

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 フロアプランナーのための熱評価手法

稲場 朋大^{1,a)} 放地 宏佳^{2,b)} 藤原 大輔^{2,c)} 眞島 一貴^{2,d)} 吉見 真聡^{2,e)} 入江 英嗣^{2,f)}
吉永 努^{2,g)}

概要 :

半導体の電力性能比を向上させる技術として、プロセスの微細化に加え、3次元積層が有効に働くことが報告されている。マイクロプロセッサの設計においても3次元化による様々な利点が期待される一方で、積層が生じさせる熱の扱いは重要な課題となっている。我々は、スイッチングアクティビティに基づいて配線長を決定し、省電力なフロアプランを得る3次元用モジュールマッパーを開発している。本研究報告では、熱密度のバランスと最大温度の削減を目的として、フロアプランナーに熱評価を導入し、得られた配置について議論する。

Introducing Thermal Cost Function to Wire-Activity-Aware 3D-stacked Processor Floorplanner

Abstract:

The 3D-stacked silicon technology is reported to improve performance-power ratio of semiconductor along with microfabrication technology. While the 3D-stacked technology is also expected various advantages to design microprocessors, it is growing important problem to control heat-generation caused by layered structure. Our research group is developing a 3D module mapper based on switching activity to obtain optimal wire length for a electrical power saving floorplan. This report discusses layouts from our floorplanner introducing evaluation of heat-generation to achieve balancing heat density and reducing maximum temperature.

1. はじめに

半導体3次元積層技術の進展により、プロセスの微細化とは異なるアプローチによる性能向上が可能となった。3次元化の利点として、集積素子数の増加、フットプリントの減少、ローカル配線の短縮による高速化、総配線長の減少によるインターコネクト消費電力の削減が挙げられる。

また、TSV(Through-silicon via)を用いることで、ロジック、メモリ、異なるプロセス世代などの異なる半導体プロセスを柔軟かつ密接に接続でき、高いメモリバンド幅の使用が可能となる [1], [5].

一方、3次元化の問題点として、熱密度の増大が挙げられる。プロセッサ中の1点でも動作温度を超えてしまうとプロセッサ全体が動作不可となるため、現在のプロセッサ設計では発熱が主要な制約要件となっている。2次元実装よりも熱密度が増大する3次元実装ではより大きな問題となる。熱の増大を防ぐ方策として、プロセッサ中の熱がヶ所に集中しないようにすること、発熱の大きなモジュールはヒートシンクの近くに配置するなどの手法がとられる。

アーキテクチャレバルのマイクロプロセッサを3次元設計では、機能モジュールの適切な3次元配置が問題となる。しかし、柔軟な3次元実装の設計空間は広く、プロセッサのような複雑な構成を設計する手法はまだ確立されてい

¹ 電気通信大学電気通信学部
Faculty of Electro-Communications, The University of
Electro-Communications
² 電気通信大学情報システム学研究所
Graduate School of Information Systems, The University of
Electro-Communications
a) inaba@comp.is.uec.ac.jp
b) houchi@comp.is.uec.ac.jp
c) Dz-Fujiwara@comp.is.uec.ac.jp
d) majima@comp.is.uec.ac.jp
e) yoshimi@is.uec.ac.jp
f) irie@is.uec.ac.jp
g) yoshinaga@is.uec.ac.jp

ない。

本研究では、3次元プロセッサ向けに開発したブロックレベルでのモジュールマッパーに熱評価関数を取り入れることで、発熱に配慮したフロアプランを行い、得られた配置を議論する。

以降、第2章では3次元フロアプランナーと3次元デバイス中の熱に関する研究を紹介する。第3章では、本稿で提案する熱を考慮したフロアプランナーについて述べ、第4章ではそのアルゴリズムを示す。第5章で実験の評価環境について述べ、第6章で得られたフロアプランを示し、議論する。第7章でまとめを述べる。

2. 関連研究

2.1 3次元プロセッサモジュールマッパー

CPUやLSIシステムの高密度化、性能向上のため、様々な実装技術が開発されてきた。しかし、トランジスタを平面上に配置する2次元的な実装では性能向上が限界に達しつつある。プロセスの微細化においては、配線がボトルネックとなるためである。微細化とは別方向の発展として3次元実装は有望な技術である [10]。

3次元積層技術の簡単な利用法として、マイクロプロセッサの上に大量のキャッシュを積層したり、個々のモジュールを3D化しフットプリントの削減を行うなどの技術が挙げられる [2]。さらに、柔軟で広い設計空間を持つ3次元実装の最適な設計により、電力削減を始めとした3次元化をより活かすため研究がなされている。プロセッサコアの3次元化では、モジュールレベルの配置を考えるためにモジュールマッパーが使用される。モジュールの配置の是非を判定する評価関数としては配線、フットプリント、熱、TSVの製造コストなどが用いられる [2], [12]。

2.2 3次元積層時の熱の問題

3次元化に伴う配線長の削減による消費電力の低減、TSVによる排熱効率の上昇が得られるが、一方で立体的な細密配置による熱密度の上昇が懸念されており、3次元積層時の熱は重要な問題として研究がなされている [1], [12]。

Lohらは3次元積層時の層数が多いほど熱が深刻な問題となることを示した [7]。また、Puttaswamyらはレイヤの素材を考慮し、配置することで排熱効率が上がることを示した [9]。Congらはシリコン、銅と比較して、絶縁体として用いられる樹脂は熱伝導率が低いため、垂直方向の熱放散を考える必要性は高いと指摘している [12]。また、花田らの報告では、3次元化の恩恵をより多く受けるためには、処理を行うモジュールが偏らないよう、実行プログラムを並列化するソフトウェア側のアプローチも必要であることが示されている [6]。

2.3 熱を考慮したフロアプラン

プロセッサにおける熱の問題は消費電力の問題と同一である [2]。熱を増大させないための操作の方向として、消費電力密度の大きなモジュール (頻繁に使用されるモジュール) をヒートシンクの近くに配置すること、熱密度が偏らないようにすること、最大温度が動作可能範囲に収まるようにすることが挙げられる [2], [6], [7], [8], [9]。

3次元積層回路中に消費電力密度、熱密度がどのように分布しているかを分析することは難しい。Congらは回路全体をグリッド分割し、電力密度、熱密度の分布の情報をグリッドに持たせることで抽象化し、フロアプラン時の熱の解析に活用するを提案した [12]。

Hungらはインターコネクットの消費電力を考慮に入れたフロアプランニングを行っている [3]。

フロアプランにおいて熱を見積もる手法として、イテレーション中に詳細な熱解析を行なうもの [2]、近似的に見積もるもの [2], [12]、得られたフロアプランに対して詳細な熱解析を行い、そのフロアプランを採用するかどうか決めるものがある。

3. 提案手法

3.1 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ー

我々はブロックの配置を層毎のシーケンスペアで表現し、シミュレーテッドアニーリング (SA) によって最適解を探索する3次元モジュールマッパーの開発を行なっている。

入力としてモジュールのフットプリント、モジュール間の通信負荷、バス幅をとり、最適解を探索したのち、そのモジュール配置を返す。評価関数としてフットプリントと配線アクティビティを使用している。

$$CostFunction = \alpha FootPrint + \beta WireActivity \quad (1)$$

我々のコスト関数では、ロングワイヤ電力の削減を最適化するために、配線長に加え、スイッチングアクティビティの重みを考慮する。スイッチングアクティビティの取得にはより上位のプロファイルが必要であるが、我々はパイプラインシミュレータによりモジュール間の通信頻度を取得することにより実現する。

3.2 熱評価フロアプランニング

本論文では、第3.1章のモジュールマッパーの評価関数に熱を加え、配線アクティビティに加え、熱を考慮したフロアプランニングを行う。追加のプロファイルとして、モジュール毎の消費電力を与え、フロアプラン中の熱の偏りを減らすような評価を行う。

$$Cost = \alpha FootPrint + \beta WireActivity + \gamma Heat$$

提案システムを図1に示す。

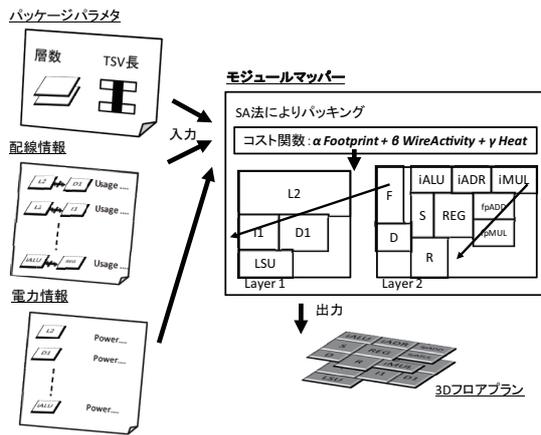


図 1 提案手法の概要

Fig. 1 The overview of proposed methodology

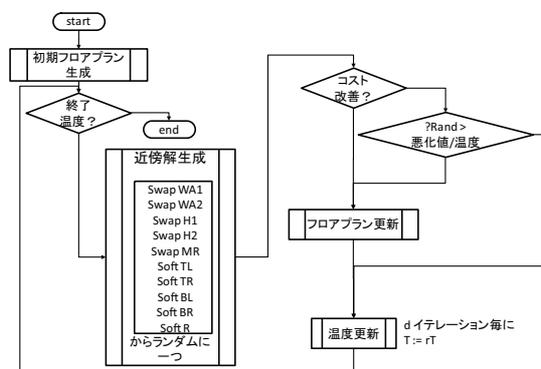


図 2 モジュールマッパーアルゴリズム

Fig. 2 The algorithm for module mapper

提案モジュールマッパーはフットプリント、TSV 長といったパッケージパラメータと、モジュール間通信負荷、モジュール毎の消費電力といったプロセッサ情報を入力とし、SA により最適解を探索する。

4. アルゴリズム

4.1 シミュレーテッドアニーリング

図 2 に本実験で使用したマッパーの概要を示す。

シミュレーテッドアニーリングは最適解が確定していない解空間を探索し、妥当な解を得るアルゴリズムである。当マッパーはシーケンスペアで表現されたブロック配置に対し、シミュレーテッドアニーリングを適用することで最適解を探索する。

はじめにランダムに解を生成し、ブロック配置をコスト関数で評価し値を得る。以後、以下に示す揺動を繰り返し、評価値が改善する場合は必ずその配置を採用、悪化する場合はその時点の温度に応じた確率で採用する。

揺動は以下の項目から採択する。

- 最も通信負荷の高いモジュールの位置を入れ替える
- 最も熱が高いモジュールの位置を入れ替える
- ランダムなモジュールの位置を入れ替える
- ランダムなモジュールのアスペクト比を調整する

これらの操作を SA の温度が終了温度に達するまで繰り返す。評価値が悪化した場合にその配置を採用する確率は式 2 で与えられる。乱数 $Rand$ は $0 < Rand < 1$ を与える。ここで、 $Cost$ は SA の評価関数であり、 $mapB$ は得られた近傍解、 $mapA$ は揺動前の配置である。この確率は実行序盤では大きく取られ、収束するに従って小さくなる。

$$Rand > \frac{Cost(mapB) - Cost(mapA)}{Temperature} \quad (2)$$

4.2 熱評価アルゴリズム

マッパーへの入力として、モジュール毎の消費電力を与える。関連研究に挙げた Cong らの方法同様、3次元プロセッサをグリッド分割し、グリッド毎に熱の評価を行う。グリッドの縦横の長さはフットプリントを 10 分割したものとした。グリッドの熱評価は、以下のように行う。まず、グリッドに含まれるモジュールについて、消費電力を足し合わせる。ここで、モジュールの配置が複数グリッドにまたがる場合は、フットプリントに応じた割合で該当グリッドへ分配する。次に、 x, y 位置を同じくするグリッドの z 方向の熱拡散に着目する。最上層のヒートシンクに向かって熱が拡散すると仮定し、ヒートシンクに至るまでの各層ごとに消費電力と熱抵抗を掛ける。式 3 は k 層目のグリッドについて、消費電力から熱を近似している。

$$GridHeat = \sum_{i=1}^k (R_i \sum_{j=i}^k P_j) + R_b \sum_{i=1}^k P_i \quad (3)$$

式 3 から求めたグリッドの熱をもとにプロセッサの熱の評価関数を作る。特に、式 4 の第一項はプロセッサ全体の熱の偏りを。第二項は層ごとの熱の偏りを表す。

$$Heat = \frac{H_{max} - H_{average}}{H_{average}} + \max\left(\frac{H_{kmax} - H_{kaverage}}{H_{kaverage}}\right) \quad (4)$$

4.3 評価関数

シミュレーテッドアニーリングで使用する評価として、フットプリント、配線アクティビティ、熱の 3 項を使用した。フットプリントはプロセッサ全体を含む最少の矩形である。配線アクティビティは配線距離 \times 通信頻度 \times バス幅である。この時、配線距離はモジュール重心間のマンハッタン距離を採用した。

各々の評価値が結果に均一な影響を与えるためには、重みを設定する必要がある。モンテカルロ法を試行回数 1,000,000 回で実行し得られた評価値が、(フットプリント):(配線アクティビティ):(熱)=1:1:1 となるように式 ?? の α, β, γ の値を決定した。

5. 評価環境

実験するにあたりプロセッサは alpha21264 を想定し、

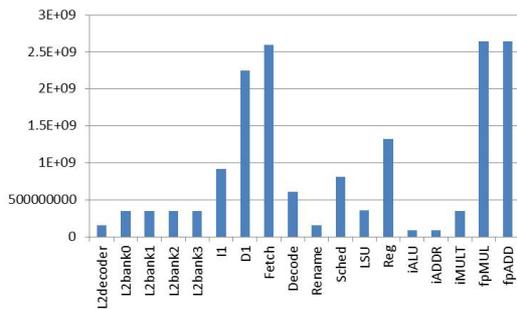


図 3 モジュール毎の消費電力

Fig. 3 The energy consumption by modules

これを java version 1.6.0.21 環境上に実装した. これを TSV100um, 3層という条件下で行った.

実験では今回提案する評価関数を用いる手法 (F+WA+H モデル) と熱を考慮するしない手法 (F+WA モデル) を比較した.

入力データとしてモジュールのフットプリント, 配線負荷, バス幅, 消費電力を用意した.

配線アクティビティを求めるための通信頻度は, プロセッサシュミレータ鬼斬 2 rev.4240 で SPECCPU2006 から 435.gromacs を選び, 先頭 10G スキップ, 続く 100M 命令を実行し, その実行統計から取得した. 消費電力はプロセッサシュミレータ sim-wattch に同じく SPECCPU2006 から 435.gromacs を先頭 10G スキップ, 続く 100M 命令を実行した. sim-wattch によるシミュレーションで得られたモジュール毎の消費電力を図 3 に示す. モジュールのフットプリントは 28nm プロセスを想定し, 設定した.

6. 評価結果

6.1 評価の流れ

まず, 熱のみを評価関数としたフロアプランニングを行い, 熱の評価関数の傾向を明らかにする. その上で, F+WA と, F+WA+H を評価関数とした場合の実行結果を示し, フットプリント, 配線アクティビティ, 熱についての比較を行う. 最後に実際に得られたフロアプランを提示する.

6.2 熱のみを評価関数とした場合のフロアプラン

図 4 は, 評価関数として熱のみを採用した場合のフロアプランである.

評価関数として熱のみを考慮する場合, フットプリントを大きくとって熱源を拡散させ, 発熱の大きいユニットがヒートシンク側に集まると考えられる. 図 4 を見ると, D1, Fetch, fpMUL, fpADD といった消費電力の大きなモジュールが最上層に集まっており, かつフロアプランのホワイトスペースが多いことから, 熱が考慮されていることが分かる.

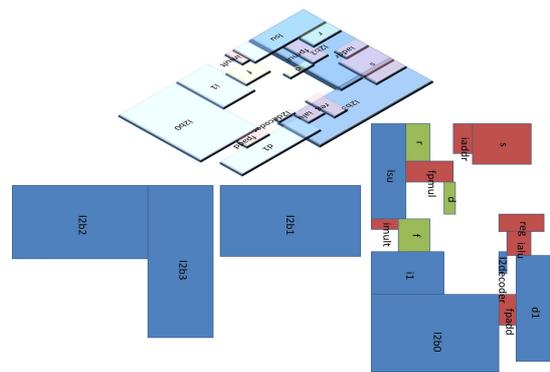


図 4 熱のみを評価関数として使用した場合のフロアプラン

Fig. 4 An floorplan by only the heat generation as an evaluation function

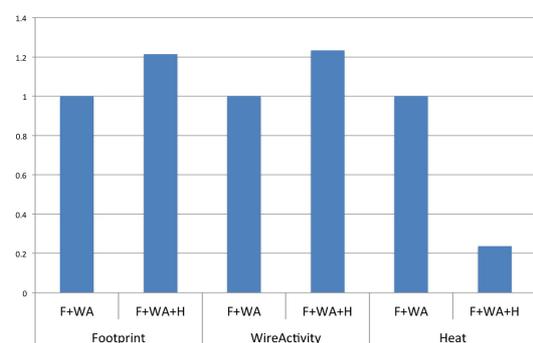


図 5 評価値の比較

Fig. 5 Comparisons of the evaluated values

6.3 熱評価を加えた場合の評価値の比較

図 5 は F+WA と F+WA+H で得られたフロアプランのフットプリント, 配線アクティビティ, 熱を比較したグラフである.

得られたフロアプランは熱要素を 76%削減する一方で, フットプリントおよび配線アクティビティの悪化は 25%以内だった. 接続位置関係を最適化したまま, ホワイトスペースなどによって適切に温度をさげるフロアプランが得られた.

6.4 実際に得られたフロアプラン

図 6, 図 7 は, それぞれ F+WA と F+WA+H の最も評価値の良かったフロアプランである.

データパスが int と fp の 2 層構成をとりつつ密集してコンパクトになっている L2 キャッシュの各バンクの大きさが揃えられ, フットプリントを最小化しようとしている

SA の収束は, 図 8 のようになった. 実行序盤は評価値の上下が激しく, イテレーションが進むにつれ振幅が小さくなっていき, 最終的に収束している.

7. まとめ

プロセッサ 3 次元化によって性能/電力バランスが改善することが期待されている. この研究では我々が提案して

