

Optimisation du partage de blocs *BIST* pour le test des mémoires d'un circuit intégré

L. Zaourar, J. Alami Chentoufi, Y. Kieffer, A. Waserhole

Laboratoire G-SCOP, 46 avenue Félix Viallet, 38031 Grenoble CEDEX 1
{Lilia.Zaourar, Jihane.Alami-Chentoufi, Yann.Kieffer}@g-scop.inpg.fr

Mots-clés : *Optimisation multicritère, méta-heuristiques, micro-électronique, DFT, mémoires Bist*

1 Introduction

Les systèmes sur puce (SoC) actuels croissent de plus en plus vite et sont de plus en plus complexes. Dans ce contexte, l'objectif général du test est d'assurer que le composant électronique mis sur le marché donnera satisfaction au client en termes de fonctionnalités et de fiabilité. Bien sûr, plusieurs niveaux de test existent et sont proposés en fonction de la nature du circuit et de sa future utilisation sur le marché.

Cette évolution conduit à une augmentation importante de la part du test dans le coût de revient du circuit. En effet, de nos jours, le coût de test d'un circuit intégré est devenu, d'une manière générale, de plus en plus important au point même de surpasser celui de la conception du circuit lui-même. Dès lors, il devient primordial d'optimiser les techniques de test pour être plus rentable. Afin d'assurer cette optimisation, il est important de considérer l'insertion des techniques de test très tôt dans le flot de réalisation d'un circuit dès les premières étapes de conception. Pour cela les techniques de conception en vue du test, (*DFT* pour Design For Test) sont apparues.

Plusieurs problèmes d'optimisation et d'aide à la décision découlent de la micro-électronique. La plupart de ces travaux traitent des problèmes d'optimisation combinatoire pour le placement et routage des circuits [1]. Nos travaux de recherche sont à un niveau de conception plus haut, la *DFT* : citons par exemple l'optimisation de l'insertion des chaînes de scan pour le test des circuits [2,3], ou l'insertion de blocs *BIST* (*Built In Self Test*) pour le test des circuits particuliers que sont les mémoires [4].

Dans ce travail, on s'intéresse à ce dernier problème. Après une brève introduction et description du problème industriel, nous présentons la modélisation ainsi qu'un algorithme de résolution.

2 Présentation du problème industriel

Les systèmes sur puce contiennent un grand nombre de mémoires. Elles représentent la grande majorité des dispositifs embarqués. En effet, les mémoires occupent très couramment plus de la moitié de la surface des SoC actuels. L'association industrielle des semi-conducteurs (ITRS) indique que ce ratio devrait dépasser 90% d'ici dix ans.

Du fait de la structure particulière des mémoires, une des solutions la plus utilisée actuellement pour tester les mémoires est la méthode *BIST* [5]. Il s'agit d'une méthode de test dans laquelle une partie du circuit est utilisée pour tester le circuit lui-même. Pour cela, il faut rajouter autour des mémoires un ensemble de circuiteries nécessaire au test, c'est ce qu'on appelle bloc *Bist* : *Colliers* et *Contrôleurs*.

L'objectif de nos travaux est de développer des solutions algorithmiques permettant d'optimiser l'insertion de bloc *BIST* pour le test des mémoires. Ces travaux ont été réalisés en collaboration avec la société ST-microelectronics.

L'insertion du bloc *BIST* consiste plus précisément à rajouter à l'architecture d'origine du circuit un *Contrôleur* et des *Colliers*. Le *Contrôleur* contrôle les instructions pour l'ensemble de l'architecture. Chaque mémoire du circuit est pilotée par un *collier* qui a pour rôle d'exécuter les instructions données par le *Contrôleur*.

La résolution de ce problème s'avère facile lorsque le système *BIST* contient un nombre réduit de mémoires et que l'on peut opter pour une architecture dédiée, c'est-à-dire que chaque mémoire est pilotée par un *Collier* qui lui est propre. Néanmoins, de nos jours, la taille des circuits entraîne, pour cette stratégie, une augmentation trop importante de la surface, d'où l'idée de partager des colliers au sein d'un groupe de mémoires compatibles.

Pour les architectures des circuits considérés, on distingue deux types de partage de colliers: colliers séries, pour le test en séquence de mémoires et colliers parallèles, pour un test simultané de mémoires.

Le partage parallèle de colliers permet de réduire la surface ainsi que le temps de test. Ce type de partage est néanmoins très coûteux en termes de consommation de puissance. Tandis que le partage séquentiel réduit le taux de surface et la puissance au détriment du temps de test.

Une solution au problème du *BIST* consiste à définir un *contrôleur* pour l'ensemble du système et des *colliers*, pour une mémoire ou un ensemble de mémoires, minimisant simultanément les trois objectifs : surface, puissance de test et temps de test. Par conséquent, on est confronté à un problème d'optimisation multi-objectif. Dans ce contexte, il est impossible de parler de solution optimale. On cherche alors à identifier un ensemble de solutions de compromis entre les trois objectifs.

La modélisation ainsi que la solution obtenue doivent être indépendants de la technologie. En effet, les circuits sont conçus par génération de nœud technologique (65nm, 45nm, 32nm...) qui correspondent à la finesse de gravure du circuit. Les générations technologiques changent en moyenne tous les 18 mois, il est alors essentiel de concevoir des modèles indépendants de la technologie.

La modélisation de ce problème ainsi que la méthode de résolution sont présentées dans la section suivante.

3 Modélisation et résolution

Le problème d'optimisation du système de mémoires *BIST* peut être formulé de la façon suivante: la donnée est un groupe de mémoires, définies par leur type et un ensemble de paramètres, ainsi que des règles de partage des colliers en série et en parallèle. Il s'agit d'identifier des solutions au problème en associant à chaque mémoire un collier. La solution obtenue doit minimiser à la fois la surface, la puissance dynamique et le temps de test.

Pour résoudre ce problème, nous avons conçu un système automatique qui permet de rechercher un ensemble de solutions valides. Il est constitué de deux phases de résolution. La première phase consiste à créer des groupes de compatibilité de mémoires en tenant compte des règles de partage et d'abstraction des technologies utilisées. La deuxième phase implémente un algorithme d'optimisation multi-objectif sur ce modèle.

Nous présentons une méthode de résolution utilisant les algorithmes génétiques basés sur la notion de dominance de Pareto [6].

Des résultats prometteurs ont été obtenus pour différents jeux de données de complexités variables. Le nombre de mémoires dans le circuit varie entre 100 et 2000 mémoires. Pour les grandes instances, le temps d'exécution est de l'ordre de quelques minutes.

Les échanges avec l'industriel montrent que les résultats obtenus sont satisfaisants en termes de qualité des solutions et de temps d'exécution.

Références

- [1] A.Schrijver, L.Lovasz, B. Korte, H.J. Promel, and R. L. Graham . *Paths, Flows, and VLSI Layout*. Springer-Verlag, 1990.
- [2] L. Zaourar, Y. Kieffer, *Chaînage des sommets d'un graphes pour le test des circuits intégrés*. Roedef 2009, 10ème congrès de la société Française de Recherche Opérationnelle et d'Aide à la Décision, France (2009).
- [3] L. Zaourar, Y. Kieffer, C. Aktouf, *A Practical Scan Optimization Algorithm at the Register Transfer Level*. Design Automation and Test in Europe Proceeding, France (2009).
- [4] Y. Kieffer, J. Alami Chentoufi, L. Zaourar, *Automatizing the sharing of Bist blocks for low power testing of embedded-memories*. 23rd European Conference on Operational Research EURO, Allemagne (2009).
- [5] C. Landrault. *Test de circuits et de systèmes intégrés*. Les éditions Hermès, 2004.
- [6] M. A. Abido, *Multiobjective Evolutionary Algorithms for Electric Power Dispatch Problem*. IEEE Transactions on Evolutionary Computation, vol. 10, no. 3, pp. 315-329, June 2006.