

3E-5 論理合成後のクリティカルパス改善手法の構築

前田 寿

橋田 関志

山本 孝司*1

津口 公一*1

日本電気エンジニアリング (株)

北陸日本電気ソフトウェア (株) *1

1. はじめに

現在のLSI設計では、開発量の増大と設計期間の短縮のため、論理合成ツールを中心としたLSI設計が主流になっている。しかしながら、論理合成ツールの遅延収束機能には限界があり、高性能を実現するために最終的に論理合成手法を諦め、人手設計手法へ変更せざるを得ない場面がある。この様な、論理合成手法から人手設計手法への移行作業をスムーズに行うことが、LSI設計期間の短縮に繋がる。

2. 現設計手法の課題

超高速LSIでは、予め論理合成部と人手設計部とに分けて設計する。本論文の改善対象は論理合成部である。論理合成ツールの機能は近年格段に向上したが、遅延収束機能はRTL記述方法や遅延制約(=スクリプト)の内容に左右されやすく、ベストなRTL記述、スクリプト記述を見つけるまでは遅延解析を含めた繰り返し処理が必要である。これ以上の最適化は難しいという場合は、クリティカルパス部分を論理合成対象回路から一部切り出しを行い、その部分は人手により設計し直し、その後、論理合成回路とマージする手法が通常の対処方法である。具体的にはRTLの修正と論理合成の再実行、切り出したクリティカルパス部分の人手回路図登録が必要となる為、設計工数が増大する。

3. 解決案

解決策として合成後の遅延解析結果で、目標を満たしていないクリティカルパスが存在する場合、そのパスを特定した上で、部分的な回路図修正を容易に行えるシステムを構築する。具体的には、合成後のネットリストを回路図エディタで直接修正する方法である。一般的な設計フローではIPO等による遅延最適化も行われているが、今回実現したフローでは、論理ゲート簡略化等の論理修正も可能である。

回路図修正による遅延収束方法には以下の3方式を採用した。

- ・中継ゲートの挿入
- ・パワーゲートへの変換 (以上図1-aで説明)

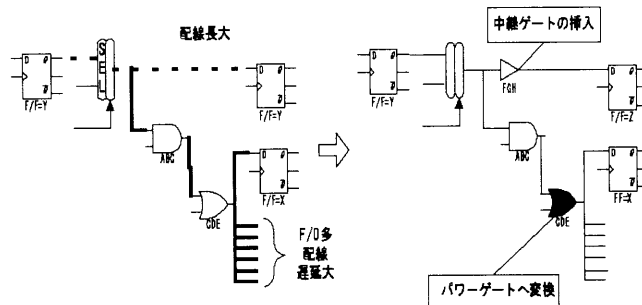


図1-a: 回路修正による配線遅延改善例

- ・段数削除などの論理構成変更 (図1-b)

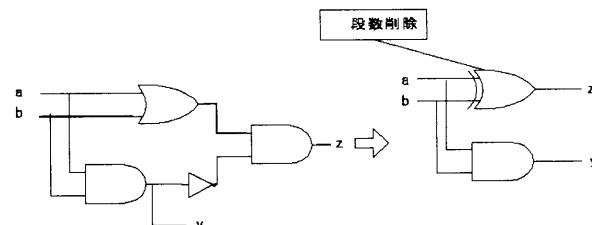


図1-b: 回路修正による遅延改善例

4. 統合ツール開発

LSI設計の為に必要なCADプログラムを集合させ、入力コマンドファイル作成や出力ファイルの管理、タイムスタンプを利用した実行制御などを自動的に行うLSI設計統合支援ツールを開発した。しかし、回路図エディット処理というインタラクティブな処理が発生する場合、統合ツールから一度離れて回路図修正作業を行い、再度統合ツールへ戻る必要があり、その場合下記の問題が発生する。

(1) 予測される問題点

- (a) インタラクティブ処理のために、ユーザ側で必要ファイルのコピーや移動、版数管理など人手による作業が必要になる。(図2-a, b)
- (b) ネットリストと遅延エラーファイルの版数の統一性を人手作業で保つ必要がある。(図2-c)
- (c) ネットリスト修正時のバグ混入に気付かないまま処理が進められる可能性がある。(図2-d)
- (d) 今回採用した回路図エディタはNT版であり、

一方、論理合成/最適化はUNIX版であるため、OSを越えたファイル管理が必要である。

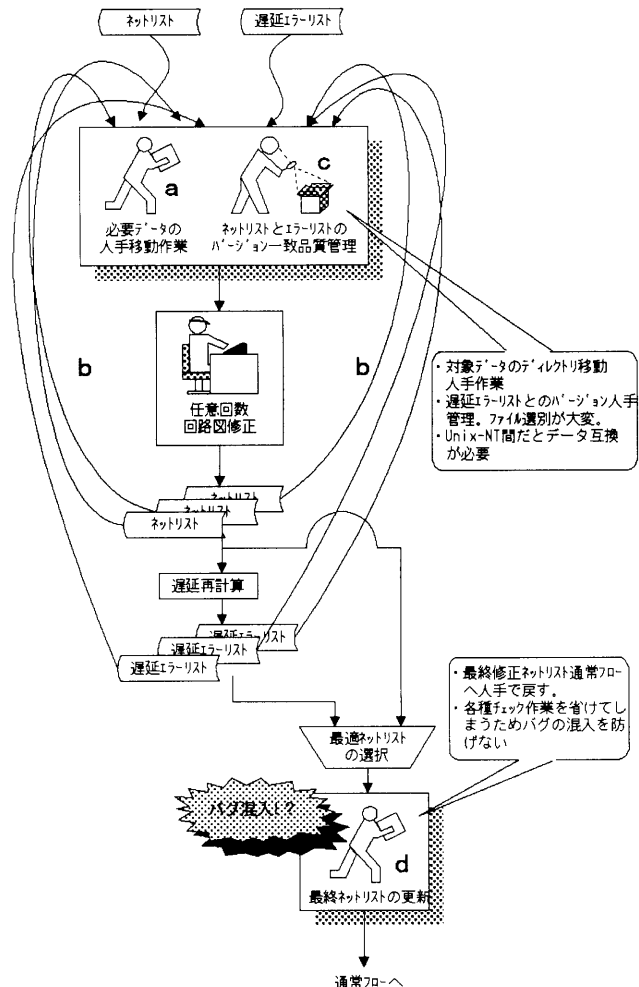


図2：予測される問題点

(2) 統合ツールへの追加機能

設計者が容易に作業を進められ且つバグ混入を最小限に押さえるため、以下の機能を実現した。(図3)

(a) 回路図エディタの立ち上げ機能
統合支援ツールから回路図エディタを立ち上げ、必要なデータを専用エリアへ自動コピーする。

(b) ネットリストの版数自動振り分け
設計者が修正を重ねる毎に版数を上げることを意識しなくて済むように、修正ファイルに的確な版数を自動的に振り分ける。

(c) ファイル更新の抑制
最終的な修正ネットリストはフォーマルベリフィケーションとルールチェックが実施されていないとファイル更新がされない抑制機能を追加。

(d) sambaによるデータ共有
UNIX上のデータをNT上で扱えるように、フリー

ソフトウェア samba を利用したデータ互換環境を構築した。

以上の機能追加により、図2で示すファイル管理の煩わしさやバグ混入の可能性を排除した。

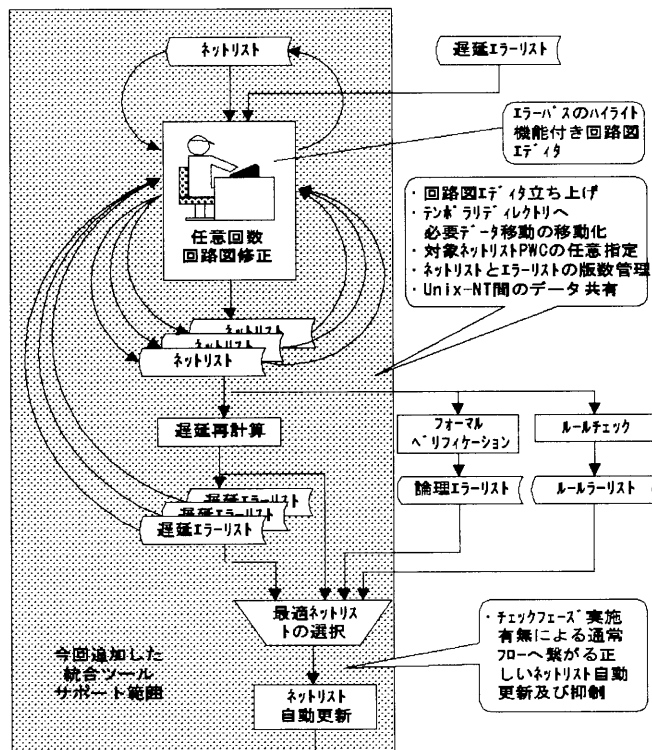


図3：機能追加による処理の流れ

4. 効果

以上説明した設計フローと統合ツールの提供により、以下の効果を得た。

- (1) マクロ切り出し工数の80%削減
- (2) 最適化繰り返し工数の90%削減
- (3) ファイル版数管理工数の80%削減

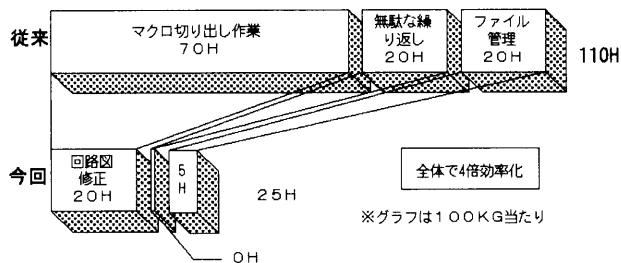


図4：設計工数比較

5. 今後の予定

今回説明したフローは、レイアウト処理前フェーズの仮想配線長予測による遅延の改善手法であるが、今後はレイアウト実施後の実配線長遅延による改善手法を検討したい。