OPTIMISATION DES TESTS DE PRODUCTION POUR LES CIRCUITS ANALOGIQUES AVEC PRISE EN COMPTE DES TOLÉRANCES

Abdelhakim KHOUAS et Anne DERIEUX

Département ASIM, laboratoire LIP6, université Pierre et Marie CURIE
4, place Jussieu, 75252 Paris Cedex 05
tel : (33) 1 44 27 71 23 , fax : (33) 1 44 27 72 80
e-mail: Abdelhakim.Khouas@lip6.fr
Anne.Derieux@lip6.fr

RÉSUMÉ : Le test des circuits analogiques nécessite un grand nombre de vérifications. Toutes les spécifications fonctionnelles doivent être verifiées et cela peut engendrer un temps de test prohibitif sur des équipement de test très coûteux. L'objectif de notre travail est de minimiser automatiquement le coût du test de production tout en tenant compte des tolérances sur les composants du circuit et des variations des paramètres du processus de fabrication.

I – INTRODUCTION

Le test des circuits analogiques nécessite un grand nombre de vérifications. Toutes les spécifications fonctionnelles doivent être verifiées et cela peut engendrer un temps de test prohibitif sur des équipement de test très coûteux [?]. L'objectif de notre travail et de minimiser automatiquement le coût du test de production tout en tenant compte des tolérances sur les composants du circuit et des variations des paramètres du processus de fabrication.

Avant la phase d'optimisation des tests, on effectue une phase de simulation de fautes pour déterminer les fautes qui peuvent être détectées par chaque test. Le problème avec les simulateurs électriques, est que les résultats de simulation ne sont pas précis. Le simulateur effectue des simulations avec des valeurs nominales des composants sans tenir compte des variations de ces valeurs duent aux fluctuations des paramètres du processus de fabrication (voir figure ??). L'optimisation et la génération des tests pour les circuits analogiques a fait l'objet de plusieurs publications ces dernieres annnées, [?], [?] et [?] pour les circuits lineaires, [?] et [?] pour les circuits mixtes, [?] et [?] pour la génération avec prise en compte des déviations sur les valeurs des composants.
Pour la simulation de fautes avec prise en compte des tolérances, il est necéssaire de déterminer les distributions des valeurs simulées pour le bon circuit et les circuits fautifs. L'évaluation exacte des distributions nécessite des simulations monte-carlo coûteuses en temps de simulation. Le but de notre approche est d'éviter le plus possible d'effectuer des simulations monte-carlo qui peuvent être inutiles, comme dans la majorité des cas de fautes catastrophiques.

II – OPTIMISATION DES TESTS AVEC PRISE EN COMPTE DES TOLÉRANCES

II–1 – Notre approche

Soit la distribution Dc d'une valeur de sortie d'un circuit correct, et Df la distribution de la même sortie pour un circuit fautif. On suppose que chaque distribution est définie par une valeur moyenne μ et un écart type σ. En fonction de la précision que l'on souhaite avoir, on peut définir un seuil S en fonction de σ_c et σ_f et on peut dire que :

$$(|\mu_c - \mu_f| > S) \Longrightarrow La\ Faute\ est\ detectable \tag{1}$$

Dans notre cas, on prendra $S = 2(\sigma_c + \sigma_f)$.
Soit V_f une valeur simulée du circuit fautif, on a $V_f \in [\mu - 2\sigma_f, \mu + 2\sigma_f]$, si on suppose que $(\sigma_c = \sigma_f)$ alors d'après la proposition (**??**) on déduit :

$$(|\mu_c - V_f| > 6\sigma_c) \Longrightarrow La\ Faute\ est\ detectable \tag{2}$$

La proposition (**??**) montre que dans le cas où la valeur simulée V_f d'un circuit fautif vérifie la condition $(|\mu_c - V_f| > 6\sigma_c)$, alors on peut tout de suite et sans évaluer la distribution de la valeur fautive, déduire que la faute en question est détectable.

II–2 – Algorithme de simulation

L'algorithme de simulation avec prise en compte des tolérances est le suivant :

pour *chaque test* $T_i,\ i \in [1..n]$ **faire**
    évaluer μ_c *et* σ_c *pour le circuit correct*
    pour *chaque faute* $F_j,\ j \in [1..m]$ **faire**
        Nstat = 0 ;
        FinSim = 1 ;
        DD_{ij} *= 0 ;*   *// initialisation*
        tant que *((FinSim == 1) et (Nstat < MAXstat))* **faire**
            effectuer un tirage et une simulation ;
            évaluer Vf_{ij} *du circuit fautif ;*
            si $(|\mu_c - Vf_{ij}| > 6\sigma_c)$ **alors**
                DD_{ij} *= 1 ;*
                FinSim = 0
            sinon
                Nstat ++
                sauvegarder Vf_{ij}
            fin si
        fin tant que
        si *(*DD_{ij} *== 0)* **alors**
            évaluer μ_f *et* σ_f *du circuit fautif*
            $I_c = [\mu_c - 2\sigma_c\ ,\ \mu_c + 2\sigma_c]$
            $I_f = [\mu_f - 2\sigma_f\ ,\ \mu_f + 2\sigma_f]$
            $DD_{ij} = \frac{(I_f - I_c)}{(I_f \cap I_c) + (I_f - I_c)}$;
        fin si
    fin pour
fin pour

Cet algorithme est basé sur la proposition (**??**). Pour chaque test T_i, on évalue $\mu_c\ et\ \sigma_c$ du circuit correct, et pour chaque faute F_j on effectue un premier tirage aléatoire et une simulation électrique pour calculer Vf_{ij}. Si Vf_{ij} vérifie la proposition (**??**), alors la faute est détectable (DD_{ij} = 1) et on passe à la faute suivante, sinon on sauvegarde la valeur, on effectue un autre tirage et on refait les mêmes étapes. Si après MAXstat tirages (MAXstat $\in [3 \cdots 30]$), la proposition (**??**) n'est pas vérifiée. alors on évalue la moyenne μ_f et l'écart type σ_f de la distribution des MAXstat valeurs simulées. Le coefficient DD_{ij} définit le degré de détectabilité de la faute F_j par le test

T_i. On remarque que plus la valeur de DD_{ij} est proche de 1 plus la faute peut être détectable, Si (DD_{ij} = 1) alors la faute est détectable et si (DD_{ij} = 0) alors la faute n'est pas détectable.
A la fin de la simulation de fautes on obtient une matrice des degrés de détectabilité DD_{ij}, Les lignes $T_1 \cdots T_n$ de la matrice représentent les n tests et les colonnes $F_1 \cdots F_m$ représentent les m fautes.

II–3 – Optimisation des Tests

La méthode d'optimisation consiste à utiliser la matrice des degrés de détectabilité DD_{ij} donnée par le simulateur de fautes pour ordonnancer les tests selon leur capacité à détecter les fautes, pour cela nous avons défini un taux de couverture TC_i de chaque test, ce taux de couverture est donné par:

$$TC_i = \frac{\sum_{j=1}^{m} DD_{ij}}{m} \tag{3}$$

On remarque que plus le taux de couverture TC_i est grand, plus le test T_i détecte un grand nombre de fautes. Ainsi, ces taux de couverture TC_i sont utilisés pour classer les tests suivant le nombre de fautes qu'ils détectent, et supprimer les tests qui ne détectent pas de fautes.
Si on suppose que l'ensemble des tests optimisés est un ensemble à l tests alors le taux de couverture global TC est donné par:

$$TC = \frac{\sum_{j=1}^{m}(Max\{DD_{kj}/k \in [1..l]\})}{m} \tag{4}$$

Le terme $Max\{DD_{kj}/k \in [1..l]\}$ détermine pour chaque faute F_j le meilleur degré de détectabilité.

III – RESULTATS

Pour valider notre approche, nous avons utilisé un amplificateur opérationnel (AOP) CMOS à 2 étages de la figure **??**. Nous avons supposé que tous les paramètres (W et L des transistors et les capacités) ont des tolérances de 5%. Nous avons considéré 25 fautes, 5 paramétriques, 10 courts circuits et 10 circuits ouverts. Nous avons effectué un test AC entre $0.5Hz$ et $5E6Hz$ avec 8 fréquences, des simulations du gain et de la phase ont été effectuées. Comme ce sont les fautes paramétriques qui sont les plus difficiles à détecter, pour illustrer nos résultats nous avons choisi les 5 fautes paramétriques données dans le tableau **??**.
Les tableaux **??** et **??**, représentent les matrices des degrés de détectabilité en considérant le gain et la phase respectivement de la sortie S des 5 circuits fautifs. D'après le tableau **??**, pour le test en gain l'ensemble du test peut être réduit au test 4 (fréquence = $5E2Hz$) avec un taux de couverture $TC = TC_4 = 4.45/5 = 86\%$. Pour le test en considérant la phase, d'après le tableau **??**, nous remarquons un meilleur taux de couverture. toutes les fautes sont détectables, et l'ensemble de test peut être réduit au test 7 suivi du test 5 car $TC_7 > TC_5$. Le taux de couverture pour le test en phase est $TC = 100\%$.
Pour les 25 fautes, le gain en temps de simulation par rapport à la simulation Monte Carlo avec 30 tirages est de l'ordre de $5,6$ avec la même précision sur les résultats. Le gain en temps CPU de notre méthode dépend des fautes simulées, Pour les fautes catastrophiques le gain est meilleur que pour les fautes paramétriques, en effet les variations dues aux fautes catastrophiques sont plus importantes que celles dues aux fautes paramétriques.

IV – CONCLUSION

Nous avons présenté une nouvelle méthode d'optimisation des tests de production pour les circuits analogiques. Dans cet article, nous avons décrit notre approche basée sur la théorie statistique et nous l'avons appliqué sur un amplificateur operationnel CMOS à deux étages.
Actuellement, nous travaillons sur la réalisation d'un outil automatique permettant l'injection de fautes, la simulation de fautes avec prise en compte des tolérances et l'optimisation des tests de production pour les circuits analogiques.

RÉFÉRENCES

[1] Linda Milor and A. L. Sangiovanni-Vincentelli. Minimizing Production Test Time to Detect Faults in Analog Circuits. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 13(6), 1994.

[2] C.-J. Richard Shi and Michael W. Tian. Automatic Test Generator of Linear Analog Circuits Under Parameter Variations. In *Proc. Asian and South Pacific Design Automation Conference*, 1998.

[3] Naveena Nagi, Abhijit Chatterjee, Ashok Balivada, and Jacob A. Abraham. Fault-Based Automatic Test Generator for Linear Analog Circuits. In *Proc. IEEE International Conference on Computer-Aided-Design*, pages 88–91, 1993.

[4] S. Mir, M. Lubaszewski, and B. Courtois. Fault-Based ATPG for Linear Analog Circuit with Minimal Size Multifrequency Test Sets. *Journal of Electronic Testing, Kluwer Academic Publishers*, 1996.

[5] H. Kerkhoff, R. Tangelder, H. Speek, and N. Engin. MISMATCH: A Basis for Semi-Automatic Functional Mixed-Signal Test-Pattern Generation. In *Proc. IEEE International Conference on Electronics, Circuits, and Systems*, pages 1072–1075, 1996.

[6] Bechir Ayari, Naim BenHamida, and Bozena Kaminska. Automatic test vector generation for mixed-signal circuits. In *Proc. European Design and Test Conference*, pages 458–463, March 1995.

[7] Zwolinski M. Spinks S.J., Chalk C.D. and Bell I.M. Generation and Verification of Tests for Analogue Circuits Subject to Process Parameter Deviations. In *IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems, Paris, France*, October 1997.

[8] Michael W. Tian and C.-J. Richard Shi. Efficient DC Fault Simulation of Nonlinear Analog Circuits. In *Proc. Design And Test in Europe*, 1998.

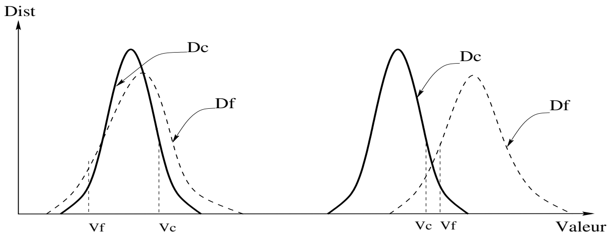

Figure 1: Comparaison des résultats de simulation avec et sans tenir compte des distributions.

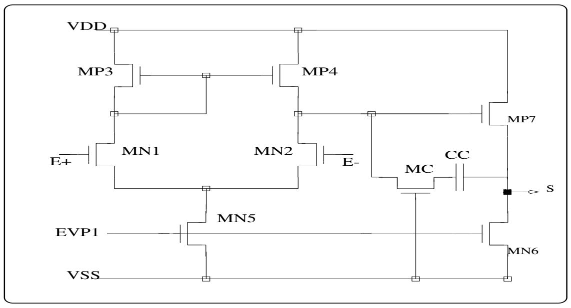

Figure 2: *AOP CMOS à 2 étages.*

faute no	description
1: MN2	modification de W, $W = W * 0.5$
2: MP3	modification de L, $L = L * 0.5$
3: MP4	modification de W, $W = W * 0.5$
4: MC	modification de L, $L = L * 0.5$
5: CC	modification de CC, $CC = CC * 0.5$

Tableau 1: *Description des fautes utilisées pour dans les tableaux* **??** *et* **??**.

test no	freq (Hz)	faute no 1	2	3	4	5	TC %
1	0.5	1	1	1	.35	.35	74
2	5.0	1	1	1	.35	.35	74
3	5e1	1	1	1	.33	.51	76
4	5e2	1	1	1	.45	1	89
5	5e3	1	1	1	.43	1	88
6	5e4	1	1	1	.43	1	88
7	5e5	1	1	1	.43	.83	85
8	5e6	1	1	1	.42	.46	77

Tableau 2: *Tableau des DD_{ij} en considérant la mesure du gain de l'AOP.*

test no	freq (Hz)	faute no 1	2	3	4	5	TC %
1	0.5	1	1	.33	.19	1	70
2	5.0	1	1	.13	.20	1	66
3	5e1	1	1	0	.17	1	63
4	5e2	1	1	0	.08	1	61
5	5e3	1	1	.43	.08	1	70
6	5e4	1	1	1	.81	.47	85
7	5e5	1	1	1	1	.50	90
8	5e6	1	.43	1	1	.60	80

Tableau 3: *Tableau des DD_{ij} en considérant le mesure de la phase de l'AOP.*