

Varchsyn(6): タイミング解析

6N-6

河内 敏彦 山内 尚 岸本 有豊 前田 直孝

NEC

1. はじめに

近年、VLSIの設計では論理合成が広く用いられている。論理合成システムでは、回路の遅延/面積を最小にするために何回も繰り返し合成を行い、そのたびに遅延解析を行う。従って、高速で詳細なタイミング解析が必要になる。

論理合成では、合成された回路がタイミング制約条件を満たさない場合、再合成を行う必要がある。最も効果的な合成方法は回路の最悪部分の改善である。最悪部分というのはクリティカルパスが最も多く通過する部分であると考えられる。最悪部分を特定するためには、タイミング解析で一度に多数のクリティカルパスを検出する必要がある。

この要求から、一度のトレースで多数のクリティカルパスを検出可能な新しいタイミング解析手法を開発し、さらに、インクリメンタル処理により高速化を実現した。

2. アルゴリズム

ここで提案する新しいタイミング解析手法の特徴は、従来の手法[1]と異なり、プライマリ入力毎にクリティカルパスを検出する場合でも、各ゲートに対し一度だけトレースをすればよいということである。ここではまず最初に新手法の基本アルゴリズムを述べ、その後でインクリメンタル処理について述べる。

新手法では、各ゲート毎に遅延値とゲートの入力端子番号を入れるテーブルを用意する。このテーブルには、各プライマリ入力からそのゲートまでの遅延値と、各プライマリ入力からのクリティカルパス上の入力端子番号を蓄える。従って、このテーブルのアイテム数はプライマリ入力の数に等しくなる。このテーブルを出力端子テーブルと呼ぶ。

図1に基本アルゴリズムを示す。第一段階では、各ゲートのトレース入力数(TIN)を0に初期化する。TINはそのゲートの入力ピンのうちパスが通った入力ピンの数を表す。第二段階では、評価すべきゲート出力端子の第一番目のグループとして、全プライマリ入力をカレントイベントテーブル(CET)に登録する。第三段階では、処理はCET上のゲート出力のファンアウト先のノードに移る。ファンアウト先のゲートでTINをインクリメントする。このとき、もしこのゲートのTINがその入力端子数に等しくなったら、処理は次の段階に進む。これはそのゲートの全ての入力端子が評価されたことを意味する。もし等しくなれば、処理は第五段階以降に移る。第四段階では、ゲートの出力端子でゲートとネットの遅延値を加え、各プライマリ入力からの遅延値を計算する。もし二つ以上のファンインゲートが同じプライマリ入力からのパスをもつとき、最悪入力端子番号と最悪遅延値を、比較後に登録する。第五段階では、評価が完了したゲートをネクストイベントテーブル(NET)に登録する。CET上のゲート出力の評価が全て終了し、NETにアイテムが残っていなければ、パストレースは完了する。そうでなければ、NETへのポインタをCETへのポインタと交換し、処理は第三段階に戻る。

要求時間を満足しなければ、論理合成プログラムはパス情報

Varchsyn(6): Timing Analysis

T. Kawachi, H. Yamauchi, A. Kishimoto, N. Maeda

NEC Corporation

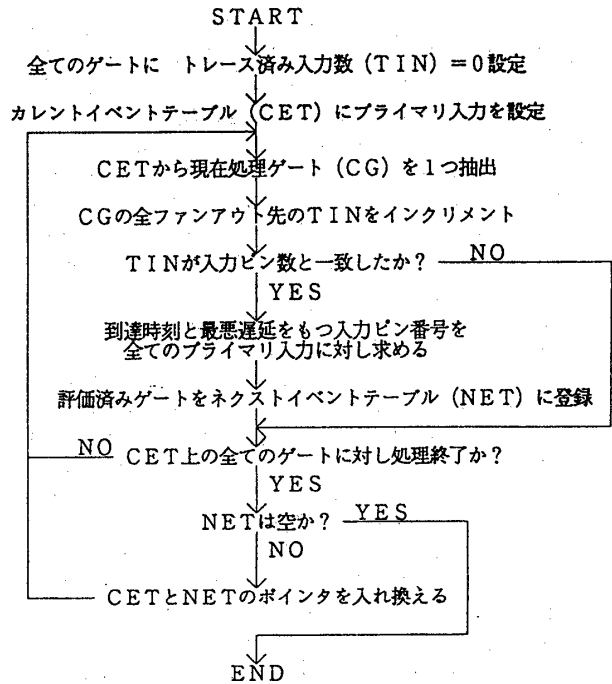


図1 基本アルゴリズム

を必要とする。この情報は、プライマリ出力から後方に最悪ゲートをトレースして決定する。各ゲートは全プライマリ入力に対する最悪入力端子をもっているため、全プライマリ入力に対するパス情報が別々に得られ、再合成によって改善すべき最悪部分が検出される。もし全ての要求時間を満たしていれば、この処理は行わない。

図2に例を示す。この図中のテーブルはそのゲートまでの到達時間と、最悪値を与えるゲート入力端子を表す。3つのカラムは、それぞれ、テーブルの上から順に、I1, I2, I3に対応する。各ゲートの入力端子番号は各ゲートの図で上から昇

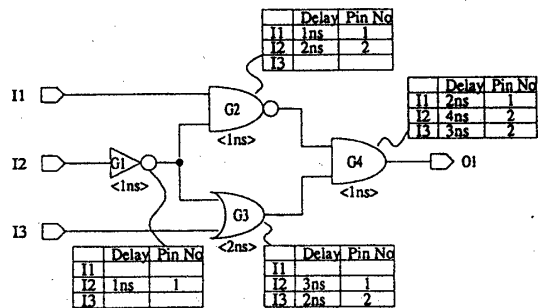


図2 新タイミング解析手法例

順に割り当てる。説明を簡単にするため、配線遅延は無視する。まず最初に、I1, I2, I3を開始ノードとして登録する。次に、プライマリ入力ファンアウトを評価する。このとき、各ゲートのTINは、G1, G2, G3で1, G4で0である。TINが入力端子数に等しいノードは処理を先に進める。従っ

て、G1のみ処理を先に進める。G1はI2からのパスしかもたないで、I2のカラムだけ評価を行う。G1は1つしか入力がないので、最悪の入力端子番号は1で、遅延値の計算結果は1nsになる。次に、G2とG3に対して処理を先に進める。G2のファンインであるI1とG1はI1とI2からのパスをもっている。遅延値の計算はI1, I2に対して行う。結果は順に1ns, 2nsである。最悪の入力端子番号はI1に対して1, I2に対して2となる。同じ操作をG3に対しても行う。遅延計算の結果は、I2から3ns, I3から2nsになる。最悪の入力端子番号はI2に対して1, I3に対して2となる。次に、G4の評価を行う。I1からのパスはG4の1番目の入力端子に、I3からのパスはG4の2番目の入力端子に到達する。しかし、I2からのパスは1番目と2番目の両方の入力端子を通してG4に達する。この場合、入力端子の遅延値の比較によって、2番目のものがクリティカルな入力端子となり、遅延値は4nsとなる。トレースはプライマリ出力端子O1に到達したときに終了する。もしI2に対するクリティカルパスを検出したければ、テーブルの第2カラムに対する入力端子番号を後方にトレースして求める。この場合は、O1-G4-G3-G1-I2が、クリティカルパスになる。

もう一つの主要な高速化手法はインクリメンタル処理である。論理合成プログラムは回路の性能を上げるため、小部分を切り出し、何回も変更する。しかし、回路のほとんどの部分は再合成の後まったく変更を受けない。インクリメンタル処理では、変更を受けない部分に対し、以前のタイミング解析の結果を可能な限り利用することにより、高速化を計る。

論理合成のマッピング処理はアップデートゲートとスタートゲートをタイミング解析処理に渡す。アップデートゲートとは変更されたゲートである。スタートゲートとはアップデートゲートの直前に位置し変更を受けていないゲートである。

図3にインクリメンタル処理の例を示す。この例ではアップデートゲートはG6とG7である。従って、G4とG5がスタートゲートになる。配線遅延計算はG4, G5, G6, G7に対して実行しなければならない。パストレースはG4, G5, G6, G7, G8, G9に対して行う。G1, G2, G3, G4, G5に対するTINは入力端子数で、G6, G7, G8, G9に対するTINは、順に1, 2, 1, 1となる。ゲートをトレースしたらTINをインクリメントする。TINが入力端子数に等しければ、全プライマリ入力からの到達時間を計算する。

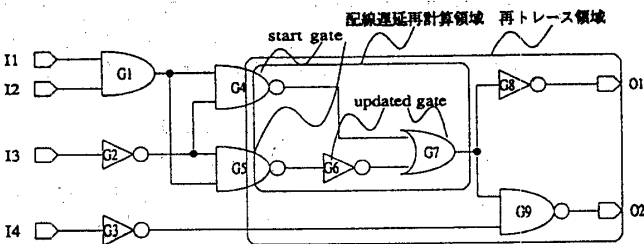


図3 インクリメンタル処理例

3. 性能

この節では、理論的な性能評価と実験結果について述べる。以下の説明中、Nはゲート数、Mはプライマリ入力数である。新手法では、全プライマリ入力に対するクリティカルパスを評価するとき、プライマリ入力からプライマリ出力までのパストレースを一回だけ行えばよい。理論的には、遅延計算を除いたトレース自体の計算時間のオーダーはNとなる。しかし、従来手法では、M回のパストレースを必要とする。従って、各ゲートに対しM回の遅延計算を行うことになる。しかし、た

とえファンインゲートがM個の異なる遅延値をもっている、各ゲートでの入出力端子間の遅延+配線遅延はM個の異なるパスに対して同じである。従って、新手法により計算時間は大幅に減少する。

新手法の欠点は余分なメモリー領域を必要とすることである。つけ加えられた情報は各プライマリ入力に対する遅延値と最悪入力端子番号である。必要なメモリーサイズは精度と基本プリミティブセルの大きさに依存する。通常、遅延値に4バイト、入力端子番号に2バイトが必要である。従って、6MNバイトが余分に必要になる。これは一見大きな量に見えるが、合成は回路の切り出し部分に対してのみ行われるので、一度に扱う回路はそう大きくならない。十分大きな回路でも1000ゲート、1000プライマリ入力の程度である。この回路の場合、0.6メガバイト余分に必要となるが、今日のコンピューターの能力からみて大きな問題ではない。

実験結果を表1に示す。F1とF2は機能マクロモジュールである。C1, C2, C3, C4は論理合成用ベンチマーク回路である。遅延値計算に関しては同じ基本ルーチンを用い、従来手法と新手法の間で比較を行った。従来手法では、前処理と配線遅延計算を一回づつおこない、残りのステップは全プライマリ入力毎に行った。CPU時間の項の数値は全パスが要求時間を満たさない場合の時間を示し、括弧内の数値は全パスが要求時間を満たしている場合の時間を示す。表1の結果から、新手法は従来手法より約4倍高速であることがわかる。

回路	ゲート数	プライマリ入力数	CPU時間 (S) 従来手法	CPU時間 (S) 新手法
F1	2160	71	19.68 (12.55)	4.16 (2.28)
F2	1912	11	5.54 (3.82)	1.21 (0.79)
C1	719	60	12.58 (5.88)	1.98 (0.98)
C2	530	22	4.93 (3.33)	1.17 (0.59)
C3	204	7	0.44 (0.16)	0.18 (0.12)
C4	123	25	1.24 (0.66)	0.29 (0.16)

表1 実験結果

インクリメンタル処理に関しては、アップデート領域の大きさや位置、要求時間を満たさない経路の全経路に占める割合が計算時間に大きく影響するため、定量的評価が難しい。定性的には、アップデート領域が小さく、その位置が出力端子に近く、要求時間を満たさない経路の全経路に占める割合が小さいほど、インクリメンタル処理による高速化の効果が大きくなることは明白である。

4. むすび

論理合成プログラムのための新しいタイミング解析手法を開発した。この論文では多重クリティカルパス検出手法と高速な解析のためのインクリメンタル処理を紹介した。この新手法により、論理合成のための高速で詳細なタイミング解析を実現した。

参考文献

[1] R. B. Hitchcock, Sr., G. L. Smith and D. D. Cheng, "Timing Analysis of Computer Hardware," IBM Journal, vol. 26, January 1982, pp. 100-105.