

シミュレーテッド・アニーリングを用いた テクノロジーマッピング

原嶋 勝美 福永 邦雄 小迫 秀夫
大阪府立大学工学部

あらまし

本稿では、高速化を目標としたテクノロジーマッピングについて述べる。従来レイアウト設計における配置手法として利用されていたシミュレーテッドアニーリング法は、逐次改善処理において局所最適解に陥りることを避けようとする方法であるが、多くの処理時間が必要である。そこで、温度パラメータを解の改善度に基づいて非線形に減少させることにより高速化を図り、テクノロジーマッピングに用いた。その結果、既存のテクノロジーマッピング・システムに比べ、処理時間は3倍～19倍になり、得られた回路の遅延時間および面積ともに、同程度以上の結果が得られた。

キーワード

テクノロジーマッピング、シミュレーテッドアニーリング、論理合成

Technology Mapping with Simulated Annealing

Katsumi HARASHIMA Kunio FUKUNAGA
Hideo KOSAKO

Faculty of Engineering, University of OSAKA Prefecture

Abstract

In this paper, we will present a technology mapping with the improved simulated annealing. Simulated annealing methods have been used for placement circuit modules in the layout phase, can get good solutions. However, they are very slowly. Therefore, we attempt this method speed-up decreasing a temperature-parameter non-lineally, and apply this one to a technology-mapping. As a result of, our method is faster than the logic synthesis system MIS and gets good solutions.

Keyword

technology-mapping, simulated-annealing, logic synthesis

1. まえがき

LSIの大規模化に伴い、設計工数が著しく増大し、その効率化が急がれている。論理回路設計においては、ブール関数の簡約化、多段論理回路の単純化、テクノロジーマッピングが効率化の手法である。この中で、テクノロジーマッピングのみが設計対象となるテクノロジーを考慮した手法であるため、生成される回路の性能を最終的に決定する重要な処理となる。テクノロジーマッピングは、製造テクノロジーに依存しない論理回路を製造用のライブラリセルへ変換する手法であり、生成回路の最大遅延の最小化および面積の最小化が目標である。

テクノロジーマッピングにおいても、他の効率化手法と同様に、高速でメモリー効率良く実行され、しかも上記の目標を満たす必要がある。このため、近年ではルールベースの手法[1, 2]からアルゴリズムベースの手法[3, 4, 5]へと研究の焦点が移ってきている。

本稿では、効率の良い実行を目指したテクノロジーマッピングについて報告する。本手法は、マッピング処理を繰り返し実行することにより解の改善を図っている。このとき、局所最適解に陥ることを極力避けるために、シミュレーテッドアニーリング法(以下、SA法)[6]を利用して解の改善率を計算し、繰り返すか否かの判定を行なっている。SA法は、焼きなまし、即ち物質が温度の低下に伴い、熱力学的平衡状態を保ちつつエネルギーの低下する方向へ移行し、最終的に基底状態において安定する現象を模擬したものである。SA法の適用例は、レイアウト設計における、逐次改善手法を用いた配置法に多数みられる。配置法では、繰り返し配置変更を行ない、配置変更前後の面積を比較する際にSA法を利用する。すなわち、配置変更後の面積が減少していればそれを受け入れ、増大していれば、増大量と、繰り返し回数を変数とした関数の値が、ある値よりも大きい時のみ配置結果を採用しそうでなければ処理を終了するものである。ここで、繰り返し回数の逆数がSA法の温度に対応している。また、改善されていない場合でも、繰り返し処理が可能とす

ることで、局所最適解の陥ることを避けている。しかしながら、この手法は逐次改善法であるため、計算時間を多大に要する方法であることから、様々な高速化が施されている。本手法では、SA法で用いる温度の低下を、解の改善状況に対応して非線形に実行することにより、処理の高速化を試みている。

以下では、初めに本手法で用いたSA法の概説、次にマッピングにおけるゲート変換規則、最後に実験結果、考察の順に述べる。

2. SA法

図1に本手法におけるSAアルゴリズムを示す。

本手法は、遅延時間および面積の各々に対し、SA法で解の評価を行なう。

実行手順は、最初に、マッピングを行なう前処理として、次の変数の初期化を行なう。

- C : 生成回路
- T1 : 温度(遅延時間)
- T2 : 温度(面積)
- R1 : 終了判定変数(遅延時間)
- R2 : 終了判定変数(面積)
- C1 : 非改善回数(遅延時間)
- C2 : 非改善回数(面積)
- D : 遅延時間
- A : 面積

Cは2段の論理回路を初期回路とし、T1, T2は初期温度を設定する。R1, R2は0のときに終了とするため、0より十分に大きな数字を初期値とする。C1, C2は解が改善されない場合でも、逐次改善を続行する際のカウント変数であり、0で初期化する。D, Aは各々遅延時間、面積を表す変数で、十分に大きな数字を初期値とする。

次に、マッピングを実行する。これは、次章に述べるゲート変換規則に基づいて遅延時間、面積が減少するようにゲート変換される。

マッピングが終了すると、マッピング前後の遅延時間および面積の比較を行なう。マッピング後の値の方が大きい場合には、関数 $F1$ を用いてマッピングを続行するか否かを判定する。

ここで、 a ($0 \leq 1$) は一様乱数、 $F1, F2$ は以下に示す評価関数である。

$$F1(t, d) = \exp(d/t)$$

$$F2(c) = \exp(-k * c)$$

ただし、

- t: 現在の温度
- d: マッピング前後のコスト差
(処理前のコスト - 処理後のコスト)
- c: コスト非改善数
- k: 正定数

tすなわち繰り返し回数が増加する、あるいはdが減少すると $F1$ の値は急速に0に近づく。したがって、繰り返しマッピングを行なっても解が改善されなければ、一様乱数 a より $F1$ の値が小さくなる可能性が増加し、繰り返しの可能性も低下する。

$F2$ の値は、 c が増加すると、徐々に0に近づく。この特徴を利用して、解が改善されない場合には $T1, R1$ ともに0に漸近するようにする。

一般にSA法では、温度は線形に低下させるが、本手法では、関数 $F2$ を使って、非線形な温度低下を実行している。この方法を用いると、解の改善が施されたなくなった直後には、温度低下幅を大きくすることで高速化を図り、解の非改善が続く場合には局所解に落ちている可能性を考慮して、温度低下を緩やかにすることで最適性を向上させている。

遅延時間に対する評価の手順は

- (1) $D > D'$ のとき、結果採用し、 C, D を更新して、 $C1$ を0にする。このとき、 $T1, R1$ は更新しない。
- (2) $D \leq D'$ のとき、 a と $F1(t, d)$ の大小比較を行う。 $a \leq F1(t, d)$ の場合には、結果を採用し、

{Algorithm SA}

```

C = C0;
T1 = T0;
T2 = T0;
R1 = R0;
R2 = R0;
C1 = 0;
C2 = 0;
D = MAX_D;
A = MAX_A;
REPEAT
  MAPPING(C', D', A');
  IF(D > D') THEN
    C = C';
    D = D';
    C1 = 0;
  IF(a > F1(T1, D - D')) THEN
    R1 = 0;
  ELSE
    C1++;
    R1 = F2(C1) * R1;
    T1 = F2(C1) * T1;
  ENDIF
  IF(A > A') THEN
    C = C';
    A = A';
    C2 = 0;
  ELSE
    IF(a > F1(T2, A - A')) THEN
      R2 = 0;
    ELSE
      C2++;
      R2 = F2(C1) * R2;
      T2 = F2(C1) * T2;
    ENDIF
  UNTIL (R1 == 0 && R2 == 0);
END;
```

図1 SAアルゴリズム

C1に1を加えて、T1, R1を更新する。 $a > F1(t, d)$ の場合には、R1を0にする。

面積についても同様に処理する。

以上の操作を終了条件 ($R1 == 0$ かつ $R2 == 0$) が満たされるまで実行する。

3. 回路変換

マッピング処理における回路変換について述べる。

基本的な考え方は、クリティカルパス(以下、CP)上に連結して存在する複数のゲートを組み合わせて、より大規模なゲートに変換する操作(Delayモード操作)、およびCP上に存在しない連結した複数ゲートの組み合わせ操作(Areaモード操作)の2つからなる。

Delayモードでの変換は、CP上に直列に連結したゲートが変換対象となる。一方、Areaモードでの変換は、CP上に存在しないゲートが対象であり、出力側のゲートを介して並列に連結したゲートの組合せを許す。

図2に、Delayモードの変換規則をしめす。Areaモードも同様な変換規則を使用する。

太線がCPを示す。Delayモードでは、面積が増加しても遅延時間の削減を優先するために、CP上のゲートが多出力であっても変換の候補とする。一方、Areaモードでは、面積の増加が起こらないように、最も後段のゲート以外は1出力でなければならない。

また、変換操作が実行されるか否かは、変換前後の遅延あるいは面積の値を比較し、改善される場合にのみ実行する。ところが、図2の(b), (d), (e)のパターンは変換後のゲートの入力にインバータが生成され、遅延時間や面積が一時的に増加する場合がある。この場合でも、後にインバータの削除操作を行なうことで、遅延時間、面積ともに減少させることができる。

マッピングでは、最初にDelayモードでのゲート変換を行ない、その後Areaモードでの変換を行なう。これを一組にして、逐次改善を繰り返す。

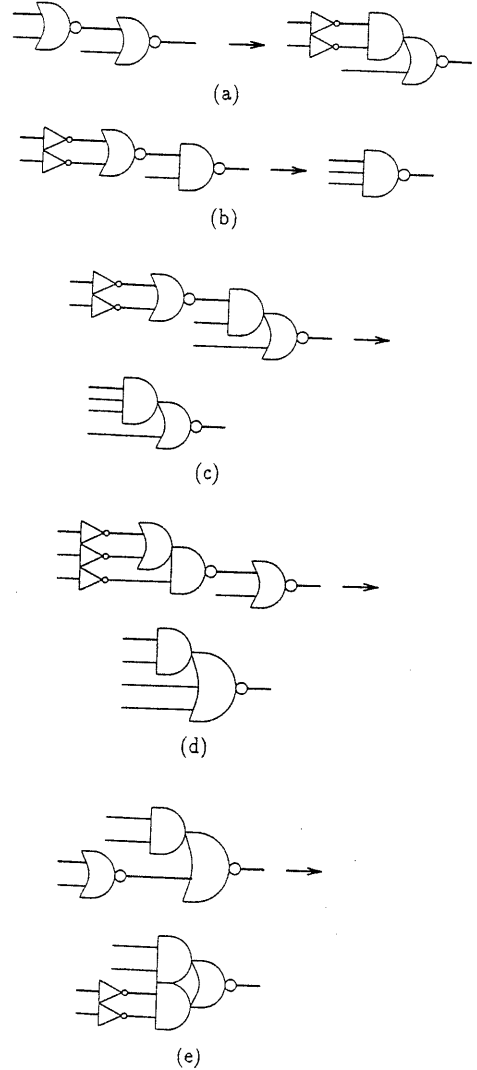


図2 ゲート変換規則

4. 実験結果

本手法を、MCNC '91の論理合成ワークショップでの2段論理ベンチマークデータおよびベンチマークセルライブラリを用いて、性能評価した。

本手法の実験結果を表1に示す。また、比較手法としてMIS[7]の実行結果を表2に示す。いずれもSUN SPARC station2で実行した。

遅延時間については、本手法は6回路のうち4回路でMISの結果を上回り、他の回路でもほぼ同程度の結果が得られている。

面積も同様に、4回路でMISの結果を本手法が上回っており、残りの回路も5%程度の増加で抑えられている。

計算時間は、すべての回路でMISの3倍～1.9倍高速に実行されている。

以上のことから、本手法で行なったSA法の改良をテクノロジーマッピングに利用することにより、良好な結果が高速に求められることがわかる。

5. 考察

実験結果から、本手法の有効性が確認されたが、今回用いたマッピング手法は比較的規則的なゲート変換操作だけであるため、大規模回路に適用した場合、必ずしも良好な解が得られる保証はない。逆に、複雑な変換操作を施していないために、高速に処理が実行できたともいえる。

また、マッピングを行なった回路の規模が大きくなるにしたがって、MISとの速度差が縮小していることが分かる。これは、ゲート変換の組合せが回路規模の増加よりも著しく大きいためであると考えられる。

したがって、多くの回路パターンでより効率良く変換が行なえるような手法を考案し、より大規模な回路において有効であるか否かを検討する必要がある。

一方、SAアルゴリズムにおいて、温度低下を非線形に行なったが、この際用いた関数は理論的に選出したものではない。このため、関数の必然性の検証を行なうと同時に、より信頼性の高い関数について考察しなければならない。

6. むすび

本稿では、従来レイアウト設計における配置手法として利用されていたSA法を改良し、テクノロジーマッピングに利用した実験結果を示した。

SA法は局所最適解に陥ることを避け易い反面、膨大な処理時間が必要であるため、温度パラメータを非線形に変更するように改良することで、高速化を実現した。また、マッピング後に得られた回路は、遅延時間および面積のいずれにおいても良好な結果が得られた。このことから、本手法の有効性が確認された。

表 1: 実験結果 - 本手法

回路	遅延時間 (ns)	面積	CPU(s)
9sym	9.78	574.4	2.7
con1	2.38	27.3	0.1
duke2	12.22	1293.1	3.9
misex1	4.52	134.5	0.4
misex2	4.32	212.5	0.5
xor5	5.42	84.9	0.3

表 2: 実験結果 - MIS

回路	遅延時間 (ns)	面積	CPU(s)
9sym	9.80	543.3	8.8
con1	2.88	26.0	1.9
duke2	12.73	1313.1	14.4
misex1	4.36	137.8	3.2
misex2	4.29	224.6	4.0
xor5	5.78	109.5	2.5

参考文献

- [1] D. Gregory, K. Bartlett, A. de Geus and G. Hachtel, " Socrates : a System for Automatically Synthesizing and Optimizing Combinational Logic ", IEEE Proc. 23rd Design Automation Conf., pp.79-85(1986).
- [2] 松本, 西山, " 論理合成システムにおける高速な回路変換手法 ", 信学技報, CAS89-24(1989).
- [3] 名古屋, 中村, 小栗, 野村, " 高位記述からの大規模論理合成における多段論理最適化 ", 信学論(A), J74-A,2, pp.206-271(1991-2).
- [4] M. Crastes, K. Sakouti, G. Saucier, " A Technology Mapping Based on Perfect and Semi-Perfect Matchings ", IEEE Proc. 28th Design automation Conf., pp.93-98(1991).
- [5] F. Mailhot, G. de Micheli, " Algorithms for Technology Mapping Based on Binary Decision Diagrams and on Boolean Operations ", IEEE Trans. Comput.-Aided Des. Integrated Circuit I& Syst., CAD-12,5,pp.599-620(1993).
- [6] C. Sechen, A. Sangiovanni-Vincentelli, " The Timberwolf placement and routing package ", IEEE J. Solid-State Circuits, SC-20,2,pp.510-522(1985).
- [7] R. Brayton, R. Rudell, A. Sangiovanni-Vincentelli and A. Wang, " MIS : Multiple-Level Interactive Logic Optimization System ", IEEE Trans. Comput.-Aided Des. Integrated Circuit I& Syst., CAD-6,6,pp.1062-1081(1987).