

3 J-03 高速回路実現のための HDL 初期品質向上手法

川口 実、 島田 哲宏、 正垣 啓幸

山下 茂、 鈴木 孝昭

日本電気エンジニアリング (株)

1. はじめに

0.35 μm プロセスまでの LSI 設計手法は、論理合成で期待遅延値を満足する回路を生成することで、レイアウト後の遅延解析で問題が発生することは少なかった。より微細なプロセスでは、論理合成を繰り返し行い、仮想配線長で遅延が満足したにもかかわらず、レイアウト後の遅延が満足しないケースが増えており、レイアウトと遅延解析の繰り返し設計期間の増大を招いている。今回、論理合成及びレイアウト結果を考慮した回路の高速化を行い、高速回路作成のための設計手法改善策を得たので以下に述べる。

2. 現設計手法の問題点

従来、回路全体の遅延値は論理ブロックの遅延が大勢を占めていたが、0.25 μm 以下では配線遅延が大勢を占める傾向にある。そのため配線遅延を統計計算で求めている論理合成後の遅延結果とレイアウト後の遅延結果で遅延値の誤差が大きくなるため、論理合成時の遅延見積もりが不正確になっている。

3. 高速化回路作成検討、試行

問題点を解決するには、レイアウト後の実配線長での遅延結果を考慮した上で早期に HDL 修正に着手した方が遅延の収束が早いと推論した。今回以下の2通りの方法で高速回路を作成し、その実現手法について検討した。論理合成による作成回路の目標性能は 166MHz(クロック周期: 6ns)とし、合成ツールは Synopsys 社の Design_Compiler を使用した。

(1) 論理合成での高速化限界試行

今回の試行回路の論理合成による性能向上の限界を調査するため、複数種類の遅延重視の合成制約を

用意し、論理合成及びレイアウトを行った。比較のための遅延値はレイアウト後の遅延値を使用している。試行の結果、図1-④が一番高速回路になったが目標性能を達成出来なかった。また④が今回の試行回路の性能限界であり、これ以上の大きな性能向上は不可能と判断した。

合成制約	①	②	③	④
違反バスの割合 (>6ns)	16.97%	20.64%	6.50%	2.23%
最大遅延 (ns)	8.52	8.85	8.10	7.80
規模: ①を1とした場合の倍率	1	1.03	1.08	1.11

①CLOCK=5ns制約
②CLOCK=5ns制約+クリティカルレンジ=2ns
③CLOCK=2ns制約
④CLOCK=2ns制約+クリティカルレンジ=2ns

図1 合成による高速回路試行結果

(2) レイアウト結果からの HDL 書き換え試行

上記(1)の試行と平行してレイアウト後の結果を元に目標性能に達していない論理に対して HDL 修正を行い、再度合成及びレイアウトを行った。HDL 書き換えのための回路を生成する合成制約は、クロックに依存せず、高速回路を生成する図2-⑤の制約を使用した。制約⑤については、上記(1)項の最良結果④と同様な回路になることを確認済みである。また HDL 修正に際してはレイテンシの変更は行わないことを条件に作業を行った。

合成制約	違反バス割合	最大遅延
④	2.23%	7.80ns
⑤	2.32%	8.48ns

⑤の制約: CLOCK制約なし、
max_delay=0ns+クリティカルレンジ=10ns

図2 高速回路スクリプトでの論理合成結果

試行の結果、レイアウト後の HDL 修正により、高速で目標性能を満足する回路が作成できた。

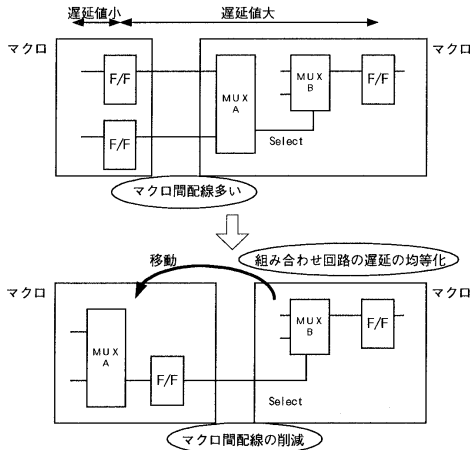


図3 HDL 修正例

4. 試行結果

3章の高速化の試行結果で得られた性能及び規模 (サイズ) の値を図4に示す。3章(2)のHDL修正を行うことにより、目標性能を達成出来た。

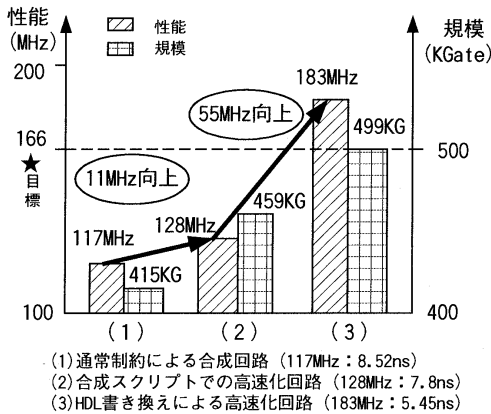


図4 高速回路作成試行結果

5. 考察

今回の高速化試行から得られた設計上の留意点を以下に述べる。

★早期レイアウト実施の重要性

HDLのコーディングが完了し、論理合成可能になった時点(論理FIXしていない状態でも)で、目標性能に回路が到達しているかチェックするために高速制約で論理合成を行い、仮のレイアウトを実施する(基本的に1回)。この時点で目標性能に達していない論理は、これ以上の改善が見られないと判断しHDLを修正する。これにより合成

及びレイアウトで遅延が収束できそうにない論理をフロントエンド処理の早期で発見可能となる。

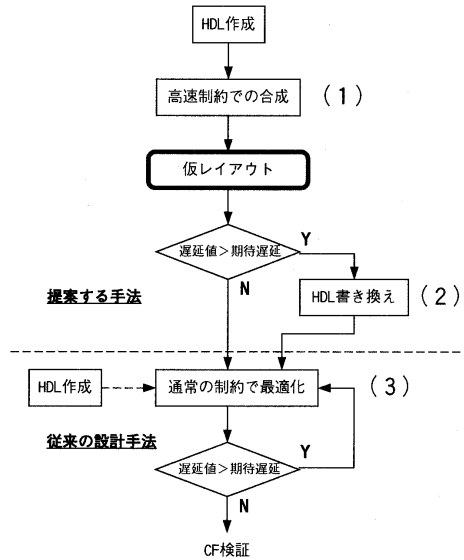


図5 仮レイアウト導入フロー

(1) 高速制約での最適化

合成ツールで最速回路を出力するには、複数回の試行工数を要するため、以下の制約で1回の合成及び最適化を行う。以下の制約で最速に近い高速回路を生成可能。(図2-⑤の制約)

- クロック制約なし、max_delay=0
- クリティカルレンジ=違反遅延を考慮した値

(2) HDLの修正

高速回路でレイアウトを行った後、性能未達パスが出た場合は、今後合成ツールでの遅延値収束が見込めないと考え、未達パスのHDL修正による高速化を行う。

(3) 通常のクロック制約での最適化

従来設計と同じ方法の制約で合成を実施する。その際、仮想配線長計算でのクリティカルパスがレイアウトでクリティカルパスになるとはかぎらないため、隠れているパスのことを考えてクリティカルレンジは大きめに設定する。

参考文献

Synopsys : Design Compiler Version1999.05
リファレンスマニュアル
CONSTRAINTS AND TIMING