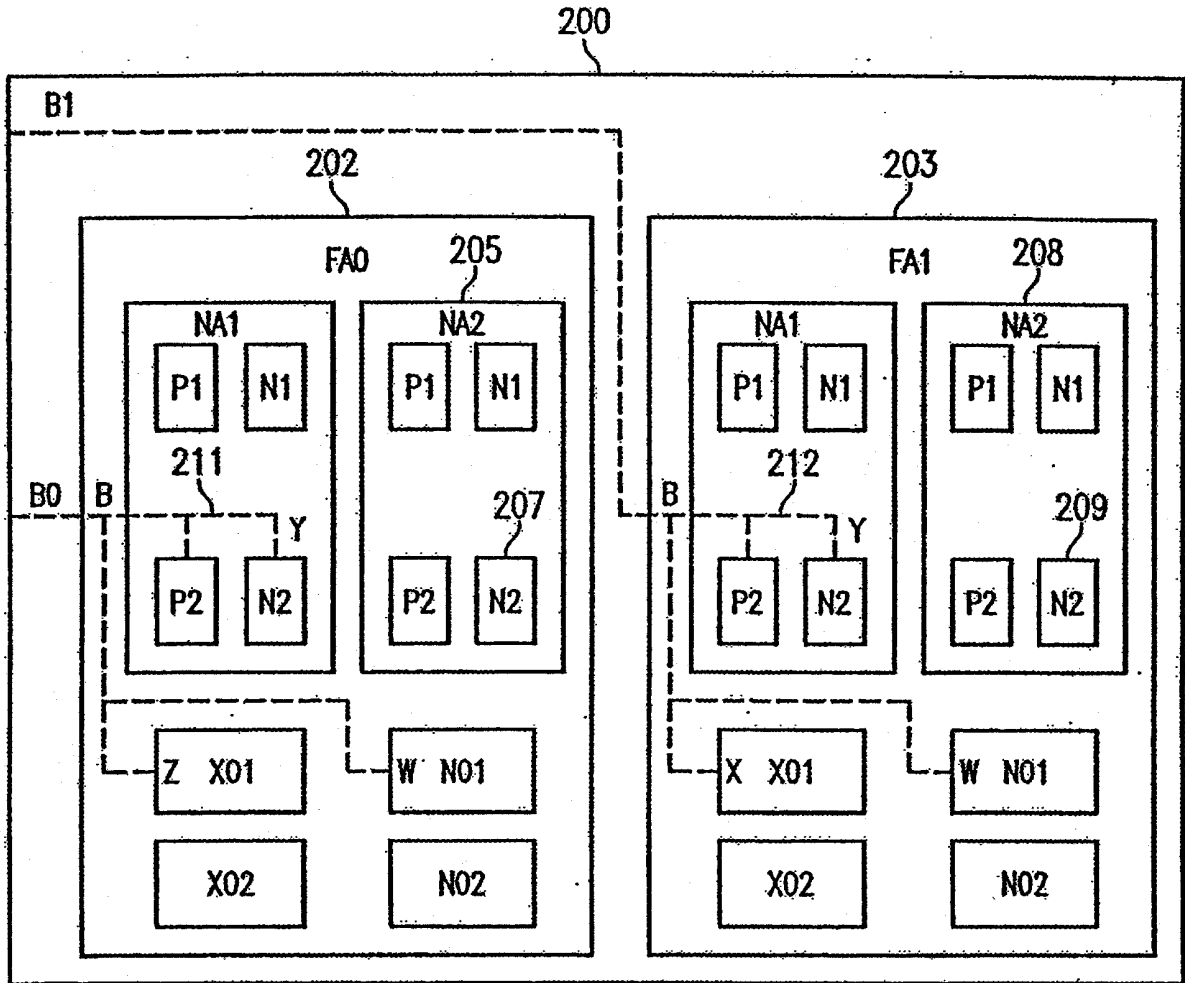


FIG. 3
(Art antérieur)

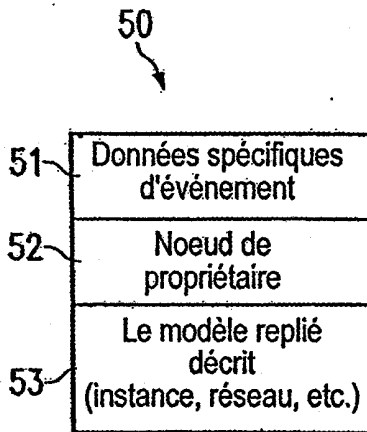


FIG. 5

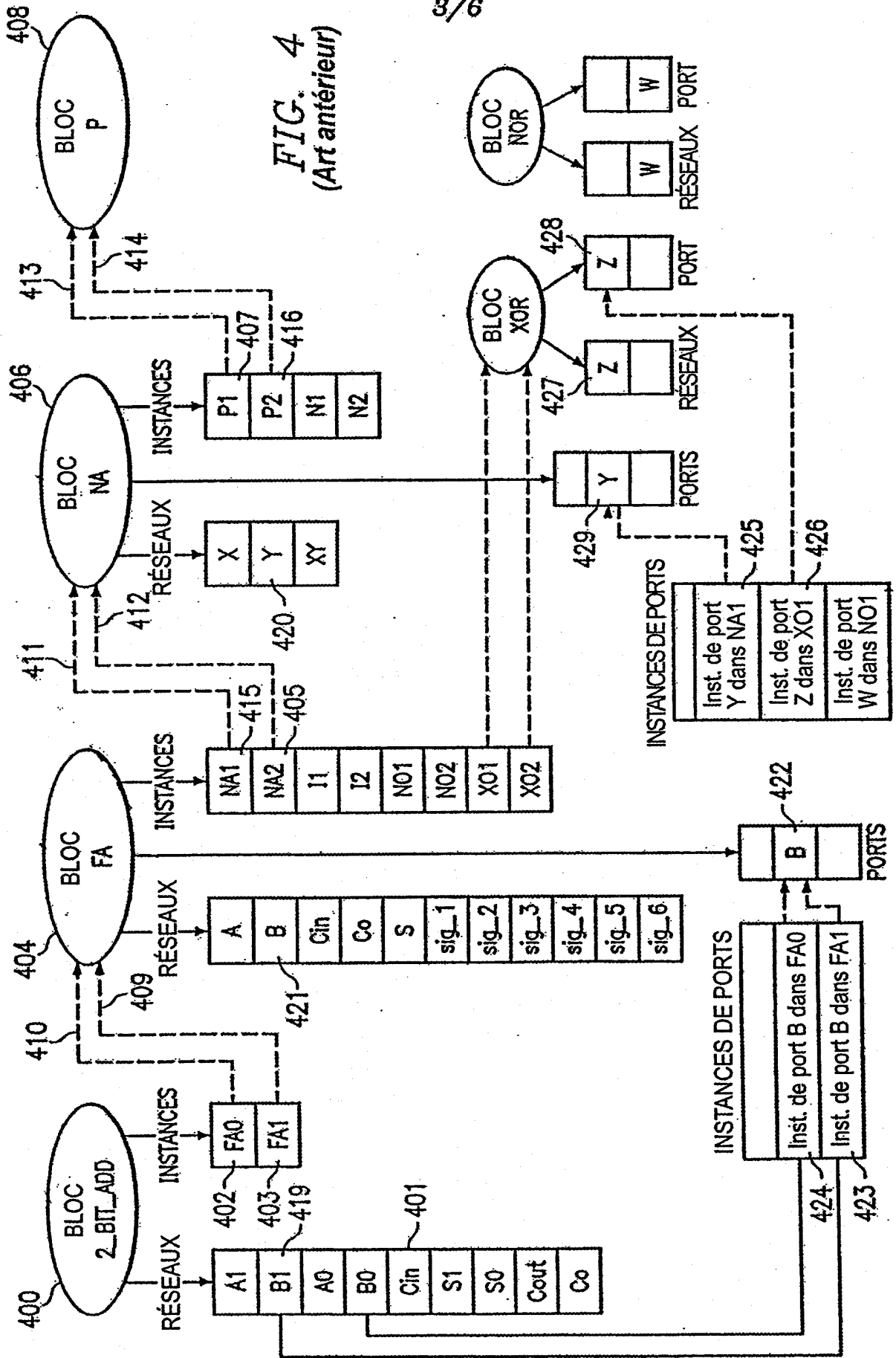


FIG. 4
(Art antérieur)

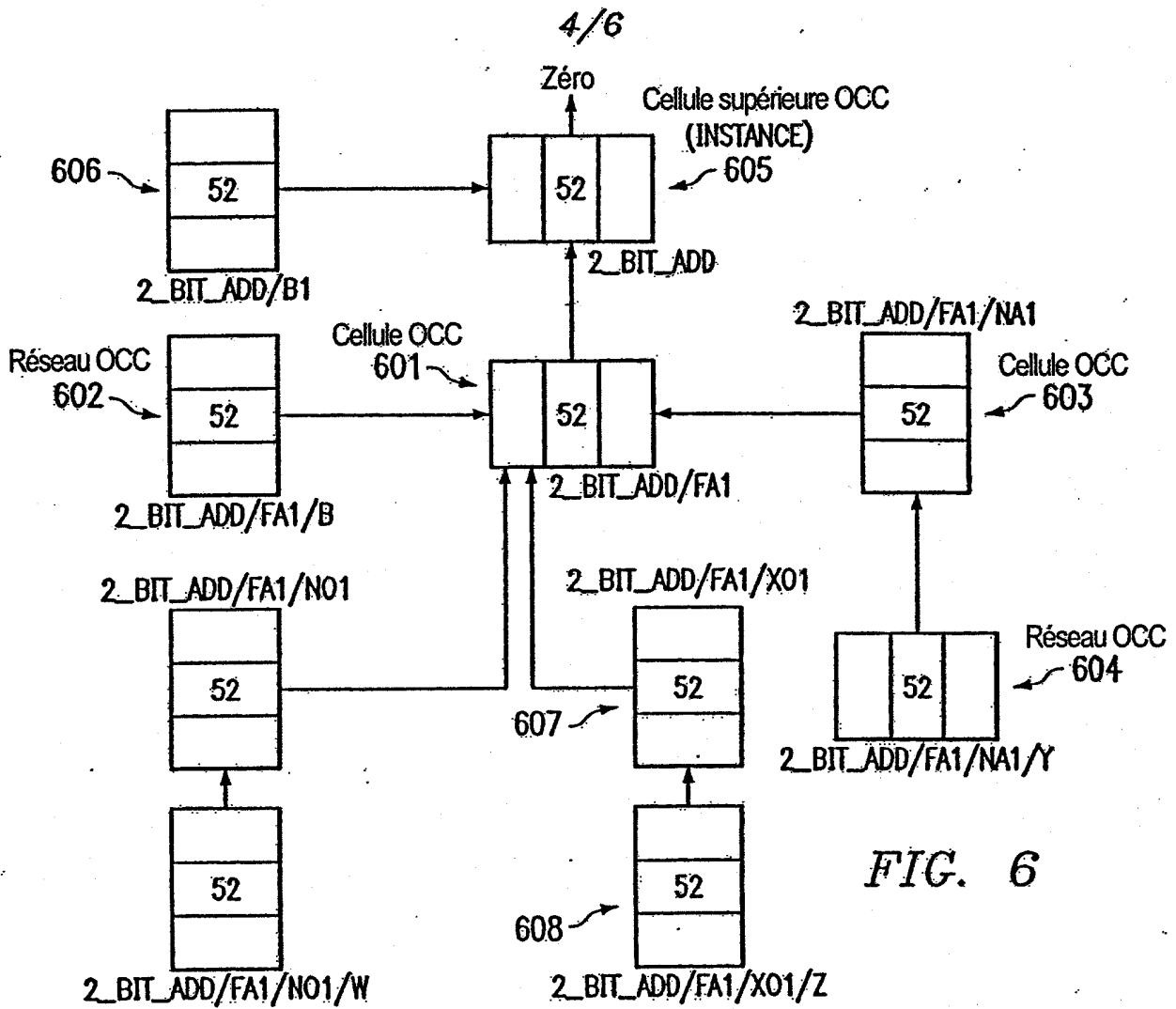


FIG. 6

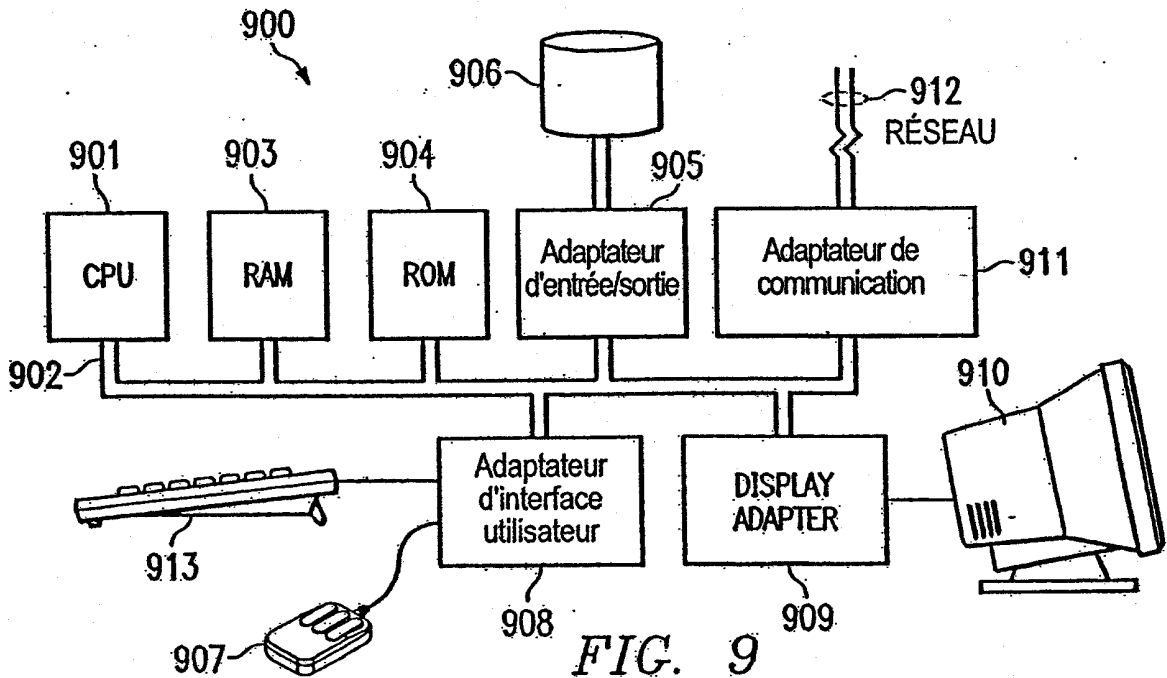


FIG. 9

5/6

