

STRUCTURATION DES DONNÉES DANS LES OUTILS DE VÉRIFICATION ET DE SIMULATION BACK-END

Dominique LE DÛ Pirouz BAZARGAN SABET

Département ASIM, laboratoire LIP6, Université Pierre et Marie CURIE
4, place Jussieu, 75252 Paris Cedex 05
Dominique.ledu@lip6.fr

RÉSUMÉ : La conception des systèmes électroniques et informatiques requièrent des circuits VLSI de plus en plus puissants. Pour répondre à ces exigences les designers intègrent toujours plus de transistors pour une surface qui reste constante avec des temps de cycle toujours plus faibles. On s'intéresse dans cet article aux problèmes induit par cette augmentation de puissance dans le cadre de la vérification back-end. Puis on présente une structure de donnée qui répond à ces difficultés nouvelles.

I – INTRODUCTION

Les circuits intégrés actuelles possèdent jusqu'à quelques dizaines de millions de transistors et atteignent une finesse de gravure de 0.13μ . Dans ces circuits sub-microniques, deux facteurs conjoints doivent être pris en compte. D'une part l'augmentation du volume de données directement liés au nombre de transistors du circuit. Cela implique par exemple de redimensionner la mémoire des ordinateurs mais aussi de restructurer la hiérarchie des informations en mémoire pour accroître l'efficacité et la pérennité des outils de simulations et de vérifications. D'autre part, de nouveaux éléments non pris en compte jusqu'alors deviennent prédominant dans le fonctionnement et les performances du circuit. Ainsi la longueur des fils constituent des réseaux RC qui augmentent considérablement le temps de cycle. Aussi la promiscuité des fils ajouté à une diminution des tensions d'alimentations conduit à accroître les bruits parasites sur les signaux. On peut enfin noter que 9% des dysfonctionnements des circuits sont dues aux bruits. De ces constatations, il ressort la nécessité de concevoir des outils de simulation et de vérification dotés de structures adaptées et efficaces aptes à détecter ces dysfonctionnements et à répondre aux nouvelles exigences des circuits VLSI.

On présente dans un premier temps l'outil yagle [1] dont la structure CNS (cone netlist support) générée nous sert de base à la naissance d'une structure plus efficace. Dans un second temps, on considère les nouveaux problèmes comme le couplage diaphonique et l'incidence sur la structure de donnée. Puis nous verrons un exemple de réalisations de structure répondant à ces nouvelles exigences de simulation et de vérification back-end.

II – YAGLE ET SA STRUCTURE DE DONNÉE CNS

Pour concevoir des outils de simulation performant et réaliste, il est obligatoire de disposer de deux paquets d'informations pour un circuit. Un paquet d'informations électriques qui donnent l'agencement des transistors dans le circuit et leurs caractéristiques physiques. Cela permet de concevoir des modèles électriques indispensable pour des calculs de temps précis. Un paquet d'informations logiques qui découpent le circuit en fonctions logiques

offrant une rapidité de simulation fonctionnelle par exemple. Yagle est un abstracteur fonctionnelle qui travaille sur une netlist transistors, elle-même extraite des masques du circuit (Fig 1). Il en résulte une structure de donnée représentant une liste de cônes organisés hiérarchiquement

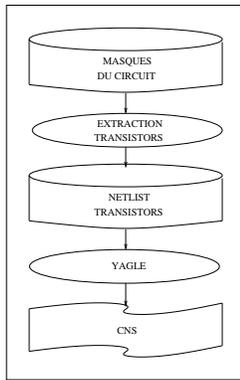


FIG. 1 – Génération de CNS

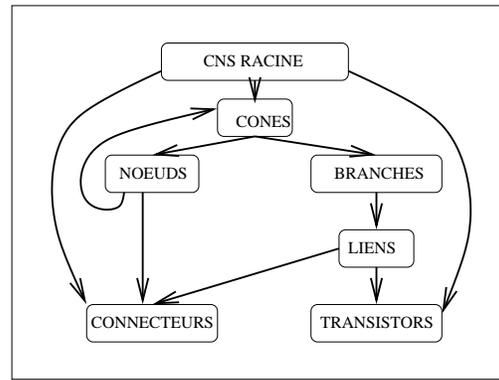


FIG. 2 – Hiérarchie de la structure CNS

La racine contient (Fig 2) la liste des cônes issue du déssassemblage de yagle. Chaque cône est associés à la liste de ses entrées et de ses sorties (noeuds) et à une liste distincte pour les différents types de branches (VDD,VSS ou EXTerne dans le cas des transistors de passages). Chaque noeud référence soit un cône, soit un connecteur externe. Ainsi, il est aisé d’effectuer un parcours du graphe de dépendance dans les deux sens.

Dans la structure CNS, un cône est un élément d’une liste chaînée. Cette organisation facilite l’allocation des structures et le parcours de l’ensemble des cônes. Elle répond donc au premier critère d’être accessible face à la complexité des circuits VLSI.

III – COUPLAGE CAPACITIFS

Un des phénomènes parasites émergeant des circuits fortement sub-microniques est la diaphonie qui se caractérise par un couplage capacitifs entre deux fils électriques voisins. Chaque fils possèdent plusieurs centaines de couplages capacitifs ce qui constitue un volume important de données à traiter.

Un outil d’évaluation Crise [2] du bruit diaphonique basé sur une modélisation simple du transistor [3] offre d’excellente performance comparé aux simulateurs électriques tel que spice.

Crise est architecturé autour d’une structure variante de CNS incluant les données utiles à une simulation électrique.

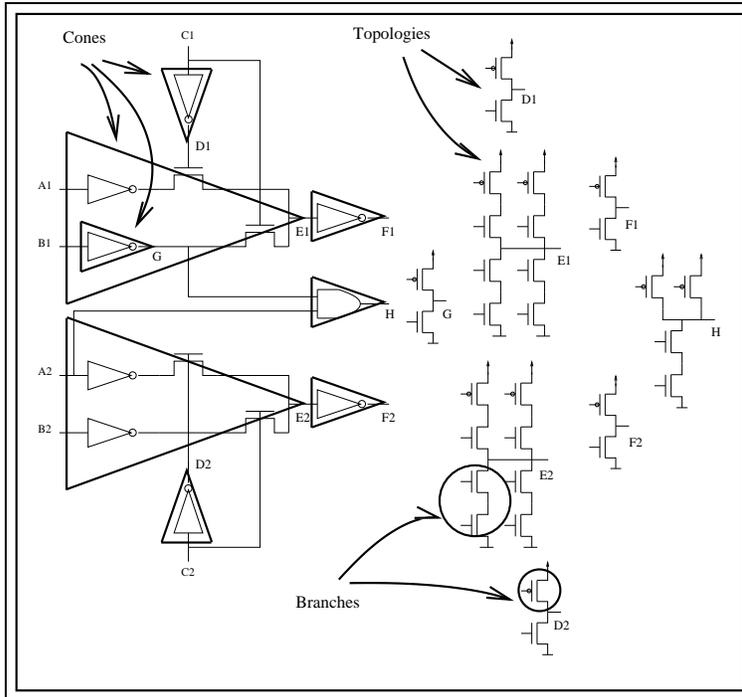
Cet outil démontre outre ses capacités de simulations performantes que la structure de donnée sous-jacente garantie un traitement rapide et efficace des informations.

Crise traite ainsi un "System on Chip" d’un million de transistors en 1h30 pour une consommation mémoire de 140Mo.

IV – STRUCTURE HIÉRARCHIQUE

La structure CNS par elle-même ne sait pas gérer les circuits hiérarchiques, c’est-à-dire, les circuits qui instancient des blocs fonctionnelles comme des additionneurs, des rams, etc... pour réaliser des fonctions plus complexes et aussi permettre de réutiliser ses blocs fonctionnelles pour d’autres circuits. Ce qu’apporte la structure de Crise comme évolution à CNS, c’est le traitement de circuits plus important en gérant l’instanciation de même blocs fonctionnelles en une seule zone mémoire. Cependant la structure interne des blocs est toujours régit par CNS. Or le gros défaut d’une telle structure est de représenter chaque cône en mémoire alors comme on peut le voir sur le circuit présenté ci-après, il serait intéressant d’effectuer des factorisations.

V – EXEMPLE DE CIRCUIT



En Observant le circuit sur la figure de gauche, on constate qu'un certains nombres d'éléments sont redondants.

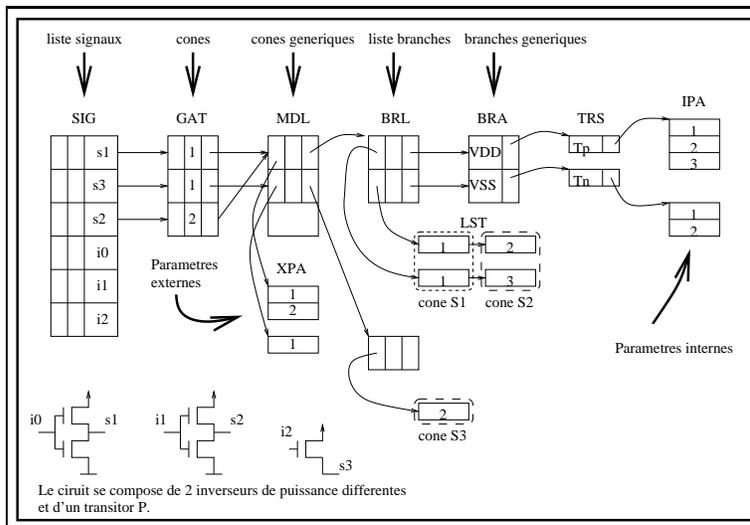
On peut voir que les cônes F1,G,D1,F2 et D2 puis E1 et E2 offrent des topologies en transistors identiques.

On remarque également que beaucoup de cônes ont des topologies de branches identiques avec les autres cônes. Ainsi E1 et E2 ont tous deux 2 branches VDD similaires mais aussi 2 branches à VSS identiques à la branche VSS du cône H.

D'où l'idée de factoriser ces paramètres que l'on désigne par la suite comme cônes génériques et comme branches génériques.

A partir de ces similitudes, il était intéressant de développer une nouvelle structure de données associant la hiérarchie d'une Cone Netlist Structure avec un compactage des informations exploitant la généralité des cônes et de leur topologie en branches.

VI – EXEMPLE DE STRUCTURE



La figure de gauche représente une implémentation de structure de donnée permettant de factoriser principalement deux éléments que sont les cônes et les branches.

On va détailler successivement les étapes amenant à la création de cette structure.

Le premier point consiste à générer dans une table la liste des signaux du circuit. Ici nous avons 3 cônes, dont 2 inverseurs (s1) et (s2) et 1 transistor P unique (s3), chacun des cônes ne possèdent

qu'une seule entrée, respectivement i0,i1 et i2. La table des **SIG**naux est donc renseignée en tout premier lieu. Il s'agit ensuite de parser la netlist de cône pour renseigner la table des cônes ou "**GAT**e". L'idée étant de définir une table de cônes génériques ou cones **MoDè**Les, il faut spécialiser ces cones génériques via un index défini dans **GAT**e. Cette index va définir pour un cône générique de **MoDè**Les 2 jeux de paramètres dont l'un caractérise le cône d'un point de vue externe, c'est la table "**eX**ternal **PA**rameters", et l'autre d'un point de vue interne, c'est la table "**I**nternal **PA**rameters".

Dans la table "**eX**ternal **PA**rameters" on peut y stocker le modèle électrique du cône, dans le cas ?? il y aurait une résistance Min et une Résistance Max.

Dans la table "**I**nternal **P**arameters" on y stocke les caractéristiques de chaque transistors du cône.

Dans ce circuit les 5 transistors sont différents, il y a donc 5 entrées dans la table "**I**nternal **P**arameters". Par conséquent les 3 cônes sont différents et donc la table "**e**Xternal **P**arameters" possèdent 3 entrées.

Cependant, d'un point de vue topologique les cônes (s1) et (s2) sont identiques, ce qui explique qu'ils se réfèrent à la même entrée dans la table des cônes génériques **M**oDèLes.

Chaque cône se caractérise par une liste de branche dans la table "**B**Ranch **L**ist". En l'occurrence, 1 branche à VDD et une à VSS pour les cônes (s1) et (s2) et 1 branche unique a VDD pour le cône (s3).

Au même titre que la table "**G**ATe" spécialisait la table **M**oDèLes, la table "**B**Ranch **L**ist" spécialise la table **B**Ranche qui n'est autre que la liste des branches génériques.

En regardant le schéma, d'un point de vue topologie il n'existe qu'une architecture de branche à VDD et une à VSS, ce qui explique qu'il n'y ait que deux entrées dans la table **B**Ranche.

En revanche, d'un point de vue électrique chaque branche est différente, aussi existe-il 3 jeux de paramètres dans la table **L**iSTE. Cette table est indexée par la même variable qui indexe le jeu de paramètre externe du cône.

Enfin, une branche se caractérise par une liste de **T**RansistorS.

VII – RESULTATS

On présente les résultats obtenus pour une implémentation de la structure ci-dessus. On observe sur le tableau ci-dessous que l'on peut diminuer de moitié la quantité de mémoire alloué comme pour le circuit mips R3000 dont le gain minimum est de 50%. Le gain maximum représente la quantité de mémoire gagné lorsque le circuit est complètement défini par la structure en mémoire.

circuit	Gain min	Gain max
barriere de registre 16 bits	79%	86%
additionneur rapide 64 bits	27%	76%
rom 32bits	47%	84%
amd2901	82%	87%
mips R3000	50%	78%

Le gain Min correspond au gain mémoire lors des phases d'allocations et de libérations de mémoire pour la construction du circuit par rapport à CNS. Le gain Max correspond au gain mémoire une fois que le circuit est totalement généré dans la structure toujours par rapport à une Cone netlist structure classique.

VIII – CONCLUSION

Des difficultés présentées en introduction qu'étaient la hiérarchisation des informations et faire face à la masse de données qu'introduit les phénomènes parasites tel que la diaphonie, on peut dire que cette structure répond bien à ces nouveaux besoins. D'autre part amélioré la structure CNS en une structure factorisée permet une représentation "à plat" de circuit comportant des millions transistors. Enfin, le fait de garder une structure en cône avec leur topologie en transistor laisse la possibilité de concevoir une modélisation électrique plus réaliste applicable dans les simulations temporelles.

RÉFÉRENCES

- [1] *Abstraction fonctionnelle des circuits numériques VLSI avec une méthode formelle basée sur une extraction de réseau de portes*, A.Lester Th. : Architecture micro-électronique, Paris, 1999
- [2] *Analyse du bruit due aux couplages capacitifs dans les circuits intégrés numériques fortement submicroniques*, F.Ilponse Th. : Architecture micro-électronique, Paris, 2002
- [3] *A model for crosstalk noise evaluation in deep submicron Processes*, P.Bazargan Sabet, F.Ilponse ISQUED 2001, p139-144