

プレフィックス展開と比較回路の併用による アクセス制御リストの効率的な管理

黄 恵聖[†] 山本 耕次^{††} 阿多 信吾^{†††} 井上 一成^{††} 村田 正幸[†]

[†] 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

^{††} 株式会社ルネサステクノロジ 〒 664-0005 兵庫県伊丹市瑞原 4-1

^{†††} 大阪市立大学 大学院工学研究科 〒 558-8585 大阪市住吉区杉本 3-3-138

E-mail: [†]{h-hwang,murata}@ist.osaka-u.ac.jp, ^{††}{yamamoto.koji4,inoue.kazunari}@renesas.com,
^{†††}ata@info.eng.osaka-cu.ac.jp

あらまし 高速なパケット分類を実現するために、TCAM (Ternary Content Addressable Memory) が使用されている。アクセス制御リストのフィールド中の範囲で表現されるポート番号をメモリに書き込む際に、複数のエントリを必要とすることが、TCAMの消費電力やコストが高くなる理由の一つになっている。本稿では、ポート番号をメモリに書き込む時、プレフィックス展開を利用してエントリ数を削減する方法、および新しいハードウェア構成を提案する。さらに実運用された ACL リストを用いた評価により、提案アーキテクチャがより効率的にエントリ数を削除できることを示す。

キーワード IP ルータ, ACL (アクセス制御リスト), TCAM, プレフィックス展開, 範囲比較回路, ハードウェアコスト

Efficient Management of Access Control List by Combining Prefix Expansion and Range Matching Devices

Haesung HWANG[†], Koji YAMAMOTO^{††}, Shingo ATA^{†††},

Kazunari INOUE^{††}, and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University

^{††} Renesas Technology Corporation

^{†††} Graduate School of Engineering, Osaka City University

E-mail: [†]{h-hwang,murata}@ist.osaka-u.ac.jp, ^{††}{yamamoto.koji4,inoue.kazunari}@renesas.com,
^{†††}ata@info.eng.osaka-cu.ac.jp

Abstract TCAM (Ternary Content Addressable Memory) is a special type of memory used in routers in order to achieve high speed packet classification. Port numbers are one of the five typical fields in an ACL (Access Control List). Those that are expressed in ranges have been the reason for raising the cost of hardware due to the requirement of multiple entries. In this paper, we suggest a method to reduce the number of entries when expressing ranges in TCAM, using prefix expansion and hardware modification.

Key words IP Router, Access Control List, Ternary Content Addressable Memory, Prefix Expansion, Range Matching Device, Hardware Cost

1. はじめに

現在 IP ルータにおいて、パケットのルーティングテーブルやアクセス制御リスト (ACL; Access Control List) を高速に検索するために、TCAM (Ternary Content Addressable Memory)

が幅広く用いられている [1]。しかしながら、TCAMには、消費電力、チップの面積、原材料コスト等の問題点があり、大容量の TCAM を搭載することが困難であることがたびたび指摘されている。特に、多くの ACL を管理するためには非常に大容量の TCAM が必要になるが、その一方で格納できる容量は

表 1 Example of Access Control List

```
access-list 101 permit tcp host 10.1.1.2 host 172.16.1.1 eq telnet
access-list 102 deny tcp any range 137 139 any
access-list 101 permit ip 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
access-list 111 deny icmp any 10.1.1.0 0.0.0.255 echo
access-list 191 permit udp any any range 16384 16483
```

限られており、管理者はハードウェア容量を意識した効率的な管理をせざるを得ず、ルータ設定を行う管理者の負担が大きい。また、TCAM 容量を超えた ACL を管理する場合、TCAM に格納できなかった ACL に対する性能が急激に劣化し、ルータの処理能力がワイヤスピードを保証できないなどの問題も生じる。

ACL が TCAM 容量を大量に消費する要因の一つとして、ポート番号の範囲指定がある。通常、ACL は送受信 IP アドレス、送受信ポート番号、およびプロトコル番号（それに加えて TCP フラグなど）の組み合わせによって表現されたルールによってパケットを識別し、パケットが指定されたルールにマッチした場合、そのルールで記述された処理（棄却、通過など）をパケットに対して行う。図 1 にアクセス制御リストの例を示す。ルータにおける ACL の指定は、完全一致だけでなく、範囲指定も可能なものが数多く存在する。特にポート番号については単一のポート番号を指定するだけでなく、任意の数値による範囲指定が頻繁に行われる。しかしながら一方で、TCAM は完全一致あるいは部分一致でなければ検索することができない。部分一致検索の場合、範囲が 2 のべき乗であれば単独エントリにより表現可能であるが、それ以外の任意の範囲については部分一致検索エントリを複数組み合わせなければならないため、ACL に対する必要 TCAM エントリ数が増大することになる。これをプレフィックス展開 (Prefix Expansion) と呼ぶ。通常、ACL の設定は人間の手によるものであり、その結果としてポート番号の範囲指定は人間が理解しやすい値 (5, 10, 100 の単位など) で指定されることが多い。このことが、2 のべき乗で表現される TCAM の範囲指定との不整合を生じさせ、結果としてプレフィックス展開によるエントリ数はより多くなる傾向になる。

ポート番号の範囲指定によるエントリ数の削減手法はこれまでも検討されているが、いずれの手法においても、TCAM デバイスはあくまでも既存のものを使用し、ソフトウェアあるいは外部デバイスの併用によりエントリ削減を実現しようとするものである。しかしながら、TCAM デバイス自体に拡張を加えることによってどの程度エントリが削減できるか、という点についてはこれまで検討されていない。

そこで本稿では、プレフィックス展開による TCAM 消費量の増大を抑制するため、範囲指定回路 (RMD; Range Matching Device) を既存の TCAM に組み込んだ改良型 TCAM チップを提案する。TCAM 内に RMD を実装することで、外部からは既存の TCAM と同様の利用法を維持しつつ任意の範囲指定を容易にサポートすることが可能となり、トータルとしてハードウェアコストを軽減できることを示す。さらに、RMD の容量が不足した場合においても、範囲指定がより柔軟に行えるよ

うにするため、TCAM に NOT および AND の演算機能を付加したものについても検討し、通常の OR 演算のみの組み合わせよりもより効率的に ACL を管理できることが可能であることを示す。

以降、2. において、範囲指定されたポート番号の取り扱いに関する問題点と既存技術について紹介する。3. では提案する RMD 組み込み型、および AND/NOT 演算機能を付加した TCAM アーキテクチャについて述べる。また、4. では RMD への格納ポリシー、および AND/NOT/OR を併用したプレフィックス展開のアルゴリズムについて述べる。そして 5. では実運用されている ACL に対して本提案手法を適用した場合の TCAM エントリ数削減効果について議論する。最後に 6. でまとめと今後の課題について述べる。

2. ポート範囲指定の問題点と既存の解決手法

TCAM とは、アドレスを入力としてそのアドレスに格納されている内容を返す通常の RAM (Random Access Memory) とは異なり、格納された内容を探索し、入力データと同じ情報が TCAM 内のどの位置に格納されているかを返す記憶デバイスである。TCAM の各ビットには 0, 1 および任意の値を表す * を格納することができる。検索は入力として探索キーが与えられたとき、そのキーの値と完全に一致する TCAM エントリのアドレスが返される。これを利用して、ルータでは ACL で制御したいパケットのヘッダ情報を TCAM に格納し、到着したパケットのヘッダ情報を入力として TCAM を検索する。そして、該当するエントリが見つければ、そのエントリに対する処理を行う。

CAM には、入力データに対して完全に一致したエントリのみを返す Binary CAM (BCAM) と、* を使用することによって * 以外の範囲の部分一致検索を行える Ternary CAM (TCAM) が存在する。ルータの経路表や ACL における IP アドレスの格納については、アドレスにおけるホスト ID 部分を * で表現することで、ネットワークの単位で集約できることなど、TCAM との親和性が高い。しかし、範囲で表現されているポート番号を格納するためには、IP アドレスのような単純な集約を行うことができず、範囲に該当するすべてのエントリを記述することになる。たとえば、1024-65535 で指定されたポート番号の範囲を格納する場合、もっとも単純な方法は、1024, 1025, ..., 65534, 65535 のすべてのエントリを個別に記述し、完全一致による検索を行うことである。しかしこの方法では 1 つのポート範囲を指定するために 64,512 行のエントリが必要になり、資源を大量に消費する。本稿ではこれを **フル展開** と呼ぶ。

この問題を解決するために、これまでにプレフィックス展開と呼ばれる方法や範囲比較のためのハードウェア回路の導入などの手法が検討されている。本章では、これら既存の解決手法について紹介し、その問題点について述べる。

2.1 プレフィックス展開

複数のポート番号を単一のエントリで記述するために、TCAM の * を用い、ポート番号の下位ビットを * で格納することが考えられる。下位 i ビットを * とした場合、 2^i の単位で範囲を記

述することが可能である。これを用いることで 1024-65535 は

```

1***** 32768-65535
01***** 16384-32767
001***** 8192-16383
0001***** 4096-8191
00001***** 2048-4095
000001***** 1024-2047

```

の 6 行で表現できる。これを**プレフィックス展開**と呼ぶ。

フル展開と比較して、プレフィックス展開は大幅に必要 TCAM エントリ数を削減することは可能であるものの、依然として単一の ACL に対して数多くのエントリ数を必要とする。このため、さらにエントリ数を削減する手法についてこれまでも検討がなされている。

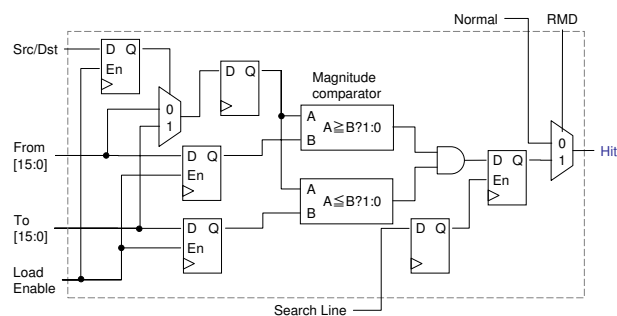
[2] では、ACL にある範囲をエンコードし、各範囲をそれぞれ一つのビットにマッピングする技術である。さらに、一つの範囲をエンコーディングした結果、他の範囲に与える影響も考慮し、ポート範囲が TCAM に格納される際の平均フル展開率 6.20 に比べ、1.23 まで削減できることを示している。TCAM の検索ビット長は限られていることから、範囲の種類が多くなるほど、TCAM でのアクセス回数が増加する。また、マッピングを行うために別のメモリが必要となり、検索回数が増加するなどの問題点が考えられる。

また [3] では、ポート番号で指定された範囲をできるだけ 2 のべき乗で表現できるように、ACL で指定されたポート範囲に、ACL の挙動に悪影響を与えない制約内でポート番号範囲の追加削除を行い、異なる ACL における範囲の共通化、簡略化をすることにより、必要エントリ数を削減する。最適化がなされることにより、エントリ数が最大 50% 削減できることを示している。しかしながら本方式では、最適化のために ACL 情報をオリジナルのものから変更しなければならず、実 ACL との整合性がとりにくい。さらに、ある ACL のポート番号範囲が変更された場合、範囲に対する依存関係が複雑になっているために、更新が必要となるエントリが多く発生するなど、ACL の更新に対して柔軟に対応できないという問題がある。

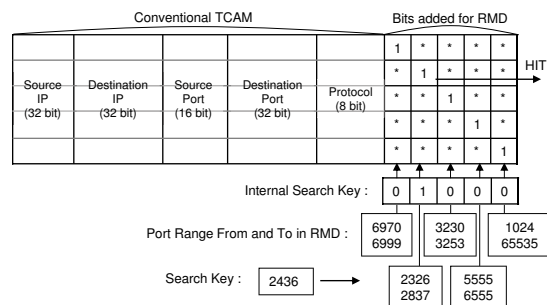
一方 [4] では、ある範囲を、プレフィックス展開するのではなく、任意の三値 (0*1*0) を使用し、範囲のコーディングを行っている。このコーディングの結果は、一つのエントリで使用されていないビット空間に書き込んでいる。結果より、エンコードに 32 ビットを使用した場合、全体のエントリ数を最大 50% 削減可能である。しかし、[4] 自体でも指摘されているが、最適なエンコード手法は ACL の内容に依存し、最適解を得るためのアルゴリズムが事前に決定できない、という問題点がある。

2.2 比較演算ハードウェアによる範囲指定

Cisco の IP ルータでは、範囲指定されたポート番号をより少ない TCAM 資源で格納するため、外部に L4Op (Layer-4 Operation Unit) を実装している [5]。これは、ポート番号に対して比較演算を可能とするハードウェアである。具体的には、L4Op にポート番号および比較演算子 (gt (greater than), lt (less than), range, neq (not equal)) を指定し、入力されたポート番号が、指定された比較演算を満足するかどうかを判定する。範囲指定された ACL のルールを満たすかどうかは、ま



(a) Example of Range Matching Device



(b) Bits for Range Matching Device in TCAM

図 1 Example of Range Matching Device with TCAM

ずポート番号を除いた情報で TCAM による完全一致検索を行い、ヒットしたエントリについて、さらに指定された L4Op の比較演算を満たすかどうかを検証することで行われる。これにより、範囲指定されたポート番号は TCAM に格納することなく独立に L4Op において管理され、ACL のルールは単一 TCAM エントリで表現することが可能となり、TCAM により多くの ACL を格納することができる。

しかしこの手法での最大の問題は TCAM の外部に L4Op ハードウェア回路を実装している点である。L4Op では TCAM への格納データが、ポート番号によるものと全くフォーマットが異なるため、外部から TCAM へ検索データを入力する際、他の方式と同じ処理レートを実現するためには、L4Op フォーマットとポート番号フォーマットによりデバイス間で消費する I/O バンドが倍増してしまう。即ち、同一動作周波数のもとではデータ入出力ピン数を 2 倍に拡張しなければならない。デバイスの物理サイズが同一の場合、外部 I/O のための入出力ピン数を増やすと、相対的にピン間隔が狭くなり、より高い精度での結線が求められる。また近年の高速ハードウェアデバイスでは、Signal Integrity およびピン間干渉 (SSO (Simultaneous Signal Output) noise) が重要なファクターとなっており、配線数が多くなることによってピン間干渉ノイズばらつきが増大するため、ノイズに対する対策も必要とされ、ボード設計の自由度が制約される、多層化が必要となるなど、ボード設計に多大な労力と、コストを払う事になる。

以上のことから、同様の動作をするハードウェアチップであれば外部ピン数がより少ない方が望ましい。

3. 範囲指定を実現する TCAM ハードウェア

2. で述べたとおり、プレフィックス展開の最適化および L4Op

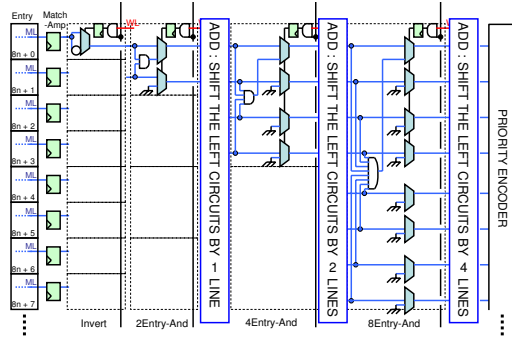


図 2 NOT/AND Operation Part

によるエントリ数の削減は、最適化が困難、更新が容易ではない、あるいはハードウェアの性能に影響を与えるなどの問題が存在する。これらの手法はあくまでも TCAM デバイスは既存のものを使用し、ソフトウェアあるいは外部デバイスの併用によりエントリ削減を実現しようとするものである。本稿では、TCAM デバイスの拡張によりポート番号の範囲指定をサポートし、エントリ数の削減を目指す。

ここでは TCAM の拡張として、範囲判定回路の追加、およびエントリ間論理演算子の追加について検討する。個別の詳細について以降の各節で説明する。

3.1 範囲判定回路 (Range Matching Device)

図 1(a) は提案する範囲判定回路 Range Matching Device (RMD) を示している。入力線として、Normal, RMD, Src/Dst, From, To, Load Enable, Search Line があり、出力線として Hit がある。Normal と RMD は、ユーザが最初にこのデバイスを普通のメモリとして使用するか、範囲を記憶するデバイスで使用するかを決定する入力である。RMD にある範囲を記憶させるには、Load Enable により、その範囲が送信元か宛先か区別する Src/Dst に信号をセットする。同時に、ポート番号範囲の From と To の値を 16 ビットで入力する。

検索時は、上記で設定された、情報を元に、送信元ポート番号、宛先ポート番号から、どちらかが選択され、From および To との大小比較が行われる。From より大きく、かつ To より小さい場合に、Hit が 1 となる。このようにして各 RMD は、検索対象のポート番号に対して範囲内かそうでないかを 0 あるいは 1 で返す。各 RMD の Hit は TCAM の範囲比較ビットに送られる。TCAM では、送受信 IP アドレスとポート番号、プロトコル番号などのフィールドに加え、RMD 判定用のビットを割り当てる。RMD ごとに 1 ビット割り当てて、使用した範囲を持つ RMD に対してビットを 1 にセットし、残りを * にセットする。図 1(b) に TCAM 内の構造例を示す。この例の場合、2 行目のエントリは 2326-2837 の範囲を指定するため、TCAM 内の RMD の部分は *1*** となる。このときポート番号 2436 は、各 RMD での検証の結果、01000 となり、2 行目のエントリにヒットすることがわかる。

3.2 エントリ間論理演算回路

TCAM エントリへのアクセスは少なくとも一つ以上一致するエントリが見つかった場合に検索が成功したと判断できることから、各エントリの結果を論理和で演算していると考えられる。

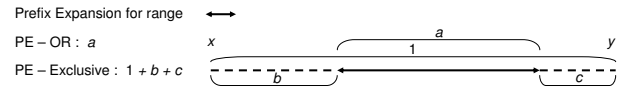


図 3 Prefix Expansion Algorithms

ともできる。本稿では従来の OR に加えて、TCAM エントリの NOT および AND を実現するための TCAM 拡張について検討する。図 2 に NOT/AND 演算を追加した TCAM 回路の修正図を示す。まず左側の回路で NOT 演算を実現する。次に、論理積である AND については、必要に応じて動的にハードウェア回路を変更することはできないため、あらかじめ 2, 4, 8 エントリの論理積をあらかじめ実装し、ACL のプレフィックス展開に応じてどの位置に記述するかを決定する。

4. TCAM デバイスの制御

4.1 RMD 格納ポリシー

RMD の回路数は有限であり、TCAM 内部の仕様で決定されたものである。したがって、エントリ数の削減を最大限にするためには、もっとも削減効果の高い範囲を RMD に入力することが望ましい。そこで本稿では、RMD へ格納するポート番号範囲の順位付けを以下で定義する Weight に基づいて決定する。

Weight = (プレフィックス展開後の行数-1) × (その範囲を参照している ACL 数)

すなわち、プレフィックス展開により多くのエントリが必要になるほど、あるいはその範囲を持つ ACL 数が多い範囲ほどより RMD へ格納される傾向が強いことを意味する。

4.2 プレフィックス展開の最適化

本節では、3.2 で拡張した、エントリ間の NOT/AND/OR を実現する TCAM を用いたプレフィックス展開の最適化アルゴリズムについて示す。ここでは、最適化の程度および実現の容易さを勘案した 3 種類のアプローチを提案する。図 3 にアルゴリズムの概要を示す。PE-OR は、従来のプレフィックス展開である。範囲を 2 のべき乗の組み合わせにより表現する。From が 2^i で、To が $2^{i+1} - 1$ の時に最も優れていて 1 行で表現できる。ただし、16385-65534 の場合は、最悪値である 29 行となる。

PE-Exclusive は、範囲全体を完全包含する $[2^i, 2^{i+1} - 1]$ を決定し、不要部分を 2 のべき乗の組み合わせにより表現する。この手法は PE-OR では表現が難しい範囲に効果的である。たとえば 16385-65534 の場合、PE-Exclusive を用いることで、16384-65535 AND (NOT 16384) AND (NOT 65535) の 3 行で表現できる。PE-OR と PE-Exclusive は補間関係にある。

PE-MIN は、総当たりによりもっともエントリが少なくなるよう最適化されたプレフィックス展開を求める。以下にアルゴリズムを示す。

- (1) まず、範囲 $[x, y]$ 内に存在する $k2^x (1 \leq x \leq 15)$ のうち x がもっとも最大になるものを選択し、それを **基準点** b とする。さらに範囲を基準点を境界として 2 つに分割する
- (2) 基準点より大きい範囲について、 $b = j2^i < x \leq$

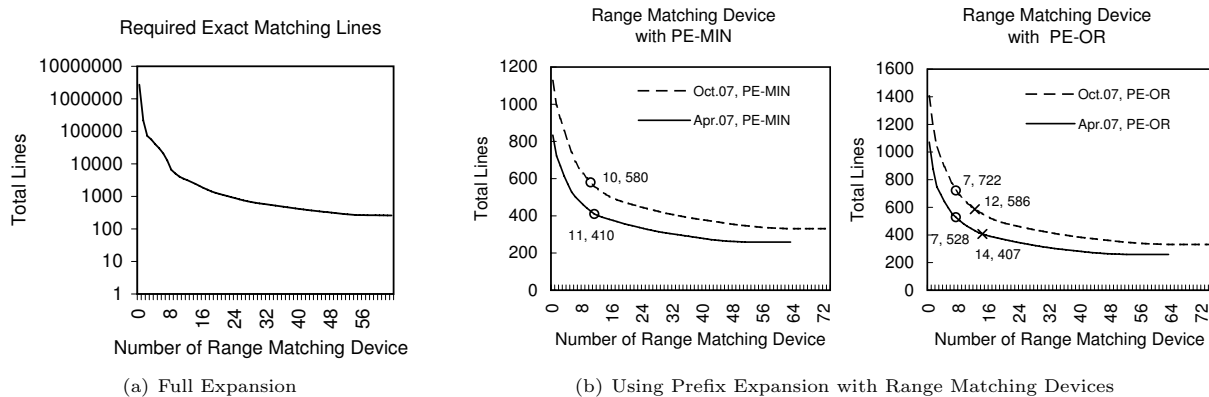


図 4 Comparisons of Total Required Entries (w/ or w/o RMD)

表 2 Two ACL Sets for Evaluation

| Date Captured | Apr. 07 | | Oct. 07 | |
|-------------------------|---------|------|---------|------|
| # of unique ACL entries | 6,440 | | 7,202 | |
| | Src | Dest | Src | Dest |
| # of ranges | 3 | 256 | 6 | 325 |
| # of unique ranges | 63 | | 74 | |

$(j+1)2^i$ を満たす最大の $i (i < x)$ を求める

(3) $[j2^i, x]$ および $[x, (j+1)2^i]$ の 2 つの範囲について、プレフィックス展開をした場合のエントリ数をそれぞれ再帰的に求め、それぞれ x_l, x_r とする

(4) $x_l < x_r$ なら、プレフィックス $[b, j2^i]$ を追加、そうでなければ $[b, (j+1)2^i]$ を追加する

(5) 基準点より小さい範囲も同様に求める

5. 評価

本章では、実運用されている ACL を用いて、実際に範囲指定されているポート番号を TCAM に格納したときに必要となるエントリ数について評価し、その結果について考察する。評価で使用した ACL の概要を表 2 で示す。評価には稼働中のキャンパスネットワークルータにおいて異なる時期に取得した 2 つの ACL を用いている。

5.1 比較判定回路によるエントリ削減効果

まず、比較回路によるエントリ数の削減効果について、図 4 により示す。図 4(a) は、範囲指定されたポート番号すべてを個別のエントリとして格納（フル展開）した場合の必要エントリ数を示したものである。この図より、比較回路が存在しない場合は必要エントリ数が 100 万オーダーになることがわかる。これは、クライアント側の動的ポート指定として使用される 1024-65535 の範囲が、フル展開により 64,512 エントリ消費することが大きな要因である。このため比較回路を 1 つ実装し、そこに 1024-65536 を格納するだけで、必要エントリ数を約 1/10 にまで軽減できる。

次にプレフィックス展開を行った場合の RMD の削減効果について、図 4(b) に結果を示す。節の RMD 格納ポリシーに基づき、Weight の値が最も大きくなるポート番号範囲より RMD に格納することを考える。その結果、プレフィックス展開を行った場合、RMD を追加することによって、必要エントリ数を大

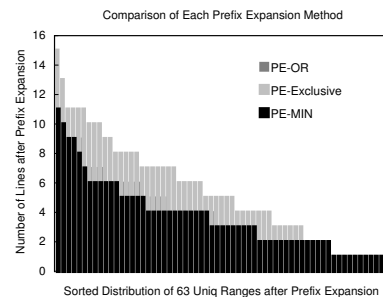


図 5 Comparisons of Prefix Expansion Algorithms

きく減少させることができる。特に、RMD の数が少ない場合の削減効果は大きく、提案方式である PE-MIN を用いた場合、Oct.07 で 10 個、Apr.07 で 11 個の RMD があれば、必要エントリ数をそれぞれ半減させることが可能となる。また、従来方式である PE-OR の場合は、7 個の RMD でそれぞれ半減できる。

さらに PE-MIN と PE-OR を比較すると、必要エントリ数は同じ数の RMD を用いた場合で、AND/NOT 比較演算子を追加することで、約 25% のエントリ数を削減することが可能である。PE-OR を用いて、PE-MIN と同程度の削減効果を得るためには、RMD をさらに 2 から 3 個追加する必要がある。

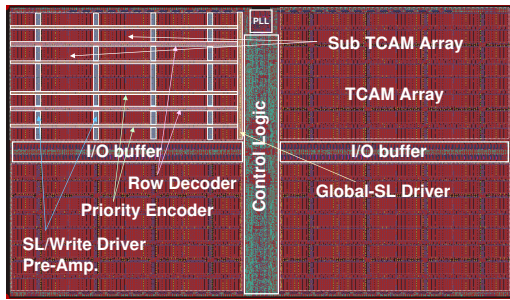
5.2 比較演算子を用いた最適化プレフィックス展開の効果

提案したプレフィックス展開アルゴリズムの違いによる影響を調べるため、各方式によって ACL の必要エントリ数がどのように変化するかを調べた。図 5 は各展開アルゴリズムを用いてプレフィックス展開を行った場合の各 ACL ルールに対する必要エントリ数を比較したものである。結果として、PE-MIN を使用することにより、PE-OR と比較して必要エントリ数を約 25% 削減できることがわかる。

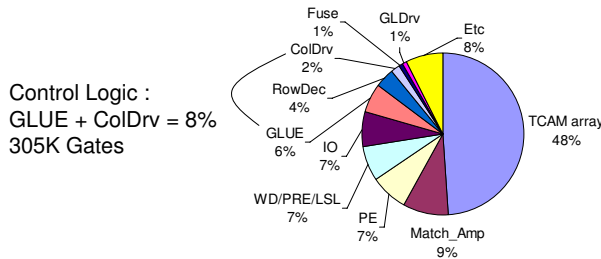
エントリ削減数のみを考えた場合、PE-MIN がもっとも少ないエントリ数を実現できているが、TCAM 内部に NOT および AND の演算回路が必要になる。

5.3 比較判定回路によるコストオーバーヘッドの検討

最後に、RMD を追加することによるハードウェアのコスト増加について検討する。図 6(a) に現在流通している 90 nm プロセスルールによる TCAM のハードウェア構成を示す。このハードウェアの要素ごとの構成比率は図 6(b) で示されている。



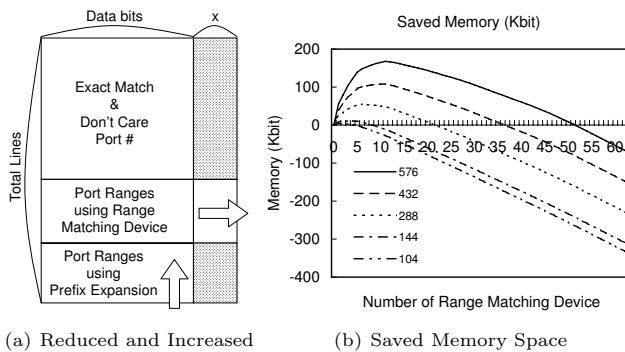
(a) TCAM VLSI in 90nm Technology



RMD: 580 Gates x 200 RMDs x 2(margin) = 232K Gates
Current TCAM : TCAM with RMD = 100 : 106

(b) Cost compared to the existing chip

図6 Existing TCAM and Cost comparison with suggested hardware



(a) Reduced and Increased Memory Space

(b) Saved Memory Space

図7 Wasted and Saved Memory when Range Matching Devices are used

このうち、全体の約 8% を占める Control Logic の構成ゲート数はおよそ 305 K である。一方 RMD は 1 つあたり 580 ゲートを使用する。この RMD を 200 個搭載するためには、レイアウトマージンを含めるとおよそ 232 K ゲート必要になる。これは、TCAM 全体の約 6% に相当する。したがって、6% 程度の製造コストの増加によって、TCAM の必要エントリ数を大幅に削減することが可能となり、より多くの ACL を格納することができる。なお、構成比率については今後 65 nm に移行しても大きく変化しないものと考えられる。

また、図 7(a) に示すとおり、RMD を 1 個増加させると、その RMD にマッチするかどうかを判定するために TCAM のビットが 1 ビット必要となる。RMD を増加させることにより必要エントリ数を削減できる一方で、RMD 判定用に TCAM の検索ビット長が増大するため、両者のトレードオフを考慮しつつ実装する RMD の数を決定する必要がある。

となる。

図 7(b) は、RMD を追加することによる節約可能なメモリを示したものである。ここでは、データビットを 104, 144, 288, 432, 576 ビットである場合、それぞれに対して RMD が増加した場合、各 RMD の数において節約できるメモリ量を示している。

この結果より、RMD の個数を増加させることによりはじめはメモリを削減することができるものの、ある値より多い RMD を導入した場合、そのオーバーヘッド増の影響が大きく、必要メモリサイズが削減できないということが分かる。また、データビットが短ければ短いほど、RMD が多くなると無駄になるメモリ空間が増加する。それぞれのデータビットに対して、最も節約できるメモリ空間の最適 RMD 数は、ACL のデータベース中に、範囲で表現されているポート番号の比率にも依存と考えられる。

6. まとめと今後の課題

本稿ではアクセス制御リストの中で特に範囲で表現されているポート番号を TCAM でより効率的に管理するため、プレフィックス展開と範囲比較デバイスの併用した、新しい TCAM アーキテクチャを提案した。またさらにエントリを削減するための範囲指定方法、および最適化プレフィックス展開手法も提案し、実運用 ACL を用いた評価により、提案アーキテクチャが既存のプレフィックス展開よりもより効率的に ACL を管理できることを示した。今後の課題としては、多様な実運用 ACL を使用して提案手法を評価することと、実際に改良ハードウェアを搭載したチップを試作し、ネットワークプロセッサ全体の性能を評価することなどがある。

謝辞

本研究の遂行において、多大なるご協力をいただいた、大阪大学サイバーメディアセンター准教授、長谷川剛博士に謝意を示す。また、本評価の一部は株式会社ルネサステクノロジインターンシッププログラムにおいて行った。関係諸氏に感謝申し上げる。

文献

- [1] H. Liu, "Reducing Routing Table Size Using Ternary-CAM," *Hot Interconnects (HoTI)*, vol. 9, no. 2001, pp. 22–24, 2001.
- [2] H. Che, Z. Wang, K. Zheng, and B. Liu, "DRES: Dynamic Range Encoding Scheme for TCAM Coprocessors," *technique report, Univ. of Texas at Arlington*, <http://crystal.uta.edu/hche/dres.pdf>, 2006.
- [3] Q. Dong, S. Banerjee, J. Wang, D. Agrawal, and A. Shukla, "Packet Classifiers In Ternary CAMs Can Be Smaller," in *ACM SIGMETRICS '06/Performance '06*, (New York, NY, USA), pp. 311–322, ACM Press, 2006.
- [4] K. Lakshminarayanan, Rangarajan, and S. Venkatachary, "Algorithms for Advanced Packet Classification with Ternary CAMs," in *ACM SIGCOMM '05*, (New York, NY, USA), pp. 193–204, ACM Press, 2005.
- [5] M. A. Ross, S.-D. Chen, and A. V. Bechtolsheim, "Logical Operation Unit for Packet Processing, US Patent 6,658,002," December 2003.