

配線アクティビティを考慮した 3次元積層プロセッサ向けフロアプランナ

入江 英嗣^{1,a)} 放地 宏佳^{1,†1,b)} 稲場 朋大^{1,c)} 眞島 一貴^{1,d)} 藤原 大輔^{1,e)}
吉見 真聡^{1,f)} 吉永 努^{1,g)}

受付日 2012年12月21日, 採録日 2013年5月30日

概要: 半導体 3次元積層技術の進展により, 3次元化したプロセッサの性能/パワーバランスが向上することが予測されている. 配置配線の 3次元化は, プロセススケールリングでは縮まないロングワイヤを幾何学的に縮め, 電力消費の主要部分を占めている配線電力を削減することができる. しかし, 既存の 3次元モジュールマップでは, 配線のコスト関数はスイッチングアクティビティを考慮しておらず, またどのようなフロアプランが得られたかが明らかにされていない. 本研究では, バイブラインシミュレータと連携して, モジュール間の通信頻度を考慮する 3次元モジュールマップを提案し, 3次元プロセッサのフロアプランおよびそのアーキテクチャへの影響を明らかにする. 提案モジュールマップの出力結果からは, 3次元構造を利用した効率的なデータパスやキャッシュ配置が確認された. 3層, TSVの配線容量を $30\mu\text{m}$ の通常配線と同等と仮定したときのフロアプランでは, 2次元実装に比べて面積を 34%, ロングワイヤ電力の近似値である配線アクティビティ値を 57%まで削減し, 従来のフロアプランナによる 3次元実装に対して 10%の配線アクティビティ値削減となった.

キーワード: 3次元積層 VLSI, マイクロプロセッサ, フロアプランナ

A Novel Wire-activity-aware Floorplanner for 3D-stacked Processor

HIDETSUGU IRIE^{1,a)} HIROYOSHI HOUCHI^{1,†1,b)} TOMOHIRO INABA^{1,c)}
KAZUKI MAJIMA^{1,d)} DAISUKE FUJIWARA^{1,e)} MASATO YOSHIMI^{1,f)}
TSUTOMU YOSHINAGA^{1,g)}

Received: December 21, 2012, Accepted: May 30, 2013

Abstract: As 3D-stacked silicon technology grows, the significant increase of performance/power balance of 3D-stacked processor is expected. Exploiting 3D-stacked design, long wires that are not shrunk by process scaling can shrink geometrically, which essentially reduce the interconnect power that is the major part of the power dissipation. However, existing 3D module-mappers have not reflected switching activity to cost functions of wires; moreover, their outputs of 3D-microprocessor floorplans have not been revealed. This study introduces novel 3D module-mapper which reflects communication patterns to the cost function by collaborating with pipeline simulator, and reveals the floorplan and its effects to the 3D-stacked processor architectures. Our result showed efficient mapping of 3D data path and cache structures. With the condition of 3-layer, assuming the wire load of TSV as same as $30\mu\text{m}$ of the normal wire load, compared to that of 2D floorplan, it requires 34% footprint and shows 57% “Wire-Activity” value that represents interconnect power dissipation, which is improved by 10% from the result of existing 3D floorplanners.

Keywords: 3D-stacked VLSI, microprocessor, floorplanner

¹ 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, The University of
Electro-Communications, Chofu, Tokyo 182-8585, Japan

^{†1} 現在, 株式会社ディー・エヌ・エー
Presently with DeNA Co., Ltd.

a) irie@is.uec.ac.jp

b) houchi@comp.is.uec.ac.jp

c) inaba@comp.is.uec.ac.jp

d) majima@comp.is.uec.ac.jp

e) Dz-Fujiwara@comp.is.uec.ac.jp

f) yoshimi@is.uec.ac.jp

g) yosinaga@is.uec.ac.jp

1. はじめに

計算機システムの中核となるプロセッサは、半導体デバイス技術とアーキテクチャ技術により急速な性能向上を続けてきた。近年、半導体デバイスの3次元実装技術が進展し、従来のテクノロジスケーリングに加え新たな性能向上の源泉が生じている。数百 μm 間隔で実装できるシリコン貫通電極 (Through-silicon via, TSV) により、積層ダイ間の高密度な接続が可能となり [1], [2], [3], 3次元設計は、従来のパッケージレベルからブロックレベル、セルレベルの細かい設計への進展が見込まれている [4]。

3次元実装の利点として、パッケージ内の有効面積を広げるほか、ロジック、メモリ、RF といった異なる半導体プロセスを柔軟かつ密接に接続できること、配線を幾何学的に縮め、配線電力を改善することがあげられる。克服すべき課題として、熱密度、製造テスト、TSVのコスト (1本あたりのコストは従来のワイヤボンディングよりも安価だが、活用により本数が増加すると考えられるため) があげられるが、モジュール配置による対策技術が提案されている [4], [5], [6]。さらに、TSVよりも高密度かつ柔軟な接続を可能とするモノリシック 3D 技術も研究が進められており [7], 半導体デバイスの3次元方向の活用はコスト・性能両面の利点からさかんに研究が進められている。

マイクロプロセッサ設計においても、3次元化によって従来の制約を超えた性能を得るための研究が進められている。DRAM プロセス積層によるキャッシュ大容量化、キャッシュの3次元積層によるフットプリント節約、3次元化したSRAMによる高性能キャッシュ、マルチコア積層など、マルチコアやメモリシステムに関する研究が多く行われている [8], [9], [10] ほか、プロセッサコアそのものの3次元化により10%以上の性能向上と10%以上の電力削減を同時に達成できるという見積りも報告されている [5], [11], [12]。プロセッサ3次元化の特に主要な利点として、ロングワイヤの電力削減があげられる。VLSIの配線はローカルワイヤとロングワイヤの2種類に分類でき、ロジック内にあってプロセス微細化に従って短くなるローカルワイヤに対し、モジュール間のインタコネクタであるロングワイヤの電力が、チップの主要な電力要因となっている [13]。配置配線の3次元化は、このロングワイヤの距離を幾何学的に縮め、チップの消費電力を本質的に削減する。このような3次元化の効果はテクノロジスケーリングとは直交しており、今後急速に活用が進むと考えられる。

しかし、プロセッサはメモリと異なりレギュラ構造ではないため、どのような3次元配置が良いかは自明でない。既存研究では、2次元設計で用いられている手法を3次元に拡張したフロアプランナをプロセッサのモジュール配置に導入しているが、どのようなフロアプランが得られたかは、我々の知る限り明らかにされていない [5], [13]。さら

に、これらの既存手法では、配線のコスト関数として総配線長が用いられている。配線電力は配線長とスイッチングアクティビティに依存するため、3次元配置による電力削減を目的とした最適化からは乖離している可能性がある。この問題に関して、プロファイルなどからスイッチングアクティビティを推定してコスト関数へ導入すれば、より良い配置結果が得られることが、いくつかの2次元の配置配線の分野では報告されている [14], [15]。

そこで本研究では、パイプラインシミュレータによる動的情報を利用し、配線のコスト関数に通信頻度の重みを加える最適化手法を提案する。このことにより、3次元化の利点の1つであるロングワイヤ電力削減をより推し進める。提案手法によって得られる3次元フロアプランを明らかにし、プロセッサモジュールの3次元配置傾向と、アーキテクチャへの効果について議論する。

以降、本論文は次のように構成される。2章では3次元フロアプランの設計技術および3次元積層プロセッサに関する研究を紹介する。3章では我々の提案する3次元モジュールマップの概要と新しいコスト関数について述べ、4章ではアルゴリズムの詳細について示す。5章では評価環境について述べる。6章では提案モジュールマップによって得られたフロアプランを示し、従来アルゴリズムとの比較および3次元化の効果について議論する。7章でまとめを述べる。

2. 3次元積層プロセッサの設計

2.1 3次元モジュールマップ

ICの3次元設計はテクノロジレベル、アーキテクチャレベル、回路レベルという各粒度の3次元分割・配置の組合せで構成される [4]。テクノロジレベルでは、平面ロジックプロセスで製造されたプロセッサにDRAMプロセスで製造されたメモリや、異なるテクノロジ世代で製造されたロジック回路を積層し、パッケージの性能バランスを最適化する。回路レベルではバンクスタッキング [9], [16] やポートスプリッティング [8] など、主にSRAM回路や多ビット演算器など規則的な構造を持つモジュールを3次元高性能化する。中間粒度であるアーキテクチャレベルでは、機能モジュールの3次元配置によりロングワイヤやバスを最適化する。アーキテクチャレベルの3次元設計では、広大な設計空間から各モジュールの配置最適化する3次元モジュールマップが用いられる。

従来用いられている2次元モジュールマップは、シーケンスペア [17] や B^* ツリーを用いて非スライス構造のフロアプランを記述し、シミュレーテッドアニーリング法 (SA) によりパッキング問題を探索する。最小化するコスト関数として面積のほか、チップの消費電力の主要部分を占めるロングワイヤ電力の削減を期して、モジュール間の総配線長が用いられる。

モジュールマップの3次元拡張では、フロアプランを3次元表現するTCG [18] やシーケンス・トリプル [19] 手法が存在するが、3次元積層IC設計の場合、積層方向(z軸)は他の方向(x, y軸)に比べて自由度が低いいため、より軽量の、層ごとの2次元配列を複数管理する手法が用いられることが多い [6], [20], [21], [22].

Shiuら [20] は3次元積層SiP (System in Package) において、シーケンスペアとSAを用いたモジュール配置手法を提案している。彼らはコスト関数として、面積、配線長、ビア数、配置制約ペナルティを組み合わせたものを提案している。Nainら [22] は3次元SRAMのような3次元モジュールに対応した配置手法、GSP (Grouped Sequence Pair) を提案している。温度の問題に関して、Congら [21] はコスト関数に温度評価を導入したCBA (Combined-bucket-and-2D-array) 方式を提案している。一方、最近のLiら [6] の手法では温度評価は計算量を増加させるとしてSAのイテレーションからは取り除き、得られたフロアプランに対して温度制約をチェックする方法を採用している。また、彼らはTSVの大きさを考慮した配置手法を提案している。

近年の高性能回路ではモジュールの使用頻度に応じてクロックゲーティングやパワーゲーティングなどが適用されるため、配線長と通信コストの相関が複雑となっている。Yehら [23] はパワーゲーティングを行うSoCを対象に、プロファイルを用いてパワーゲーティングを適用するモジュールを選択し、パワートランジスタの配置を最適化する手法を提案しているが、彼らのモジュールマップでは、従来手法と同様、配線長をコスト関数として用いている。

2.2 プロセッサコアの3次元化

プロセッサ3次元化に関しては、3次元SRAMやDRAMプロセス積層を利用してキャッシュやレジスタなどのメモリ要素を強化するLiuら [24] やPuttaswamyら [8], [9], 上野ら [10] の研究が行われている一方で、プロセッサコアのロジックを3次元化する研究は、設計の困難さからまだ優位性が一般的となっておらず、多くない。

Xieら [5] は3次元積層プロセッサについて、SRAM最適化、3次元モジュールマップ、IPCなどのアーキテクチャ評価、チップ試作を通じた総合的な議論を行っている。彼らの評価ではIA32アーキテクチャを2層で3次元実装した場合、従来の平面実装に比べて、性能/パワー効率率は50%以上向上すると見積もられている。また、彼らはCongら [21] の3次元ICモジュールマップをもとにして、温度評価の代わりにパワー密度の偏りをコスト関数に導入したモジュールマップを提案し、プロセッサ設計に適用している。しかし、彼らの性能評価ではヒューリスティックによって得たフロアプランが用いられており、提案されたモジュールマップによってどのようなフロアプランが得られたかは明らかでない。

Hungら [13] はパワー密度の偏り評価において、モジュールの消費電力だけでなくロングワイヤの消費電力を考慮する手法を提案している。彼らは、配線による発熱を考慮することで90nmプロセスで最大15度の温度上昇を防ぐことができるとしているが、Xieら同様、どのようなフロアプランが得られたかは明らかにされていない。コスト関数は、面積、配線長、電力密度の偏りの要素を足しあわせたものとなっている。手法の特徴である配線電力の見積りはDesign Compilerのプロファイルをもとに算出されているが、スイッチングアクティビティは考慮されず、総電力を配線長に応じた比で分配することにより、各モジュール間配線の電力を推定している。

3次元プロセッサのチップ試作はBlackら [11], [12] によっても報告されており、平面実装に比べて、最高温度を14度上昇させる代わりに15%の性能向上と15%の電力削減、あるいは最高温度を変化させずに8%の性能向上と34%の電力削減が見込まれるとしているが、フロアプランの設計ツールは明らかにされていない。

2.3 配線スイッチングアクティビティの考慮

チップ電力消費の主要部分を占めるロングワイヤ電力は一般に式(1)に比例する [15]。ここで SW_{ij} はモジュール*i*とモジュール*j*の間の配線のスイッチングアクティビティであり、配線の信号が0から1、あるいは1から0へ変わる頻度を示している。 C_{ij} は配線容量を示すが、モジュールマップの段階では主にモジュール間の配線長(モジュール間距離と配線ビット幅の積)によって決定される。これはゲート負荷容量はモジュール間配線容量に比べて無視できるほど小さいためである [25]。一方、 SW_{ij} 項について、配置配線のレイヤでは取得できない情報であるため、ロングワイヤ電力解析の典型的な手法では、この値を0.05から0.2の間の固定値、あるいは乱数とすることが多い [26], [27], [28].

$$\sum SW_{ij} \times C_{ij} \quad (1)$$

しかし、スイッチングアクティビティの傾向はモジュールによって異なるため、スイッチングアクティビティを一律と仮定する手法、つまり配線のコスト関数として総配線長で近似する手法では、実際の電力最適化からは乖離が生じる。このため、上位レイヤツールのプロファイルを利用してスイッチングアクティビティを推定し、コスト関数に組み込む手法が有効であることが2次元モジュールマップでは報告されている。Ekpanyapongら [14] は、SPECPCU2000から選んだ10本のプログラム実行時のプロファイルからスイッチングアクティビティを推定している。彼らは、重み付き配線遅延を線形計画法により最小化して、最大40%の性能向上を得る配置を得たとしている。また、Liuら [15] は、専用回路の配置配線において、デー

タフロアプランを解析して各エッジのスイッチングアクティビティが少なくなるように回路合成を行い、さらに配線のコスト関数にスイッチングアクティビティの重みを付けてSA法による配置を試みることで、ロングワイヤ電力を従来手法に比べて15%削減したとしている。

3. 3次元プロセッサ向けモジュールマップの提案

3.1 配線長へ重み付け

前章で見たように、現在、3次元積層チップのフロアプラン問題では2次元モジュールマップを3次元拡張した手法が様々に研究されており、プロセッサコアの3次元化についても研究が行われ始めている。しかし、既存の3次元モジュールマップでは配線のコスト関数にスイッチングアクティビティが考慮されておらず、また、どのようなフロアプランが得られたかも明らかになっていないため、将来の3次元プロセッサのアーキテクチャを議論することが難しくなっている。

プロセッサ3次元化の主要な利点の1つは、チップの主要な消費電力や配線遅延を占めているロングワイヤを幾何学的に縮め、チップの性能/電力バランスを本質的に改善することである。配置のコスト関数として、ロングワイヤの消費電力に近い値を用いることで、より良いフロアプランを得ることができる。このため、従来手法では一律に近似されている、各モジュール間のロングワイヤのスイッチングアクティビティ傾向を考慮することが有効である。そこで、我々はプロセッサシミュレータを用いて各ロングワイヤのスイッチングアクティビティを推定し、配線長に重み付けた“配線アクティビティ”を配線のコスト関数とするモジュールマップを提案する。また、提案コスト関数や従来のコスト関数によって、実際にどのようなフロアプランや効果が得られるかを明らかにし、将来の3次元プロセッサについて議論を行う。

プロセッサの各モジュール配置では、互いに多くのモジュールが接続して密な接続網を形成し、それらの通信頻度は一様ではない。モジュール間のスイッチングアクティビティ推定について、我々はこのモジュール単位の通信頻度に着目した。一般に、高性能プロセッサでは、モジュール間通信が必要ないサイクルにクロック/パワーゲーティングや値のホールドが適用されるため、通信頻度が低い配線はスイッチングアクティビティも小さくなる。モジュール間通信頻度はパイプラインシミュレータを利用すれば設計の初期段階から見積り可能であり、これをモジュール間配線長に対して重み付けた関数、配線アクティビティとして提案し、配線消費電力を反映する指標として用いる(式(2))。

$WireActivity$

$$= \sum Distance_{ij} \times Bitwidth_{ij} \times Usage_{ij} \quad (2)$$

ここで $Distance_{ij}$ はフロアプラン上のモジュール M_i とモジュール M_j 間の距離、 $Bitwidth_{ij}$ は M_i と M_j 間の接続ビット幅であり、 $Usage_{ij}$ が我々が新たに導入する、パイプラインシミュレータから得た $M_i M_j$ 間の通信頻度である。式(2)から $Usage_{ij}$ の積算を取り除くと、従来の配線長の式となる。

式(2)における $Usage_{ij}$ はモジュール間の有効な通信の有無で与えられるため、厳密にはスイッチングアクティビティとは異なり、近似値となる。たとえば、2サイクル連続でモジュールから同じ値を通信する場合、ビット単位で見れば反転がなく、ロングワイヤ電力が発生しない場合がありうるが、配線アクティビティでは、2回の通信があったとして重みが計測される。これは、ビットレベルの反転傾向を重みに反映しようとする、プロファイルしたベンチマーク入力値の影響が大きくなること、また高速性のためにラッチやダイナミック回路を用いる部分では同じ値が連続してもビット反転がありうることから、設計の初期段階の見積りとしては、プロファイリングをビット単位としても精度向上が少ないためである。一方、クロック/パワーゲーティングや値のホールドによるスイッチング頻度の現象は、モジュール間通信の有無で決定されるため、配線アクティビティ値はこれらの傾向を反映する。

3.2 提案システムの概要

提案システムの構成図を図1に示す。提案モジュールマップは、プロセッサを構成するレジスタ、キャッシュ、ALUなどのモジュールの粒度で最適な3次元フロアプランを探索、出力する。各モジュールは平面矩形で表され、同じ層の中では重ならないように配置される。

モジュールマップは大きく2種類の入力が必要とする。1つはパッケージのパラメータで、積層する層数およびTSVの配線コストを与える。もう1つはプロセッサ情報で、モジュールのリストとそれぞれの面積、およびすべてのモジュール間接続について、接続しているモジュール対、

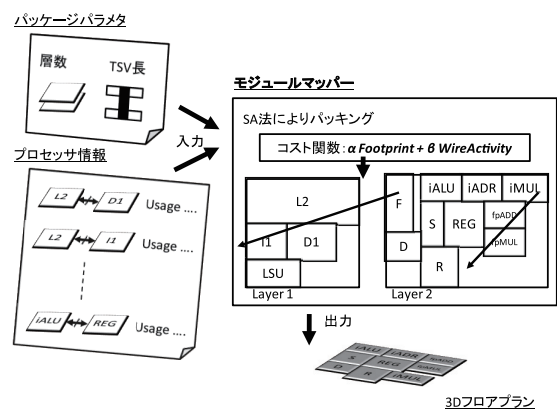


図1 提案手法の概要

Fig. 1 Outline of our Floorplanner.

ビット幅、通信頻度のリストで構成される。これらの情報はテクノロジーパラメータ、アーキテクチャモデル、パイプラインシミュレーション結果を用いてあらかじめ生成する。

モジュールマップは入力情報をもとにSAを用いてパッキング問題を最適化する。フロアプランを各層のシーケンスペアで表現し、イテレーションごとに揺動させて最適解へ接近する。最小化するコスト関数は、式(3)に示すように、面積とロングワイヤ負荷の双方を考慮する。

$$CostFunction = \alpha Footprint + \beta WireActivity \quad (3)$$

提案手法では従来手法に比べて、さらにロングワイヤ電力を削減したフロアプランを得ることができる。また、新しいプロセッサ設計について、アーキテクチャレベルの設計段階で3次元向けのフロアプラン傾向を得て、アーキテクチャパラメータにフィードバックすることが可能となる。

4. 配置アルゴリズム

4.1 SAの制御

提案モジュールマップは、関連研究[5], [13]同様、3次元フロアプランを各層のシーケンスペアで表現する。最初にランダムにシーケンスペアを作成してモジュールを各層に配置し、イテレーションごとにフロアプランへの小さな改良を企図した揺動を繰り返し最適化を進めていく。SAの特徴として、揺動によって解が一時的に悪化しても、「温度」に応じて確率的に悪化を受け入れることで局所最適解を避けることがあげられる。

提案モジュールマップのSAの制御の流れを図2に示す。まずランダムに初期配置を生成し、近傍解の生成とその採択の判断を繰り返しながら徐々に温度を下げていき、終了温度に達したときの配置を出力とする。各イテレーションでは、スワップ操作とソフトモジュール操作の中からランダムに1つを選び、近傍解を生成する。スワップ操作は、シーケンスペアの配置を交換することでモジュールの位置関係を変更する。ソフトモジュール操作は、面積を保ったままモジュールのアスペクト比を変形して周りの境界線と合わせ、ホワイトスペースを少なくする[29](図3)。なお、各モジュールは横:縦および縦:横の比が1:4を超えない範囲で変形されることとした。Xieら[5]の手法同様、配置探索の終盤になるほどソフトモジュール操作が選択される確率が高くなるように、アルゴリズム選択の確率に重みを付ける。揺動操作の詳細は以下の8種類となっており、この中から各イテレーションで1つが選択される。ここでは、その時点のフロアプランについて、最もWireActivity(式(2))の増加に寄与しているエッジの両端のモジュールを M_1 および M_2 と呼ぶことにする。

- M_1 を同じ層または別の層のランダムなシーケンスペア位置へ移動させる。
- M_2 を同じ層または別の層のランダムなシーケンスペア

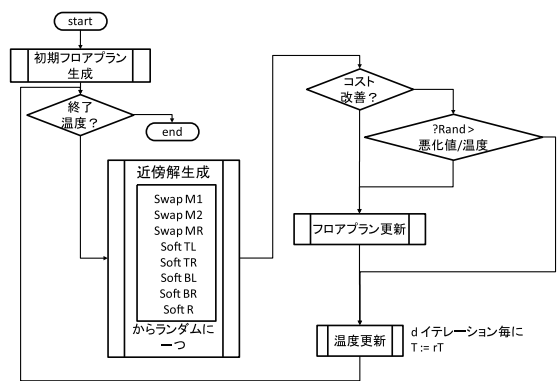


図2 モジュールマップアルゴリズム

Fig. 2 Flowchart of our module mapper algorithm.

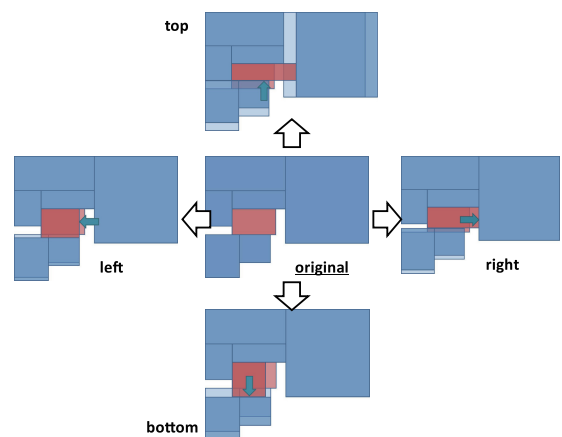


図3 ソフトモジュール最適化

Fig. 3 Soft module optimization.

ア位置へ移動させる。

- ランダムに選ばれたモジュール M_R を同じ層または別の層のランダムなシーケンスペア位置へ移動させる。
- M_R の下辺が上方向に移動するようアスペクト比を変更し、境界線を揃える。
- M_R の右辺が左方向に移動するようアスペクト比を変更し、境界線を揃える。
- M_R の下辺が下方向に移動するようアスペクト比を変更し、境界線を揃える。
- M_R の右辺が右方向に移動するようアスペクト比を変更し、境界線を揃える。

この入れ替え操作によって得られた遷移候補のコスト関数が改善された場合はフロアプランを更新する。悪化した場合は、 $0 < Rand < 1$ なる乱数 $Rand$ を生成し、式(4)が満たされれば採択、更新する。ここで、 $Cost()$ はSAで最小化を目指すコスト関数、 $mapB$ は揺動によって得られた近傍解、 $mapA$ は揺動前の配置とする。温度が高いほど、悪化したフロアプランでも採択する確率が高くなる。冷却スケジューリングは典型的な冷却関数に従い、 d 回イテレーションごとの $T := rT$ 更新とした。 d および初期温度の決定法については評価環境の章で述べる。

$$Rand > \frac{Cost(mapB) - Cost(mapA)}{Temperature} \quad (4)$$

4.2 コスト関数

提案モジュールマップはSAのコスト関数として面積とロングワイヤ負荷の双方を考慮した式(3)を用いる。式(3)中のWireActivityは式(2)で与えられるので、代入して、

$$CostFunction = \alpha Footprint + \beta \left(\sum Distance_{ij} \times Bitwidth_{ij} \times Usage_{ij} \right) \quad (5)$$

を各フロアプランについて計算する。

面積は図4のように、各層を重ねあわせたときにプロセッサ全体を含めることができる最小の矩形として計算する。各モジュール間の配線距離について、まずxy平面上の配線長は、広く用いられている half-perimeter bounding box モデルを用い、配線の周囲を囲う矩形の外周長の半分とした。次に、z軸方向の配線長は、またぐ層の数にTSVの配線コストを表すパラメータ、TSVweight をかけることにより与えられる。TSVによる配線長には、TSVそのものの長さに加えてバンプや再分配層、TSV間隔の制限によるオーバーヘッドが加算される。また、電気特性について、他のTSVとの間に生じる容量やTSVを囲う絶縁体を通して基板との間に生じる容量が発生し、さらに高周波帯ではL成分の影響が大きくなるなど、通常の配線とは異なる特性を持つ[2]。TSVweightはこれらのオーバーヘッドを考慮し、TSV1本あたりの配線コストが、どれだけの長さの通常配線と等しいかを表すパラメータとなる。

イテレーション内演算の軽量化のため、ロングワイヤはつねに最短距離で配線されると仮定し、2モジュール間(M_i, M_j)の配線長は以下の式(6)で計算される。ここで M_{ix}, M_{iy} は M_i の重心のx座標およびy座標、 M_{iz} は M_i の配置される層番号を示している。 M_j についても同様に定義する。

$$Distance_{ij} = |M_{ix} - M_{jx}| + |M_{iy} - M_{jy}|$$

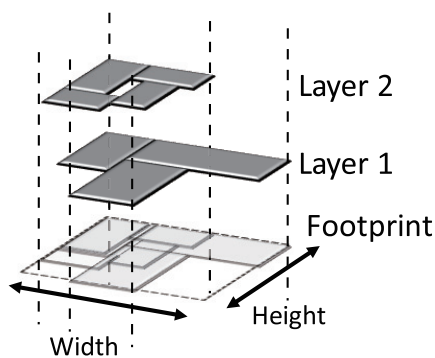


図4 フロアプラン面積の見積り方
Fig. 4 The estimation of the footprint.

$$+ TSVweight \times |M_{iz} - M_{jz}| \quad (6)$$

残るBitwidth_{ij}とUsage_{ij}はプロセッサ情報として入力から与えられた値を利用する。Usage_{ij}はパイプラインシミュレータを利用し、各モジュールのヒット/ミスや処理命令数の統計から、全モジュール間エッジの通信頻度を推定して作成する。 α と β は面積とロングワイヤの最適化の重みを調整する係数である。

提案アルゴリズムではアウトライン制約は用いず、フットプリントの大きさやアスペクト比はフロアプランナが最適配置とした値に合わせる設計を仮定している。チップ内のプロセッサ領域が決まっているアウトライン制約が必要な場合は、Unified convex optimization (UFO) [30]のような設計方法が知られている。

5. 評価環境

5.1 比較モデル

提案するモジュールマップをJava version 1.6.0.21環境上に実装し、フロアプランの取得を行った。実験に用いたパッケージの設定パラメータを表1に示す。従来モジュールマップで用いられているコスト関数との比較のため、以下の3種類のコスト関数について実験を行った。比較対象となる2種類は、いずれも2次元および3次元のICモジュールマップにおいて、一般的に用いられているコスト関数である。

- F：面積のみを考慮
- F+WL：面積と配線長を考慮
- F+WA：提案モデル、面積と通信頻度の重み付き配線長を考慮

式(5)における α と β の係数は、最適化が進んだときの面積と配線長要素の重みが等しくなるよう、コスト関数とパッケージパラメータごとに決定した。具体的にはモンテカルロシミュレーションによるランダム配置を行い、各要素について最も良かった数値の比を用いる。係数決定のためのモンテカルロ試行回数は10,000,000回とした。

5.2 アーキテクチャパラメータ

配置を行うプロセッサモデルとして、DEC Alpha 21264 [31]を想定し、入力ファイルを作成した。配置するモジュールの接続図を図5に示す。モジュール間通信頻度の情報は、プロセッサシミュレータ鬼斬2 rev.4240 [32]上に集計ルーチンを実装し、SPEC CPU2006ベンチマーク全プログラムについて先頭10G命令スキップ、続く100M命令の実行について各モジュール間の通信頻度を計測し、

表1 パッケージ設定

Table 1 Settings of parameters.

層数	1 - 3
TSVweight	30 μ m, 100 μ m4

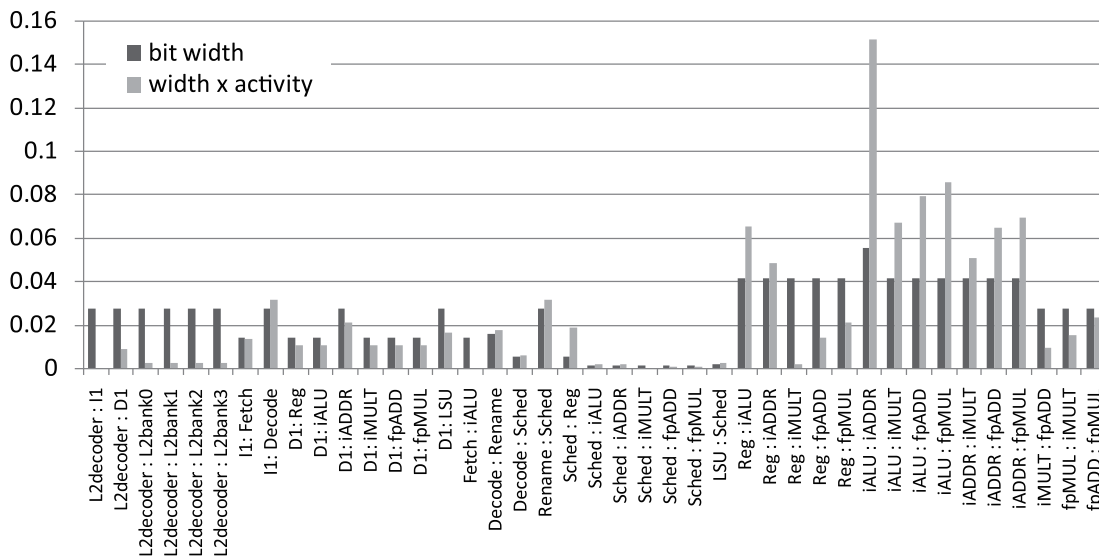


図 6 モジュール間配線の重み
Fig. 6 Weight of the long wires.

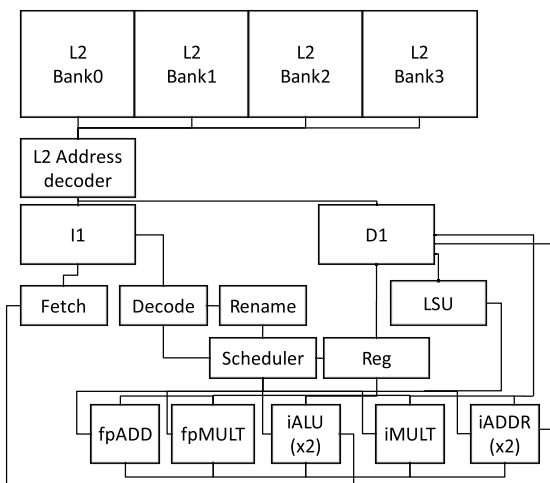


図 5 配置するプロセッサモジュールと接続図
Fig. 5 Modules and their connection diagram.

作成した。

図 6 は、得られた通信頻度の重みを示したものである。横軸に通信エッジのあるモジュールペアを列挙し、縦軸にそのエッジの重みの相対値（全エッジの重みの総和を 1 とする）を示している。グラフでは、従来手法であるエッジビット幅の重みと、提案手法である通信頻度が考慮された重みの 2 つの値が示されている。図 6 の L2decoder:L2bank0 などの値を見ると、ビット幅の重みとアクティビティの重みでは差があり、低次キャッシュでは確保されたビット幅に対して通信頻度は多くないことが分かる。一方で、Reg:iALU 以降のデータパスの項目に着目すると、通信頻度を考慮した場合に重みが顕著に増加するエッジがあることが分かる。各モジュールのサイズは Kessler [31] および Lewis ら [4] の報告をもとに、アーキテクチャ構成はそのまま、28 nm プロセスにスケールしたと見積もった値を設

表 2 SA パラメータ
Table 2 SA parameters.

	開始温度	終了温度
F (1 layer)	5000000	0.0125
F (2 layer)	4000000	0.01
F (3 layer)	3000000	0.0075
F+WL (1 layer)	10000000	0.025
F+WL (2 layer)	8000000	0.02
F+WL (3 layer)	8000000	0.02
F+WA (1 layer)	150000000000	375
F+WA (2 layer)	60000000000	150
F+WA (3 layer)	40000000000	100

定した。

5.3 SA パラメータ

実験では同じパラメータのものについてそれぞれ 100 回のフロアプラン生成を行い、最もコスト関数の低い結果を採用した。初期温度はパラメータごとにコスト関数の収束の様子を予備取得し、最も深い局所解のコスト関数値から十分抜け出せるような値をそれぞれ設定した。パラメータが異なっても SA イテレーション回数を同じとするため、終了温度と開始温度の比はすべてのパラメータで一定とした。冷却スケジューリングについて、 r の値を 0.9 とし、温度更新間隔 d は、近傍解の数のオーダーに従い、モジュール数の 3 乗と揺動数の積とした。設定パラメータを表 2 に示す。

6. 評価結果

6.1 2次元フロアプランにおける検証

まず、現行のプロセッサと同様の 1 層、平面構成について、我々のモジュールマップによって得られたフロアプラン結果を図 7 に示す。フロアプラン図は、メモリサブアー

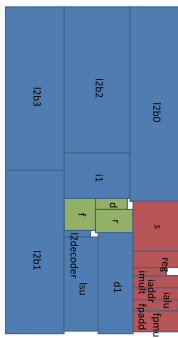
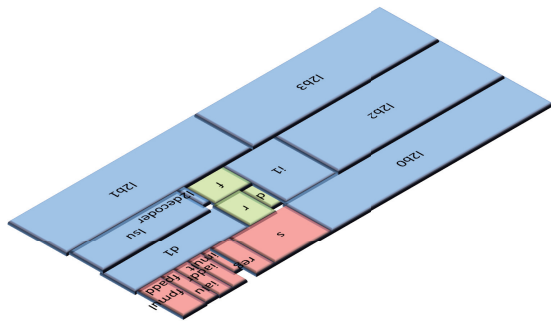


図 7 1層, F+WA
Fig. 7 1 layer, F+WA.

キテクチャ、フロントエンド、バックエンドについてそれぞれ色分けし、密接に関係するブロックどうしの配置が分かりやすいようにした。最も密な通信が行われるバックエンド部分が密集して配置され、効率の良いデータパスを形成している。またフロントエンド部分もまとめて配置されている。命令キャッシュはフロントエンドに、データキャッシュはデータパスに隣接して配置され、後方のL2キャッシュへと続いている。ホワイトスペースも少なく、パッキングの最適化が行われていることが分かる。L2のサブアレイに特別な制約をいれていないため、それぞれ独立したアスペクト比で配置されている点を除けば、図7は2次元プロセッサに典型的なブロック配置が得られている。

同様に、2次元のプロセッサについて、面積のみ (F) および面積と配線長の考慮 (F+WL) のコスト関数によって取得したフロアプランをそれぞれ図8と図9に示す。面積のみを考慮して得られた図8では提案手法に比べてさらに少ないホワイトスペースを実現している。しかし、モジュール間の接続を考慮していないため、フロントエンドおよびバックエンドは広がって配置されており、プロセッサの配置としては効率が悪くなっている。配線長が考慮された図9では関連する機能モジュールがまとめて配置されているが、提案手法ほど徹底した配置とはなっておらず、フロントエンドとバックエンドが入り交じっている。これは、通信頻度の重みをいれない従来手法では、分岐予測ミス時のみにアクティブとなるパス (バックエンドとフェッチロジック) の配線コストが、提案手法に比べて相対的に大きく見積もられているためと考えられる (図6)。また、

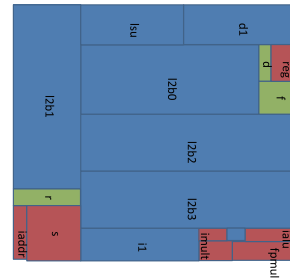
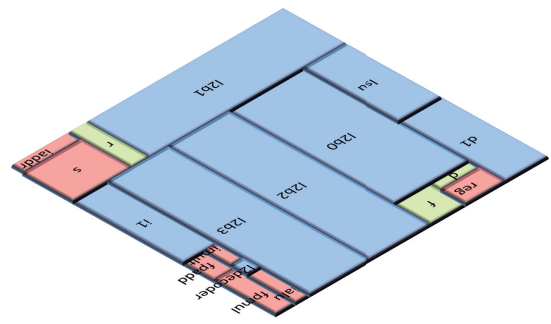


図 8 1層, F
Fig. 8 1 layer, F.

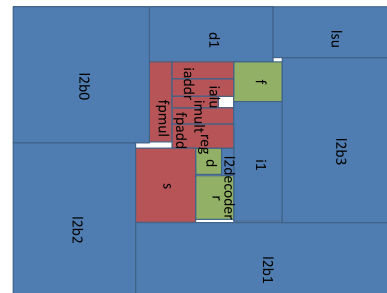
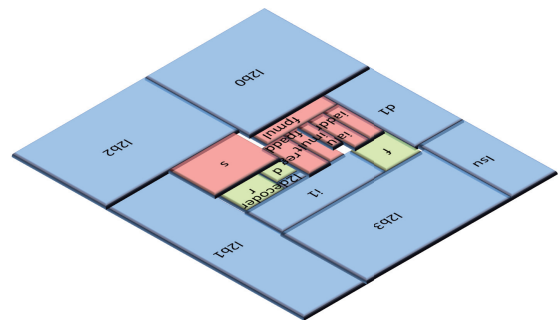


図 9 1層, F+WL
Fig. 9 1 layer, F+WL.

提案手法に比べL2キャッシュとの配線の重さが大きくなるため、すべてのL2バンクの距離を縮めるために対称的な配置が選択されている。以上より、提案手法は既知のプロセッサフロアプランと同様のモジュール配置を出力でき、プロセッサ向けの最適化は他のコスト関数に比べて優れていることが分かる。

6.2 提案手法によるロングワイヤ電力削減

次に、3次元化および提案コスト関数による効果を確認する。図10は各モデルについて得られたフロアプランの配線アクティビティ値 (式(2)) の値を示している。1層,

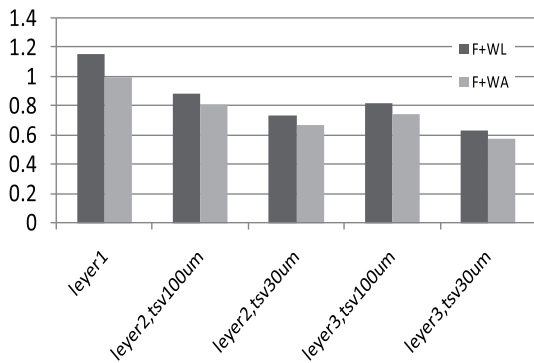


図 10 3次元フロアプランの効果 (配線アクティビティ)
Fig. 10 The effect of 3D floorplan (Wire activity).

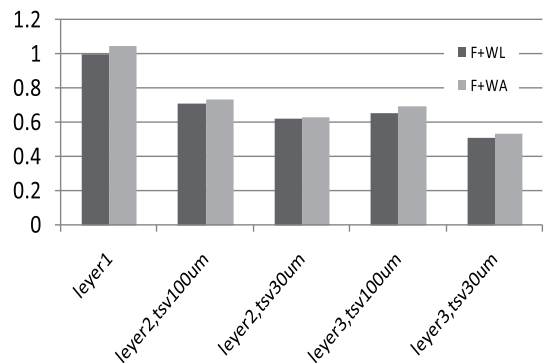


図 11 3次元フロアプランの効果 (総配線長)
Fig. 11 The effect of 3D floorplan (Total longwire length).

2層, 3層, および2層以上の場合は2種類のTSVweightの条件をX軸に示し, また, Y軸には配線アクティビティ値を, 1層のときのF+WAモデルによる値を1とした相対値で示している. 前述のように, この値はチップの電力消費の主要部分を占めるロングワイヤ電力 [15] を近似している. まず, 提案モデルF+WAについて, 3次元化の効果に着目すると, 1層のときに対して, 2層TSVweight 100 μm のモデルでは80%, TSVweight 30 μm のモデルでは66%にまで削減している. 2層から3層への改善は緩やかだが, 3層TSVweight 30 μm のときに57%となっている. 次に, 従来手法F+WLと比較すると, 提案手法はすべてのパッケージパラメータについて10%から14%の改善となっており, 3次元化の利点であるロングワイヤ電力削減に適した手法であることが分かる.

一方, 提案手法では配線長に通信頻度の重みを考慮するため, 純粋な総配線長については悪化することが予想される. 図 11 は得られたフロアプランの配線長コストを, 1層F+WLモデルのときの値を1として正規化して示したグラフである. 悪化量はすべてのパラメータについて6%以内であった. なお, 1層に対し2層, 3層の配線長は30%から50%の短縮となっており, Xieら [5] がAlphaアーキテクチャに対して行った見積りにおける短縮値38%や, Hungら [13] が行った同様の見積りにおける38%と同様な結果となっている.

フロアプランの配線領域について考えた場合, 総配線長からz軸方向の寄与分を除いた値によって近似される. 図 11 において, z軸方向の寄与分はF+WAモデル, F+WLモデルともに各パラメータで同じ値を示し, 2層TSVweight 100 μm のときに15ないし19%, 2層TSVweight 30 μm のときに7%, 3層TSVweight 100 μm のときに20%, 3層TSVweight 30 μm のときに12%などとなった. このため, 配線領域に寄与するxy方向の総配線長に注目した場合も, 提案手法による悪化は少なかったといえる. また, モジュール間配線の重なりを確認したところ, 積層される各チップにおいて, それぞれ3層の配線層をモジュール間通信で使用できれば, 最短距離での接続が可能であり, 最

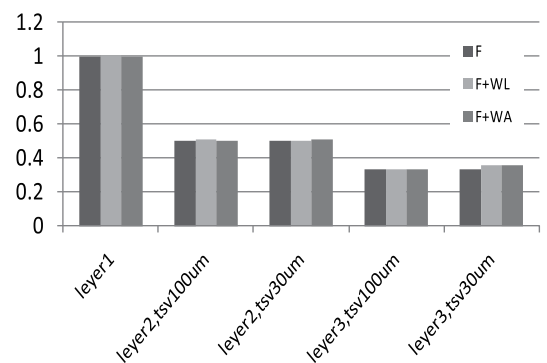


図 12 3次元フロアプランの効果 (面積)
Fig. 12 The effect of 3D floorplan (Footprint).

終的な配置配線結果でも同様の傾向と考えられる. ビット幅の重みと配線アクティビティ値の重みの乖離が大きい場合は, 配線領域の最適化品質を維持するために, コスト関数として配線アクティビティ値と配線長の双方を用いる手法も考えられる.

最後に, 面積について確認する. 図 12 はフロアプランの面積コストを, 1層, Fのときの値を1とした相対値で示している. 値が小さいほど面積を小さくパッキングできていることになる. 面積のみを優先するため, 各設定でFが最も良い結果となっているが, 他のコスト関数でも同様の面積効果が得られている. モデルF+WAでは3次元化することにより面積は2層で51%, 3層で34%まで縮小されており, 層数に応じた縮小効果が得られており, 配線アクティビティを考慮することによって面積効率が劣化することとはなっていない. この値もXieら [5] の見積りにおける50%, Hungら [13] の見積りにおける52.6%と同様の結果が得られている.

6.3 取得された3次元フロアプラン

ここでは, 得られたフロアプランを確認する. 3次元モジュールマップによるマイクロプロセッサの配置結果は, 我々が知る限りこの報告が初めてである. 2層, TSVweight 100 μm の設定時に得られたフロアプランを図 13 に示す. 2層になってもホワイトスペースの少ないパッキングが得

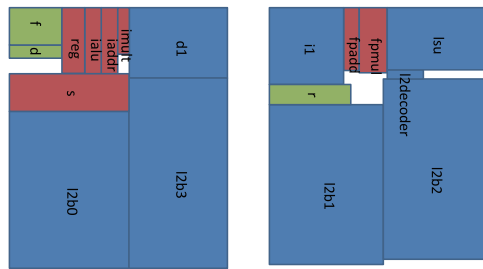
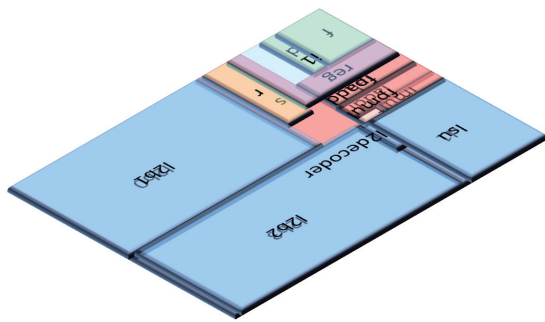


図 13 2層, F+WA, TSVweight 100 μm
 Fig. 13 2 layer, F+WA, TSVweight 100 μm .

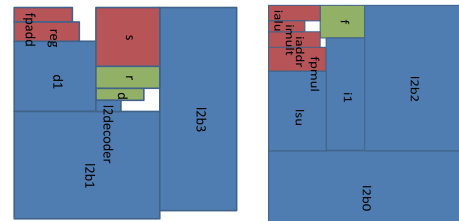
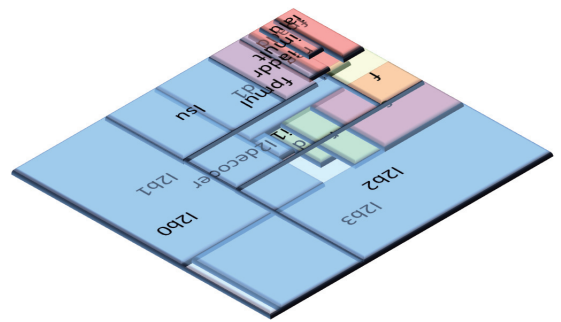


図 15 2層, F+WL, TSVweight 100 μm
 Fig. 15 2 layer, F+WL, TSVweight 100 μm .

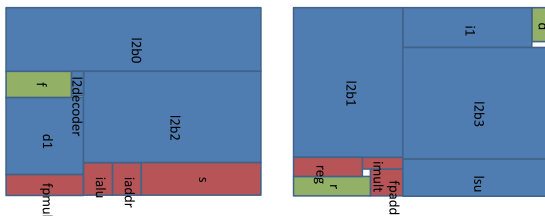
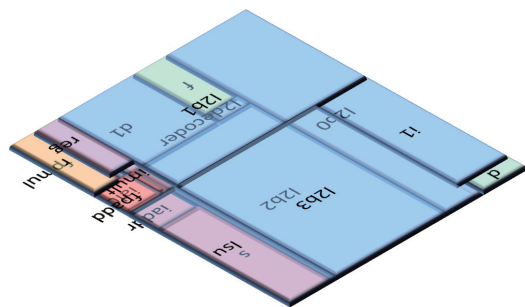


図 14 2層, F, TSVweight 100 μm
 Fig. 14 2 layer, F, TSVweight 100 μm .

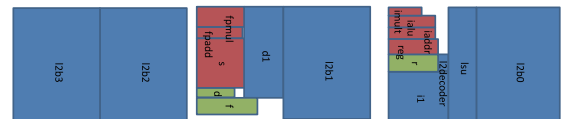
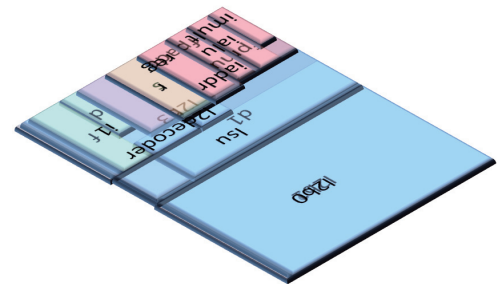


図 16 3層, F+WA, TSVweight 100 μm
 Fig. 16 3 layer, F+WA, TSVweight 100 μm .

られており、面積が効率良く使われることが分かる。バックエンド部分は fp と int に分かれ、2層に重なったコンパクトな形成が得られている。

同様に、2層、TSVweight 100 μm 設定時の従来コスト関数フロアプランについて、図 14 にコスト関数 F、図 15 にコスト関数 F+WL の結果を示す。1層のとき同様、面積のみを考慮した場合（図 14）ではさらにホワイトスペースの少ない配置が得られているが、フロントエンドおよびバックエンドは広い範囲に配置され、通信効率が悪くなっている。面積と配線長を考慮した場合（図 15）では、バックエンドが集中していることが確認できるが、提案手法による図 13 の方が密集度は高い。1層時同様、フェッチロ

ジックがデータバスの近くに寄せられ、また L2 キャッシュのバンクの等距離配置が優先されていることが分かる。

図 16 は、3層、TSVweight 100 μm としたときの提案手法によるフロアプランである。3層の場合もホワイトスペースは少なく、面積が効率良く使われている。z 軸方向を利用することにより狭い面積にすべてのデータバスが構築され、配線が短縮されている。また、L2 キャッシュのサブアレイは積み重ねられ、大容量のキャッシュを効率良く実装できることが分かる。SA によるコスト関数の収束の様子を図 17 に示す。開始時直後は振れ幅が大きく、イテレーションの進展に従って最適コストを徐々に下げながら振れ幅が収束していく SA の挙動となっている。

6.4 プロセッサ 3 次元化とアーキテクチャ

以上に述べた 3 次元フロアプランおよびコスト関数の結果をもとに、プロセッサ 3 次元化によるアーキテクチャへの影響について議論する。関連研究 [5], [11], [12] でも指摘

されているように、3次元化によりパッケージ内で利用できる面積はほぼ層数に比例して増加する。また、隣接するモジュール数を増やすことができ、ルーティングが向上している。このことはコア数の増加や大容量キャッシュ、メインメモリ積載に有利に働き、メニーコアアーキテクチャをより推進する。

一方、面積の使い方として大容量キャッシュは性能向上への寄与が低いことが指摘されている [33]。また、コア数がメニーコアと呼ばれるオーダへ入ってくると、十分なスレッド並列性が存在する場合でも、従来のようにコア数に比例した性能向上が得られなくなるという報告も行われている [34]。キャッシュ容量、コア数のほかに3次元化が寄与する要素としては、データパスの配線が効率化することによるバックエンドのレイテンシ短縮、あるいはバック

エンド幅の向上があげられる。また面積の使い方としてキャッシュではなくレジスタやリオーダーバッファへ資源を振り分ける設計が考えられる [35]。

6.5 アーキテクチャによるフロアプラン見積りの変化

本手法の特徴の1つとして、プロセッサ・アーキテクチャによるフロアプランの変化を設計初期段階で見積もることができる点があげられる。ここでは、これまでの評価に用いた図5のプロセッサにプリフェッチを導入した場合の変化を観察する。簡単のため、プリフェッチャはLSUに含まれ、このことによるLSUの面積増はないものとする。プリフェッチではキャッシュの転送頻度は増えるが、接続のビット幅は変化しない。このため、転送頻度を考慮しない従来のフロアプランの場合、プリフェッチの有無にかかわらず同じフロアプランが出力される。

I1, D1, L2 キャッシュにそれぞれプリフェッチを適用し、図6と同様にシミュレータによりモジュール間配線の重みを取得したものを図18に示す。プリフェッチアルゴリズムは単純なシーケンシャルプリフェッチとし、変化を分かりやすくするために、積極的な先読みを行った。具体的にはI1, D1 キャッシュではミスアドレスの16ライン先まで順次読み出しを行い、L2 キャッシュではミスアドレスの64ライン先までプリフェッチを行うものとした。図18からは、プリフェッチの導入によって、キャッシュを含むエッジの重みが増加していることが分かる。

提案手法によって得られたフロアプランを1層(図19)、2層(図20)、3層(図21)についてそれぞれ示す。これらはTSVweight 100umとした場合のフロアプランであ

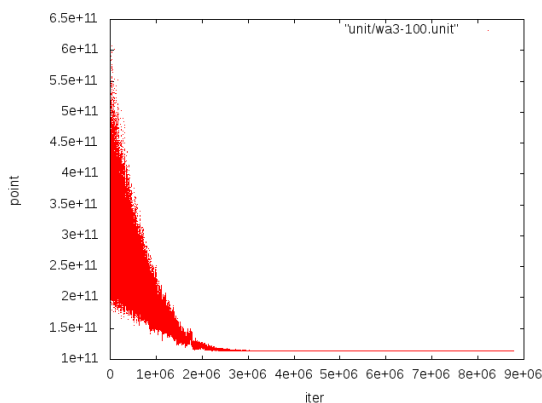


図17 コスト関数の収束の様子(3層, TSVweight 100 μ m, F+WA)
 Fig. 17 The Convergence of Cost Function (3layer, F+WA, TSVweight 100 μ m).

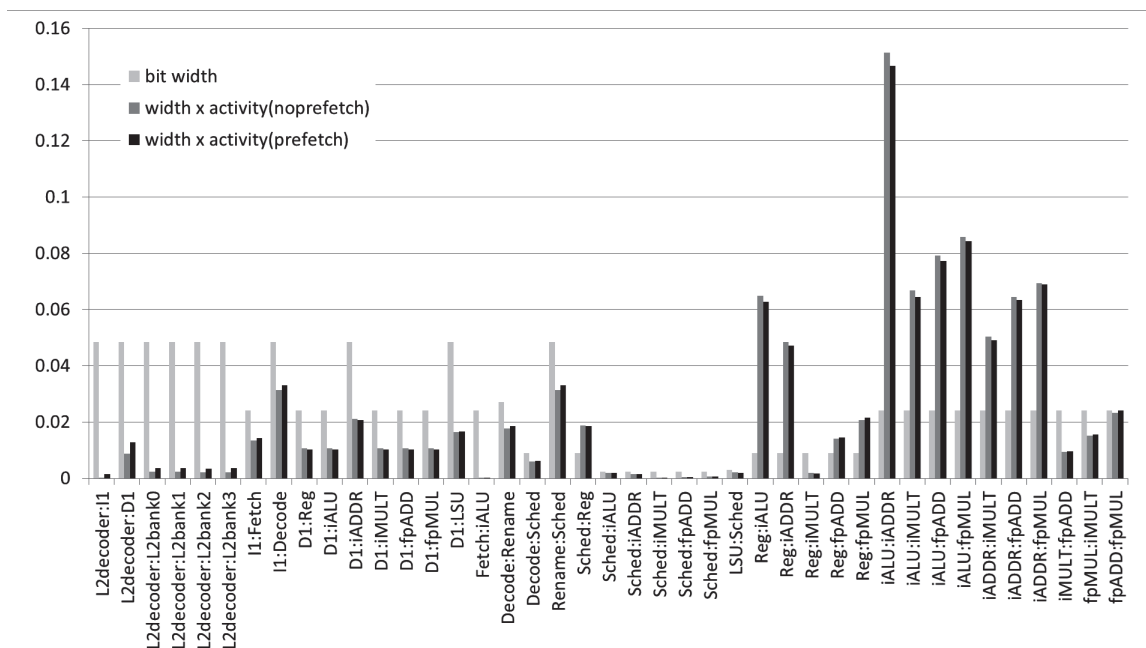


図18 モジュール間配線の重み(プリフェッチ有り)
 Fig. 18 Weight of the long wires (with prefetch).

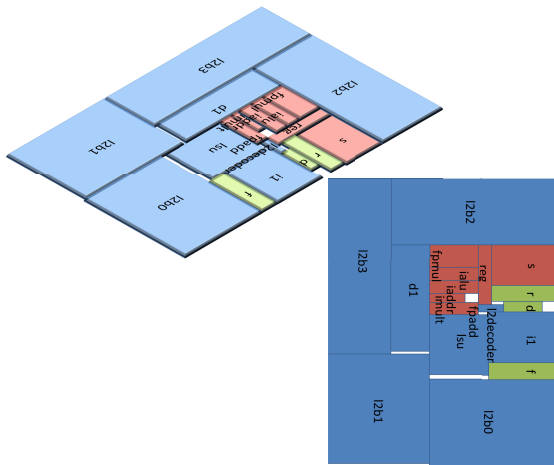


図 19 プリフェッチ有り, 1層, F+WA
 Fig. 19 With prefetch, 1 layer, F+WA.

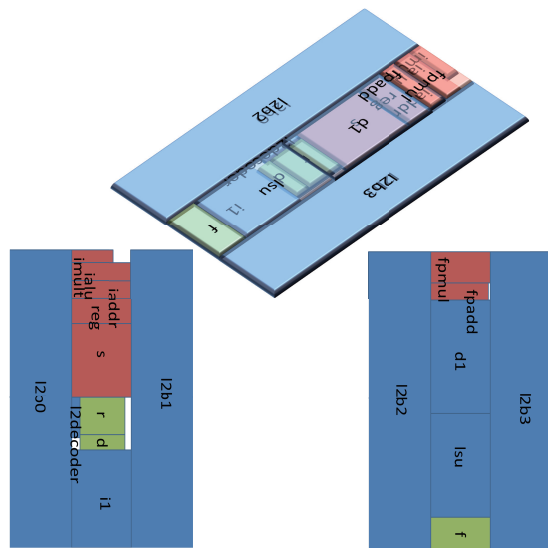


図 20 プリフェッチ有り, 2層, F+WA, TSVweight 100 μm
 Fig. 20 With prefetch, 2 layer, F+WA, TSVweight 100 μm.

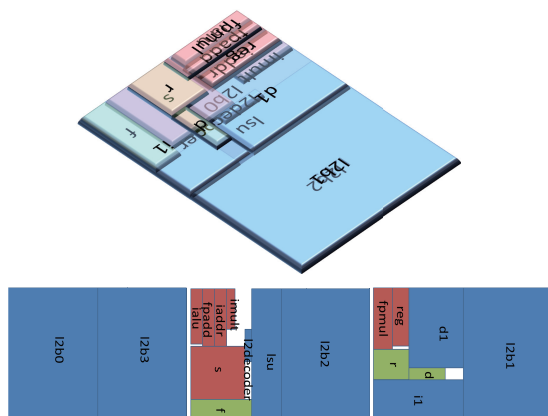


図 21 プリフェッチ有り, 3層, F+WA, TSVweight 100 μm
 Fig. 21 With prefetch, 3 layer, F+WA, TSVweight 100 μm.

り, それぞれ図 7, 図 13, 図 16 に対応している。
 これらの配置では, プリフェッチが適用されないときのフロアプランに比べると, キャッシュの配置が優先されて

いる様子が分かる. たとえば, 1層や2層のフロアプランでは, L2 キャッシュの各バンクが実行コアから等距離になるように配置され, 極端に遠いバンクが存在しない. 2層のフロアプランでは特に顕著な最適化が行われている. また3層のフロアプランにおいても, L2の各バンクに接続される L2decoder が中央の層に配置されるように変化しており, よりキャッシュが優先された配置となっている. これらのフロアプランの変化はプリフェッチによるキャッシュトランザクション増に対応した変化となっており, シミュレータによる重み取得により, 適したフロアプランが得られることが分かる.

7. まとめ

本論文では 3D プロセッサのためのモジュールマップを提案し, プロセッサ 3次元化のアーキテクチャへの影響を議論した. 半導体 3次元実装技術の進展は著しいが, プロセッサモジュール配置の 3次元化はまだ研究は多くない. 既存の 3次元モジュールマップでは配線のコスト関数として総配線長が用いられているが, 電力解析の研究では, チップ電力消費の主要部分を占めるロングワイヤ電力の削減のために, 配線のスイッチングアクティビティを考慮することが有効とされている. 本論文では, プロセッサシミュレータと連携して, モジュール間の配線頻度を推定し, 配線長への重みとするモジュールマップを開発した.

Alpha21264 アーキテクチャと SPEC CPU2006 実行時のワークロードを入力とし, 提案システムによる配置最適化を行った. 得られた 3層, TSVweight 30 μm フロアプランは, 2次元構成に対して面積を 34%, ロングワイヤ電力を 57%にまで削減した. また, 提案コスト関数は従来の最適化に対して 10%以上のロングワイヤ電力を削減し, 従来の 3次元プロセッサの見積りよりも最適化を進めることが分かった. また本論文では, 自動配置によって得られる 3次元プロセッサフロアプランを, 我々の知る限り初めて明らかにした. 提案モジュールマップはアーキテクチャ設計の早い段階でのフロアプラン見積りが可能であり, 新アーキテクチャの見積りに有用である.

今後の課題として, マルチコアプロセッサの最適配置や温度シミュレーションの反映, 遅延スラックの最適化があげられる. 提案システムはシミュレータと連携しているため, 既存のシミュレータと組み合わせることで実装の拡張が可能である.

謝辞 本研究の一部は, 平成 24 年度服部報公会工学研究奨励援助金による.

参考文献

[1] Hsieh, A. and Hwang, T.: TSV Redundancy: Architecture and Design Issues in 3-D IC, *Trans. on Very Large Scale Integration Systems*, Vol.20, No.4, pp.711-

- 722 (2012).
- [2] Kim, J., Pak, J., Cho, J., Song, E., Cho, J., Kim, H., Song, T., Lee, J., Lee, H., Park, K., Yang, S., Suh, M., Byun, K. and Kim, J.: High-Frequency Scalable Electrical Model and Analysis of a Through Silicon Via (TSV), *Trans. on Components, Packaging, and Manufacturing Technology*, Vol.1, No.2, pp.181–195 (2011).
- [3] Pak, J., Kim, J., Cho, J., Kim, K., Song, T., Ahn, S., Lee, J., Lee, H., Park, K. and Kim, J.: PDN Impedance Modeling and Analysis of 3D TSV IC by Using Proposed P/G TSV Array Model Based on Separated P/G TSV and Chip-PDN Models, *Trans. on Components, Packaging, and Manufacturing Technology*, Vol.1, No.2, pp.208–219 (2011).
- [4] Lewis, D. and Lee, H.-H.: A scanisland based design enabling prebond testability in die-stacked microprocessors, *Test Conference, 2007, IEEE International*, pp.1–8 (2007).
- [5] Xie, Y., Loh, G., Black, B. and Bernstein, K.: Design space exploration for 3D architectures, *ACM Journal on Emerging Technologies in Computing Systems*, Vol.2, No.2, pp.65–103 (2006).
- [6] Li, C., Mak, W. and Wang, T.: Fast Fixed-Outline 3-D IC Floorplanning With TSV Co-Placement, *IEEE Trans. Very Large Scale Integration Systems*, pp.1–10 (2012).
- [7] Batude, P., Vinet, M., Pouydebasque, A., Royer, C.L., Previtali, B., Tabone, C., Hartmann, J., Sanchez, L., Baud, L., Carron, V., Toffoli, A., Allain, F., Mazzocchi, V., Lafond, D., Delenibus, S. and Faynot, O.: 3D monolithic integration, *Int. Symp. on Circuits and Systems*, pp.2233–2236 (2011).
- [8] Puttaswamy, K. and Loh, G.: Implementing Register Files for High-Performance Microprocessors in a Die-Stacked (3D) Technology, *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures*, pp.384–389 (2006).
- [9] Puttaswamy, K. and Loh, G.: 3D-Integrated SRAM Components for High-Performance Microprocessors, *IEEE Trans. Computers*, Vol.58, No.10, pp.1369–1381 (2009).
- [10] 上野伸也, 橋口慎哉, 福本尚人, 井上弘士, 村上和彰: 3次元積層 LSI 向け SRAM/DRAM ハイブリッドキャッチャー・アーキテクチャ, 情報処理学会論文誌 コンピューティングシステム (ACS), Vol.5, No.1, pp.41–52 (2012).
- [11] Black, B., Nelson, D., Webb, C. and Samra, N.: 3D Processing Technology and Its Impact on iA32 Microprocessors, *Int. Conf. on Computer Design*, pp.316–318 (2004).
- [12] Black, B., Annavaram, M., Brekelbaum, N., DeVale, J., Jiang, L., Loh, G.H., McCauley, D., Morrow, P., Nelson, D., Pantuso, D., Reed, P., Rupley, J., Shankar, S., Shen, J. and Webb, C.: Die Stacking (3D) Microarchitecture, *Int. Symp. on Microarchitecture*, pp.469–479 (2006).
- [13] Hung, W., Link, G., Xie, Y., Vijaykrishnan, N. and Irwin, M.: Interconnect and thermal-aware floorplanning for 3D microprocessors, *Int. Symp. on Quality Electronic Design*, pp.104–109 (2006).
- [14] Ekpanyapong, M., Minz, J., Watwai, T., Lee, H. and Lim, S.: Profile-guided microarchitectural floorplanning for deep submicron processor design, *Design Automation Conference*, pp.634–639 (2004).
- [15] Liu, Z., Bian, J., Zhou, Q., Yang, L. and Wang, Y.: Interconnect Power Optimization Based on the Integration of High-level Synthesis and Floorplanning, *Int. Conf. on Communications, Circuits and Systems*, pp.2286–2290 (2006).
- [16] Li, F., Nicopoulos, C., Richardson, T., Xie, Y., Vijaykrishnan, N. and Kandemir, M.: Design and Management of 3D Chip Multiprocessors using Network-in-Memory, *Int. Symp. on Computer Architecture*, pp.130–141 (2006).
- [17] Murata, H., Fujiyoshi, K., Nakatake, S. and Kajitani, Y.: VLSI module placement based on rectangle-packing by the sequence-pair, *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.15, No.12, pp.1518–1524 (1996).
- [18] Bazargan, K., Kastner, R. and Sarrafzadeh, M.: 3D Floorplanning: Simulated Annealing and Greedy Placement Methods for Reconfigurable Computing Systems, *Int. Workshop on Rapid System Prototyping*, pp.38–43 (1999).
- [19] Yamazaki, H., Sakanushi, K., Nakatake, S. and Kajitani, Y.: The 3D-packing by Meta Data Structure and Packing Heuristics, *IEICE Trans. Fundamentals*, pp.639–345 (2000).
- [20] Shiu, P., Ravichandran, R., Easwar, S. and Lim, S.: Multi-layer Floorplanning for Reliable System-on-Package, *Int. Conf. on Circuits and Systems*, Vol.5, pp.69–72 (2004).
- [21] Cong, J., Wei, J. and Zhang, Y.: A thermal-driven floorplanning algorithm for 3d ics, *Int. Conf. on Computer-Aided Design*, pp.306–313 (2004).
- [22] Nain, R. and Chrzanowska-Jeske, M.: Fast Placement-Aware 3-D Floorplanning Using Vertical Constraints on Sequence Pairs, *IEEE Trans. Very Large Scale Integration Systems*, Vol.19, No.9, pp.1667–1680 (2011).
- [23] Yeh, C., Chen, H., Huang, L., Wei, W., Lu, C. and Liu, C.: Using power gating techniques in area-array SoC floorplan design, *Int. SOC Conference*, pp.233–236 (2007).
- [24] Liu, C., Ganusov, I., Burtscher, M. and Tiwari, S.: Bridging the processor-memory performance gap with 3D IC technology, *Design Test of Computers, IEEE*, Vol.22, No.6, pp.556–564 (2005).
- [25] Predictive Technology Model (PTM), available from (<http://www.eas.asu.edu/~ptm/>).
- [26] Sylvester, D. and Keutzer, K.: Getting to the bottom of deep submicron, *Int. Conf. on Computer-Aided Design*, pp.203–211 (1998).
- [27] Zhang, L., Chen, H., Yao, B., Hamilton, K. and Cheng, C.: Repeated On-Chip Interconnect Analysis and Evaluation of Delay, Power, and Bandwidth Metrics under Different Design Goals, pp.251–256 (2007).
- [28] Qiu, X., Ma, Y., He, X. and Hong, X.: IPOSA: A Novel Slack Distribution Algorithm for Interconnect Power Optimization, *Int. Symp. on Quality Electronic Design*, pp.873–876 (2008).
- [29] Chi, J. and Chen, M.: An Effective Soft Module Floorplanning Algorithm Based on Sequence Pair, *Int. Conf. on ASIC/SOC*, pp.54–58 (2002).
- [30] Lin, J. and Hung, Z.: UFO: Unified Convex Optimization Algorithms for Fixed-Outline Floorplanning Considering Pre-Placed Modules, *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.30, No.7, pp.1034–1044 (2011).
- [31] Kessler, R.: The Alpha 21264 Microprocessor, *IEEE Micro*, Vol.19, No.2, pp.24–36 (1999).
- [32] 塩谷亮太, 五島正裕, 坂井修一: プロセッサ・シミュレータ「鬼斬式」の設計と実装, 先進的計算基盤システムシン

ポジウム 2009 ポスター (2009).

- [33] Lotfi-Kamran, P., Grot, B., Ferdman, M., Volos, S., Kocberber, O., Picorel, J., Adileh, A., Jevdjic, D., Idgunji, S., Ozer, E. and Falsafi, B.: Scale-Out Processors, *Int. Symp. on Computer Architecture* (2012).
- [34] Bhadauria, M., Weaver, V. and McKee, S.: Understanding PARSEC Performance on Contemporary CMPs, *Int. Symp. on Workload Characterization*, pp.98-107 (2009).
- [35] Irie, H., Fujiwara, D., Majima, K. and Yoshinaga, T.: STRAIGHT: Realizing a Lightweight Large Instruction Window by using Eventually Consistent Distributed Registers, *Int. Workshop on Challenges on Massively Parallel Processors*, pp.336-342 (2012).



入江 英嗣 (正会員)

1999年東京大学工学部電子情報工学科卒業。2004年同大学院情報理工学系研究科電子情報学専攻博士課程修了。博士(情報理工学)。2004年科学技術振興機構CREST研究員, 2008年東京大学情報理工学系研究科助教を経て,

2010年より電気通信大学大学院情報システム学研究科准教授。計算機システムおよびその応用の研究に従事。特に汎用プロセッサアーキテクチャ, ディベンダブルシステム, HCIの研究を進めている。情報処理学会山下記念研究賞(2010), 同学会論文賞(2010)。電子情報通信学会コンピュータシステム研究会幹事(2011~)。電子情報通信学会, IEEE, ACM各会員。



放地 宏佳

2009年富山商船高等専門学校情報工学科卒業。2011年豊橋技術科学大学知識情報工学課程卒業。2013年電気通信大学大学院情報システム学研究科修士課程修了。現在, 株式会社ディー・エヌ・エーにて開発者支援に従事。



稲場 朋大

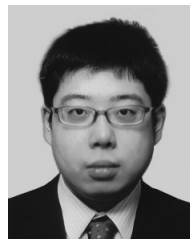
2013年電気通信大学電気通信学部電子工学科卒業。同年より電気通信大学大学院情報システム学研究科情報ネットワークシステム学専攻修士課程にて3次元積層プロセッサの研究に従事。2013年度スーパー連携大学院プログラ

ム受講生。



眞島 一貴

2011年電気通信大学電気通信学部情報通信工学科卒業。2013年同大学院情報システム学研究科情報ネットワークシステム学専攻修士課程修了。プロセッサアーキテクチャの研究に従事。



藤原 大輔

2012年電気通信大学電気通信学部情報通信工学科卒業。現在, 同大学院情報システム学研究科情報ネットワークシステム学専攻修士課程在学中。プロセッサアーキテクチャの研究に従事。



吉見 真聡 (正会員)

2004年慶應義塾大学理工学部情報工学科卒業。2009年同大学院理工学研究科開放環境科学専攻後期博士課程修了。博士(工学)。2006年度より日本学術振興会特別研究員(DC1)。2009年より同志社大学理工学部助教を経て,

現在, 電気通信大学大学院情報システム学研究科助教。リコンフィギャラブルシステム, 並列処理の研究に従事。人工知能学会会員。



吉永 努 (正会員)

1986年宇都宮大学工学部情報工学科卒業。1988年同大学院修士課程修了。同年より宇都宮大学工学部助手。1997年から翌年にかけて電子技術総合研究所客員研究員。2000年電気通信大学大学院情報システム学研究科助教授。

現在、教授、博士(工学)。計算機アーキテクチャ、並列分散処理等に興味を持つ。ACM, IEEE, 電子情報通信学会各会員。