設 計 自 動 化 80－6
（1996. 5. 24）

マルチスキャンチェイン最適化手法

小林 進　枝廣 正人

NEC　C&C 研究所

概要

本論文では、スキャンパスが複数存在する回路のためのスキャンチェイン最適化手法を提案する。提案手法は、まずスキャンイン端子・スキャンアウト端子の組合せを端子位置情報から決定し、次にグラフ理論的手法を用いてフリップフロップの分割を行う。さらに3種類の巡回セールスマン問題の解法を用いてフリップフロップの接続順序を最適化する。実設計データを用いた計算機実験により、提案手法の有効性を確認した。

Abstract

This paper presents a scan-chain optimization method for multiple scan-paths. The proposed method first determines pairs of scan-in and scan-out pins using pin locations. Then, flip-flops are assigned to the pairs by a graph theoretical method, and three TSP methods optimize connection-order of flip-flops. Experimental results show the effectiveness of the proposed method.

1　はじめに

スキャン方式 [1] のテスト容易化回路において、スキャンパス上の配線はレイアウトの収容性に大きな影響を与える。スキャンパスにおけるフリップフロップ (以下、FF) の接続順序は、入れ換えても機能的には問題ないため、各 FF の配置位置が確定した後 (配置処理終了後)、FF の接続順序 (スキャンチェイン) の付け替えを行って、スキャンパス上の配線長を短くすることができる。このように FF の接続順序を付け替えてスキャンパス長を最小化することをスキャンチェイン最適化と呼ぶ。

従来のスキャンチェイン最適化手法としては、巡回セールスマン問題 (以下、TSP) の解法を利用した手法 [5] がある。TSP は対称 TSP と非対称 TSP に分類できるが、この手法では、FF 間の配線長を正確に見積もることを目的として、非対称 TSP の解法を用いている。しかし、非対称 TSP 解法は対称 TSP 解法と比較して、解の探索範囲が狭いという問題点がある。

また、スキャンパスは回路中に通常1本だけ存在するが、動作テスト時間を短縮するため、複数のスキャンイン・スキャンアウト端子を設けて、スキャンパスを複数に分割するマルチスキャン方式も将来的には必要性が増大すると考えられる。このマルチスキャン方式に対するスキャンチェイン最適化手法は、現在までのところ報告されていない。

本論文ではまず、解の探索範囲の広い対称 TSP 解法と、配線長見積り精度の高い非対称 TSP 解法の両方の利点を利用するため、これらの解法を組み合わせたスキャンチェイン最適化手法を提案し、実設計データを用いた計算機実験によりその有効性を示す。次に、上記最適化手法とグラフ理論的手法を組み合わせたマルチスキャンチェイン最適化手法を提案する。本手法は、まずスキャンイン・アウト端子の組合せを端子位置情報から決定し、次に FF の分割をグラフ理論的手法を用いて行う。さらに、対称 TSP 解法および非対称 TSP 解法を用いて各スキャンパスにおける FF の接続順序を最適化するとともに、スキャンパス間での FF の交換による改善も行う。実設計データを用いた評価の結果、大規模な問題に対して本手法が

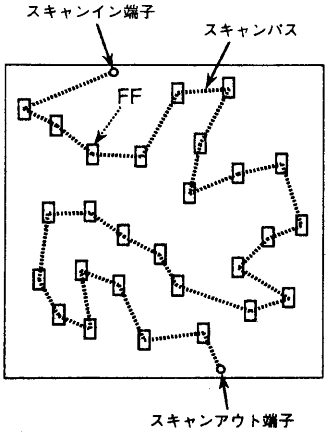

図 1: スキャンチェイン最適化問題

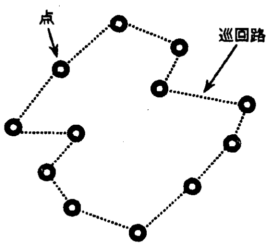

図 2: 巡回セールスマン問題

有効であることが確認された。

2 スキャンチェイン最適化

スキャンチェイン最適化問題は図1のようにスキャンパスの始点であるスキャンイン端子から始まり全てのFFを一回ずつ経由してスキャンアウト端子に至る最短の経路を求めることである。

これは図2のように全ての点を一回ずつ経由する巡回路を求める巡回セールスマン問題(以下、TSP)に類似している。

そこで、TSPの解法をスキャンチェイン最適化問題に応用することを考える。これら2つの問題の違いは、スキャンチェイン最適化問題では経路の始点と終点が異なるのに対し、TSPでは経路の始点と終点が一致していることであるが、これは、始点・終点間に仮想の枝を設け、その枝が必ず解に含まれるようにすれば良く、この制約は後述のTSP解法に容易に取り込むことができる。

TSPは、対称TSPと非対称TSPに分類することができる。対称TSPは、2点間の距離が方向に依存しないTSPであり、それ以外のTSPを非対称TSPという[2]。

スキャンチェイン最適化問題は、FF間の距離の計算方法により、対称TSPとしても、非対称TSPとしても解くことができる。FF間距離の計算方法として、次の2つの方法が考えられる。

対称方式： 各FFの代表点を入力端子と出力端子の中点とし、FF間の配線の長さは、代表点間のマンハッタン距離とする。(図3(a))

非対称方式： 2つのFF AからBまでの距離は、Aの出力端子とBの入力端子との間のマンハッタン距離とする。(図3(b))

対称方式の場合、2つのFF間の距離は信号の方向に関係なく一定であり、対称TSPの解法が使用できる。非対称方式の場合、2つのFF間の距離は図3(b)のように信号の方向によって異なるため、非対称TSPの解法を使用しなければならない。しかし、後者の方が配線長の見積りとしてはより正確である。特にFFのサイズが大きい場合(入出力端子の位置のずれが大きい場合)に、対称方式では距離の見積り誤差が大きくなる。

これに対し本論文では、対称TSP解法と非対称TSP解法を組み合わせた方法を提案する。この方法は、まず対称TSP解法で解を求め、その解を非対称TSP解法でさらに改善するものである。最初に対称TSP解法を用いることにより、広範囲の探索を行うことができ、続いて非対称TSP解法を用いることにより、FF間距離を正確に見積もった改善が可能である。

以下では、対称TSP解法、非対称TSP解法、そして対称TSP解法と非対称TSP解法を組み合わせた手法を、スキャンチェイン最適化問題に適用した場合の評価結果について述べる。

(1) 対称TSP解法の利用

対称TSP解法を利用する場合は、距離の計算方法として対称方式を採用する。対称TSP解法としては、様々な手法が提案されているが、ここでは次の2つの手法を組み合わせて用いることにする[3]。

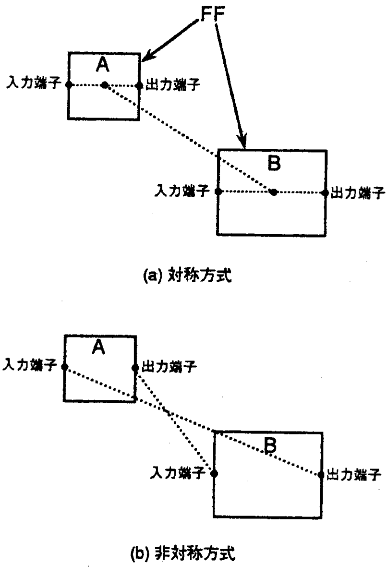

図 3: FF 間距離の計算方法（実際にはマンハッタン距離で計算）

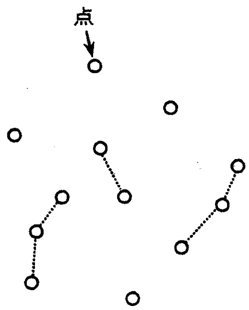

図 4: Greedy 法

Greedy 法 短い枝から順に経路に加えていく (但し、部分的な閉路ができないように、そして 1 つの点から 3 本以上の枝が出ないようにする) 手法 (図 4)。

3-opt 法 巡回路中の 3 本以下の枝の交換による改善を繰り返す逐次改善法。枝の交換は、例えば図 5のように行われる。

ここでは Greedy 法でまず初期解を求め、それを 3-opt 法で改善するという手法を用いる。

(2) 非対称 TSP 解法の利用

非対称 TSP 解法を用いる場合は、FF 間距離の計算方法を非対称方式にする。

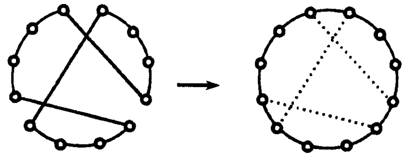

図 5: 3-opt 法

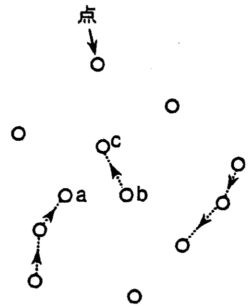

図 6: 非対称 Greedy 法

上記の Greedy 法や 3-opt 法は FF 間の距離が信号の方向によって変化しないことを利用した解法であるため、非対称 TSP には使用できない。

しかし、これらの手法を変形することにより、非対称 TSP に対応できる解法にすることができる。変形した解法をそれぞれ非対称 Greedy 法、非対称 3-opt 法と呼ぶことにする。

非対称 Greedy 法 上述の Greedy 法で、2 つの部分経路を接続する際に信号の方向を考慮する。例えば、図 6の場合には点 a と点 b は接続できるが、点 a と点 c は接続できない。

非対称 3-opt 法 上述の 3-opt 法で、枝の方向が逆転しない交換のみ採用するもの。例えば、図 7のように 3 本の枝を交換した場合はいずれの枝の方向も変化しない。このような枝交換を繰り返すことにより改善を行う。

対称 TSP の場合と同様に、非対称 Greedy 法

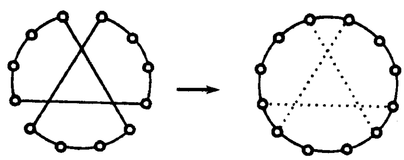

図 7: 非対称 3-opt 法

でまず初期解を求め、それを非対称 3-opt 法で改善するという手法を用いる。

(3) 対称 TSP 解法と非対称 TSP 解法の組合せ

本論文では、対称 TSP 解法と非対称 TSP 解法を組合せた手法を提案する。本手法では、まず Greedy 法で初期解を求め、それを 3-opt 法で改善し、得られた解を非対称 3-opt 法でさらに改善するものである。

実設計データによる評価結果を表 1に示す。尚、この表では比較のため、対称 TSP 解法を用いた場合でも結果のパス長は非対称方式で計算している。

[評価データ]
データ名：S1　FF 数：7437

	S1	
	パス長	計算時間
付け替え前	1685.1	—
(1) 対称 TSP	893.7	208
(2) 非対称 TSP	926.9	3259
(3) 対称 TSP+ 非対称 TSP	881.9	428

パス長：FF 間距離の合計 [mm]
FF 間距離の計算方法は非対称方式
計算時間の単位：sec
使用計算機：EWS4800/360EX (179MIPS)

表 1: 評価結果（スキャンチェイン最適化）

対称 TSP 解法、非対称 TSP 解法をそれぞれ単独で用いた場合は、対称 TSP の方が解が良く、しかも計算時間が短かった。また、対称 TSP 解法で解いた後、非対称 TSP 解法を用いることにより、さらに解を改善でき、計算時間は非対称 TSP 解法を単独で用いた場合よりかなり短い (対称 TSP 解法の場合の 2 倍程度) ことを確認した。

3 マルチスキャンチェイン最適化

3.1 マルチスキャンチェイン最適化問題

マルチスキャンチェイン最適化問題は、図 8のように、スキャンイン端子、スキャンアウト端子が複数存在するとき、スキャンパス長の合計の最

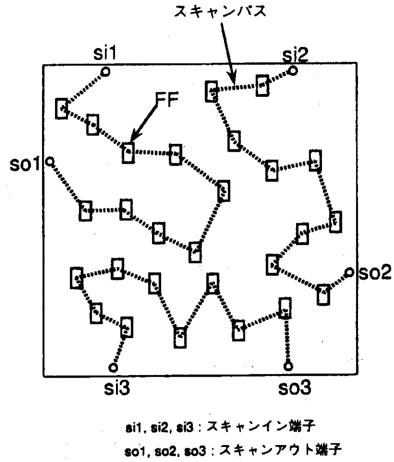

図 8: マルチスキャンチェイン最適化問題

小化を目的として、スキャンイン端子・スキャンアウト端子の組合せ、FF のスキャンパスへの割当、および FF の接続順序を決定する問題である。

ただし、動作テストにかかる時間を短くするため、各スキャンパスに含まれる FF 数はできるだけ均等になっていることが望ましい。ここでは、各スキャンパスに含まれる FF 数の最大値と最小値との差が 1 以下でなければならない、という制約を設けることとし、以下ではこの制約を FF 数制約と呼ぶ。

3.2 最適化アルゴリズム

提案アルゴリズムの概要を以下に示す。

Step1 スキャンイン端子、スキャンアウト端子の組合せの決定

Step2 FF のスキャンパスへの割当の決定

Step3 対称 TSP 解法による各スキャンパス内の FF の接続順序の最適化

Step4 非対称 TSP 解法による各スキャンパス内の FF の接続順序の最適化

Step5 スキャンパス間での FF の交換

図 9は提案アルゴリズムの処理フローである。以下に各ステップについて説明する。

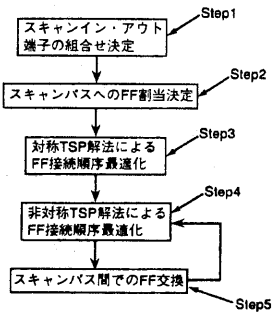

図 9: 提案アルゴリズムの処理フロー

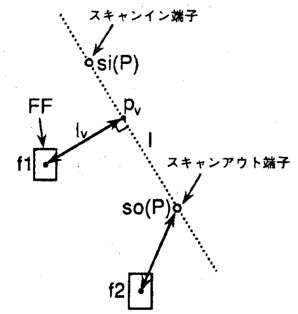

図 10: FF とスキャンパスとの距離

3.2.1 Step1: スキャンイン端子、スキャンアウト端子の組合せの決定

スキャンイン端子、スキャンアウト端子の組合せが予め決定していない場合には、適切な組合せを決める必要がある。最終的に短いスキャンパスを得るためには、直観的にスキャンイン端子、スキャンアウト端子間の距離が短い方が良いことがわかる。そこで、基本的に各スキャンイン端子に最も近いスキャンアウト端子を割り当てるが、重複する場合には、スキャンアウト端子からより遠い方のスキャンイン端子を採用する。すると、スキャンパスのスキャンイン端子、スキャンアウト端子間の距離が極端に大きくなることを防止することができ、スキャンイン・アウト端子間距離を平均化 (最大値の最小化) することができる。また、その後で、スキャンイン・アウト端子間距離の合計の最小化を目的として、各スキャンパスのスキャンアウト端子の交換による改善を行う。アルゴリズムの詳細は以下の通りである。

Step1.1 SI をスキャンイン端子の集合、SO をスキャンアウト端子の集合とする。

Step1.2 SI の中の各スキャンイン端子 s について、SO の中で距離が最も近いスキャンアウト端子 $t(s)$ を求め、s と $t(s)$ との距離を $d(s)$ とする。

Step1.3 SI の中で、$d(s)$ が最も大きい s を s_{max} とし、s_{max} と $t(s_{max})$ の組合せを確定として、それらを 1 つのスキャンパスに割り当てる。

Step1.4 SI の中から s_{max} を、SO の中から $t(s_{max})$ をそれぞれ削除する。

Step1.5 SI が空集合ならば Step1.6 へ進む。そうでなければ Step1.2 へ戻る。

Step1.6 2 本のスキャンパス P, Q の間でスキャンアウト端子を交換することにより、$\{(si(P)$ と $so(P)$ 間の距離$)+(si(Q)$ と $so(Q)$ 間の距離$)\}$ が小さくなるならば、その交換を行う。ここで、$si(K)$ はスキャンパス K のスキャンイン端子、$so(K)$ はスキャンパス K のスキャンアウト端子とし、距離はマンハッタン距離とする。この交換は全てのスキャンパスの組合せについてそれ以上改善ができなくなるまで繰り返す。

3.2.2 Step2: スキャンパスへの FF 割当決定

最終的なスキャンパス長が短くなるためには、各スキャンパス中の FF はできるだけ小さな領域に収まっていることが望ましい。そこで、まず FF とスキャンパスとの距離を後述のように定義して各 FF を最も距離的に近いスキャンパスに割り当てる。ここでは、FF 数制約は考慮していないので、通常スキャンパス間で FF 数は均等になっていない。この不均等を是正して FF 数制約を満足させるため、各スキャンパスを点とするグラフを作成し、そのグラフ上で最短経路問題を解くことにより、適切な FF の移動手順を導出し、それに従って FF を移動するという処理を繰り返すことにより、最終的に FF 数制約を満足させる。

FF とスキャンパスとの距離 D を以下のように定義する (図 10)。

FF f からスキャンパス P のスキャンイン端子 $si(P)$、スキャンアウト端子 $so(P)$ を通る直線 l へ下ろした垂線 l_v と l との交点を p_v とすると、

i) p_v が $si(P)$、$so(P)$ 間の線分上にある場合 (例：図 10の FF $f1$ の場合)
$$D(f,P) = (f \text{ と } p_v \text{ との距離})$$

ii) 上記以外の場合 (例：図 10の FF $f2$ の場合)
$$D(f,P) = \min((f \text{ と } si(P) \text{ との距離}), (f \text{ と } so(P) \text{ との距離}))$$

ここで、FF の位置は入力端子と出力端子の中点とする。また、距離はユークリッド距離とする。

以下は Step2 を詳細化したものである。

Step2.1 すべての FF とスキャンパスの組合せについて、D を求める。

Step2.2 各 FF を最小の D を与えるスキャンパスに割り当てる。

Step2.3 各スキャンパスを点 (以下ではパスノードと呼ぶ) とする有向完全グラフ G を作成する (図 11)。

Step2.4 グラフ G 上の各枝 (n,m) について、重みと移動候補を求める。ここで、枝 (n,m) とは、パスノード n からパスノード m への有向枝のことである。枝 (n,m) の重み $W(n,m)$ と移動候補 $c(n,m)$ は次のように定義される。

n に含まれる FF f_n の中で、増分

$$Inc(f_n,n,m) = D(f_n,P(m)) - D(f_n,P(n))$$

が最小になるものを枝 (n,m) に関する移動候補 $c(n,m)$ とする。ここで、$P(k)$ はパスノード k に対応するスキャンパスである。また、そのときの増分 Inc の値を枝 (n,m) の重み $W(n,m)$ とする。

Step2.5 FF 数制約を満たしていないスキャンパスがある間、以下の Step2.5.1 から Step2.5.3 までを繰り返す。

Step2.5.1 FF 数の最も多いパスノードを n_{max}、FF 数の最も少ないパスノードを n_{min} とする。

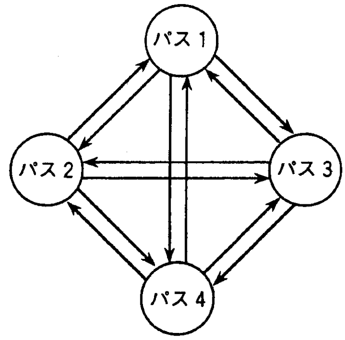

図 11: 各スキャンパスを点とする有向グラフ

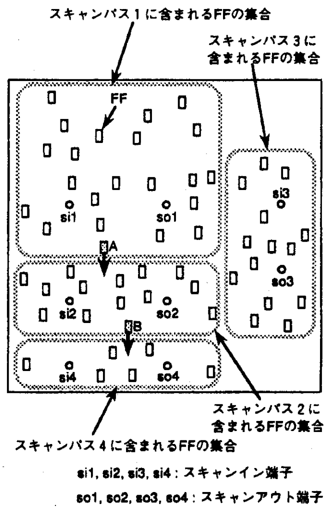

図 12: スキャンパス間の FF の移動

Step2.5.2 n_{max} から n_{min} への (グラフ G 上での) 最短経路を求め、それに沿って FF を移動させる (その経路上の枝について移動候補を移動させる)。最短経路を求める際には重み W を考慮する。

Step2.5.3 Step2.5.2 で FF の追加・削除を行った各パスノードを始点とする各枝について、重み W と移動候補 c を更新する。

図 12は Step2 の動作を説明した図である。

この図の例では Step2.2 が終了した時点を表している (スキャンパス数は 4 本)。この時点ではスキャンパス 1 の FF 数が多く、スキャンパス 4 の FF 数が少ない。FF 数制約を満足させるため

の方法として、スキャンパス1からスキャンパス4へFFを移動させることが考えられるが、この図から明らかなように、その方法ではスキャンパス4はそれが含むFF群からかなり離れた位置にあるFFを受け取ることになり、このようなFFはスキャンパス長増大の原因となる。この図から直観的にわかるように、まずスキャンパス1の中のFF(例えばA)をスキャンパス2に移動し、続いてスキャンパス2の中のFF(例えばB)をスキャンパス4に移動というような、段階的な移動を行うのが適切である。有向グラフG上で最短経路を求めることは、このような段階的な移動手順のうち、スキャンパス長の増大を最小にするような手順を導き出すことに相当している。

3.2.3 Step3: 対称TSP解法による各スキャンパス内のFF接続順序最適化

各スキャンパスをGreedy法および3-opt法を用いて最適化する。これについては、すでに第2節で述べた。

3.2.4 Step4: 非対称TSP解法による各スキャンパス内のFF接続順序最適化

各スキャンパスを非対称3-opt法を用いて最適化する。これも第2節で述べた通りである。なお、このステップを2回目以降に実行する場合で、どのスキャンパスについても改善が全く行われなかった場合には、(Step5へ進まずに)処理を終了する。

3.2.5 Step5: スキャンパス間でのFF交換

2本のスキャンパスの間で、FFの1対1の交換を行うことによりスキャンパス長の合計が改善されるならば、その交換を行う。ここで、改善されるかどうかの判断を行うために、FFの削除・追加によるスキャンパス長の増減を計算しなくてはならない。各交換毎にTSP解法を用いてスキャンパスの最適化を行った方がより正確に増減を求められるが、計算時間が膨大になる。そこで、元のスキャンパスについて、FFが削除された部分はその前後のFF(またはスキャンイン端子、スキャンアウト端子)を直接接続するものとし、追加されるFFは挿入によるスキャンパス長の増加が最も少ない位置に挿入されるものとする。

FFの交換は、すべてのスキャンパスの組合せについて、それ以上改善ができなくなるまで繰り返す。

Step5でFFの交換が全く行われなかったならば、処理を終了する。そうでなければStep4へ戻る。

4 計算機実験による評価結果

提案したマルチスキャンチェイン最適化アルゴリズムを計算機上にC言語で実現し、実設計データを用いて評価を行った。ただし、評価データはスキャンパスが1本のデータであるので、スキャンイン端子・スキャンアウト端子はそれぞれ1個しか存在しない。そこでこの他にスキャンイン端子・スキャンアウト端子をそれぞれ9個追加してスキャンパス数10のマルチスキャンチェイン最適化問題として解いた。なお、追加したスキャンイン端子・スキャンアウト端子の位置は乱数により決定した。

評価結果は表2の通りである。比較のため、スキャンチェイン付け替え前(スキャンパス1本)のパス長と、スキャンパス数1のスキャンチェイン最適化(第2節で提案した手法)の結果を示してある。

表2の評価結果より、10本のマルチスキャン方式を採用してテスト時間を削減する場合において、提案アルゴリズムを用いることにより、スキャンパス長の増加を1割程度に抑えることができることを確認した。スキャン1本当たりのFF数が1/10になっていることから、1割程度のスキャンパス長増加によって大幅にテスト時間が削減できることがわかる。

また、計算時間の点でも、FF数8千～9千程度という大規模な問題で約2分という高速な最適化が可能であることがわかった。

5 まとめ

複数のスキャンパスが存在するスキャンチェイン最適化問題を効率的に解くアルゴリズムを提案した。

提案アルゴリズムは、まずスキャンイン・アウト端子の組合せを端子位置情報から決定し、次にFFの最適な分割をグラフ理論的な手法を用いて行う。さらに、TSPの解法を用いて各スキャンパスにおけるFFの接続順序を最適化するとと

[評価データ]

- データ名：M1　FF 数：8128
- データ名：M2　FF 数：9053

		M1	M2
付け替え前	パス長	1806.9	2185.6
	パス数	1	1
スキャン	パス長	888.7	1009.6
	パス数	1	1
	計算時間	545	737
マルチスキャン	パス長	956.5	1127.3
	パス数	10	10
	計算時間	120	136

スキャン：スキャンパス数1のスキャンチェイン最適化
マルチスキャン：スキャンパス数10のマルチスキャンチェイン最適化
パス長：FF 間距離の合計 [mm](全てのスキャンパスについて)
FF 間距離の計算方法は非対称方式
計算時間の単位：sec
使用計算機：EWS4800/360EX (179MIPS)

表 2: 評価結果（マルチスキャンチェイン最適化）

もに、スキャンパス間の FF の交換による改善も行うものである。

実設計データを用いた評価の結果、FF 数8千～9千程度の大規模回路において、10本のマルチスキャン方式を採用して提案アルゴリズムを用いることにより、スキャンパス長を1割程度増加させるだけで、テスト時間を大幅に削減することができることを確認した。

今後の課題としては、運搬経路問題 (VRP) の解法を利用することが挙げられる。提案アルゴリズムでは、スキャンパス間での FF の交換に関しては極めて単純な手法を用いており、この部分に VRP 解法を適用することにより、さらに高性能なアルゴリズムの実現が可能である。

謝辞

本研究を行うにあたり、多大なる御支援、御助言をいただいた東京商船大学の久保幹雄助教授に深謝いたします。

参考文献

[1] 樹下行三, 藤原秀雄,「ディジタル回路の故障診断 (上)」, 工学図書, 1983.

[2] 久保幹雄、「巡回セールスマン問題への招待 I」, オペレーションズ・リサーチ, Vol.39, No.1, pp.25-31, 1994.

[3] 久保幹雄、「巡回セールスマン問題への招待 II」, オペレーションズ・リサーチ, Vol.39, No.2, pp.91-96, 1994.

[4] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, and D. B. Shmoys, editors, "The Traveling Salesman Problem", John Wiley and Sons, 1985.

[5] 中村, 小林, 後藤, 多和田,「スキャンパスの線長最適化とホールドタイム補償の一手法」, 情処春季全国大会, 1996.