Vol.2011-ARC-195 No.10
Vol.2011-OS-117 No.10
2011/4/13

3次元積層向けブロック配置問題の検討

松村雄貴[†1]　三好健文[†1]
吉永努[†1]　入江英嗣[†1]

貫通電極を用いた3次元積層技術の進歩により，z軸方向を活用したプロセッサが現実的となってきた．面積増，配線短縮などによるプロセッサ性能のブレイクスルーが期待されている一方，プロセッサコアの3次元化やその設計手法，効果の研究は始まったばかりである．本研究では，プロセッサパイプラインシミュレータの実行結果をもとにブロックレベルの初期検討を行った．各ブロック及びブロック間配線の消費電力を見積もり，これを最小化することで3次元プロセッサ内のブロック配置を最適化する手法により，3次元プロセッサコアのデザインを得た．ブロックレベルで3次元化したプロセッサコアの簡単な配置最適化と見積もりでは，平面配置に対してブロック間通信負荷を約半分に削減できることが分かった．またブロックレベルであれば，最適化も現実的であることが分かった．

Examination of block arrangement problem on 3D integrated microprocessor

Yuki MATSUMURA ,[†1] Takeufmi MIYOSHI ,[†1]
Tsutomu YOSINAGA [†1] and Hidetsugu IRIE [†1]

By progress of 3D integrated technology with Through - Silicon - Via (TSV), processor which exploits z-axis becomes practical. It is expected that processor performance improves with increase of available hardware resource and reduction of wiring length. Meanwhile, the studies of 3D processors integration, designing methods for them, and the efficiency are just beginning. In this paper, discussion of block-level for 3D integrated processor is studied by the execution result of processor pipeline simulator, preliminarily. Power consumption of each block and wiring between blocks is estimated, and a 3D integrated processor core design is generated by an optimizing method for allocating blocks in the processor in manner of minimizing the estimated power consumption. It is shown that the estimated communication cost of blocks in the block-level optimized 3D integrated processor is about half as much as the cost of 2D integrated processor core. It is also found that the optimization of allocation at the block-level is practical.

1. はじめに

貫通電極 (TSV:Through - Silicon - Via) を利用した半導体3次元積層技術の進歩により，プロセッサやSoCの3次元化が注目を集めている．従来のワイヤボンディングによるマルチダイパッケージ と比べ貫通電極によるz軸方向の通信は，通常のビアサイズでの高密度実装が可能であり，また経路を短縮できるため通信負荷やレイテンシを大幅に効率化させることができる．このことから，3次元活用によるブレイクスルーが期待されている．例えば，3次元活用の例として，プロセッサに大容量メモリを積層するアプローチが研究されている[1)]．

3次元活用の利点は，単純にパッケージ内の面積が増えることだけではなく，3次元方向の接続性を活用して，ブロック間をより高速かつ低負荷に結び付けることができる点にある[2)]．プロセッサコアには，データパスをバイパスしあう演算器や多くのアクセスが集中するレジスタファイルなど，配線が集中するブロックが多数存在するため，適切な3次元近接配置がもたらす性能・電力上の利点は計り知れない．新しい積層技術による3次元化がどのような効果をもたらすかという見積もりは，プロセッサアーキテクトにとって今後のアーキテクチャ研究を進める上で重要な興味の対象である．

しかし，柔軟な3次元実装の設計空間は広く，プロセッサコアをどのように3次元配置すればよいかという問題にはまだ定まった答えがない．そこで本論文では，ブロックレベルの簡単な3次元配置を検討し，初期見積もりを行う．パイプラインシミュレータを用いて各ブロック間の通信回数を計測し，ブロック間通信負荷を通信回数と配置距離の積和として近似する．その上で，この通信負荷を最小化するように配置探索問題をモンテカルロ法により解き，3次元プロセッサの予想される配置とその効果を検討する．

以降，本論文は次のように構成される．第2章では貫通電極による3次元積層技術について述べる．第3章では3次元プロセッサコアの有効性を述べ，本論文における3次元配置の検討手法を述べる．第4章では実装した見積もりアルゴリズムについて詳細を述べる．第5章で実験パラメータ等の環境を述べ，第6章で結果を示す．第7章で関連研究について紹介し，第8章でまとめを行う．

2. 3D ブロック配置の概要

メモリやコアを貫通電極を用いダイを積層する事で高効率の3次元積層が実現出来るよう

†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

© 2011 Information Processing Society of Japan

Vol.2011-ARC-195 No.10
Vol.2011-OS-117 No.10
2011/4/13

になった。貫通電極とは半導体基板を貫通して形成するものであり、TSV(Through Silicon Via）とも呼ばれる。日本では、1998 年頃から研究されており近年実用化が注目されている技術であるが、複数のチップをワイヤボンディングで接続する従来の手法に比べて、配線距離を短縮できるため、高速化、省電力化、小型化などの面に向上が見込める。現在実用化に向けて開発されている貫通電極の直径及び長さは、10nm 程度及び 10μm 程度であり、数 μm -200 μm 間隔で配列出来る。国内では貫通電極を利用して大容量メモリを積んだ研究があるが、橋口ら[1] によると従来のプロセッサにくらべてアプリケーションの特性の違いに応じて性能向上があることが明らかにされている。

3. プロセッサコアの三次元化フロアプラン

3 次元積層プロセッサでは熱の問題が指摘されているが、その一方で 3 次元配置によって配線が短くなり、配線による電力消費が減って効率化するという主張もある。Loh ら[2] によるメモリのバンクを 3 次元化した研究や Black ら[3] によるプロセッサコアを 3 次元化した研究では消費電力の低下と性能向上が得られている。しかし，Loh，Black らのプロセッサコアの 3 次元化の研究では、どのようにフロアプランを設計し、どのようなフロアプランとなったかの詳細は議論されていない。また、その設計の難しさも指摘されている。

そこで初期検討として、自動探索によりブロックレベルでの 3 次元配置を求め、3 次元プロセッサのフロアプランがどのようになるかを調べる。また、3 次元化の効果を評価し、3 次元プロセッサに適したアーキテクチャの議論を行う。図 1 にその概要を示す。モジュール間の通信の緩和に注目し、ブロック間配線の長さと使用回数の積和を通信負荷として近似する。シミュレータで得た実行ログや通信回数に加え予め決定しておいた既存のブロックサイズを用いて見積もる。モジュール間の通信の緩和に注目し、ブロック間配線の長さと使用回数の積和を通信負荷として近似する。ブロック間配線の使用回数はパイプラインシミュレータにより求める。原始モンテカルロ法により通信負荷が小さくなる 3 次元ブロック配置を求める。

4. 実装したモデル

4.1 プロセッサモデル

out-of-order のスーパースカラプロセッサである Alpha 21264[4] を参考にプロセッサモデルを決定したものを図 2 に示す．各ブロックの大きさは，ダイサイズを従来の 180nm から現在プロセスの 28nm に換算するため 20mm × 20mm を 28/180 倍した場合の比から求めた．プロセッサは L1 データキャッシュ(D1)・L1 命令キャッシュ(I1)・L2 キャッシュ(L2)・レジスタ (REG)・演算器 (ALU)・フェッチャ(F)・デコーダ (D)・リネーマ (R)・スケジューラ (S)・ロードストアユニット (LSU) から構成される．

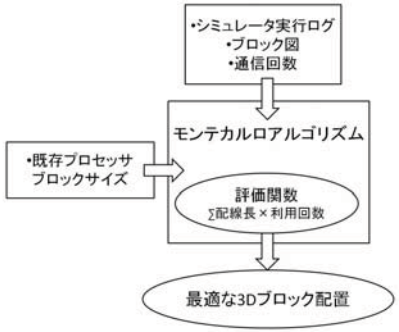


図 1 評価手法の概要
Fig. 1 Outline of evaluation approach

このプロセッサモデルをプロセッサシミュレータに対応させた．4.2 シミュレータログの対応で求めることができる．

4.2 シミュレータログの対応

プロセッサシミュレータの出力により各ユニット間でどれだけの通信が行われたかを計測することができる．例えば L1 データキャッシュのミスが発生すれば 8B のデータが 8 回 L2 から読みこまれることとなる．同様に，演算器の結果は各演算器やレジスタ，LSU へ転送される．このようにして，前節で挙げた全てのブロック間同士の通信回数を，シミュレータのログから得た．

4.3 評 価 関 数

プロセッサの消費電力は，プロセッサを構成する各ブロックモジュールで消費される電力とブロック間のデータ転送に必要な電力の和で得られる．本論文では，これらの消費電力のうち，ブロック間のデータ転送にかかる消費電力を 3 次元空間上でのモジュール間配置によって最小化することを目標とする．

ここで，モジュール M_i と M_j 間のデータ転送回数を $N_{active}(i,j)$，データ幅を $BW(i,j)$，

ⓒ 2011 Information Processing Society of Japan

Vol.2011-ARC-195 No.10
Vol.2011-OS-117 No.10
2011/4/13

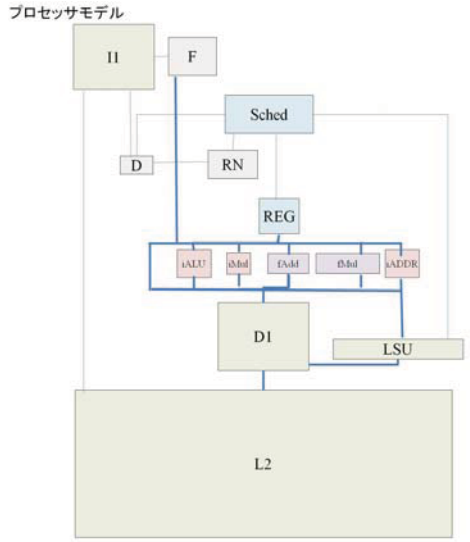


図 2 プロセッサモデル
Fig. 2 Processor model

モジュール間の長さを $WL(i,j)$ とするとき，モジュール間のデータ転送電力 $P(i,j)$ は，

$$P(i,j) \propto N_{active}(i,j) \times BW(i,j) \times WL(i,j) \quad (1)$$

と近似する．実際には通信される値等によっても通信電力が変動するが，本論文では簡略化する．このとき，プロセッサ全体のモジュール間のデータ転送電力の和 P_{all} は，

$$P_{all} \propto \Sigma_{i,j} N_{active}(i,j) \times BW(i,j) \times WL(i,j) \quad (2)$$

と見積ることができる．すなわち，データ転送に必要な消費電力を最小化する配置は，$\Sigma_{i,j} N_{active}(i,j) \times BW(i,j) \times WL(i,j)$ を評価関数とし，この値を最小化することで得られる．

モジュール M_i と M_j 間の距離 $length(i,j)$ は，簡単のため，モジュール M_i と M_j の重心間のマンハッタン距離で近似する．すなわち，3 次元空間上に配置されるモジュール M_i，M_j の重心の座標が (x_0, y_0, z_0)，(x_1, y_1, z_1) でそれぞれ与えられるとき $WL(i,j)$ は，

$$length(i,j) = |x_0 - x_1| + |y_0 - y_1| + \alpha \times |z_0 - z_1| \quad (3)$$

で算出する．ここで，z_i，z_j はそれぞれモジュールが配置される層を示すインデックス値であり，α は層を接続する貫通電極の長さに相当する．

```
WHILE(stop_condition)
 all_modules := all_modules
 FOR modules IN divide_layers(all_modules)
  pattern := random_sequence_pair(modules)
  decode_sequence_pair!(all_modules, pattern)
 END
 value := evaluation_function(all_modules)
 IF (value < saved_value) THEN
  saved_value := value
  saved_pattern := all_modules
 END
END
```

図 3 探索アルゴリズム

4.4 3 次元最適モジュール配置探索アルゴリズム

第 4.3 節で述べた評価関数を最小化する配置を探索する手法を述べる．ここで，配置パタンはモジュールの数に対して指数関数的に増加するため，全パタンの評価値を求めて比較することは現実的ではない．

2 次元上での最適な配置を求めるための手法としては，焼きなまし法や遺伝的アルゴリズムが一般的である．本論文では，初期評価として原始モンテカルロ手法によって配置を探索する．原始モンテカルロ手法では，ランダムに選択した配置パタンの評価値を求め比較することで，それらの中から評価値が最小となるパタンを解として得る．焼きなまし法に必要な，あるパタンにあたる近傍パタンや，遺伝的アルゴリズムで考えなければならない交配法や突然変位に相当するパタン生成手法を考える必要がないため初期評価として手軽に傾向を得ることができる．

本論文で用いた探索アルゴリズムを**図 3** に示す．まず，プロセッサ中のすべてのモジュール `all_modules` をランダムに各層に分けて割り当てる (`divide_layers`)．次に，各層に割り当てたモジュール `modules` の平面上の配置を決定する．平面上のモジュールの配置は，sequence-pair[5] を用いて表現できる．ここでは，乱数を使って `modules` の sequence-pair 上の順序を決定することで，モジュールのランダムな配置パタン `pattern` を得る (`random_sequence_pair`)．各モジュールの幅と高さを用いて sequence-pair をデコードすることで，配置パタンにおける各モジュールで座標が決定できる (`decode_sequence_pair!`)．全モジュールの座標の決定後，第 4.3 で定義した評価関数を用いて，ランダムに決定したモジュール配置の評価値 `value` を求める (`evaluation_function`)．ここで求めた評価値

ⓒ 2011 Information Processing Society of Japan

Vol.2011-ARC-195 No.10
Vol.2011-OS-117 No.10
2011/4/13

value が過去に得た評価値 saved_value より小さい場合，新しく求めた配置を最適解として保存する．解が十分に収束したと判断 (stop_condition が真になった) 場合に，探索アルゴリズムは終了する．

4.5 制　　約

今回の最適配置探索手法では，簡単のため，次の 2 点の前提を置いている．

配置は矩形のモジュール単位で決定

チップ上のトランジスタでプロセッサを構成する各モジュールを構成する場合，それらは，必ずしもモジュール同士がはっきりとした区切りをもつ矩形を保持するわけではない．しかし，探索の対象となるモジュール数の削減と sequence-pair での取り扱い易さのために，各モジュールは矩形領域を保持したまま取り扱う．

また，モジュール内のトランジスタが三次元空間に展開されて配置されることは考慮せず，モジュール内/モジュール間のクリティカルパス長は配置によらず一定であるとする．

各層の配置を原点で位置合わせ

今回の探索アルゴリズムでは，図 3 に示したように，配置すべきモジュールを，まずどの層に割当てるか決定し，同じ層に割り当てられたモジュールの配置を決定している．配置は，ランダムに決定した sequnce-pair をデコードすることで得ている．ここで，sequence-pair ではモジュール間の相対位置だけを表現できるのみであり，絶対位置を決定するためには，基準点が必要となる．今回は，一回の探索コストの軽減のため，すべての層の原点を固定している．ただし，層の原点を考慮することで，その配置の評価値が減少する，あるいは，より最適な配置が得られる可能性がある．

5. 評 価 環 境

cycle accurate なプロセッサシミュレータ「鬼斬弐」[6)7)] を用いてブロック間エッジの使用回数を計測した．使用したベンチマークは，SPECCPU2006INT 及び SPECCPU2006FP から全てのベンチマークプログラムについて，先頭 1G 命令をスキップし，続く 100M 命令をシミュレーションして計測した．それぞれにおいて貫通電極を 10 μm・25μm・50μm の 3 パターンと，3 次元積層するフロアのレイヤーを 1 層，2 層，3 層の 3 パターンについて行った．プロセッサシミュレータの設定は表 1 に示す．

6. 評　　価

得られたフロアプランの例を**図 4** から**図 7** に示す．図 7 は 3 層使用可能なときの様子で，どの TSV 長の時も同じ配置が得られた．フロントエンド，データパス，キャッシュ等が命令の動線に従って配置されており，ブロックレベルのモンテカルロ探索でも，ある程度意味のある自動配置ができていると考えられる．

表 1　ベースラインプロセッサ緒元
Table 1　Microarchitectural Parameters for Baseline Processor

命令セット	Alpha ISA
フロントエンド	4way，7cycle
命令ウィンドウ	i64 entry，f32 entry
LSQ	32entry
機能ユニット	2 iALU，1 iMUL/DIV，2 LD/ST，1 fpADD，1 fpMUL/DIV/SQRT
L1 ICache	32KB，LRU，8way，64B line，1cycle latency
L1 DCache	32KB，LRU，8way，64B line，1cycle latency
L2 I/DCache	256KB，LRU，16way，64B line， 20cycle latency
memory access	200cycle + arbitration latency

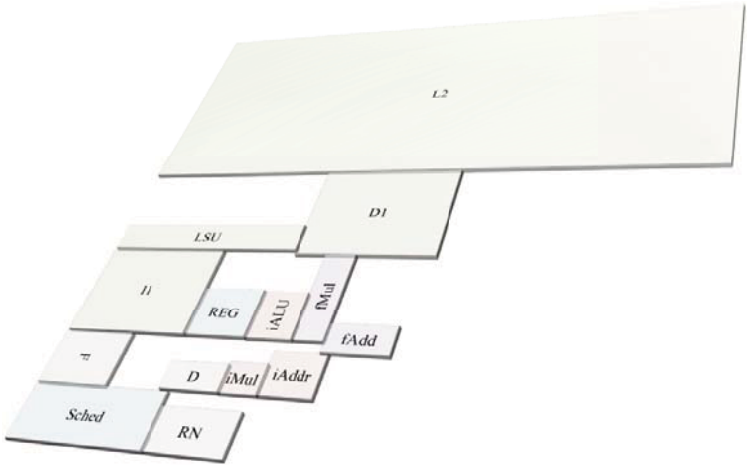


図 4　1 層のフロアプラン

3 次元配置が特に顕著な部分として，演算器を結ぶデータパスが挙げられる．演算器が z 軸方向にも並んで配置され，密なデータパスが形成された．ま た平面配置と異なり，キャッシュのような大きいブロックがあっても配置の自由度が失われにくいことが分かる．このため，キャッシュポートに隣接し て配置できるブロックが増え，このことにより通信負荷を削減する配置となっている．

2 層使用可能なときの配置結果を**図 5**，**図 6** に示す．10μm，25μm のときと 50μm のときでは異なる結果となったが いずれも，3 層配置と同様，データパスが 3 次元方向を利用

ⓒ 2011 Information Processing Society of Japan

Vol.2011-ARC-195 No.10
Vol.2011-OS-117 No.10
2011/4/13

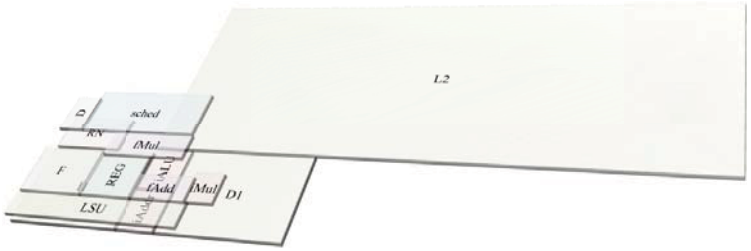


図 5 2 層のフロアプラン

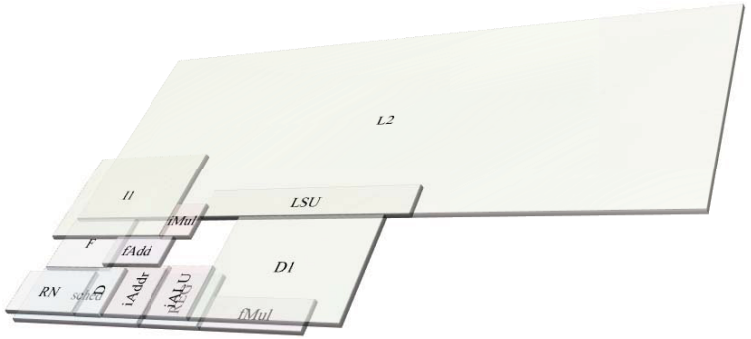


図 6 2 層のフロアプラン

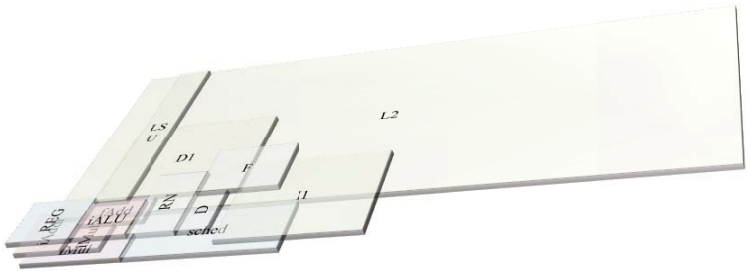


図 7 3 層のフロアプラン

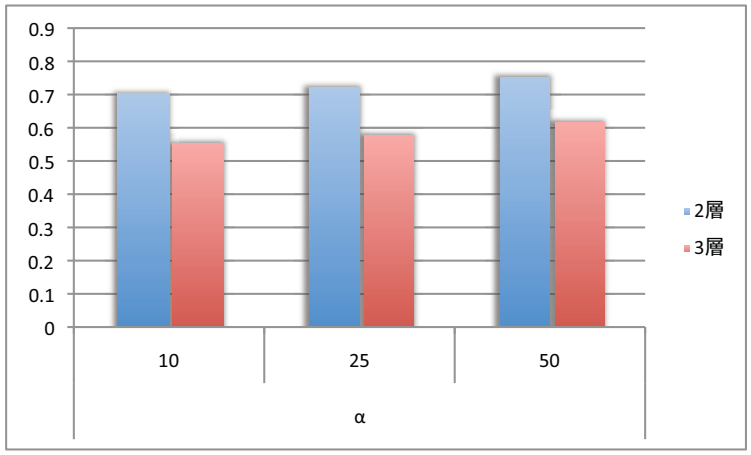


図 8 通信負荷

して形成され，またキャッシュのような大きいブロックでも隣接配置がされやすくなっていることが分かる．また，同じ手法を用いて平面配置を試みた場合の結果を図 4 に示す．ブロックの隙間や動線の冗長さなど，最適化の余地 が見られるが，フロントエンド，データパス，キャッシュ等のグループ配置が実現していることが分かる．

6.1 3 次元配置の効果

図 8 は得られたフロアプランの通信負荷をグラフで示したものである．通信負荷として，4.3 章における評価関数でみつもった値を相対値で示している．図 8 では，使える層が大きいほど，また TSV の長さが短いほど良い結果が得られており，3 次元化は効果があることが分かる．配置の 3 次元化によりプロセッサコアのブロック間通信負荷が半分近く削減する見積もり結果となった．

6.2 3 次元配置の最適収束

図 9 に各パラメータにおけるモンテカルロ試行時の評価関数の収束の様子を示す．3 次元配置最適化は探索空間が広く，簡単な探索では収束が困難と予想されたが，今回のような簡単なブロック配置では原始モンテカルロで比較的素早く収束することが分かった．また動線的に理にかなっている配置が得られている．

以上のように，ブロックレベルで 3 次元化したプロセッサコアの簡単な配置最適化と見積もりでは，データパスのコンパクト化やキャッシュへのアクセ シビリティ向上などにより，平面配置に対して通信負荷を半分に削減できることが分かった．またブロックレベルであれば原始モンテカルロでも収束 し，最適化も現実的であることが分かった．

ⓒ 2011 Information Processing Society of Japan

Vol.2011-ARC-195 No.10
Vol.2011-OS-117 No.10
2011/4/13

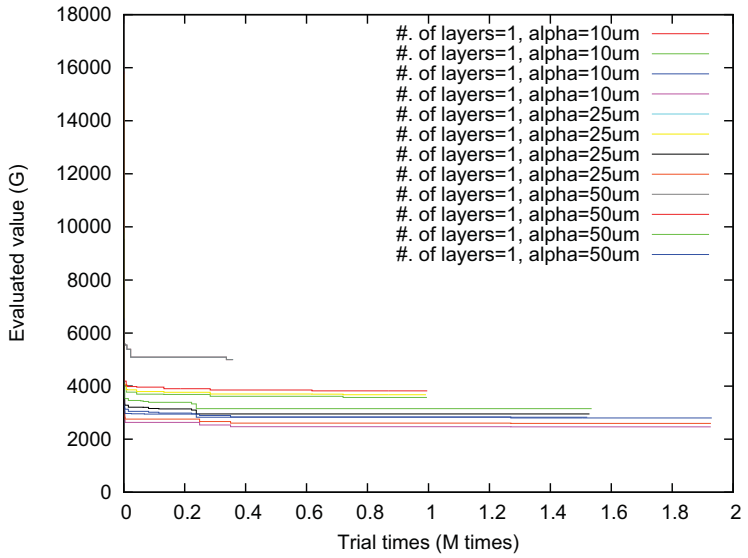


図 9 探索アルゴリズムの収束

7. 関 連 研 究

3 次元積層化されたプロセッサの問題は様々な既存研究において議論されてきた．Black ら[3] は，3 次元積層プロセッサの消費電力解析と温度解析を行っている．3 次元積層により配線長が短くなり，その結果として消費電力が削減される事を示している．3 次元積層化の問題点の一つに発熱が挙げられるが，通信間で削減出来る電力に伴い，温度の向上も冗長出来る事を示している．

橋口ら[1] は大容量の DRAM を LLC(Last Level Cache) として上層に積層し，プログラムの特性によってキャッシュの動作モードを切り替えるアーキテクチャとその決定方法を示している．本研究のように 3 次元積層プロセッサ内の 3 次元配置手法とそのフロアプランに取り組んだ既存研究はない．本研究では 3 次元積層マルチコアにおいて電力消費の観点からユニット配置問題を検討し，ブロック配置問題を解く手法が有効であるという結果を新たに得た．

8. ま と め

本論文では、3 次元積層のメリットを活用したプロセッサを実現するための各ブロックの適切な配置問題を解く手法を検討した。パラメタ算出手法と解探索手法について述べ、それらを用いて得られたブロック配置を示した。得られたフロアプランからフロントエンド、データパス、キャッシュ等が命令の動線に従って配置されており、ブロックレベルのモンテカルロ探索でも、ある程度意味のある自動配置ができていると考えられる。ブロックレベルで 3 次元化したプロセッサコアの簡単な配置最適化と見積もりでは、データパスのコンパクト化やキャッシュへのアクセシビリティ向上などにより、平面配置に対して通信負荷を半分に削減できることが分かった。またブロックレベルであれば原始モンテカルロでも収束し、最適化も現実的であることが分かった。今後の課題として、例えば遺伝的アルゴリズムを用いるなどモンテカルロ探索以外以外による、更なる配置最適化が必要であると考えられる。

参 考 文 献

1) 橋口慎哉, 福本尚人, 井上弘士, 村上和彰. SRAM/DRAM ハイブリッド・キャッシュにおける実行時動作モード決定法の提案. 情報処理学会研究報告. 計算機アーキテクチャ研究会報告, Vol. 2011, No.9, pp. 1–6, 2011-01-13.
2) Gabriel H. Loh. 3D-Stacked Memory Architectures for Multi-core Processors. *SIGARCH Comput. Archit. News*, Vol.36, pp. 453–464, June 2008.
3) Bryan Black, Murali Annavaram, Ned Brekelbaum, John DeVale, Lei Jiang, Gabriel H. Loh, Don McCaule, Pat Morrow, Donald W. Nelson, Daniel Pantuso, Paul Reed, Jeff Rupley, Sadasivan Shankar, John Shen, and Clair Webb. Die stacking (3d) microarchitecture. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 39, pp. 469–479, Washington, DC, USA, 2006. IEEE Computer Society.
4) Kiran Puttaswamy and Gabriel H. Loh. Thermal analysis of a 3d die-stacked high-performance microprocessor. In *Proceedings of the 16th ACM Great Lakes symposium on VLSI*, GLSVLSI '06, pp. 19–24, New York, NY, USA, 2006. ACM.
5) Hiroshi Murata, Kunihiro Fujiyoshi, Shigetoshi Nakatake, and Yoji Kajitani. Rectangle-packing-based module placement. In *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*, ICCAD '95, pp. 472–479, Washington, DC, USA, 1995. IEEE Computer Society.
6) 渡辺憲一, 一林宏憲, 五島正裕, 坂井修一. プロセッサ・シミュレータ「鬼斬」の設計. 先進的計算基盤システムシンポジウム, pp. 194–195, 2007.
7) 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」の設計. 先進的計算基盤システムシンポジウム, pp. 120–121, 2009.

ⓒ 2011 Information Processing Society of Japan