

## ソフトウェア・ハードウェア連動配置手法

## A novel placement technique based on collaboration of hardware and software

嶋本 俊彦†      吉川 雅弥†      寺井 秀一†  
Toshihiko Shimamoto      Masaya Yoshikawa      Hidekazu Terai

## 概要

VLSI 設計におけるセルの配置問題において、遺伝的アルゴリズムを用いた配置手法がある。遺伝的アルゴリズムは強力な解探索能力を有する。しかし、対象の大規模化によって処理時間が増加する問題がある。この問題の解決のために、本研究ではソフトウェアの柔軟性と、ハードウェアの高速性を兼ね備えたハードウェア・ソフトウェア連動配置手法を提案する。ハードウェアとソフトウェアの利点を効果的に使用できるように処理の切り分けを行い、FPGA に実装した。また、比較実験を行い、提案手法の有効性を確認した。

## 1.はじめに

近年の情報化社会の急速な発展に伴い、LSI の大規模化・高集積化による回路の複雑さは増大の一途を辿っている。また微細化の弊害でこれまで問題にならなかった物理現象なども考慮した設計が必要となっている。複数のトレードオフ関係にある要因を複合的に考慮しながらセルを配置していく手法として遺伝的アルゴリズム (*Genetic Algorithm:GA*)<sup>[1]</sup>を用いた配置手法があり、その有効性が示されている<sup>[2][3]</sup>。しかし、GA は多点探索アルゴリズムであるために、評価対象である回路規模の増大に伴って、適応度評価にかかる時間が著しく増加してしまうという問題がある。

GA の処理時間の問題を解決する方法としては、ハードウェアを用いて処理の高速化を図るという手法が考えられるが、全ての処理をハードウェアで実現した場合、汎用性が無くなり、現実的ではない。そこで本研究ではこの問題に対し、ハードウェアの高速性とソフトウェアの柔軟性を兼ね備えたソフト・ハード連動配置手法を提案する。

## 2.GA を用いたセル配置手法

## ・2.1 階層化配置手法

本研究では、基本アルゴリズムとして GA を用いた階層化配置手法を用いている。階層化配置手法では、処理を「概略配置」、「詳細配置」の2ステップに分割し、対象が大規模な間は概略配置を繰り返し行い、概略配置により対象がある程度小規模になってきたら厳密にセルの位置を決定する詳細配置を行う。本研究では、概略配置手法として4分割配置法を用い、詳細配置では部分領域を2次元のグリッド領域とみなし、セルを最適な位置へ配置する。

## ・2.2 評価関数

本研究では評価関数として、仮想配線長、電力指向配線長、消費電力の分散 (電力分散係数) を用いる。以下それぞれについて簡単に説明する。

## (i)配線長

LSI の微細化が進むに連れて、配線遅延の影響が大きくなってきている。配線遅延を改善するためには、配線長を短くする必要がある。しかし、配線アルゴリズムのみで、配線を短くするには限界があるため、配置段階から配線長を考慮する必要がある。そのため、配置結果の評価基準として配線長の総和が多く用いられる。本研究では仮想配線長の値を求める手法として幹線支線方式を採用した。

## (ii)電力指向配線長

CMOS LSI の消費電力のうち、信号線のスイッチングによる消費電力が全体の70%を占めており、これを削減することが回路全体としての消費電力の削減に有効である。信号線のスイッチングによる消費電力は

$$P = \alpha \times C \times f \times V^2 \quad (1)$$

( $\alpha$  は活性化率、 $C$  は負荷容量、 $f$  は周波数、 $V$  は電源電圧) で求めることが出来る。そこで、本研究では $\alpha$  が大きいセル (活性化率の高いセル) の負荷容量  $C$  を下げる (配線の負荷容量を下げる) ために、活性化率が閾値以上のネットの仮想配線長を優先的に短くする目的で電力指向配線長として評価に用いる。

## (iii)電力の分散 (電力分散係数)

消費電力の大きなセルを1箇所に配置すると、発熱などの弊害を引き起こす可能性がある。そこで概略配置 GA では“チップ全体での消費電力の分散化をはかる目的関数”を、詳細配置 GA では“各セル列での消費電力の合計の標準化をはかる目的関数”をそれぞれ取り入れている。

概略配置 GA では、各分割領域に割り当てたセルの消費電力の総和  $P_i$  (4分割を行っているので  $i = 1, 2, 3, 4$ )

† 立命館大学

を求める。ここでは、outline Pi の最大値から最小値を引いた値 P を評価値としている。また、詳細配置 GA では、同じ列 (同じ電源ラインから駆動する) に配置したセルの消費電力の総和 detail Pi ( $i = 1 \sim n$ ) を求める。ここでは、概略配置 GA と違い最大値 P を評価値としている。

### 3. ハード・ソフト連動配置手法

GA を用いた配置手法では、配置対象となる LSI の回路規模が大きくなるにつれて、処理時間の増加が問題となる。検証のために、これまでのソフトウェアのみで実現した配置手法を用いて時間を計測した。実験条件は CPU : PentiumIV 1.7GHz, メモリ 512M, OS : WINDOWS2000 のマシンを用い、入力として  $0.35 \mu\text{m}$  CMOS セルライブラリ、および規模の違う 6 つのネットリストを用いた。その結果を表 1 に示す。表 1 からわかるように回路規模の増加に伴って、処理時間が急激に増加しているのがわかる。また、今後も回路規模は増加し続けることが予想され、処理時間の問題もより深刻になってくるものと考えられる。

#### ・3.1 処理の分割

本研究では効率的な高速化を実現するために、“処理時間”と“柔軟性”の 2 点に着目し、ハードウェア化する部分を検討した。

まず処理時間の内訳について考える。ここでは表 1 の示した回路の中から最も処理時間のかかった回路 F について検討することにする。図 1(a) は概略配置と詳細配置の処理時間の内訳を示したものである。図を見ると、詳細配置にくらべ概略配置で処理時間が多くかかっていることがわかる。さらに概略配置を 1 回目とそれ以降に分割した図を図 1(b) に示す。これを見ると、1 回目の概略配置が処理全体の 39% を占めていることがわかる。これは、1 回目の概略配置、すなわち最初の 4 分割が、配置結果に与える影響が最も大きいからであって、そのため、今後、対象の回路規模が大きくなりセル数が増えると、1 個体当たりの計算時間が増え、さらに多くの処理時間が必要になる。これらの事より、1 回目の概略配置に要する時間はさらに増大すると思われる。

次に、各処理に要する時間を同じく回路 F について比較した。結果を図 2 に示す。

図 2 を見ると、遺伝処理が全体の 23% なのに対し、評価演算が全体の 77% を占めている。中でも配線長演算が全体の 68% を占めており、処理時間の改善という観点から見れば、この部分をハードウェア化するのが最も適当であると考えられる。また、柔軟性の維持の観点から見れば、遺伝処理を行う部分では、柔軟にパラメータを変更することが必要となり、ハードウェア化には適当ではない。1 回目の概

表 1 回路規模の違いによる処理時間比較

	セル数	ネット数	処理時間[s]
回路A	186	223	114
回路B	407	489	646
回路C	1512	1521	2309
回路D	2136	2157	23525
回路E	3109	3122	27650
回路F	7992	8133	55053

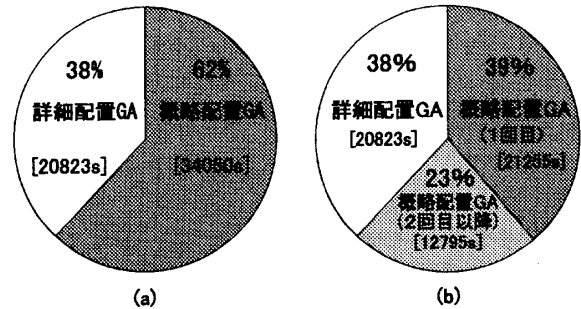


図 1 (a) 概略配置と詳細配置の時間比較

(b) 1 回目の概略配置に着目した時間比較

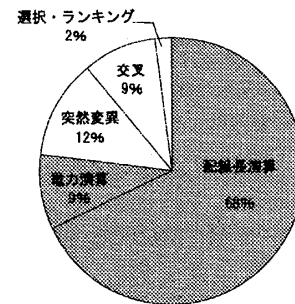


図 2 処理時間比率

略配置 GA、すなわち最初の 4 分割では、配線長演算部分の遺伝子数は固定され、その他のパラメータにも影響されることが無い。そのためハードウェア化しても配置プログラム全体の柔軟性を保つことが出来る。そのために、従来のソフトウェアの機能を削減することなく、ハードウェアとソフトウェアの連携が実現する。以上より、本研究では初回概略配置 GA の配線長演算部分をハードウェアで実装し、従来のソフトウェアと連動させる。

#### ・3.2 ハード・ソフト連動システム

ハードウェア・ソフトウェア連動のシステムについて説明する。ソフトウェア側では、1 回目の概略配置 GA の遺伝処理と電力演算、そして 2 回目以降の概略配置 GA および詳細配置 GA 処理を行い、ハードウェア側では 1 回目の概略配置 GA の配線長演算 (仮想配線長・電力指向配線長) を行う。ソフトウェアとハードウェアの間にはデータの送受信が必要である。配線長演算に必要なセルの配置位置を

表す“遺伝子データ”とネットリストやネットの信号遷移確立を表す“ネットデータ”の2種類をソフトウェア側から送信し、ハードウェア側から“仮想配線長”と“電力指向配線長”の2種類のデータを受信する。ネットデータは不変なので最初に1度送信するだけ良いが、遺伝子データは個体ごとに異なるので毎回送信する必要がある。図3にハード・ソフト連動配置手法のシステムフローを示す。ハード・ソフトを連動させた場合には通信時間が生じるため、それを考慮した設計が必要である。今回計測した結果、FPGAボードの仕様に応じた32bit unsigned int型データを送信するのに $2.4 \times 10^{-7}$  [s]、またHW応答時間として1送信毎に $3.2 \times 10^{-4}$  [s]かかることがわかった。ソフトウェアで初回概略配置GAの配線長演算1回に要する時間は $1.24 \times 10^{-2}$  [s]であり、1回の評価演算のためにソフトウェア側から固体情報を送るために $0.22 \times 10^{-2}$  [s]必要であるため、1回の評価演算に対するHWの処理時間は $1.02 \times 10^{-2}$  [s]以下に抑える必要がある。

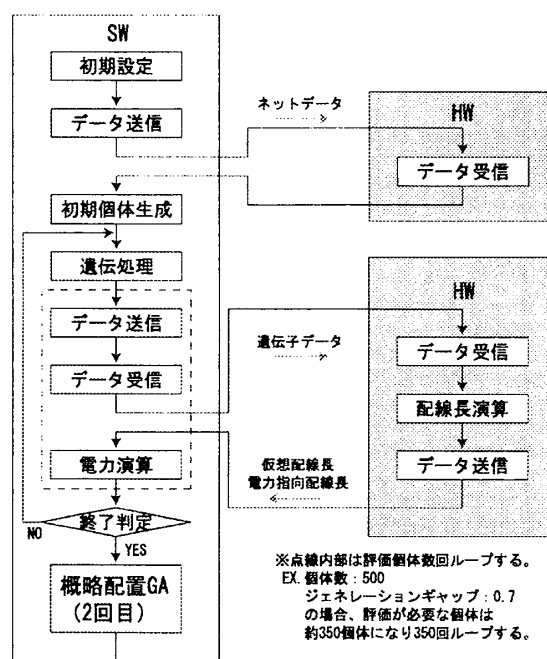


図3 HW・SW連動処理フロー

#### 4.回路構成

今回設計した回路のブロック図を図4に示す。今回作成した回路は大きくインターフェース回路、演算回路、データ格納制御回路、データ出力制御回路、ネットデータ格納RAM、遺伝子データ格納RAMで構成されている。演算回路内部の配線長計算回路およびネットデータ格納RAM、遺伝子データ格納RAMは3つで1セットになっており、これらを並列に72個並べ、並列処理を行うことで処理時間の削減を図った。(今回は使用できるRAMの個数の限界から72並列とした)

#### 5.実験・考察

今回の実験では、 $0.35 \mu\text{m}$  CMOSセルライブラリおよびセル数7992、ネット数8133のネットリストを用いた。また、ソフトウェアはCPU:PentiumIV1.7GHz、メモリ512M、OS:Windows2000のマシンを用い、ハードウェアには株式会社ネサステクノロジ社製VirtualTurboPCIボード、およびLogicBench:HF06Kを用いた。世代モデルは連続世代モデルとし、トーナメント戦略およびエリート保存戦略を採用した。GAの各パラメータはジェネレーションギャップ0.7、トーナメントサイズ4、初期個体数500、突然変異率0.005、世代数4500とした。

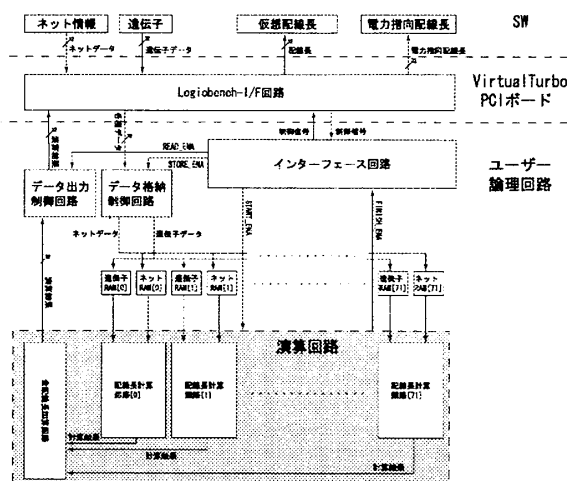


図4 回路構成

ソフトウェアを用いた配置手法とハード・ソフト連動(初回概略配置GAの配線長評価部をハードウェア化した)配置手法のそれぞれについて5回計測を行い、それぞれの平均値を実験結果とした。平均値を用いたのは、本研究では基本アルゴリズムに確率的探索を行うGAを用いているために、処理ごとに配置結果および処理時間が異なるからである。表2に配置処理全体にかかった処理時間比較を、表3に商用ツールを用いて配置したものも含めた最終配置の評価値比較を示す。

表2より、評価演算部のHW化を行った1回目の概略配置では、ソフトウェア単体で処理した場合と比較して、処

理時間において 22381[s]の削減に成功し、約 77%の改善が見られた。配置処理全体での処理速度比較をみると、概略配置 GA (2回目以降)と詳細配置 GA においては、ソフトウェア単体での処理とハード・ソフト連動処理の間に、大きな違いは見られない。これは配置処理が全く同じアルゴリズムを元に完全にソフトウェアで行われているからである。ソフトウェアでは概略配置 GA(1回目)に時間を要しており、これを大幅に削減したハード・ソフト連動手法は、合計処理時間からもソフトウェアに対して 35%の改善を実現し、十分な有効性が確認できた。

次に、配置結果から評価値として用いた仮想配線長、電力指向配線長、電力の 3 つの値について比較を行った。ソフトウェア単体での処理、ハード・ソフト連動での処理ともに、ほぼ同じ値になっており、元の配置プログラムの機能が損なわれていないことが確認できる。また商用ツールで配置を行い、それを本研究で用いた評価値演算を用いて計算した結果、仮想配線長、電力指向配線長ともほぼ同じ値を示し、GA を基本アルゴリズムに用いた配置手法の有効性が確認できる。しかし、電力分散係数を比較してみると商用ツールとの差がみられる。これは配置領域の差が原因で、商用ツールの配置に対し本研究で用いた配置手法の方が 1 列あたりのセル数が多くなってしまふことが問題で起こっている。

さらにハードウェアでの演算、通信時間、オーバーヘッド、通信用のデータ変換(ソフトウェア処理)の 4 つの処理の比較を行った。結果、内訳は以下の表 4 のようになった。

この結果から 1 個体分の仮想配線長および電力指向配線長を計算し結果を受け取るのに必要な処理の中で通信が 58%を占めていることがわかる。ハード・ソフト連動配置手法

では、データ通信の回数が非常に多く、今後はデータを送受信する際のデータ格納方式などの改善を行うことで処理時間の削減を図ることが有効であることが確認できた。

6.まとめ

本研究では、高速化と柔軟性の維持という問題を、ソフトウェアとハードウェアを連動させることによって解決を試みた。そして、提案手法によりソフトウェアの柔軟性を失わないまま、全体として約 35%の処理時間の短縮に成功した。

今後の課題としては、初回の概略配置 GA を HW 化することの有効性が確認できたので、同じアーキテクチャを使用できる 2 回目以降の概略配置 GA の配線長演算の HW 化を行いさらなる時間短縮を図ること、SW・HW を連動させる上で、通信時間が問題となっているのが確認されたため、インターフェース回路やデータ送信方式を見直すことにより通信時間の削減を図ること等があげられる。

7.参考文献

[1] J.Holland. "Adaptation in Natural Artificial Systems", the University of Michigan Press (Second Edition : MIT Press1992).  
 [2] 吉川雅弥, 寺井秀一, 「遺伝的アルゴリズムをベースとする LSI セルの並列配置手法」, 電気学会論文誌 C, Vol.120-C, No.11, pp.1528-1538, 2000  
 [3] 吉川雅弥, 河村渡, 寺井秀一, 「配線混雑度・消費電力分散化を考慮した VLSI 配置手法」, DA シンポジウム 2002 論文集, pp95-100, 2002

表 2 処理時間比較

	概略配置GA (1回目)	概略配置GA (2回目以降)	詳細配置GA	合計処理時間
ソフトウェア	29825[s]	13151[s]	19420[s]	62196[s]
ハード・ソフト連動	6960[s]	12991[s]	19864[s]	39815[s]

表 3 評価値比較

	仮想配線長[μm] (幹線支線)	電力指向配線長 [μm]	電力
ソフトウェア	1216136	244397	6.39
ハード・ソフト連動	1226996	245951	6.45
商用ツール	1286971	246328	5.37

電力：同じ電源ラインから駆動するセルの消費電力の総和

表 4 処理時間内訳

	処理時間[s]	割合[%]
演算	$9.0 \times 10^{-4}$	28
通信	$1.91 \times 10^{-3}$	58
オーバーヘッド	$3.2 \times 10^{-4}$	10
SWでのデータ処理	$1.1 \times 10^{-4}$	4