

パフォーマンス指向論理合成技術

Technologies of Performance-driven Logic Synthesis

あらまし

本稿では、大規模回路の遅延改善を行うための論理合成で使われる、二つの技術について解説する。大規模回路の遅延改善で一般的に行われる「局所的な遅延改善の繰返し」では、回路の「どこを」「どのように」変更するかが重要な技術課題になっている。本稿の前半では、「どのように」に関する技術であるテクノロジマッピングについて解説する。ここでは特に、tree-covering法を用いて与えられたライブラリセルの最適な組合せを求める手法について紹介する。これによって、要求される遅延特性と面積特性を持つ回路を自動で生成することが可能となった。また後半では、「どこを」に関する技術について紹介する。富士通研究所で開発した「分離集合」を用いる方法により、遅延を改善すべきか所を最適に特定することが可能となった。これら二つの技術を組み合わせることにより、大規模な回路に対して安定した遅延改善機能を実現できることが実験的に分かった。

Abstract

This paper describes two logic synthesis technologies used to reduce delay in large-scale circuits. In the repetitive local delay improvement generally done for large-scale circuits, it is important to know where and how to change a circuit.

The first half of this paper describes technology mapping for solving the problem of “how.” Particularly, a method of obtaining an optimal combination of library cells given by a tree-covering algorithm is introduced. This method allows automatic generation of a circuit that has the required delay and area characteristics. The second half of this paper introduces a technology developed by Fujitsu for solving the problem of “where.” This technology uses “separator sets” to optimally specify locations where delay needs to be improved. Experiments have shown that a combination of these two technologies can reliably improve delays in large-scale circuits.



田宮 豊 (たみや ゆたか)

1990年東京大学工学部精密機械工学科修士課程了。同年(株)富士通研究所入社。以来論理合成CADの研究・開発に従事。1994年情報処理学会第47回全国大会奨励賞受賞。コンピュータシステム研究所CAD研究部



松永裕介 (まつなが ゆうすけ)

1987年早稲田大学大学院理工学研究科修士課程了。同年(株)富士通研究所入社。以来LSIの論理設計CADの研究に従事。1998年から慶應義塾大学大学院理工学研究科非常勤講師。1994年情報処理学会山下記念研究賞受賞。1999年情報処理学会システムLSI設計技術研究会優秀論文賞受賞。コンピュータシステム研究所CAD研究部

まえがき

近年のLSIは微細化が進み、1チップに数百万ゲートの大規模回路が搭載されるようになってきた。このような大規模回路の設計では、まず、遅延制約を考慮せずに回路をとりあえず合成し、そして、回路が遅延制約を満たすまで局所的な遅延改善を繰り返すという手法が一般的である。

局所的な遅延改善では、まず、回路の遅延解析を行って、クリティカルパス(与えられた遅延制約を満たさない経路)を見つける。そして、このクリティカルパスの一部を含む局所回路を元の回路から切り出す。この局所回路中について、切り出されたクリティカルパスが短くなるように回路を変更する。最後に、変更した局所回路を元の場所に戻す。このような局所的な遅延改善をすべての遅延制約を満たすまで繰り返す。

局所的な遅延改善の繰返しでは、回路の「どこを」「どのように」変更するかが重要な技術課題になっている。これによって出来上がる回路の性能が大きく変わるからである。本稿では、まず、「どのように」に関する技術である「テクノロジマッピング」を解説する。そして、「どこを」に関する技術である「分離集合」を用いる方法を解説する。

テクノロジマッピングによる遅延改善

テクノロジマッピングは、回路の論理ゲートにライブラリセルを割り当てる処理である。同じ論理回路を実現するライブラリセルの割り当て方は複数通り存在するのが普通である。テクノロジマッピングの目的は、それらの中から要求される遅延特性と面積特性を持つものを選び出すことにある。

古くは回路の遅延時間はゲートの段数のみに左右されていたので、ゲート段数削減が遅延時間の改善手法であった。現在はゲート出力の駆動すべき負荷容量やゲート入力の信号遷移時間(slew rate: 信号のなまり)を考慮した、より詳細な遅延モデルが用いられており、より高度なアルゴリズムが必要となっている。

ここでは、まず負荷容量などに依存しない簡単な遅延モデルに対するテクノロジマッピングのアルゴリズムについて述べ、その後で負荷容量に依存した遅延モデルに対するアルゴリズムへの拡張について述べる。

(1) 簡単な遅延モデルに対するテクノロジマッピング

テクノロジマッピングのアルゴリズムとしてはKeutzerの提案したtree-covering法を基にした手法が広く用いられ

ている。⁽¹⁾これはマッピング対象回路を木状回路(すべてのゲートの出力数が1の回路)に分割し、それぞれの木状回路に対するマッピングを行うもので、分割を超えたマッピングができないという欠点があるが、木状回路の性質を生かして効率の良い処理が行えるという利点がある。

Tree-coveringでは、まず与えられた回路を2入力NANDゲートとNOTゲートのみからなる回路に変換する(図-1)。任意の論理演算はNANDとNOTのみで表現可能であり、また、入力数が3以上のNANDは複数の2入力NANDとNOTから構成可能なので、この変換は必ず行える。この変換は分解(decomposition)と呼ばれ、初期回路に依存しない形でマッピングを行うための前処理となっている。分解された回路は「サブジェクトグラフ(subject graph)」と呼ばれる。一般に、一つの初期回路に対してサブジェクトグラフへの分解は複数通り存在する。サブジェクトグラフの形は最終的なマッピングの結果を左右するので、良い解を得るような分解を行うことが望ましいが、マッピング前にどのような分解が良いかを判断することは難しいので、通常は適当な分解が用いられる。

一方、マッピングに用いられるライブラリセルも2入力NANDとNOTゲートを用いた形で表現される(図-2)。こちらは「パタングラフ(pattern graph)」と呼ばれる。パタングラフの場合はサブジェクトグラフと異なり、可能なすべての分解を列挙する。

パタングラフを複数個つなぎ合わせることによって、サブジェクトグラフのすべての節点を包含するようなグラフ(被覆)を作ることができる。こうして、テクノロジマッピングの問題は、グラフの節点被覆問題ととらえることができる。

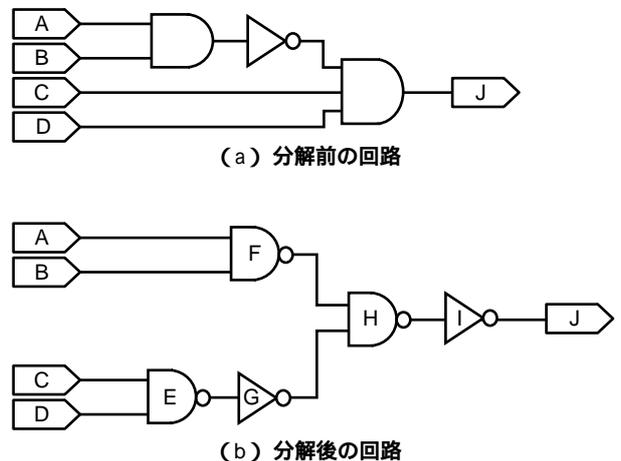


図-1 サブジェクトグラフの生成
Fig.1-Generating subject graph.

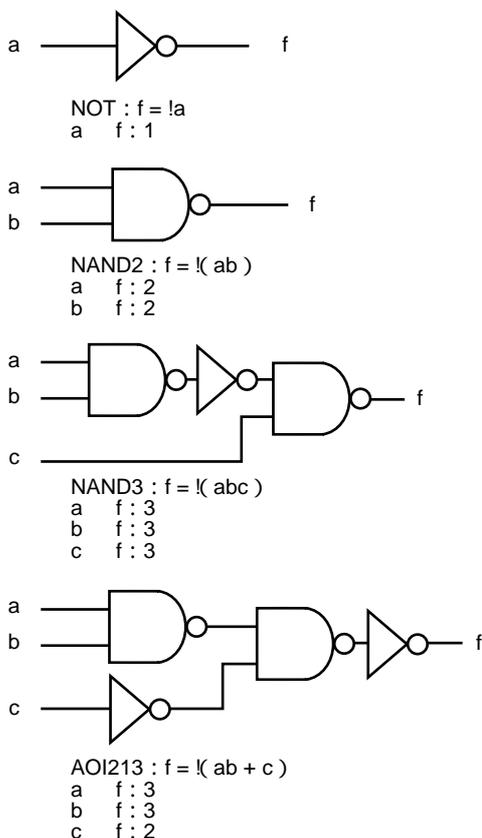


図2 パタングラフの生成
Fig.2-Generating pattern graphs.

一般のグラフ(出力数が2以上の節点を含むグラフ)の場合に節点被覆の最適解を求めることは容易ではないが、木状のグラフ(出力数が1のみの節点からなるグラフ)の場合には、グラフのサイズに比例した時間で最適解を求めるアルゴリズムが存在する。図1のサブジェクトグラフに対して図-2のパタングラフ(ライブラリセル)を用いてマッピングを行う例を図-3に示す。図-2においてa f:1のような表記はaからfへ至る経路の遅延時間が1であることを示す。ここでは負荷容量に依存しない簡単な遅延モデルを用いている。また、図-3中、入力節点の横の数字は信号の到達時刻を表している。ここでは、出力の遅延時間が最小となることを目的関数とする。

まず、Eの節点は、C、Dを入力としたNAND2の被覆しかないで、Eの出力の到達時刻は2となる。Fも同様にNAND2の被覆で到達時刻は6となる。Gに対してはEを入力としたNOTの被覆で到達時刻は3となる(Eの到達時刻2 + 1 = 3)。Hに対しては複数の被覆が考えられる。まず、F、Gを入力としたNAND2の被覆の場合には到達時刻は8となる。F、C、Dを入力としたNAND3の被覆の場合には到達時刻は9となる。この場合、到達時刻の早

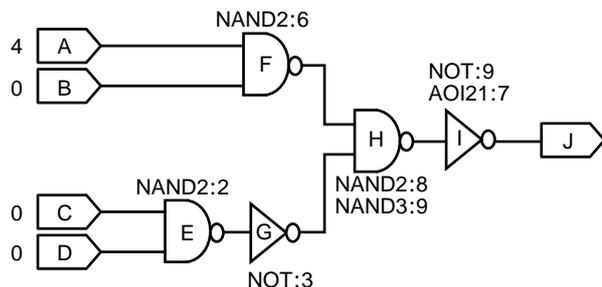


図3 マッピング
Fig.3-Mapping.

いNAND2の被覆が選ばれる。最後にIに対する被覆は、A、B、Eを入力としたAOI21の被覆(到達時刻7)とHを入力としたNOTの被覆(到達時刻9)があるので、AOI21による被覆が選ばれる。結局、IにおいてA、B、Eを入力としたAOI21の被覆が最適解であることが分かる。その結果、EにおいてC、Dを入力としたNAND2の被覆も必要であることが分かる。その他の節点(例えばF、Hなど)に対する最適被覆は結果として用いられない。

(2) 複雑な遅延モデルへの拡張

上記のような簡単な遅延モデルを用いた場合には、サブジェクトグラフの入力側から順番に部分最適解を求めていくことによって全体の最適解を求めることができるが、遅延時間が負荷容量に依存するモデルの場合には少し工夫が必要となる。例えば、図-3の例において、Hにおける最適な被覆を選ぶためにはNAND2による被覆とNAND3による被覆のどちらの遅延時間が短いかを比較する必要がある。しかし、遅延時間がHの出力の負荷容量に依存している場合、Hの出力先がどのような回路になるのかが分からない時点では遅延時間が確定せず、比較が行えない。この問題に対する解決策はあらかじめいくつかの負荷容量の値をサンプル値として用意しておいて、それぞれのサンプル値における最適解を記録する方式である。最後に実際の出力先の負荷容量が確定した時点で、それぞれのサンプル値における最適解の中で最も実際の遅延時間が短いものを選べばよい。

分離集合を用いた遅延改善か所の特定

上記のテクノロジマッピングは、メモリ量と計算時間の制約から大規模回路を一度に扱うことは不可能である。そこで、テクノロジマッピングを、回路を局所的に遅延改善する「局所変換(local transformation)」として用いる。ただし、局所変換は面積などのコスト増加を招くうえ、必ずしも全体回路の遅延が改善するとは限らないため、局所変換の選択には工夫が必要である。本章で

は、複数か所の局所変換を組み合わせ、効果的に全体回路の遅延を改善する手法について述べる。

(1) 選択集合

一般の回路ではクリティカルパスが1本ではなく、複数存在するのが普通である。回路遅延はクリティカルパスの中で一番遅いもので決まる。そのため、クリティカルパスが複数ある場合は、そのすべてを遅延改善しなくては意味がない。したがって、それぞれのクリティカルパスに対応した局所変換を複数か所選び出す必要がある。このように、回路遅延の改善のために同時に選ばなくてはならない局所変換の集合のことを「選択集合(selection set)」と呼ぶ。

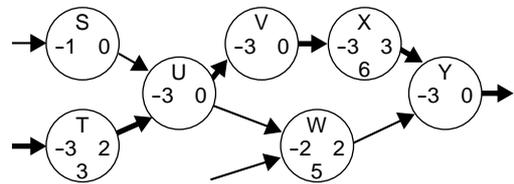
選択集合の選定の際は、互いにつながっている2本のクリティカルパスを1か所の局所変換で同時に遅延改善できるならば、それを考慮した方がよい。さらに、単に遅延を改善するだけでなく、局所変換に掛かるコスト(面積や消費電力の増加分)が少なくなる方がよい。

(2) 選択集合の選択手法

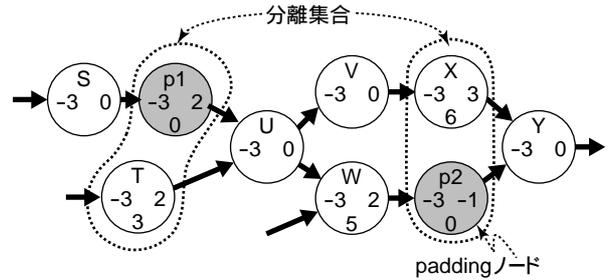
著者らは、クリティカルパスすべての遅延を改善する選択集合で最小のコストのものを求める新しい手法を開発した⁽²⁾。本手法を従来の最良手法⁽³⁾と比較すると、同じ遅延改善能力を持ちながら、とくに大きな回路に対して、少ないメモリ量で高速に遅延改善を行えるという特徴を持つ。

本手法は、まず遅延解析を行って、回路中のノードについて、スラック(slack:遅延制約の余裕)を計算する。そして、それぞれのノードにおいて、局所変換を適用した場合の遅延減少分と、局所変換に掛かるコストを評価する。図-4(a)はある回路例である。一つのノードを表し、ノード固有の遅延を持っている。矢印はノード間の信号の流れを表している。各ノードの左側の数字はスラックの値である。一番クリティカルなパスは、T-U-V-X-Yで、そのスラックは-3となっている。各ノードの右側の数字は、それぞれに局所変換を適用した場合のノード遅延の減少分である。これが0より大きい場合は、局所変換に掛かるコストを下側の数字で表している。遅延改善可能なノードは、T、W、Xの三つだけであることが図から分かる。

著者らは、回路のすべてのパスが等しいスラックを持つように、回路のクリティカルでないパス上に「paddingノード(padding node)」を挿入する。paddingノードは、遅延を持つが、コスト0の局所変換によってノード遅延を0までに改善できる仮想的なノードである。図-4(a)のすべてのパスが等しいスラック値(すなわち-3)を持つよ



(a) 局所変換の評価



(b) paddingノードと分離集合



図-4 選択集合の選び方
Fig.4-Finding selection set.

うにするため、ノード遅延が2と1のpaddingノードp1およびp2を図-4(b)のように挿入する。

このようにpaddingノードが挿入された回路において「分離集合(separator set)」を見つける。分離集合とは、回路のすべてのパスを入力側と出力側に二分するノード集合のことである。この分離集合を選択集合として選ぶと、必ず全体回路遅延が改善できることが知られている。そして、その遅延改善量は、分離集合に含まれるノードの遅延改善量の最小値に等しくなる。paddingノードのコストは0であるため、これが分離集合に含まれても選択集合のコストは増加しない。

回路中に分離集合は複数個存在する。著者らは、複数の分離集合「多重分離集合(multi-separator-set)」でコスト和が最小のものを求めれば、これがコスト最小の選択集合に一致することを証明した。残念ながら、コスト和最小の複数の分離集合を求めることは難しい問題(NP-hard問題)であるため、解くことは現実的に不可能である。そこで著者らは、コスト最小の分離集合を順次複数個求め、これを選択集合の近似解に使うことにした。

その理由は、コスト最小の分離集合を一つ求めるだけならば、ネットワークフローアルゴリズムによって、線形オーダのメモリ量と多項式オーダの計算時間で容易に求まることが知られているからである。そして、この近

似でも十分な遅延改善能力があることを実験的に証明したからでもある。

図-4(b)において、分離集合をコストが小さい順に求めると、 $\{T, p1\}$ と $\{X, p2\}$ になる。それぞれの分離集合は、遅延改善量が2と1、コストが3と6である。これよりpaddingノードを削除すると、選択集合として $\{T, X\}$ が得られ、その遅延改善量は $1 + 2 = 3$ 、コストは $3 + 6 = 9$ である。

もしもpaddingノードなしで分離集合を求める場合、paddingノードを挿入した場合より遅延改善量が減少するか、そうでなくともコストが大きくなる。図-4(a)における分離集合は $\{W, X\}$ である。この遅延改善量は3で先程と同じだが、コストが $5 + 6 = 11$ で大きくなっている。

著者らの遅延改善手法と従来法との比較結果を表-1に挙げる。著者らの手法は、従来法がメモリ不足で実行できない回路に対しても適用可能なことが分かった(実験での最大メモリ使用量は85 Mバイトであった)。

む す び

近年はLSIの微細化が進み、回路遅延の中でゲート遅延よりも配線遅延が占める割合が大きくなってきている。これは論理設計段階でゲートの配置やネットの経路を考慮して、正確な配線遅延を見積らなくては回路が設計で

表-1 遅延改善の実験結果

回路	初期回路		本手法			従来法の最良法		
	遅延	面積	遅延減少	面積増加	時間(秒)	遅延減少	面積増加	時間(秒)
s13207	31	5,045	7	-75	228	メモリ不足(>1Gバイト) >1,812		
s15850	38	6,648	13	-188	390	13	126	8,367
s35932	11	16,302	3	2,460	776	2	2,848	409
s38417	26	23,828	6	-2,067	1,306	メモリ不足(>1Gバイト) >858		
s38584	29	23,178	7	-1,348	1,984	メモリ不足(>1Gバイト) >15,937		

きないことを意味している。現行のCADツールでは、論理合成とレイアウト合成が別々に分かれているので、この問題に対処できない。今後、富士通研究所では、論理合成とレイアウト合成を同時進行させる新しい設計手法の研究・開発に取り組んでこの問題を解決していくつもりである。

参考文献

- (1) K. Keutzer : DAGON : Technology Binding and Local Optimization by DAG Matching . Proc. of 24th DAC , 1987.
- (2) Y. Tamiya : Performance Optimization Using Separator Sets. Proc. of ICCAD-99 , 1999 .
- (3) K. J. Singh : Performance Optimization of Digital Circuits. Ph.D. thesis , Univ. of California , Berkeley , 1992.