

動的クロストーク解析手法の改良と評価

Improvement and Evaluation of Dynamic Crosstalk Analysis Method

丸岡 翔 豊永 昌彦 村岡 道明

Sho Maruoka Masahiko Toyonaga Michiaki Muraoka

高知大学 理学部 応用理学科 情報科学コース 村岡研究室

1. まえがき・背景

近年の半導体の微細化技術の進歩に伴う集積回路の増大と微細化により、クロストーク問題が顕著になってきた。クロストークとは、信号変化の際に配線間容量を介して隣接する配線の信号にノイズが現れる現象のことであり、信号遅延を増減させ、回路の誤作動を引き起こす場合もある。

従来のクロストーク解析手法は静的なタイミング解析手法を使用していたが、多くの擬似エラーを含む問題点があった。研究室では、テストパターンを用いて動的なクロストークのみを解析する手法によりこの問題を解決する方法を提案し、そのプロトタイプを開発したが、解析に時間がかかる、回路制限があるなどの問題が発生していたため、大規模回路での実験は困難だった。

2. 研究目的

従来のプログラムや手法の問題点を洗い出して、解析に時間がかからない、ファイルサイズが拡大したとしても分割しなくても良いような、回路規模の制限などを極力無くすような解析手法の改良、それに沿ったプログラムの設計、開発を行う。

3. 提案手法の改良

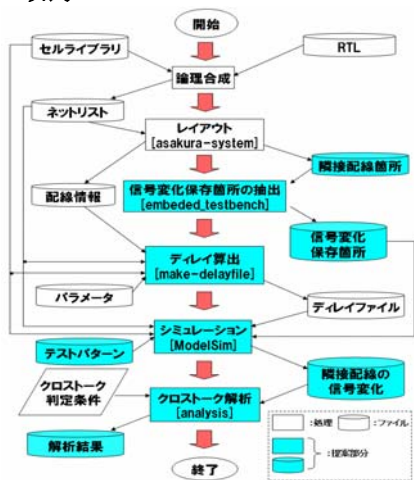


図1. 動的クロストーク解析手法のフロー図

図1の中の提案手法の中の「信号変化保存箇所の抽出」「ダイレイ算出」「クロストーク解析」の部分について改良を行った。どのような改良を行ったかを以下に示す。

— 先行研究では処理の遅いインタープリタ言語である perl で実装されていたプログラムを、動作の速いコンパイラ言語である C 言語で実装した..

— 先行研究では要素を取り込む配列を静的に用意していたので、回路が大きくなると対応できなくなっていたが、提案手法では大規模回路に対応できるよう、要素を大量に格納できるテーブルを動的に確保できるようにした。

4. 提案手法による大規模回路の評価

表1. 提案手法での動的クロストーク結果

回路	ビット数	セル数	隣接総配線数	静的CT数 (a)	動的CT数 (b)	動的CTの割合(b/a)
加算器	64	320	1,843	512	280	54.7%
	128	640	4,204	1,831	947	51.7%
	256	1,280	10,115	4,672	2409	51.6%
乗算器	16	1,488	3,698	189	73	38.6%
	32	6,048	21,342	1,410	568	40.3%
	48	13,680	45,214	3,990	1682	42.2%
	64	24,384	83,397	7,601	2667	35.1%
1コアCPU	2,108	5,470	716	102	14.2%	
2コアCPU	4,216	10981	1349	99	7.3%	
4コアCPU	8,432	23631	3325	194	5.8%	
8コアCPU	16,864	48839	8395	419	5.0%	

表1より、加算器は回路規模にかかわらず 50%の割合になり、乗算器も加算器よりは振幅が大きいのが 30%後半の割合になることが判明した。対して、マルチコア CPU では回路規模が拡大するにつれて動的クロストークの割合が減少することが判明した。

5. 従来手法と提案手法の比較

表2. 従来手法と提案手法の解析時間短縮率

回路	ビット数	解析時間[a] (先行研究)[秒]	解析時間[b] (提案手法)[秒]	解析時間短縮率[b/a]	高速化率 [a/b]
加算器	64	18	10	55.6%	1.8倍
	128	147	13	8.8%	11.3倍
	256	777	23	3.0%	33.8倍
乗算器	16	11	10	90.9%	1.1倍
	32	368	25	6.7%	14.7倍
	48	1,505	114	7.6%	13.2倍
	64	3,483	339	9.7%	10.3倍
8ビットCPU	19	12	63.2%	1.6倍	

表2より、加算器の64ビットは1.8倍までしか高速化できなかったがそれ以上のビットでは10倍以上高速化できた。乗算器も16ビットは1.1倍しか高速化できなかったがそれ以上のビットでは10倍以上高速化できた、8ビットCPUは1.6倍まで高速化できた。加算器の64ビットと乗算器の16ビットは高速化率が低いのが、これは解析時にプログラムに何らかのオーバーヘッドが起こっていると考えられる。

6. まとめ

今回、先行研究の改良を行い、それを用いて先行研究との比較、大規模の評価を行った。動的クロストークの割合は、加算器は50%程度、乗算器は30%後半、マルチコアCPUは14.2%から減少する傾向にあった。解析時間は、加算器と乗算器は10倍以上高速化できた。CPUは1.6倍しか高速化できなかったが他回路の傾向より、回路規模が拡大すれば高速化率は10倍以上になると予想できる。

7. 今後の課題

今後の課題としては解析時間の更なる短縮、更なる大規模な回路での評価、本手法の自動化というものが挙げられる。