

Etude comparative de deux algorithmes de placement de systèmes intégrés sur puce

Damien Dupuis
Laboratoire UMPC / LIP6 / ASIM
12 rue Cuvier
75252 Paris Cedex 05

E-mail: damien.dupuis@lip6.fr

Résumé

Cet article présente une étude comparative entre deux algorithmes de placement pour les systèmes intégrés sur puce. Un algorithme de quadri-partitionnement d'une part et un algorithme de placement analytique d'autre part. Le but de cette étude est d'évaluer les performances des nouveaux algorithmes de placement analytique. La comparaison porte sur le temps d'exécution et l'estimation de la longueur des fils d'interconnexion.

1. Introduction

L'utilisation des technologies fortement sub-microniques pousse les concepteurs de circuits VLSI à modifier profondément l'organisation du flot de conception back-end classique.

La multiplicité des problématiques à traiter remet en cause la possibilité de considérer chaque traitement algorithmique (placement, routage, analyse temporelle, ...) indépendamment les uns des autres.

Dans ce cadre, il est nécessaire de réévaluer les solutions algorithmiques existantes les unes par rapport aux autres. Ce papier s'intéresse aux solutions de placement.

Plusieurs algorithmes de placement existent, tels que : le *recuit simulé*, le *partitionnement* ou encore le *placement analytique*. Aujourd'hui, beaucoup de gens s'intéressent au placement analytique car il permet non seulement d'obtenir une solution analytique mais il a aussi une vision globale de toutes les interconnexions du circuit. C'est pourquoi on souhaite pouvoir évaluer les performances du placement analytique.

Pour pouvoir facilement utiliser et comparer les deux algorithmes de placement, on utilise la plate-forme Coriolis [1].

2. Présentation de la plate-forme Coriolis

La plate-forme Coriolis est l'une des premières plates-formes académiques permettant aux différents outils et algorithmes d'interagir autour d'une base de données C++ commune.

Une telle approche permet une implémentation sim-

plifiée des algorithmes ainsi qu'une comparaison de leurs performances dans des conditions similaires.

Dans la suite de l'article, on présentera deux outils de la plate-forme Coriolis : Metisse un quadri-partitionneur basé sur hMetis [2] et Blizzard un placeur analytique basé sur fastPlace [3].

3. Présentation de Metisse

Comme dit précédemment, Metisse est un quadri-partitionneur multi-niveaux basé sur hMetis [2].

Le principe de cet outil est de préciser progressivement les positions des instances en alternant des phases de quadri-partitionnement et des phases de routage global. Initialement, le circuit est divisé en 4 cases et les instances sont réparties dans ces cases de façon à réduire le nombre de liaisons entre les cases (problème de coupe minimale). Ensuite, à chaque niveau, chaque case est subdivisée en 4 et les instances sont réparties dans ces 4 sous-cases en fonction du routage global précédent. Le quadri-partitionneur divise une case tant que celle-ci a plus de 100 instances.

Au final, on obtient un placement tel que celui présenté à la figure 1.

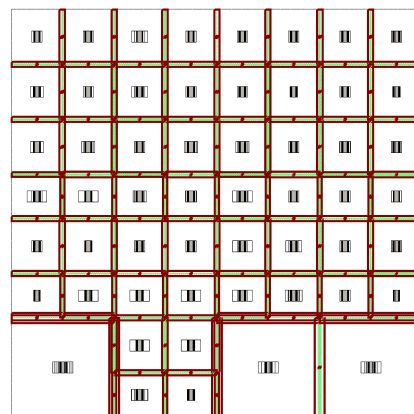


FIG. 1. Exemple de placement obtenu par quadri-partitionnement.

4. Présentation de Blizzard

Blizzard est un outil de placement analytique, basé sur fastPlace [3], encore en développement mais permettant déjà des comparaisons correctes.

Le but du placement analytique est de résoudre analytiquement l'ensemble du problème de placement. Pour cela, on modélise le problème sous forme quadratique convexe, tel qu'expliqué dans [3] : l'ensemble des interconnexions est modélisé par une matrice qui décrit les liaisons entre instances et deux vecteurs décrivent les composantes x et y des points fixes.

On obtient alors un système d'équations linéaires dans lequel les inconnues sont les coordonnées en x et y des instances. Ce système peut être résolu grâce aux méthodes traditionnelles de résolutions de systèmes linéaires.

La simple résolution du système d'équations linéaires n'est pas suffisante : dans le placement résultant, les instances ne sont pas correctement réparties sur l'ensemble de la surface du circuit. Il faut donc à la fois chercher à réduire la longueur des fils mais aussi répartir le plus possible les instances sur toute la surface. Les techniques utilisées sont décrites dans [3].

La figure 2 présente un exemple de résultat de placement analytique.

Pour pouvoir comparer correctement les deux algo-

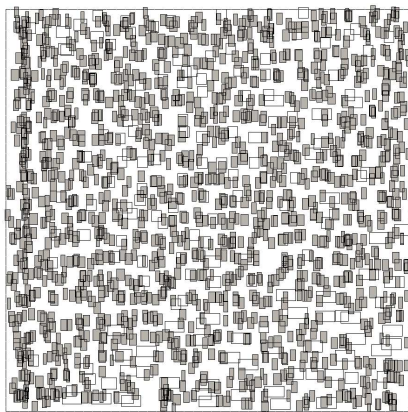


FIG. 2. Exemple de placement obtenu par placement analytique.

rièmes, on ramène le placement analytique à un placement comparable à celui fourni par le quadri-partitionnement. Pour cela, on construit une grille de quadri-partitionnement (cf figure 3) sur le placement et on regroupe les instances au centre des cases (cf figure 4).

5. Comparaisons

Les deux algorithmes de placement testés fournissent un placement global sur une grille de quadri-partitionnement avec les instances regroupées au centre des cases.

Pour avoir une bonne estimation de la longueur des fils, on procède, ensuite, à une phase de placement détaillé qui répartit les instances dans chaque case. Ce placement détaillé se fait par recuit simulé.

Du fait du regroupement qui suit le placement analytique, il peut y avoir une légère détérioration du placement

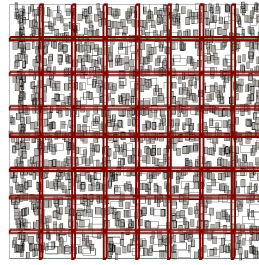


FIG. 3. Grille de quadri-partitionnement.

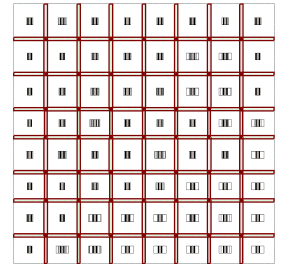


FIG. 4. Regroupement des instances.

puisque les instances ne seront pas placées de la même façon par le placement détaillé.

Pour ces tests, on utilise un circuit modélisant la partie de contrôle du microprocesseur MIPS R3000. Ce circuit est constitué de ~ 1700 instances.

5.1. Comparaison temporelle

Dans un premier temps, on cherche à comparer les temps d'exécution des deux algorithmes.

On mesure d'abord les temps d'exécution du placement analytique suivi du routage sur une grille de quadri-partitionnement de taille 8×8 . Puis, les temps cumulés des exécutions successives de quadri-partitionnement suivi du routage pour obtenir une grille de taille 8×8 .

On obtient les résultats suivants :

	Placement	Routage	Cumul
Blizzard	10,55	2,79	13,34
Metisse	3,02	4,34	7,36

TAB. 1. Temps comparés du placement / routage (en secondes).

En se référant au tableau 1, on constate que le placement par quadri-partitionnement est environ trois fois plus rapide que le placement analytique. Cet écart se réduit lorsque l'on prend en compte le temps de routage mais le quadri-partitionnement reste deux fois plus rapide.

	Initialisation placement détaillé	Placement détaillé	Cumul
Blizzard	1,09	5,79	6,88
Metisse	1,24	9,83	11,07

TAB. 2. Temps comparés du placement détaillé (en secondes).

En comparant le temps d'exécution du placement détaillé suivant les deux placements (cf tableau 2), on constate que ce placement détaillé est plus rapide s'il est exécuté après le placement analytique. Cette différence est tout à fait normale puisque le placement détaillé cherche à optimiser les nets locaux aux cases de quadri-partitionnement. Hors le placement analytique génère moins de nets locaux et donc l'exécution du placement

détaillé est plus rapide.

Au final, si on cumule les temps de placement, de

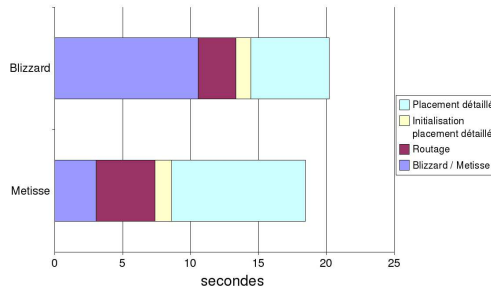


FIG. 5. Temps d'exécution comparés.

routage et du placement détaillé (cf figure 5), on constate que les temps d'exécution sont très proches :

- 20,22 secondes pour le placement analytique
- 18,43 secondes pour le quadri-partitionnement

Ces temps d'exécution sont relativement faibles, mais ils augmentent quasiment linéairement avec la complexité des circuits (cf figure 6).

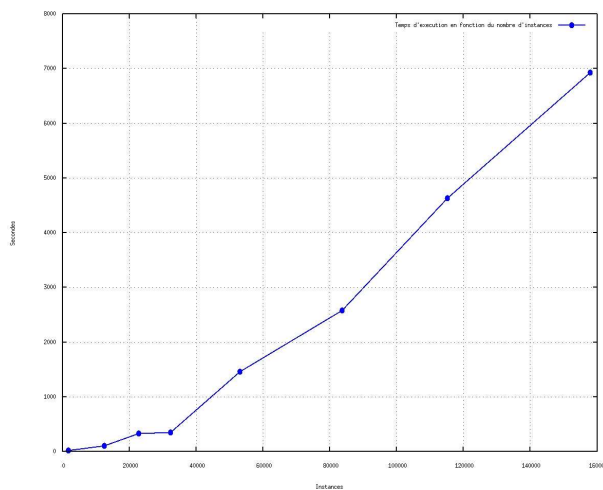


FIG. 6. Temps d'exécution en fonction du nombre d'instances.

5.2. Comparaison de la longueur des fils

Pour la comparaison des performances en termes de longueur des fils, on compare la somme des demi-périmètres des boîtes englobantes des interconnexions, la longueur moyenne des interconnexions et la longueur maximale des signaux. L'unité des longueurs n'étant pas significative pour la comparaison, elle n'est pas indiquée dans le tableau.

On obtient les résultats suivants :

	Demi périmètre	Longueur moyenne	Plus long signal
Blizzard	985 012	522	9 944
Metisse	905 105	444	9 319
% différence	8,83	17,57	6,71

TAB. 3. Résultats comparés pour la longueur des fils.

On constate que l'estimation de la longueur totale des fils est plus importante pour le circuit placé avec Blizzard de 9%. De plus, les résultats sur la longueur moyenne montrent bien que globalement Blizzard génère des signaux plus longs. Mais bien que les signaux soient plus longs, ils restent dans le même ordre de grandeur.

6. Conclusion

Le but de cette étude était d'évaluer les performances d'un algorithme de placement analytique dans un flot de conception VLSI.

Pour cela, on l'a comparé à un algorithme de placement par quadri-partitionnement. Les critères de comparaison choisis sont la vitesse d'exécution et l'estimation de la longueur des fils d'interconnexion.

Les résultats présentés montrent que, dans l'état actuel, l'algorithme de placement analytique est moins performant que le quadri-partitionnement. Mais l'ordre de grandeur des résultats est le même et l'implémentation de l'algorithme de placement analytique présentée ici est largement améliorable.

Références

- [1] Christophe Alexandre, Hugo Clément, Jean-Paul Chaput, Marek Sroka, Christian Masson, Rémy Escassut. *TSU-NAMI : An Integrated Timing-Driven Place And Route Research Platform*. DATE 2005.
- [2] George Karypis and others. *Multilevel hypergraph partitioning : application in VLSI domain*. DAC 1997, pages 526-529.
- [3] Natarajan Viswanathan, Chris Chong-Nuen Chu. *FastPlace : efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model*. ISPD 2004, pages 26-33.