

1. アーキテクチャ基盤技術

6

タイルプロセッサ

吉瀬 謙二

電気通信大学 大学院情報システム学研究科  
kis@is.uec.ac.jp

計算機アーキテクチャの新しい流れ

マイクロプロセッサは、1971年に発表された4004の誕生以来、半導体製造技術とアーキテクチャの進歩により劇的な性能向上を達成してきた。半導体製造技術の進歩（デバイスの微細化）により動作周波数は劇的に向上し、チップに集積するトランジスタ数は、Mooreの法則として知られるように、18から24カ月で2倍というペースで増加を続けている。これらの豊富なハードウェア資源を活用して、スーパースカラやVLIW (very long instruction word) などを採用する高速なプロセッサが開発されてきた。最近では、高速性という指標に加えて、消費電力の低減や信頼性の向上といった新しい性能指標が重要視されるようになってきている。プロセッサを評価するための指標が多様化しているとはいえ、依然として、高速なプロセッサへの強い要求がある。世界中の企業研究所や大学で、5年、10年先を見据えた新しい高性能プロセッサの開発が進められている。

現在の主流となっているスーパースカラやVLIWを拡張することだけで10年先の高性能プロセッサを実現することは難しい。この原因の1つに配線遅延の問題がある。図-1は、チップを製造するためのデバイス技術の進歩に伴って、1クロックサイクルで信号が伝わるチップの範囲が小さくなる様子を示している<sup>1)</sup>。1クロックサイクルで信号が伝わる範囲はチップの動作周波数に依存する（プロセッサの動作を遅くすれば、1クロックで到達できる範囲は大きくなる）。ここでは、プロセッサ性能とパイプライン段数とのバランスから、1クロックサイクルの時間をファンアウト4のインバータ回路が8回動作する時間としている。デバイスの微細化により、高性能プロセッサのチップ面積が大きく影響を受けることはない。20×20mmのチップを想定して図が描かれ

ている。

Pentium 4などの商用プロセッサが採用している90ナノ・メートルのテクノロジーの場合にも、1クロックサイクルですべての領域に信号を伝えることが難しくなっており、配線遅延の問題を無視して設計することができなくなりつつある。ただし、現在のデバイス技術を用いて設計されるスーパースカラやVLIWにおいては、配線遅延が深刻な問題になっているわけではない。

一方、将来の35ナノ・メートルのデバイス技術では、1クロックサイクルで信号を伝えることができる範囲はチップの1%以下と極端に小さくなる。従来方式を用いたプロセッサでは、チップ内のあらゆる場面で情報を伝達するための遅延が挿入される。特に、最も距離の離れた2点間の通信には、数10サイクルという大きな遅延が生じ、深刻な性能低下を引き起こす。このため、5年先のプロセッサ構成を考える場合には、アーキテクチャ設計の上位のレベルから配線遅延を考慮して方式を検討することが必要となる。

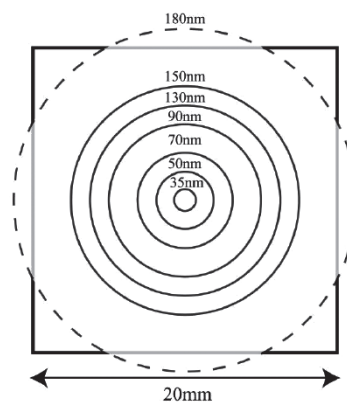


図-1 デバイス技術の進歩とクロックサイクルの間に信号が伝わるチップ内の範囲

## タイルアーキテクチャ

配線遅延が問題とならない小さいサイズの機能ブロックを考え、この機能ブロック (タイル) を規則的に敷きつめることで高速なプロセッサを構成する方式をタイルアーキテクチャと呼んでいる<sup>3)</sup>。このような方式を利用することで、前提としている小さいサイズの機能ブロックという制約から、タイルの内部で発生する配線遅延の問題を軽減できる。また、物理的に近いところに配置されているいくつかのタイル間でのみデータの送受信を行うことで、タイル間の通信遅延を軽減することができる。

スーパースカラとタイルアーキテクチャとの違いは、チップ写真を比べてみると分かりやすい (図-2)。スーパースカラの Pentium 4 プロセッサの写真を左に、タイルアーキテクチャの Raw プロセッサの写真を右に示す。Raw プロセッサでは、16 個のプロセッサコア (正方形の形をしたタイル) が格子状に整然と敷きつめられている。それぞれのタイルは同一の設計の機能ブロックが複製されたもので、計算ユニット、タイル間を接続する配線、ルータなどから構成される。

タイルアーキテクチャは、タイルとして実現される小さい機能ユニット (プロセッサ) を多数集積するという構成から、バスやリングで接続されたチップマルチプロセッサと共通点がある。しかしながら、その設計思想は大きく異なっている。チップマルチプロセッサでは、できるだけ従来のプロセッサ設計を再利用して、キャッシュやネットワークの構成を最適化することで、高い性能を狙うものが多い。一方、タイルアーキテクチャでは、配線遅延を克服するために、従来のプロセッサにとらわれることなくタイルの内部構成を工夫する。ただし、チップの上に、同じ構成のタイルを複製して配置することで設計の再利用性を高め、検証などの作業の軽減を狙う。タイルアーキテクチャでは、タイルの数を増やしたときに動作周波数が低下しない近接接続のネットワークを採用することが多い。

大量の画像データや音声データなどを扱う一部のアプリケーションを高速化するメディア処理専用プロセッサとして、タイルアーキテクチャのように多数のプロセッサを 2 次元メッシュに配置する構成を持つものがある。しかし、タイルアーキテクチャは、汎用プロセッサとしてさまざまな種類のアプリケーションを効率よく処理することを目指している点で挑戦的である。従来のプロセッサと大きく異なる構成を持つタイルプロセッサを有効に利用するために、独自の命令セットアーキテクチャを採用することが多い。このため、多くのタイル

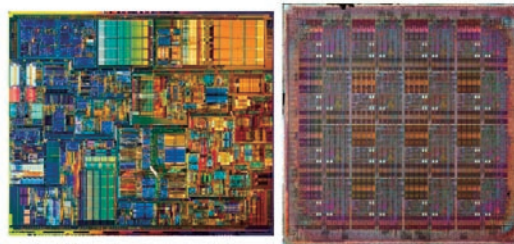


図-2 スーパースカラの Pentium 4 プロセッサ<sup>2)</sup>とタイルアーキテクチャの Raw プロセッサ<sup>3)</sup>

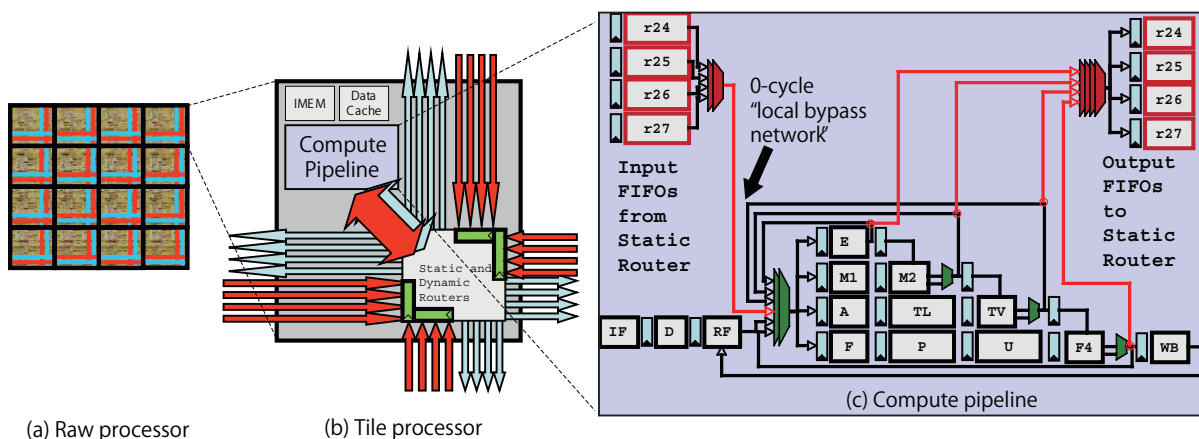
プロセッサは、伝統的な CISC (complex instruction set computer) や RISC (reduced instruction set computer) プロセッサとのコード互換性を放棄している。この場合、アプリケーションプログラムは C や Fortran などの高級言語で記述され、独自のコンパイラがタイルプロセッサのためのオブジェクトコードを生成する。

タイルプロセッサの研究開発は、現在、米国の大学を中心とする研究機関において進められている。5年、10年先を見据えた新しい高性能プロセッサ研究の試みの1つといえる。

## Raw プロセッサ

マサチューセッツ工科大学 (MIT) で開発が進められている Raw プロセッサは、タイルアーキテクチャのさきがけである<sup>3)</sup>。配線遅延の克服に加えて、豊富なハードウェア資源の活用、限られたピンの有効利用といった問題の解決を目指して、チップの試作とシステムレベルの評価が行われている。

図-3 (a) に示す Raw プロセッサは、16 個の同じ構成を持つタイルを敷きつめる構成をとる。図-3 (b) に示すそれぞれのタイルは、MIPS プロセッサに近い単命令発行のインオーダー処理の計算パイプライン、キャッシュ、コンパイル時にルートが決められるスタティックネットワーク、割り込みやメインメモリ参照のためなどに利用される動的なネットワークにより構成される。それぞれのタイルは独自のプログラムカウンタを持つプロセッサとして動作し、命令やデータキャッシュにミスした場合には、チップの外に配置されるメインメモリからデータを取得する。タイル間の通信には必ずレジスタが介在し、すべての配線長は、タイルの一辺の長さより短くなるように設計されている。このため、アプリケーションからの性能要求や、利用できるトランジスタ数の増加に応じて集積するタイルの数を増やしたとしても、動作周波数が低下することはない。一方で、試作チップの構成では、



(a) Raw processor

(b) Tile processor

(c) Compute pipeline

図-3 Rawプロセッサ

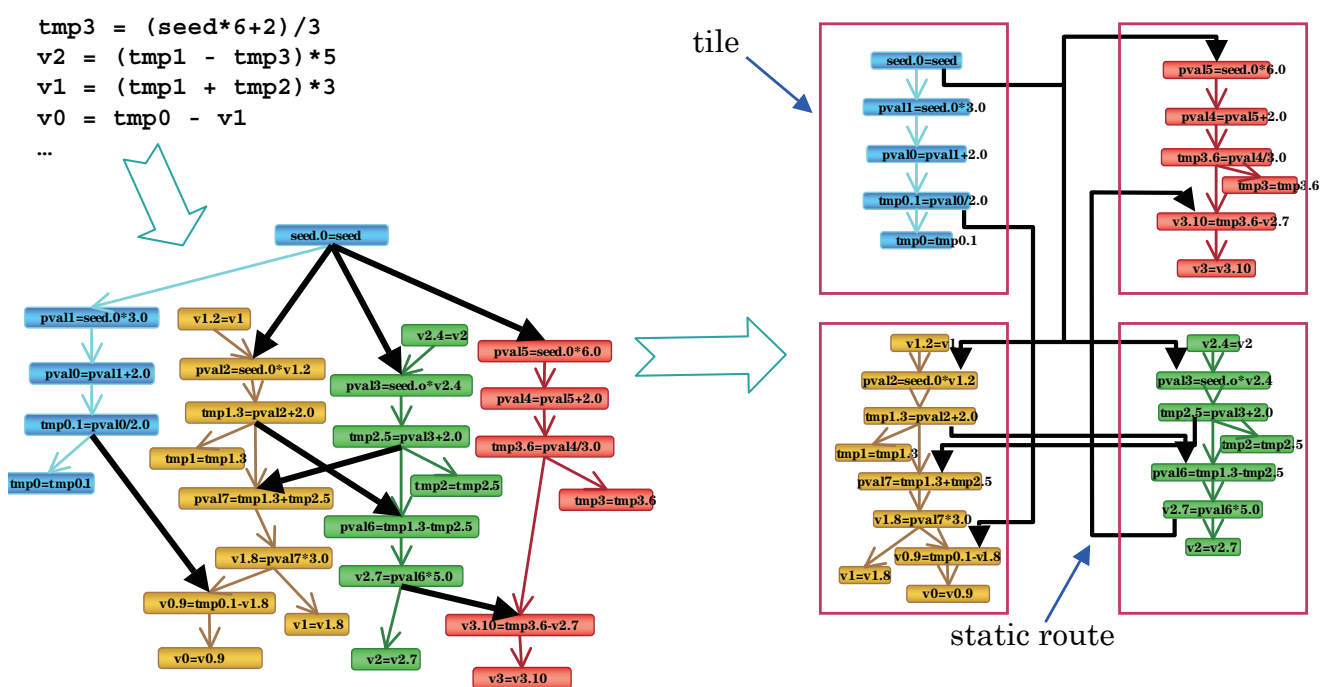


図-4 RawCCコンパイラによる命令の割り当て

タイルを経由するたびに1サイクルの遅延が生じるため、たとえば、左上のタイルが生成したデータを右下のタイルが利用するためには6サイクルの通信遅延が発生する。

1つのタイルが持つ計算パイプラインの構成を図-3(c)に示す。8ステージの命令パイプラインの構成を採用する。個々の計算パイプラインは単命令発行の単純な構成をとり、1つのタイルでは、サイクル当たりたかだか1命令しか処理することができない。しかし、16個のタイルがすべて同時に計算を行うことで、チップとしてサイクル当たり16命令という高いピーク性能を達成する。

タイル間の通信遅延を小さくするために、計算パイプラインのデータパスに通信のための機構が組み込ま

れている(図-3(c))。レジスタの24番(r24)から27番(r27)が、通信の入出力バッファに割り当てられており、特別な命令を必要とすることなくタイル間でデータの送受信を行うことができる。たとえば、レジスタ24番からの読み出しは、通信バッファからの値の読み込み(データの受信)となる。レジスタ24番への書き込みは通信バッファへの書き込み、すなわち、他のタイルへのデータ送信となる。

開発を行っているRawCCコンパイラは、Cで書かれた逐次プログラムのデータ依存関係を解析し、タイルに命令を割り当て、静的なネットワークを構築する。図-4の例では、4つの色で示した命令がそれぞれの色

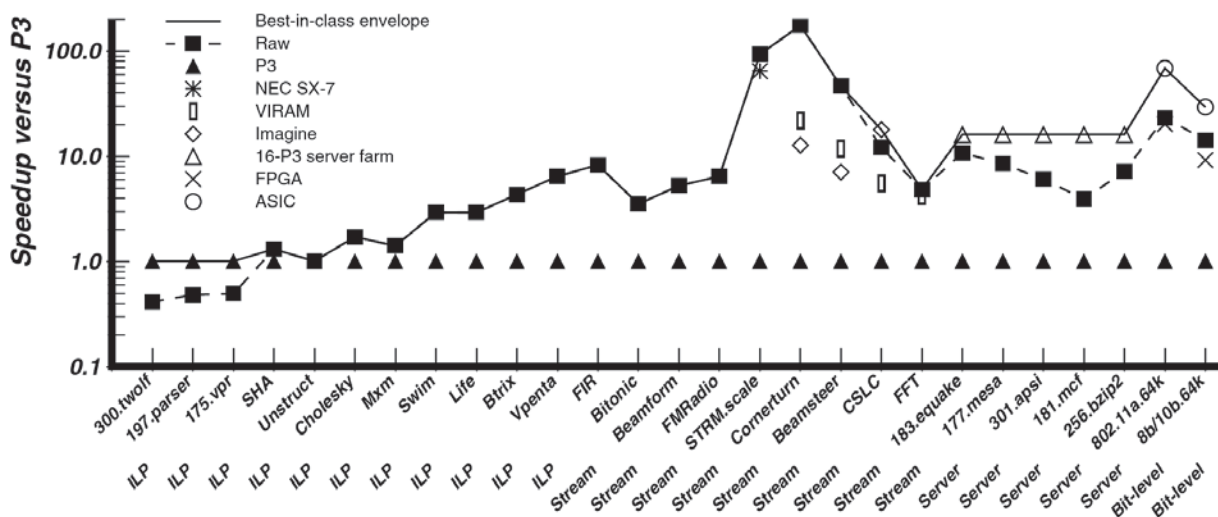


図-5 Pentium IIIプロセッサの性能を1とするRawプロセッサの性能比較<sup>4)</sup>

付けられたタイルに割り当てられる命令である。異なる色をまたがるデータ通信に対応する太い矢印が、静的なネットワークの経路を表している。RawCC コンパイラは、命令レベルの並列性を利用して、並列に実行できる部分を異なるタイルに静的に割り当てる。このため、命令レベルの並列性を十分に持たないアプリケーションでは、16個のタイルを有効に利用することは難しい。

Raw プロセッサの性能を図-5にまとめる。Pentium III プロセッサ (P3) の性能を1とする速度向上率を示している。グラフの縦軸は対数になっている点に注意されたい。命令レベル並列性を利用するベンチマークプログラム (図では ILP と記述) の中で利用できる並列性が少ないものでは、Raw の性能は Pentium III に及ばない。一方、SPEC92 ベンチマークに含まれる Vpenta のように並列性の豊富なアプリケーションでは、Pentium III を凌ぐ性能を達成する。ストリーム系のベンチマーク (図では Stream)、サーバ (図では Server)、ビット演算 (図では Bit-level) のベンチマークでは、Raw はこれらのアプリケーションに強いアーキテクチャ (スーパーコンピュータの SX-7 や、ビット演算に強い FPGA など) に匹敵する性能を達成できる。ただし、一部のベンチマークでは手作業による最適化が施されている。また、図-5に示す速度向上には、サイクル当たり16命令を実行できるという理由のほかに、メモリバンド幅の改善やキャッシュを経由しないダイレクトメモリアクセスなどの効果が含まれる。これらの工夫により、豊富な並列性が利用できる場合に、Raw プロセッサは Pentium III の数倍から数十倍の高速化を達成する。

Raw プロセッサのハードウェア構成は洗練されているが、それほど複雑というわけではない。通信と計算を

明示的に記述できる命令セットを提供して、計算パイプラインのデータパスに通信のための機構を組み込むことにより、タイル間の通信に必要となる遅延を小さくする。このことが、タイル間の通信が頻繁に発生するタイルアーキテクチャにおいて、Pentium III を大きく上回る性能を引き出す鍵になっている。

## TRIPS プロセッサ

テキサス大学では、挑戦的な構造を採用するタイルアーキテクチャとして TRIPS プロセッサ<sup>☆1</sup>の開発を行っている。単命令発行の単純な整数演算ユニット、浮動小数点演算ユニット、命令バッファ、オペランドバッファ、オペランドルータから構成される計算ノード (タイル) を格子状に配置する。そこに、ブロックとよばれるコンパイラが生成する TRIPS プロセッサの実行に適する複数の命令のかたまり (ベーシックブロックのことではない) を割り当て、その中のデータが揃った命令から処理を開始する。このようなマクロデータフロー方式の実行モデルの採用が TRIPS プロセッサの特徴である。

計算ノードを4×4の格子状に配置する TRIPS プロセッサの構成を図-6に示す。計算ノードを格子状に配置して、近傍のノード間のみでデータを受け渡すことにより配線遅延の問題を緩和する点は Raw プロセッサと同様である。一方、Raw プロセッサとは異なり、格子

☆1 以前はグリッドプロセッサと呼ばれていた。広域ネットワークを介する複数のコンピュータを利用して仮想的な高性能コンピュータを構築するグリッドコンピューティングとは関係ない。



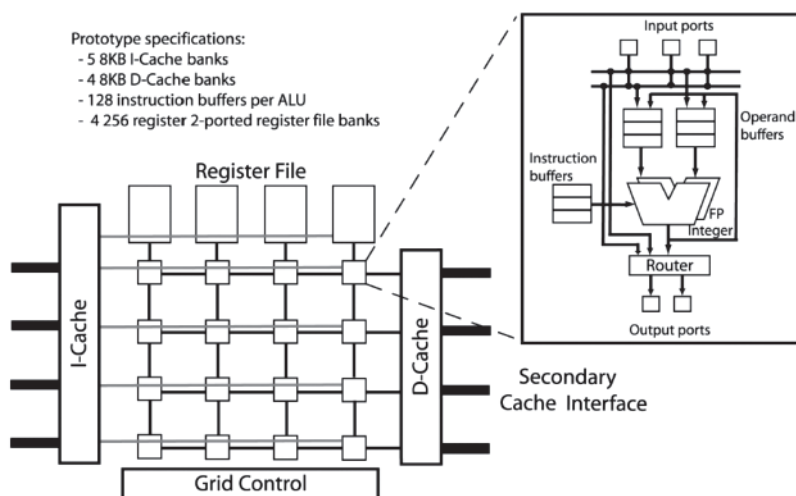


図-6 4×4の計算ノードを持つTRIPSプロセッサと計算ノードの構成<sup>1)</sup>

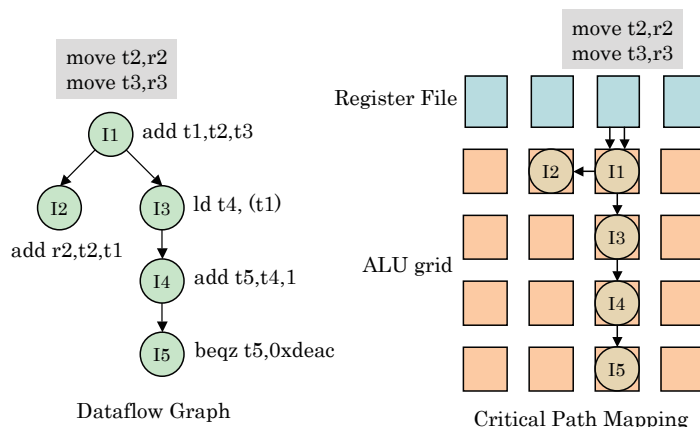


図-7 データフローグラフに基づく計算ノードへの配置<sup>1)</sup>

状に敷きつめられた計算ノードを囲む形に、命令キャッシュ、データキャッシュ、レジスタファイルを配置する。計算ノードでは、自律的にデータの到着を検出して、計算のためのデータが揃った命令から実行を開始する。これにより、スーパースカラが持つ発行ウィンドウやデータフォワーディングのパスといった高速化が困難となる回路を利用することなく、大規模なアウトオブオーダーの仕組みを実現する。

TRIPS プロセッサは、個々の命令単位ではなく、ブロックを単位として計算ノードを割り当てる。16個の計算ノードには識別子が付けられており、この情報を用いてコンパイラが静的にブロック内の命令を計算ノードに割り当てる。TRIPS プロセッサは、ブロック単位で命令

をフェッチし、ブロックを構成するすべての命令の処理が完了した時点でブロックのために割り当てられた資源を解放する。それぞれの計算ノードは複数のフレーム（命令を格納するバッファ）を持っており、複数のブロックの命令をフェッチして割り当てることができるようになっている。各計算ノードは複数のフレームからフェッチしたいくつかの命令の中から、処理が可能となった命令を発火する。

TRIPS プロセッサの各計算ノードにブロックを割り当てる様子を図-7に示す。左がブロックを構成する命令のデータの流れを表現するデータフローグラフで、右が4×4の計算ノードを持つTRIPSプロセッサにブロックを割り当てた様子である。2つのレジスタ r2, と r3 は、

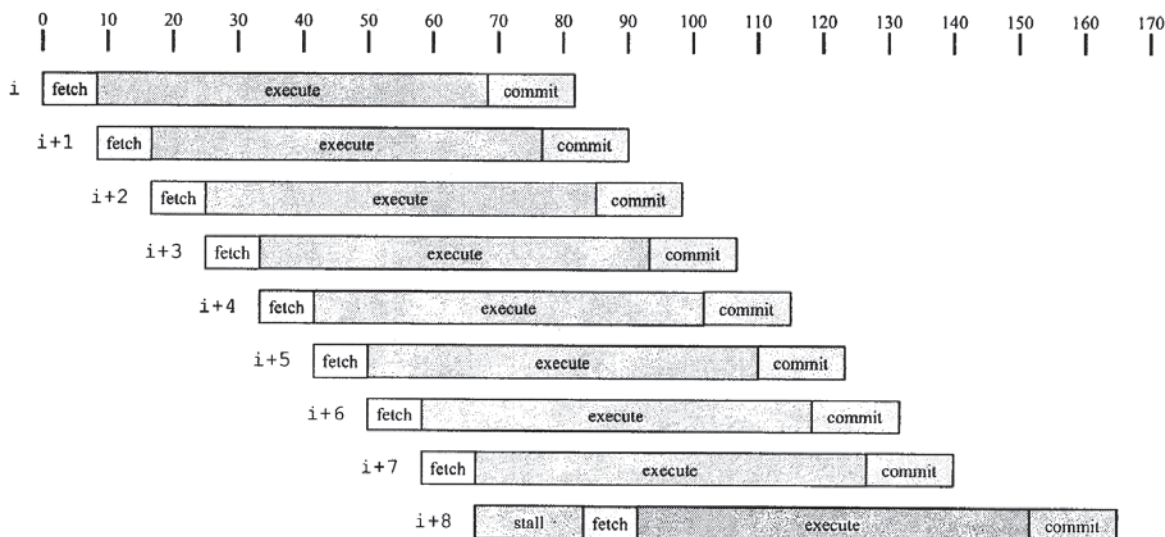


図-8 ブロック処理のオーバーラップによる遅延の隠蔽<sup>5)</sup>

ブロックが利用する入力データで、これは `move` 命令により明示的にレジスタから読み出され、計算ノードへと送られる。データは生成された計算ノードから利用される計算ノードに転送され、必要がなければレジスタに書き戻されることはない。ブロックのすべての命令の実行が完了した時点でブロックの出力をレジスタファイルに書き戻し、ブロックのために割り当てた計算ノード（厳密には、計算ノードの命令バッファ）を解放する。

TRIPS プロセッサでは、Raw プロセッサのように個々のタイルがキャッシュを持つ構成ではなく、タイルの周辺に命令やデータキャッシュを配置する。このため、必要とする命令やデータをフェッチするために遅延が発生する。たとえば、TRIPS プロセッサでベクトル加算を計算する例では、1つのブロックの処理が始まってから完了するまでに80サイクル程度の時間を必要とする<sup>6)</sup>。命令をフェッチするだけでも10サイクル程度が必要となる。このようにブロックを処理するための長い遅延を隠蔽するために、最大で8個までのブロックの処理をオーバーラップさせる工夫を施している (図-8)。

8×8の計算ノードを持つTRIPS プロセッサ (GPA) とスーパースカラのAlpha 21264 プロセッサとの性能比較を図-9に示す。これは、ソフトウェアシミュレータを構築して、得られる性能を見積もったものである。TRIPS プロセッサは、Alpha 21264 プロセッサの1.1倍から14倍の高速化を達成する。8×8の計算ノードで8フレーム構成 (8個のブロックまでオーバーラップし

て実行できる構成) のTRIPS プロセッサでは、最大で、512個の命令をフェッチすることができる。これは、直ちに実行を開始することができる命令を検索するウィンドウのサイズが劇的に大きくなることを意味し、抽出できる命令レベルの並列性が向上することとなる。このことがTRIPS プロセッサの性能向上をもたらす要因の1つになっている。

テキサス大学では、32個のタイルを集積するTRIPS チップの設計が進められている。2005年の12月にはこのチップを4個利用する (システムとして128個のタイルを集積する) プロトタイプシステムを動作させ、タイルアーキテクチャの評価を行う予定となっている。

## タイルプロセッサの課題と今後の展望

デバイス技術の進歩に伴って、配線遅延の問題が深刻になる。この問題を解決するために、配線遅延の影響を軽減する小さいタイルとしての機能ブロックを敷きつめるタイルアーキテクチャの開発が進められている。現在のタイルアーキテクチャに関する研究は、汎用アプリケーション、サーバアプリケーション、メディア処理などを含むさまざまな種類のアプリケーションを高速に処理する基本方式の開発に取り組んでいる段階にある。

今後の課題として、スーパースカラにおいて検討が進んでいるメモリ参照遅延の隠蔽技術や投機技術といった

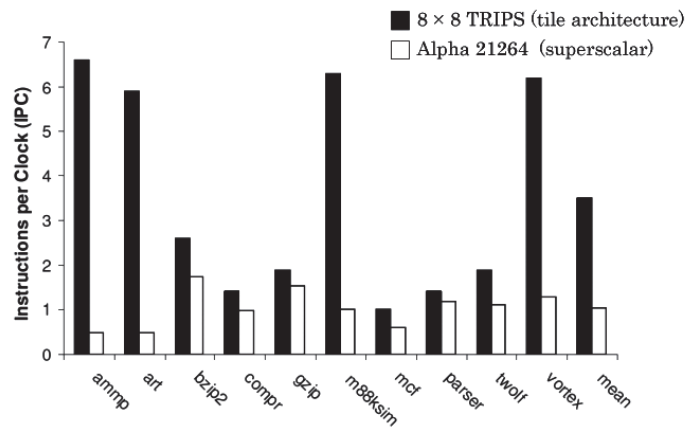


図-9 TRIPSプロセッサとAlpha 21264との性能比較<sup>1)</sup>

先端の技術を組み込んでタイルの機能やサイズを最適化する必要がある。また、高速な処理を維持しながら、消費電力を低減する技術や、高信頼性を達成する技術を融合する必要がある。タイルアーキテクチャの本質は、高い動作周波数を維持しながら、配置された多数のタイルを並列に動作させることで高速化を達成することにある。長期的な課題として、アプリケーションに内在する並列性の抽出を助けるプログラミング手法あるいは汎用の並列プログラミングの開発と普及が期待される。

タイルアーキテクチャは独自の命令セットを利用する斬新なアーキテクチャであり、その実現と普及には解決すべき課題が多い。しかしながら、10年先のプロセッサアーキテクチャという視点で捉えると、タイルアーキテクチャが魅力的な選択肢の1つであることに間違いはない。

現在は、米国を中心に研究開発が進められているタイルアーキテクチャであるが、日本でそのような研究が行われてこなかったわけではない。我々の研究グループにおいても、世界に先駆けて、ALU-Netと呼ばれる多数の演算器の相互接続網にデータフローグラフとして表現されるプログラムの断片を割り当てて処理を進める大規模データパスプロセッサ<sup>6)</sup>の研究を行った経験がある。21世紀のプロセッサ開発は始まったばかりである。プロセッサというメインストリームの市場を失わないためにも（あるいはその市場を奪回するためにも）、タイルプロセッサを代表とする斬新なアーキテクチャの研究を国内でも活発に進めることが必要である。

#### 参考文献

- 1) Keckler, S. W., Burger, D., Moore, C. R., Nagarajan, R., Sankaralingam, K., Agarwal, V., Hrishikesh, M. S., Ranganathan, N. and Shivakumar, P.: A Wire-Delay Scalable Microprocessor Architecture for High Performance Systems, International Solid-State Circuits Conference (ISSCC), pp.1068-1069(Feb. 2003).
- 2) Carmean, D., Upton, M., Hinton, G., Sager, D., Boggs, D. and Roussel, P.: The Pentium 4 Processor, HOT CHIPS 13 (Aug. 2001).
- 3) Taylor, M. B., Kim, J., Miller, J., Wentzlaff, D., Ghodrati, F., Greenwald, B., Hoffmann, H., Johnson, P., Lee, W., Saraf, A., Shnidman, N., Strumpfen, V., Amarasinghe, S. and Agarwal, A.: A 16-Issue Multiple-Program-Counter Microprocessor with Point-to-point Scalar Operand Network, International Solid-State Circuits Conference (ISSCC)(Feb. 2003).
- 4) Taylor, M. B., Lee, W., Miller, J., Wentzlaff, D., Bratt, I., Greenwald, B., Hoffmann, H., Johnson, P., Kim, J., Psota, J., Saraf, A., Shnidman, N., Strumpfen, V., Frank, M., Amarasinghe, S. and Agarwal, A.: Evaluation of the Raw Microprocessor: An Exposed-Wire-Delay Architecture for ILP and Streams, Proc. of The 31st Annual International Symposium on Computer Architecture, pp.2-13 (June 2004).
- 5) TRIPS Project Team: Design and Implementation of the TRIPS EDGE Architecture, ISCA-32 Tutorial (June 2005).
- 6) 中村友洋, 吉瀬謙二, 辻 秀典, 安島雄一郎, 田中英彦: 大規模データパスプロセッサの構想, 情報処理学会研究報告 計算機アーキテクチャ研究会, 97-ARC-124, Vol.97, No.61, pp.13-18 (June 1997).

(平成 17 年 8 月 2 日受付)

