

VLSI スキャンチェーン最適化手法

小林 進[†] 枝 廣 正 人[†]

本論文では、スキャンパスが複数存在する LSI のためのスキャンチェー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グループ分けを行う。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。

Scan-chain Optimization Methods for VLSI

SUSUMU KOBAYASHI[†] and MASATO EDAHIRO[†]

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²⁾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。しかし、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 (スキャンチェーン) 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、これによりチップ面積の削減、配線容易性の向上が可能である。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ーン最適化と呼ぶ。

従来のスキャンチェー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^{1),6)}があるが、これらはスキャンパスが回路中に 1 本だけあるシングルスキャン方式のための手法である。

しかし、近年テスト時間短縮の必要性の増大にともない、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が一般的に採用されるようになってきている。このマルチスキャン方式に対するスキャンチェーン最適化手

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ー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ー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、大規模回路に対して本手法が有効であることが確認された。

なお、スキャンパスに関しては、FF 間の遅延が小さい場合に起こりうるホールドタイムエラーに対処するため、バッファ挿入などによりホールドタイム補償を行う場合⁶⁾があるが、この点については本論文では考慮していない。

本論文では、まず 2 章でシングルスキャン方式におけるスキャンチェーン最適化手法とその評価結果を述べる。続いて、3 章ではスキャンパスが複数存在する場合のスキャンチェーン最適化問題とそれに対する提

[†] NEC C&C メディア研究所

C&C Media Research Laboratories, NEC Corporation

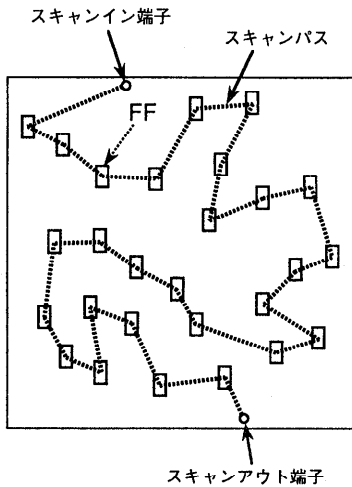


図1 スキャンチェーン最適化問題
Fig. 1 Scan-chain optimization problem.

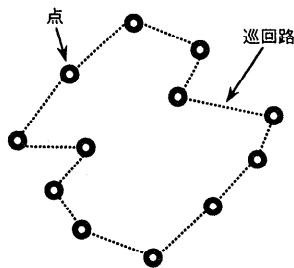


図2 巡回セールスマン問題
Fig. 2 Traveling salesman problem.

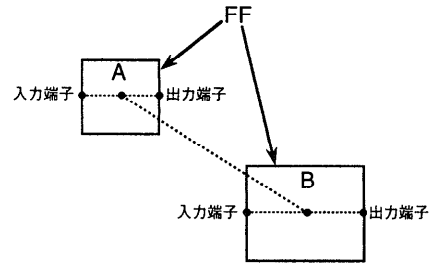
案アルゴリズムを説明し、4章で提案アルゴリズムの評価結果を示す。最後に5章でまとめと今後の課題を述べる。

2. スキャンチェーン最適化

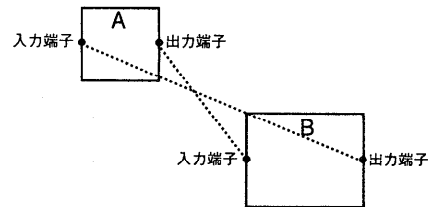
スキャンチェーン最適化問題は図1のようにスキャンパスの始点であるスキャンイン端子から始まりすべてのFFを1回ずつ経由してスキャンアウト端子に至る最短の経路を求めることである。

これは図2のようにすべての点を1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ー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、スキャンチェー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



(a) 対称方式



(b) 非対称方式

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）
Fig. 3 Methods for estimating the distance between FFs (Manhattan distance is adopted).

ればよく、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点間の距離が枝の方向に依存しないTSPであり、それ以外のTSPを非対称TSPという³⁾。

スキャンチェーン最適化問題をTSPとして定式化する場合、2つのFF間の距離を定義する必要がある。ここでは、FF間距離の定義として、次の2つの方式を採用する。

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間の距離は、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, B（信号の方向はA → B）間の距離は、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)）。

ここで、信号の方向とは、スキャンイン端子からスキャンアウト端子に向かってスキャンパス上を信号が進む方向である。

上記の対称方式の場合、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の解法を使用できる。一方、非対称方式の場合は、2つのFF間の距離は信号の方向によって異なる。その理由を以下に説明する。

図3(b)で、2つのFF A, B間の信号の方向がA

から B に向かう方向であれば、非対称方式におけるこれら 2 つの FF 間の距離は、 A の出力端子から B の入力端子までのマンハッタン距離である。逆に、信号の方向が、 B から A に向かう方向であれば、FF 間距離（非対称方式）は、 B の出力端子から A の入力端子までの距離である。したがって、非対称方式においては、2 つの FF 間の距離は信号の方向に依存することになる。

このことにより、非対称方式で FF 間距離を算出する場合には、非対称 TSP の解法を使用しなければならないが、この方式の方が配線長の見積りとしてはより正確である。特に FF の面積が大きい場合（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。

本論文では、対称 TSP 解法と非対称 TSP 解法の両者の利点を利用するため、これら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、スキャンチェー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ることにする⁴⁾。

Greedy 法 短い枝から順に経路に加えていく（ただし、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、たとえば図 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。

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。

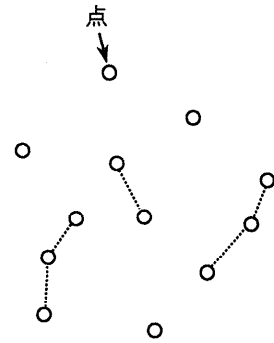


図 4 Greedy 法
Fig. 4 Greedy Method.

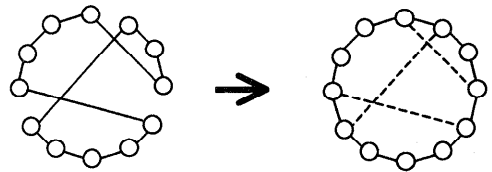


図 5 3-opt 法
Fig. 5 3-opt Method.

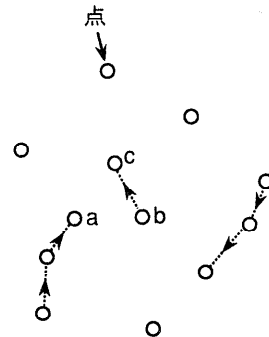


図 6 非対称 Greedy 法
Fig. 6 Asymmetric Greedy Method.

しかし、これらの手法を变形することにより、非対称 TSP に対応できる解法にすることができる。变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。たとえば、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。たとえば、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

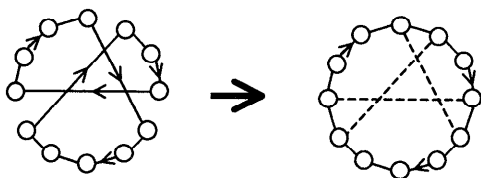


図 7 非対称 3-opt 法
Fig. 7 Asymmetric 3-opt Method.

表 1 評価結果 (スキャンチェーン最適化)
Table 1 Experimental results (scan-chain optimization).

[評価データ] データ名: S1 FF 数: 7437

	S1	
	バス長	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

バス長 : 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)

ことにより改善を行う。

ここでは、対称 TSP の場合と同様に、非対称 Greedy 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 解法を組み合わ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。なお、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バス長は非対称方式で計算している。

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ーン最適化

3.1 マルチスキャンチェーン最適化問題

マルチスキャンチェーン最適化問題は、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、スキャンバス長の合計の最小化を目的として、スキャンイン端子・スキャンアウト端子の組合

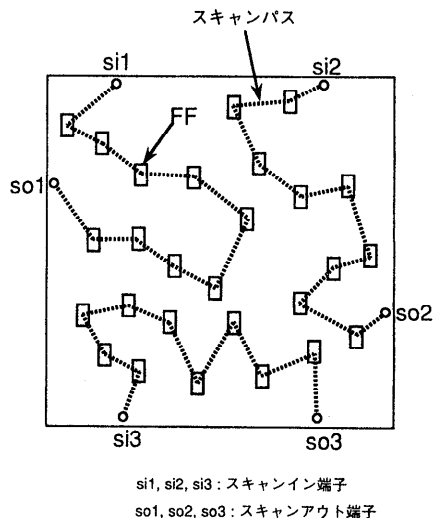


図 8 マルチスキャンチェーン最適化問題
Fig. 8 Multi-scan-chain optimization problem.

せ、FF のスキャンバスへの割当て、および FF の接続順序を決定する問題である。

ただし、テスト時間を短くするため、各スキャンバスに含まれる 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、各スキャンバスに含まれる 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 のスキャンバスへの割当て
- Step3: 対称 TSP 解法による各スキャンバス内の 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バス内の FF の接続順序の最適化
- Step5: スキャンバス間での 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。

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あらかじ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バスを得るためには、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分かる。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

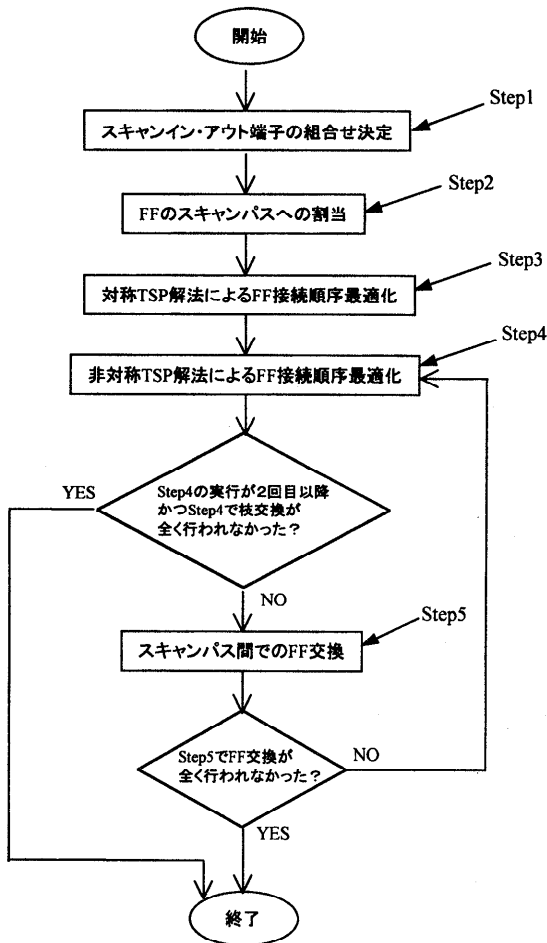


図9 提案アルゴリズムの処理フロー
Fig. 9 Flow of the proposed algorithm.

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とおりである。

- 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 中の各スキャンイン端子 s について、 SO 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 中で、 $d(s)$ が最も大きい s を s_{max} と

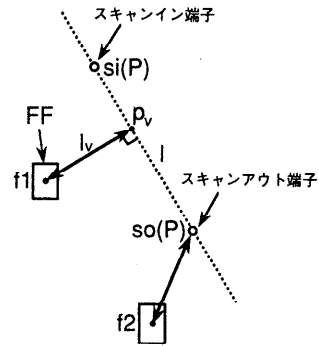


図10 FFとスキャンパスとの距離
Fig. 10 Distance between FF and scan-path.

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。

3.2.2 Step2: FFのスキャンパスへの割当て

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、

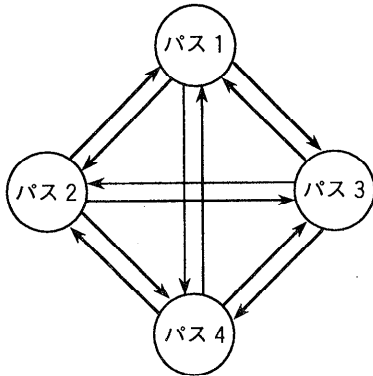


図 11 各スキャンパスを点とする有向グラフ

Fig. 11 Directed graph with nodes corresponding to scan-paths.

- i) p_v が $si(P)$, $so(P)$ 間の線分上にある場合 (例: 図 10 の FF f_1 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 f_2 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 midpoint 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 $W(n, m)$

* 実際のインプリメントの際には, 処理時間短縮のため, 各パスノードから他のパスノードへの移動候補の数を限定した. これにより, 移動候補が存在しない枝が生じることがあるが, このような枝は Step2.5 において最短経路に含まれないようにした.

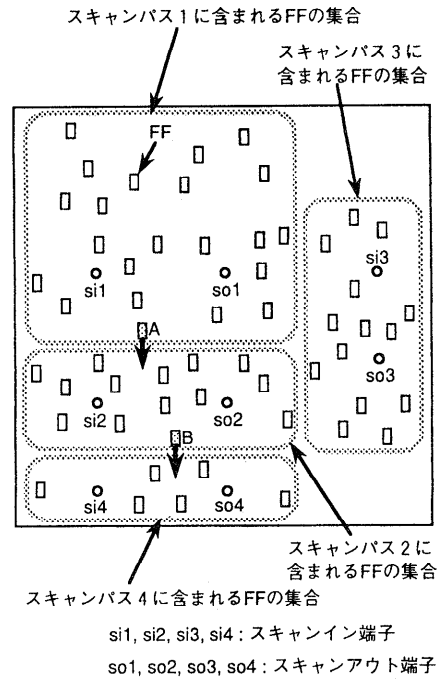


図 12 スキャンパス間の FF の移動
Fig. 12 FF movement among scan-paths.

- 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.
 - 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 Step2.2 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

ら直観的に分かるように、まずスキャンパス 1 の中の FF (たとえば 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 (たとえば B) をスキャンパス 4 に移動するというような、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 2 章で述べた。

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 2 章で述べたとおりである。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まった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。各交換ごとに 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 の交換がまった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ー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 本のデータであるため、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ーン最適化問題として解いた。なお、追

表 2 評価結果 (マルチスキャンチェーン最適化)

Table 2 Experimental results (multi-scan-chain optimization).

[評価データ]

- データ名: M1 FF 数: 8128
- データ名: M2 FF 数: 9053
- データ名: S1 FF 数: 7437

		M1	M2	S1
付け替え前	パス長	1806.9	2185.6	1685.1
	パス数	1	1	1
シングルスキャン	パス長	888.7	1009.6	881.9
	パス数	1	1	1
	計算時間	545	737	428
ランダム	パス長	7454.0	9383.6	5806.1
	パス数	10	10	10
マルチスキャン	パス長	969.1	1109.7	977.8
	パス数	10	10	10
	計算時間	99	119	80

シングルスキャン: スキャンパス数 1 のスキャンチェーン最適化

ランダム: ランダムに 10 本のスキャンパス数を生成した場合

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ーン最適化

パス長: 全スキャンパスの 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)

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 のとおりである。比較のため、以下の 3 種類のデータもあわせて示してある。

- スキャンチェーン付け替え前 (スキャンパス 1 本) のパス長
- スキャンパス数 1 のスキャンチェーン最適化 (2 章で提案した手法) の結果
- 与えられたスキャンイン・アウト端子および FF を用いてランダムに (乱数を使用) 10 本のスキャンパスを生成した場合の結果

本実験では、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、(スキャンパス数 1 のスキャンチェーン最適化と比較して) 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 本あたりの 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分かる。

また、計算時間の点でも、FF 数約 7,000~9,000 の大規模な問題に対して 1~2 分という高速な処理が可能であることが分かった。

最後に、本実験における処理経過に関するデータを

表 3 処理経過における各種データ
Table 3 Detailed information on experiments.

		M1	M2	S1
Step2	移動回数	5069	6044	5177
Step3	3-opt 法実行前のパス長	1118.3	1311.0	1164.2
	枝交換回数	1936	2337	2018
	実行後のパス長	989.6	1145.7	999.8
Step4 (1回目)	枝交換回数	490	720	393
	実行後のパス長	972.8	1112.4	986.3
Step5	FF 交換回数	10	10	9
	実行後のパス長	969.1	1109.7	977.8
Step4 (2回目)	枝交換回数	0	0	0
	実行後のパス長	969.1	1109.7	977.8

表 3 に示す。

表 3 の各項目について以下に説明する。

- Step2 における移動回数: Step2.5.2 において FF が移動した回数
- Step3
 - 3-opt 法実行前のパス長: 3-opt 法を実行する直前, すなわち, Greedy 法実行直後の総スキャンパス長
 - 枝交換回数: 3-opt 法における 2 本あるいは 3 本の枝交換の回数
 - 実行後のパス長: Step3 終了後の総スキャンパス長
- Step4
 - 枝交換回数: 非対称 3-opt 法における 2 本あるいは 3 本の枝交換の回数
 - 実行後のパス長: Step4 終了後の総スキャンパス長
- Step5
 - FF 交換回数: スキャンパス間の FF 交換の回数
 - 実行後のパス長: Step5 終了後の総スキャンパス長

なお, ここでの FF 間距離の計測方法はすべて非対称方式, 単位は [mm] である。また, いずれのデータにおいても, 2 回目の Step4 で枝交換が行われず, そこで処理が終了している。

5. まとめ

複数のスキャンパスが存在するスキャンチェー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もに, スキャンパス間の 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, FF 数約 7,000~9,000 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, まず, 運搬経路問題 (VRP) の解法を利用することがあげられる。提案アルゴリズムでは, スキャンパス間での FF の交換に関してはきわ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と考えられる。

また, スキャンパスにおいては, 一般にホールドタイム補償のためのパッファ挿入が行われ, それによるチップ面積増大も大きな問題となる。したがって, この点についても考慮しながらスキャンチェーン最適化を行うことも今後の課題である。

謝辞 本研究を行うにあたり, 多大なるご支援, ご助言をいただいた東京商船大学の久保幹雄助教授に感謝いたします。

参 考 文 献

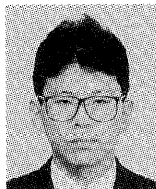
- 1) Chen, C.-S., Lin, K.-H. and Hwang, T.T.: Layout Driven Selecting and Chaining of Partial Scan Flip-Flops, *Proc. Design Automation Conference*, pp.262-267 (1996).
- 2) 樹下行三, 藤原秀雄: デジタル回路の故障診断 (上), 工学図書 (1983).
- 3) 久保幹雄: 巡回セールスマン問題への招待 I, オペレーションズ・リサーチ, Vol.39, No.1, pp.25-31 (1994).
- 4) 久保幹雄: 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).
- 5) Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. and Shmoys, D.B. (Eds): *The Traveling Salesman Problem*, John Wiley and Sons

(1985).

- 6) 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, 情報処理学会春季全国大会 (1996).

(平成 10 年 9 月 21 日受付)

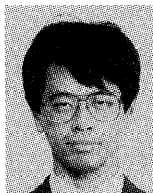
(平成 11 年 2 月 8 日採録)



小林 進 (正会員)

1967 年生. 1989 年東京大学工学部機械工学科卒業. 1991 年同大学院工学系研究科船用機械工学専攻修士課程修了. 同年日本電気 (株) 入社. 以来, C&C システム研究所に

て VLSI CAD の研究に従事. 現在 NEC C&C メディア研究所主任.



枝廣 正人 (正会員)

1960 年生. 1983 年東京大学工学部計数工学科卒業. 1985 年同大学院計数工学専攻修士課程修了. 同年日本電気 (株) 入社. 以来, C&C システム研究所にて VLSI CAD, 計算幾何学等の研究に従事. 1991 年プリンストン大学計算機科学科博士課程入学. 1993 年修士取得. 現在 NEC C&C メディア研究所主任研究員. 1993 年情報処理学会第 2 回坂井記念特別賞, 1994 年情報処理学会山下記念研究賞等を受賞. 電子情報通信学会, 日本オペレーションズリサーチ学会, IEEE 各会員.