

Title	VLIWアーキテクチャを適用した高速、低消費電力ネットワークプロセッサの研究
Author(s)	五由出, 将嗣
Citation	
Issue Date	2006-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1976">http://hdl.handle.net/10119/1976</a>
Rights	
Description	Supervisor:日比野 靖, 情報科学研究科, 修士

# VLIW アーキテクチャを適用した高速、低消費電力ネットワークプロセッサの研究

五由出 将嗣 (410048)

北陸先端科学技術大学院大学 情報科学研究科

2006 年 2 月 9 日

キーワード: ネットワークプロセッサ、パトリシアツリー、ルーティング、VLIW アーキテクチャ.

## 1 はじめに

近年、ネットワーク利用者の急激な増加や大容量データ転送に伴い、ネットワークのさらなる高速化が求められている。ネットワーク高速化のネックのひとつとして、パケットフォワーディングにおける経路探索作業があげられる。また、ネットワークの中核である高速ルータには汎用プロセッサと ASIC を利用した構成が大半であるが、高性能 ASIC チップの開発には莫大な費用と時間がかかってしまう。

こういった背景からソフトウェアラブルで、ネットワーク機器に特化したネットワークプロセッサが Intel 社から開発された。ソフトウェアによる実装が可能で、並列化による性能向上が容易なネットワークプロセッサを用いることによって、柔軟性ととんだ高機能なルータを短期間でかつ安価に開発することを期待されている。

ネットワークプロセッサではパケットフォワーディングの処理が大部分であり、データ移送 (load/store)、条件分岐、論理演算の並列実行が可能な VLIW アーキテクチャに適している。本論文で提案する手法は、このネットワークプロセッサに対してアドレステーブルの探索作業に特化した高速に並列実行が行える VLIW アーキテクチャの命令フォーマットを提案し、高速化と、VLIW を適用することによる回路単純化による消費電力を低減する。

## 2 ネットワークルーティング

代表的なネットワークプロセッサに、Intel 社のネットワーク機器向け半導体アーキテクチャである、IXA(Internet eXchange Processor) アーキテクチャに基づいたネットワーク機器に特化したプロセッサがある。このネットワークプロセッサの主な動作はパケット

フォワーディング（パケットの受信、テーブルルックアップ、パケットの送信）の処理に費やされる。特にテーブルルックアップとは、ルータがルーティングテーブルを検索することであり、このテーブルは動的に変化する。そしてこの処理は毎パケットに実行されるために一秒間に数千回から数百万回に及ぶ。またこのテーブルルックアップの作業に時間を費やしてしまうために、ネットワーク全体のボトルネックとなる。ルーティングの主な役割は、入力されたパケットのヘッダーに格納されている宛先 IP アドレスを読み取り、目的のアドレスへ適した出口へと導くことである。

ルーティングテーブルは、概念的には、宛先 IP アドレスと出口ポート番号の対の表である。表の検索には、表の大きさに比例した時間がかかり効率が悪い。

そこでパトリシアツリーと呼ばれる表現が用いられる。パトリシアツリーは、IP アドレス表現の各ビットの 0/1 に対応して、ツリーの左右に分岐し、ツリーのリーフに目的のアドレスを出口の対に格納しておく。

パトリシアツリーのメリットとしては、各ビットの OFF/ON を判定するだけなので非常に高速に出口を探し出すことができる。また動的に変化するルーティング情報に対応するには、ツリーの構造を変更することにより、効率の良い検索ツリーを保持できることである。

### 3 VLIW 命令セット

パトリシアツリーの探索を行っているプログラムは、現在のノードがリーフであるかないかを判定し、リーフでなければノード内の指定ビットに従ってパケットのビットを判定し左右に分岐する。この動作の繰り返しであり、ツリーの下位にいくほどループ回数も増加する。この時、ループ内でパトリシアツリーを探索している間、次に入力されてきたパケットは待機していなければならない無駄が多い。

そこで、パトリシアツリーの探索に特化した VLIW 命令セットを考案した。並列処理を行わない場合、リソースは  $ld \times 1$ 、 $ldd \times 1$ 、 $st \times 1$ 、 $alu \times 1$ 、 $bra \times 1$ 、の合計 5 個のユニットを使用する。このリソースをそのまま使用するのが VLIW A 命令セット。頻繁に使用されるユニットを追加し、 $ld \times 2$ 、 $ldd \times 1$ 、 $st \times 1$ 、 $alu \times 2$ 、 $bra \times 1$ 、の合計 7 個のユニットを使用する VLIW B 命令セットとする。

### 4 評価

提案した手法の命令セットを使用し、パトリシアツリーの探索に対する高速化の是非と、リソースを有効に利用できているかを評価する。

並列化を実行していない時にかかるサイクルと、そのときに使用する各ユニットの使用回数によって、同時に処理できるパケットの数は決まってくる。この探索動作においては 51 サイクル中 15 サイクルを  $ld$  命令が使用しているため最高で 3 個のパケットの処理がで

きることになる。しかしながら、これは理想的な場合であり、実際には適度にストールを挿入しユニットの待ちに対処しなければならない。

VLIW A に関しては、normal に比べパケット 1 個当たりにかかる処理サイクルは level 1 の時に、54.3% となり高速化を達成できている。しかし、リソースをパケット 1 個の時と同じだけしか使用していないため、ループ部分においてユニットの待ちが多く、ユニットの使用率は normal の時と比較し、level 1 のときの最大 31.1% をピークに減じていく。

VLIW B に関しては、normal に比べパケット 1 個当たりにかかる処理サイクルは level 1 の時に、35.4% となり高速化を達成できている。さらにユニットの使用率においても normal の時と比較し、level 1 のときに最小 33.0% を底に level が深くなるにつれて増加しており、パトリシアツリーの探索に適していると言える。

## 5 結論

VLIW B の命令セットにおいて、使用頻度の高いユニットを追加することによって、待ちを極力へらすことにより、リソースの有効活用と、パトリシアツリーの高速化という両面を達成することができた。