社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＳＬＤＭ－119（５）
2005／3／17

フロアプランと高位合成を同時に行うLSI設計手法

大塚 正臣　　伊藤 和人

埼玉大学工学部電気電子システム工学科

〒338-8570 埼玉県さいたま市桜区下大久保255

TEL/FAX 048-858-3731

E-mail: {ootsuka,kazuhito}@elc.ees.saitama-u.ac.jp

あらまし　近年の半導体製造技術の進歩により集積回路の微細化が進み，配線を用いた通信遅延が総遅延時間に占める割合が相対的に増加してきている．そのため，微細化によって向上した性能を十分に生かして高速処理実行可能なLSIを従来の設計手法で得ることが困難になってきている．本論文では，従来手法の問題点を改善するためフロアプランと高位合成を同時に考慮するLSI設計手法を提案する．この手法は，(1)与えられた処理中の各演算を演算器に割り当てるバインディングを仮定，(2)演算器とレジスタのフロアプラン生成，(3)演算スケジューリングとレジスタバインディング，からなり，Simulated Annealingによって最適なバインディングとフロアプランを探索する．提案手法をプログラム実装し計算機実験によって従来手法との比較を行い，その有効性を示す．

キーワード　LSI設計手法，　高位合成，　フロアプラニング，　配線遅延

An LSI Design Method with Simultaneous Processing of Floorplanning and High-Level Synthesis

Masafumi Otsuka　　Kazuhito Ito

Department of Electrical and Electronic Systems, Saitama University

225 Shimookubo, Sakura-ku, Saitama 338-8570, Japan

Tel/FAX +81-48-858-3731

E-mail: {ootsuka,kazuhito}@elc.ees.saitama-u.ac.jp

Abstract With the recent advances in the semiconductor manufacturing technologies, the delay of communication on wires becomes dominant among the total delay. Therefore it is getting difficult to obtain high-speed LSIs by using the conventional LSI design methods. In this paper, to overcome the problem of the conventional design methods, we propose an LSI design method where floorplanning and high-level synthesis are performed simultaneously. This method consists of (1) functional-unit binding, (2) floorplanning the functional-units and registers, and (3) scheduling and register binding, and the best solution is searched by using simulated annealing. We show effectiveness of the proposed method through experiments.

Keywords　LSI design method, High-level synthesis, Floorplanning, Interconnect delay

1. まえがき

近年の半導体製造技術や材料技術の進歩により集積回路の微細化が急速に進んでいる．回路が微細化されるとゲート遅延は減少する一方で，配線による通信遅延(配線遅延)はほとんど変化しないため総遅延時間に占める通信遅延の割合が相対的に増加しており，今後もこの傾向は続くと予想される[1].

これまでのLSI設計手法では，まず入力データとしてLSI化する処理内容とそれに対する性能要求が与えられると，基本的処理アルゴリズムを選択する．コストや性能の要求に応じてHW/SW分割などを行う場合もある．高位合成として，与えられた処理中の各演算について開始時刻を決めるスケジューリングと演算器に割り当てるバインディングを行う．この際，配線遅延を見積もることにより最終的な性能が予測される．その後論理合成，フロアプラン，および詳細レイアウト設計を行う．この時点で初めて演算器の位置が決定され，実際に配置配線が判明し最終的なLSIの処理実行時間が決定される．高位合成後にフロアプランを行うために各素子の位置に基づく正確な配線遅延の情報を高位合成のときに考慮することが出来ない．逆にフロアプランを先に行いフロアプランの情報をもとに高

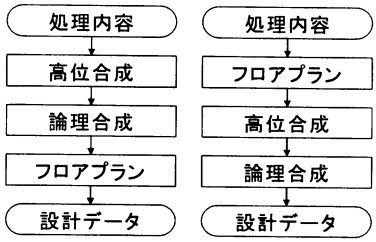
位合成を行う手法では，フロアプランの最適性を保証出来ない．これらの手法ではいずれも配線遅延が原因となって設計したLSIの処理速度が低下する．以上のように配線遅延の影響が大きい場合には，微細化によって向上した性能を十分に生かして高速処理実行可能なLSIを従来の設計手法で得ることが困難になってきている．この問題の原因は高位合成とフロアプランを分けて行うことにある．

以上を踏まえ本稿では，従来手法の問題点を改善するためフロアプランと高位合成を同時に考慮するLSI設計手法を提案する．この手法は，(1)与えられた処理中の各演算を演算器に割り当てるバインディングを仮定，(2)演算器とレジスタのフロアプラン生成，(3)演算スケジューリングとレジスタバインディング，からなり，Simulated Annealingによって最適なバインディングとフロアプランを探索する．先にバインディングを仮定することで演算器間の通信の状況が明らかになり，通信を行う演算器どうしを隣接させるなど，与えられた処理に最適なフロアプランを生成することが可能となる．また生成したフロアプランから抽出する正確な配線遅延情報を用いることで最適なスケジューリングとレジスタバインディングを可能としている．さらにスケジューリングでは，フロアプラン結果から求めた正確な配線遅延に基づいて通信専用クロック挿入やレジスタ転送を行うことで設計するLSIの動作可能クロック周期の短縮を図る．

2. 問題点および関連する研究

2.1. これまでの設計手法の問題点

これまで用いられてきた高位合成を行った後にフロアプランを生成するという設計手法(図 1(a))では，高位合成の基準となる性能予測の精度が低下してきている．遅延時間に占める配線遅延時間の割合が増加すると配線遅延時間が最終的な性能に与える影響が大きくなるため，正確な性能予測には正確な配線遅延時間が重要である．ところが，正確な配線遅延時間が判明するのはフロアプラン生成後であり，高位合成時には配線遅延時間の見積もり値を利用せざるを得ない．不正確な配線遅延時間見積もり値に基づいて高位合成を行う結果，フロアプラン後に予期しなかった長い配線遅延時間が発生すると予測よりも低い性能しか達成できない．仮に，まずフロアプランを生成した後に高位合成を行うと(図 1(b)),フロアプランに基づく正確な配線遅延時間を用いて高位合成が実行できるので，そのフロアプランに対して最適な高位合成を行うことが可能となる．しかし，先に求めたフロアプランがそもそも与えられた処理に対して最適である保証がなく，結果として最適な性能を達成する保証もない．



(a) (b)

図 1 従来手法. (a)高位合成を先に行う手法. (b)フロアプラン生成を先に行う手法.

以上のように配線遅延の影響が大きい場合には，微細化によって向上した性能を十分に生かして高速処理実行可能なLSIを従来の設計手順で得ることが困難になってきている．その問題点は，高位合成とフロアプランを独立して行うことにある．この問題を解決するために，これまでにいくつかの研究が行われている．

2.2. 関連する研究

演算器バインディングとフロアプランを同時に行う手法[2]では，与えられたフロアプランについて，クリティカルパス中の配線遅延の割合を削減する演算器バインディングを行って高速処理を達成する．さらに最適なフロアプランを探索することで，処理性能を最適化する．すなわち，図 1(b)の従来手法を繰り返し行うことで最適設計を探索する方法である．レジスタ間データ転送は考慮せず，場合によっては長い通信経路を持つ可能性がある．

基本セルを用いた手法[3]では，与えられた信号処理に対して，頻繁に行われる通信の配線遅延を最適化した配置を予め基本セルとして用意し，必要個数の基本セルを並べることで最適な設計を得ようとする．基本セルの概念により設計を容易化し，レジスタ間データ転送も考慮しているが，セル単位の設計となるため応用が難しく設計の最適性が保証できない問題点がある．

レジスタ分散型アーキテクチャを対象とする高位合成手法[4]の研究では，フロアプランを考慮したレジスタ間のデータ転送を利用する手法が用いられた．フロアプランから得られた配線遅延情報をフィードバックして，レジスタ間データ転送を行っている．スケジューリングとバインディングを行い，それに基づいてフロアプランを改善する，という手順を繰り返すことで，最適なバインディング，スケジューリング，フロアプランを求めている．すなわち，図 1(a)の従来手法を繰り返し行うことで徐々に最適な設計に近づける方法である．

本研究では，演算器とレジスタの配置に特に制約を

設けず，まず演算バインディングを仮定し，そのバインディングに対して最適なフロアプラン，レジスタバインディング，スケジュールを求めることで，フロアプランと高位合成を同時に行う手法を提案する．最適な演算バインディングを探索することで最適設計を得る手法を検討する．

3. 提案手法

3.1. モデル

設計のモデルを定義する．演算器とレジスタを合わせてモジュールと呼ぶ．図2に示すように，各モジュールは種類によって大きさが予め決められた長方形とする．モジュールのデータ入力と出力の端子は対向する2辺のそれぞれの中央に位置すると仮定する．演算器は入出力にデータ記憶機能を持たない．外部のレジスタからデータを供給して演算を行い，結果は外部のレジスタが保持する．パイプライン化された演算器では，演算の途中結果を記憶するパイプラインレジスタを演算器内部に有する．

配線を用いたデータ通信の遅延時間については，モジュールのフロアプランが与えられたとき，データの送信元と受信先の位置および配線形状に基づいて何らかの方法で一意に遅延時間が与えられるとする．方法については問わない．簡単には送信元と受信先の位置間のマンハッタン距離に比例する遅延時間を用いることが考えられる．

3.2. 設計の方針

高位合成は，演算を演算器に割り当てる演算器バインディング，演算実行時刻を決定するスケジューリング，データ保持区間をレジスタに割り当てるレジスタバインディングからなる．フロアプランでは頻繁に通信を行う演算器同士を隣接して配置することで通信時間を最短化する．このとき，どの演算器間に通信が頻繁であるかは演算器バインディングに依存する．スケジューリングでは，演算器間の通信時間に基づいて，処理完了までに要する時間が最短となるようにクロック周期と演算の実行時刻を決定する．そこで，高位合成のうちの演算器バインディングをまず仮定し，必要なレジスタ数を見積もり，モジュール(演算器とレジスタ)の最適フロアプランを求める．次にフロアプランから抽出したモジュール間通信時間に基づいて，高位合成の残りのスケジューリングとレジスタバインディングを行う．このようにフロアプランと高位合成を同時に行うことで，配線による通信時間を正確に考慮した設計を行う．

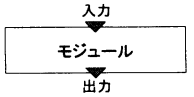


図2 モジュールモデル

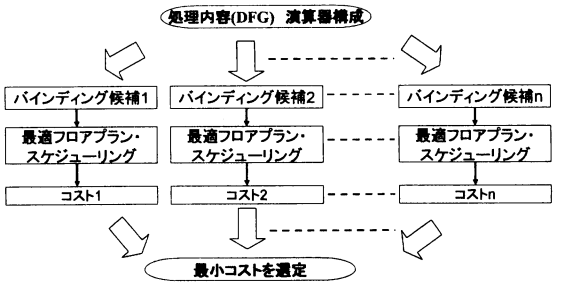


図3 提案手法概要

3.3. 設計手順

提案手法の設計手順を説明する．図3に示すようにまず，複数の演算器バインディング候補を生成し，それぞれの候補における最適なフロアプランと演算スケジュールを導いた後，コストを算出する．最小コストのバインディング，フロアプラン・スケジュールを出力する．

最適な演算器バインディングの探索は Simulated Annealing(SA)を用いて行う．演算器バインディングの各候補について最適なフロアプランとスケジュールを得るためにも SA を用いる．すなわち SA が2重の入れ子となっている．

入力： 信号処理アルゴリズム(DFG)
　　演算器・レジスタのサイズと演算処理時間
　　配線遅延情報
出力： スケジュール
　　バインディング
　　フロアプラン

<設計手順>

手順 1. スケジューリングレンジ図を用い使用する演算器数の上限を仮定する．

手順 2. 加算器・乗算器のバインディング x'を生成し，使用するレジスタ数を決める．

手順 3. 必要となる最小レジスタ数を算出し，必要レジスタ数とレジスタの上限数の間で実際に使用するレジスタ数を決定する．

手順 4. 与えられたモジュール数において新しいフロアプラン y'を生成する．

手順 5. レジスタのバインディングを行う．

手順 6. 演算器・レジスタの位置情報から配線遅延・クロック周期を求め通信専用クロック挿入や

レジスタ転送を行いスケジューリングとレジスタバインディングを修正する．

手順 7. 出力されたスケジューリング・バインディング・フロアプランよりコスト関数を求める．

手順 8. コストが改善されていれば現在のフロアプランを採用する $(y \leftarrow y')$．もし改悪となっている場合でも温度に依存した受理確率により採用する．

手順 9. N_2 回のループ毎に温度 t_2 の値を下げ終了温度に達していればループを終了する．達していなければ設計手順 4 に戻り新しいフロアプランを生成する．

手順 10. 得られたフロアプランとスケジュールに基づいてコストを算出する．前回のバインディング x のコストと比較し，改善されていればその割り当てを採用する$(x \leftarrow x')$．もし改悪となっている場合でも温度に依存した受理確率により採用する．

手順 11. N_1 回のループ毎に温度 t_1 の値を下げ終了温度に達していればループを終了する．達していなければ設計手順 2 に戻り新しい素子割り当てを行う．

手順 12. 全ての構成で最小のコストのスケジューリング・バインディング・フロアプランを出力する．

図 3 で示した提案手法を構成する 2 つの SA の作業と詳細な設計手順の流れを図 4, 5 に示す．図 4 中の SA②と示した部分で，図 5 の SA②を行っている．SA②で求めた最適なフロアプランとレジスタバインディングを用いて SA①の解評価を行う．

設計手順 1 のスケジューリングレンジは ASAP・ALAP スケジューリングを行って算出する．繰返し周期は最短繰返し周期とする．

設計手順 2, 10, 11 では，使用可能な演算器数(上限)内で各演算をバインディングする．1 個以上の演算が割り当てられた演算器数として，実際に使用する演算器数が決まる．最適なバインディングの探索を SA で行う．演算器バインディングはループ毎に一部を変更し，レンジチャートガイドスケジューリング[6]を行い演算実行が可能かを判断する．もし指定した繰返し周期で実行が不可能なら，バインディングをやり直す．これは不必要なループを取り除くためである．

設計手順 3 では，手順 2 で求めたスケジュールで必要とするレジスタ数を求め，そのレジスタ数とレジスタ数の上限との間で実際に使用するレジスタ数を決定する．余分なレジスタを用意することでレジスタバインディングのときに過度の共有を防ぎ，通信時間の短縮・平均化とともにレジスタ転送を容易にし，クロック周期の削減を図る目的がある．

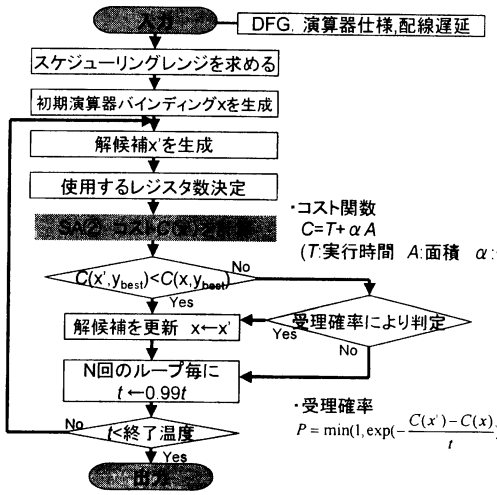


図 4　SA①バインディング探索

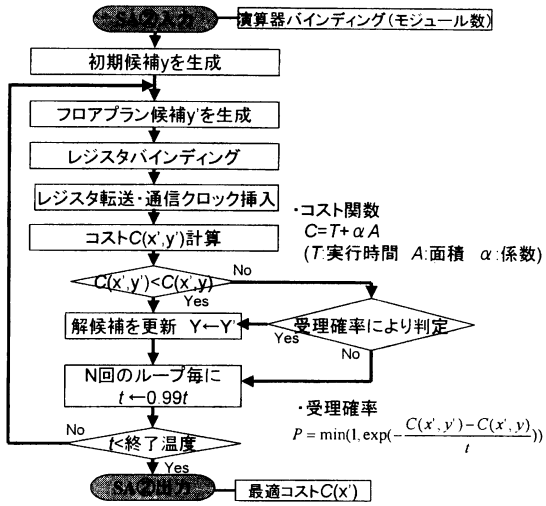


図 5　SA②フロアプラン探索

設計手順 4, 8, 9 では，フロアプランを Sequence-Pair[5]で表現し，最適なフロアプランを探索する．Sequence-Pair は 2 つの文字列によってモジュール間の位置関係を表すデータ構造である．モジュールの入出力端子位置を考慮して，90 度単位で 4 通りのモジュール回転も探索する．

設計手順 5 では，演算器バインディングによる通信状況とフロアプランによる配線遅延の情報を考慮してクロック周期が最短となるレジスタバインディングを行う．レジスタバインディングは，レジスタに格納す

るデータを生成する演算器とそのデータを消費する演算器間の通信時間に基づき，クロック周期が最短になるようデータを格納するレジスタを選ぶ．

設計手順6では，次節に述べる通信専用クロック挿入やレジスタ転送を行うことによりレジスタバインディングとスケジューリングを修正することで，クロック周期短縮を図る．

コスト関数は，T を処理実行時間，A を面積としたとき，

$$T+\alpha A \qquad (1)$$

とする．ただしαは任意の係数である．T は処理実行に必要なクロックステップ数とクロック周期の積で与えられる．A は，全てのモジュールを内包する長方形の最小の面積とする．

3.4. 通信専用クロックの挿入

配線遅延がボトルネックとなりクロック周期に影響を及ぼしている演算に注目し，対象に演算を含まない通信専用のクロック期間を挿入する．長い通信を演算実行クロック周期と分けることによりクロック周期の短縮を図る．

例として図6のデータフローグラフ(DFG)を考える．図6には枝ごとに演算結果を格納するレジスタを示している．演算e, d間に依存関係があるが，演算eは演算d実行前に処理が終了すればよく，必ずしも演算a実行終了直後に実行する必要はない．図7上のようにレジスタReg1と演算eの間の配線遅延がクロック周期のボトルネックとなっている場合，図7下のようにReg1からデータを演算eの演算器の近くに配置されているReg3に転送してから演算eを実行する．つまり，Reg1からReg3へレジスタ転送を行う通信クロック周期と演算eの演算実行クロック周期に分ける．演算eの終了は1クロック遅れるが全体のクロック数は変化しておらず，演算eと入力レジスタ間の配線距離を削減することでクロック周期を短縮し，より高速な処理実行が可能となる．提案手法では設計手順6で可能な状況なら通信専用クロックの挿入を行う．

4. 計算機実験

提案手法と従来手法により設計を行うプログラムをC言語を用いて計算機上に実装した．計算機環境はIntel PentiumIV 2.6GHz，メモリ容量512MBである．対象アプリケーションとして処理1(ノード数8)と処理2(ウェーブデジタル楕円フィルタ，ノード数34)を用意し各手法による合成結果を比較した．

使用するモジュール(演算器，レジスタ)の仕様を表1に示す．乗算器は2段パイプライン化されている．なお，プログラム中では用いるデザインルールに応じて

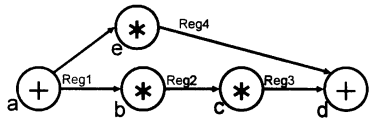


図6　データフローグラフ(DFG)

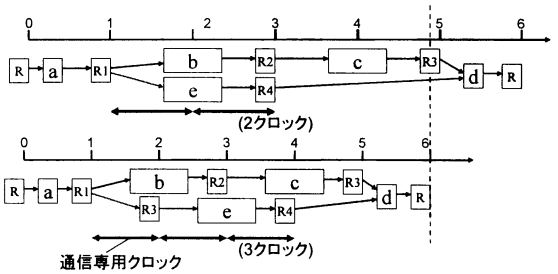


図7　通信専用クロックの挿入

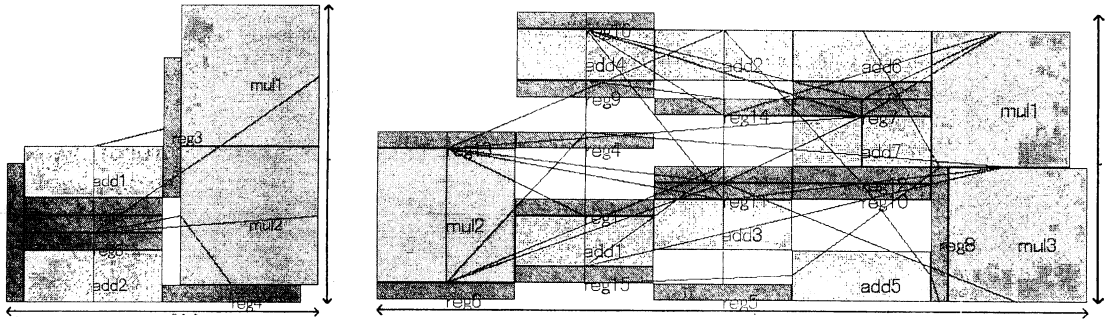
表1　モジュール仕様

モジュール	面積	処理時間
加算器	24	30
乗算器	64	50
レジスタ	8	7

演算器と配線遅延の各パラメータは変更が可能になっている．データ通信には，配線のマンハッタン距離の1単位長さあたり1単位時間の遅延が生じるとする．

処理1に対する設計結果を表2に，処理2に対する設計結果を表3に示す．表中の"従来手法①"は高位合成後にフロアプランを生成する手法(図1(a))，"従来手法②"はフロアプランを生成した後に高位合成を行う手法(図1(b))である．"提案手法①"は3.2節の高位合成とフロアプランの同時設計の探索アルゴリズムのみの設計結果を表し，"提案手法②"は探索アルゴリズムに3.3節の通信専用クロック挿入を実装した設計結果を表している．SAの開始温度は20，終了温度は1，各温度でのループ回数はN_1=5，N_2=10とした．コスト関数は，処理1ではα=0.1，処理2ではα=0.4とした．

処理1，処理2においてそれぞれ提案手法が従来手法と比較して30%または20%以上実行時間が短縮しLSIの性能が改善されていることが分かる．また提案手法において，通信専用クロックの挿入を実装した手法の実行時間が短縮されており，挿入の効果が示されている．処理1ではあまり顕著でないが，従来手法に比べ提案手法で設計した結果の演算器やレジスタが増加しているのはそれまで演算器を共有して行っていた演算を分散させることでボトルネックとなっていた長い配線遅延が削減されたためである．図8に処理1，処理2について提案手法で生成したフロアプランを示す．各モジュール間を結ぶ線が演算器間の通信を表してい



(a) (b)

図 8 フロアプラン結果. (a)処理 1, (b)処理 2.

表 2 処理 1 の設計結果 1-1

設計手法	加算器	乗算器	レジスタ数	面積	実行時間	CPU時間[sec]
従来手法①	2	2	5	234	204	0
従来手法②	2	4	6	456	219	1
提案手法①	2	2	8	300	135	2739
提案手法②	2	2	6	306	129	2745

表 3 処理 2 の設計結果 2-1

設計手法	加算器	乗算器	レジスタ数	面積	実行時間	CPU時間[sec]
従来手法①	3	2	10	560	1296	0
従来手法②	4	4	11	676	1712	111
提案手法①	5	3	13	551	1008	9629
提案手法②	7	3	16	697	992	9636

表 4 実行時間最小化の結果

処理	加算器	乗算器	レジスタ数	面積	実行時間	CPU時間[sec]
処理1	2	2	6	324	132	2720
処理2	4	3	18	800	976	9608

る. フロアプラン中でデッドスペースが多いのは実行時間の最小化を優先して探索を行ったためである.

表 4 は, コスト関数の α=0 として処理実行時間 T を最小化した結果を示す. 面積を評価しないため, 処理実行時間 T は最小化されるが, 無駄に面積の大きなフロアプラン結果となる場合がある. そこで, 得られた最適演算バインディングに対して, 後処理として処理 1 では α=0.1, 処理 2 では α=0.4 として繰り返し回数 N_2=500 とする SA②を行ってフロアプランの最適化を行う. 表 2, 3 に示した"提案手法②"と比べると, 処理 2 では実行時間が改善されているが, 処理 1 では若干実行時間が長くなっている. これはモジュール数が比較的少ない設計では, フロアプランの変化が通信距離, すなわち通信遅延時間に与える影響が大きく, 最初から面積を考慮した演算バインディング探索がよりよい結果を与えたと思われるが, 一般的に規模の大きな設計では処理 2 と同じ傾向になると考えられる.

5. 結び

微細化が進み配線遅延が相対的に大きい状況下での LSI 設計において, 従来は独立して行っていたフロアプランと高位合成を同時に考慮することで正確な配線遅延情報に基づいて最適な LSI 設計を行う手法を提案した. また, 配線遅延がボトルネックとなりクロック周期に影響を及ぼす場合を考慮して通信専用クロックを挿入する手法を考案した. 最後に計算機実験により提案手法と従来手法を比較し, その設計結果は実行時間が 20%以上短縮されており, 提案手法の有効性が確認できた.

今後の課題としては, フロアプランの改善, マルチプレクサを用いた point-to-point 通信への対応, 演算器バインディングの探索手法の改善などが挙げられる.

文 献

[1] International Technology Roadmap for Semiconductors, http://www.itrs.net/

[2] Y. M. Fang and D. F. Wong, "Simultaneous Functional-Unit Binding and Floorplanning," in Proc. ICCAD 1994, pp. 317-329.

[3] K. Ito and D. Suzuki, "A High-Level Synthesis Method for Simultaneous Placement and Scheduling Considering Data Communication Delay," in Proc. APCCAS 2002, pp. 149-154.

[4] 田中, 内田, 宮岡, 戸川, 柳沢, 大附, "レジスタ分散型アーキテクチャを対象とするフロアプランを考慮した高位合成手法," in Proc.IPSJ SIJ Technical Reports 2004, pp. 197-202.

[5] H. Murata, K. Fujiyoshi, and S. Nakatake, "Rectangle packing-based module placement," in Proc. ICCAD 1995, pp. 472-479.

[6] S. M. Heemstra de Groot, S. H. Grez, and O. E. Herrmann, "Range-Chart-Guided Iterative Data-Flow Graph Scheduling", IEEE Trans. Circuit Syst.- I: Fund. Theory & Appl., vol. CAS-39, pp. 351-364, 1992.