

面積制約を考慮したCAMプロセッサ向け ハードウェア/ソフトウェア協調設計手法

石川裕一朗[†] 宮岡祐一郎[†] 戸川 望^{†,††} 柳澤 政生[†] 大附 辰夫[†]

[†] 早稲田大学理工学部電子・情報通信学科

^{††} 北九州市立大学国際環境工学部情報メディア工学科

^{†††} 早稲田大学理工学総合研究センター

〒 169-8555 東京都新宿区大久保 3-4-1

Tel: 03-5286-3396, Fax: 03-3203-9184

E-mail: ishikawa@ohtsuki.comm.waseda.ac.jp

あらまし 我々はCAM(連想メモリ)を使用するプロセッサを対象としたハードウェア/ソフトウェア協調合成システムを構築中である。現在のシステムはC言語で記述されたアプリケーション記述を入力としてそのアプリケーションを実行するプロセッサの最適なハードウェア構成を出力する。本稿では、現在のシステムを拡張し、面積制約機能を付加したCAMプロセッサ向けハードウェア/ソフトウェア協調合成システムを提案する。提案手法では面積制約を満足した上で実行時間を最小化するCAMワード数を導出し、CAMの一部をRAMに置換してプロセッサの面積を削減する。計算機実験により、面積制約を満たした上で、システムに入力されたアプリケーションを最速に実行するプロセッサの構成を出力できる事を確認した。キーワード 連想メモリ, CAMプロセッサ, 一致検索, 面積制約, ハードウェア/ソフトウェア協調設計,

A Hardware/Software Cosynthesis Method for CAM Processor with Area Constraints

Yuichiro ISHIKAWA[†], Yuichiro MIYAOKA[†], Nozomu TOGAWA^{†,††}, Masao YANAGISAWA[†], and
Tatsuo OHTSUKI[†]

[†] Dept. of Electronics, Information and Communication Engineering, Waseda University

^{††} Dept. of Information and Media Sciences, The University of Kitakyushu

^{†††} Advanced Research Institute for Science and Engineering, Waseda University

3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan

Tel: +81-3-5286-3396, Fax: +81-3-3203-9184

E-mail: ishikawa@ohtsuki.comm.waseda.ac.jp

Abstract We have been building the hardware/software cosynthesis system for a processor core with a content addressable memory (CAM). We input a description of an application program written in C language into the system, and the system outputs an optimal hardware configuration of a CAM processor which executes an inputted application program. This paper extends our hardware/software cosynthesis system which incorporates area constraints for a CAM processor. The system computes the number of CAM words which minimizes the execution time with meeting the area constraints. We reduce the CAM processor's area by replacing CAM with RAM according to the word number that the system computed. Experimental results for practical application program show that the system can output a configuration of the processor which executes the application program fastest with meeting the area constraints.

Key words Content Addressable Memory, equivalent search, area constraint, HW/SW cosynthesis

1. まえがき

Content Addressable Memory (CAM: 連想メモリ) は機能メモリ的一种である。通常の RAM が持つアドレスによるアクセス機能に加え、メモリ内に保持しているデータに対して並列に一致検索あるいは閾値検索する機能を持つ。CAM セルアレイの周辺に検索結果のデータを処理する簡単な回路を付加することで、SIMD 型プロセッサとして利用できる。CAM はネットワークアプリケーション [1], [2], キャッシュメモリ [10], 画像プロセッサ [3], [4] などのアプリケーションに利用されている。CAM の並列処理機能を有効に利用するためには、目的とするアプリケーションごとに CAM を使用したプロセッサ (以下, CAM プロセッサ) を設計する必要がある。

我々は、CAM プロセッサを対象とした自動合成システムおよびハードウェア/ソフトウェア協調合成システムを構築している [5]–[8]。既存のシステムは C 言語で記述されたアプリケーションと時間制約を入力として、時間制約を満たす範囲で面積が最小となる CAM プロセッサのハードウェア構成を出力する。CAM プロセッサの面積はシステム使用者が指定できないので、システムの出力が使用者の要求を満たさない場合がある。本稿では、面積制約機能を付加した CAM プロセッサ向けハードウェア/ソフトウェア協調合成システムを提案する。

提案手法では C 言語によるアプリケーション記述と CAM プロセッサの面積制約 (単位: gate count (注1)) を入力とする。CAM セルアレイの種類を選択と RAM への置き換えによって、与えられた面積制約を満たす範囲で入力されたアプリケーションを最速で実行する CAM プロセッサのハードウェア構成を出力する。

2. CAM プロセッサの構成

CAM プロセッサは、マイクロプロセッサユニット (MPU), CAM ユニット, 命令 RAM, データ RAM の 4 要素で構成される (図 1)。命令 RAM およびデータ RAM は既存の SRAM の使用を仮定する。対象となる CAM ユニットおよびマイクロプロセッサユニットのアーキテクチャを定義する。

2.1 CAM ユニット

CAM ユニットは、図 2 のようにデータの内容を保持し検索を実行する CAM セルアレイ、ワード単位の検索結果を並列処理するワード処理系およびアドレスによるセルアレイへのアクセスを実現するアドレスデコーダで構成される。

2.1.1 CAMセルアレイ

CAMセルアレイは通常の RAM が持つワード単位のデータの読み出し/書き込み機能に加え、ワード並列の検索機能と並列書き込み機能を持つ。

検索機能として一致検索に加え、閾値検索機能 (greater than 検索, less than 検索, 以上検索および以下検索) が使用できる。表 1 に示すように実行可能な機能の異なる 10 種類の CAM セルアレイを用意している。複数の機能を持つセルアレイは、 W_{CS} ビットの信号線 CS によって機能を選択し、目的の検索を実行する。

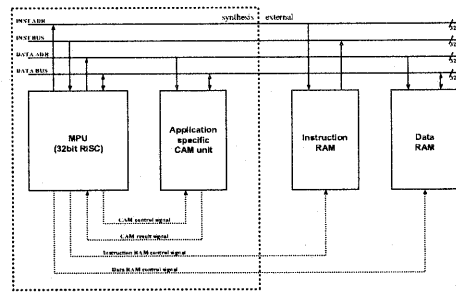


図 1 CAM プロセッサ

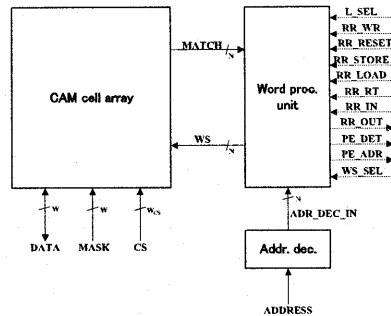


図 2 CAM ユニット

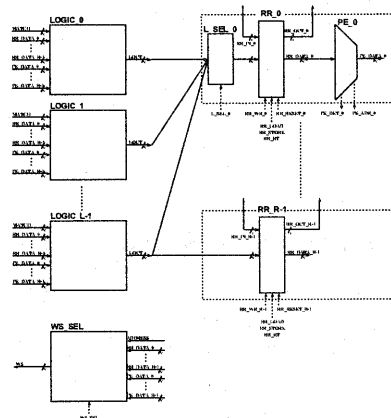


図 3 ワード処理系アーキテクチャ

2.1.2 ワード処理系

ワード処理の構成を図 3 に示す。ワード処理系は 1 つ以上の算術論理演算ユニット (LOGIC_x), 1 つ以上のレスポンスレジスタユニット (RR_x) およびワードセレクト線セレクトクタ (WS_SEL) で構成される。

a) 算術論理演算ユニット

算術論理演算ユニットは CAM ユニット内における算術論理演算を実行する。入力 は CAM セルアレイの出力 MATCH, レスポンスレジスタの出力 RR_DATA_x, およびプライオリティエンコーダの出力 PE_DATA_x から任意に使用できる。これらの入力の演算結果を LOU_T に出力する。

b) レスポンスレジスタユニット

レスポンスレジスタユニット RR_x は算術論理演算ユニットの演算結果を保存する。レスポンスレジスタユニットは算

(注 1): gate count は 2 入力 NAND ゲートの面積を 1 とした相対値である。

表 1 CAM セルアレイにおける機能

種別	WCS [bit]	機能
CAM1	0	一致検索
CAM2	0	greater than 検索
CAM3	0	less than 検索
CAM4	1	greater than 検索, 一致検索
CAM5	1	less than 検索, 一致検索
CAM6	1	greater than 検索, less than 検索
CAM7	2	greater than 検索, less than 検索, 一致検索
CAM8	0	以上検索
CAM9	0	以下検索
CAM10	1	以上検索, 以下検索

術論理演算ユニットセレクトラ L_SEL_x, レスポンスレジスタ RR_x およびプライオリティエンコーダ PE_x の 3 要素で構成される。入力として任意の術論理演算ユニットの出力を使用できる。

c) ワードセレクト線セレクトラ

ワードセレクト線セレクトラはセルアレイのワードセレクト線 WS を駆動する信号を選択する。

2.1.3 アドレスデコーダ

アドレスデコーダはアドレスによってセルアレイへアクセスするために使用される。アドレスの値からワード処理系のワードセレクト線セレクトラへ入力する ADR_DEC_IN を生成する。

2.2 マイクロプロセッサユニット

マイクロプロセッサは、CAM ユニットの制御および CAM ユニットで実行される SIMD 型並列処理以外のアプリケーション処理を担当する。C 言語で記述されたアプリケーションプログラムを実行するのに適した一般的な 32bit RISC 型 MPU アーキテクチャを採用する。

3. 面積制約を考慮した CAM プロセッサ 協調設計手法

本研究室で研究しているハードウェア/ソフトウェア協調合成システム (以下本システム) の面積制約機能は、面積制約を満たす範囲でアプリケーションを最速に実行する CAM プロセッサの構成を決定する。アプリケーション記述と面積制約値を入力とし、面積制約を満たす範囲で実行時間が最速の CAM プロセッサのハードウェア構成と、入力されたアプリケーションを実行した際のアセンブリコードを出力する。

CAM 機能を実現するため、アプリケーション記述には 5 つの CAM 検索関数 (match, greater_than, less_than, greater_equal, less_equal) と、rr_drive (レスポンスレジスタ書き込み), ws_drive (並列書き込み), word_read (セルアレイ書き込み), word_write (セルアレイ書き込み), rr_reset (レスポンスレジスタリセット) などの CAM 関数を使用する [6]。CAM 検索関数 match は一致検索を実行する。greater_than, less_than は閾値検索を実行し, greater_equal は以上検索を, less_equal は以下検索を実行する。既存のシステム [7] は CAM プロセッサのハードウェア構成を出力する時、全ての CAM 関数を CAM のハードウェアによって実現する。

既存のシステムは入力されたアプリケーションに対して CAM 関数の記述から実行時間が最速となる CAM セルアレイを自動的に選択する。例えばアプリケーション内に CAM 関数 match と greater_than が記述されていた場合、最適な CAM セルアレイは CAM4 となる。

この機能に加えて、既存のシステムはアプリケーションを入力する際に使用する CAM セルアレイを指定することができる。上記の例の場合、CAM1 のセルアレイを指定すると、システムは greater than 検索を一致検索のみによって実現する。CAM セルアレイを指定しなかった場合はシステムが実行時間が最速となる CAM セルアレイを自動的に選択する。

面積制約を考慮した CAM プロセッサ向けハードウェア/ソフトウェア協調設計手法を提案する。

1. アプリケーションを既存のシステムに入力する。使用する CAM セルアレイとして CAM1 を指定する。既存のシステムはアプリケーションを実行時間が最速で実行する CAM プロセッサのハードウェア構成を出力する。
2. 1. で出力されたハードウェア構成に対して面積見積もり式 [9] を参照して CAM プロセッサの面積見積もり値を導出する。
3. 2. で出力された面積見積もり値が面積制約値より大きい場合、CAM セルアレイの一部を RAM に置き換える事で CAM プロセッサの面積を削減する。置き換えるワード数は本システムの面積制約機能が自動的に導出する。
4. 3. で導出された RAM へ置き換えるワード数に従って、入力されたアプリケーションを自動的に書き換える。
5. 書き換えたアプリケーションを再度既存のシステムに入力し、アプリケーションを実行した際のアセンブリコードを再生成する。
6. 以上の作業を CAM2 から CAM10 について実行する。それぞれの場合のアセンブリコードから実行時間を比較し、最速のものをシステムの出力とする。

面積制約の手順 1. で既存のシステムの CAM セルアレイを指定できる機能は 3.1 章で紹介する。面積制約の手順 3. で CAM セルアレイの一部を RAM に置き換えた時の検索方法は 3.2 章で説明する。RAM へ置き換えるワード数を導出する手法は 3.3 章で説明する。提案手法による利点を 3.4 章で説明する。

3.1 CAM セルアレイの指定

既存のシステムはアプリケーション記述を入力する際に、使用する CAM セルアレイを指定することができる。一致検索と greater than 検索を実行するアプリケーションを入力する場合、CAM4 のセルアレイを使用する CAM プロセッサが最速にアプリケーションを実行できる。CAM1 のセルアレイを指定した場合、CAM1 は一致検索のみ実行できるので、CAM プロセッサは greater than を一致検索で実現する。CAM1 のセルアレイを使用した CAM プロセッサは、CAM4 を使用した場合よりアプリケーションの実行時間は遅いが、面積は小さい。

面積制約において、CAM4 を使用した CAM プロセッサの面積が面積制約値より大きい時、CAM1 を使用した CAM プロセッサに代替することで CAM プロセッサの面積を削減できる。CAM セルアレイを指定できる既存のシステムの機能は面積制約に利用できる。

以下のアルゴリズムによって、一致検索による greater than 検索等の閾値検索を実現する。アルゴリズムの実行時間はセルアレイのビット幅に比例し、ワード数には依存しない。

greater than 検索

1. レスポンスレジスタを 0 にクリアする。

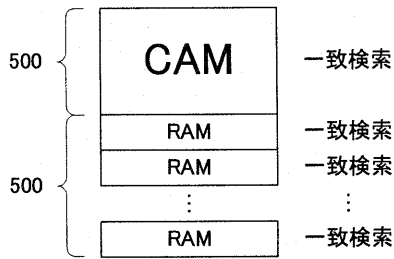


図4 CAMをRAMに置き換えた時の一致検索

2. 検索データを上位から下位へと走査し、最初の0のビットをターゲットビットとする。
3. ターゲットビットを1にして新しい検索データを作る。
4. ターゲットビットより下位のビットをマスクして一致検索する。その結果とレスポンスレジスタの論理和を取り、レスポンスレジスタに格納する。
5. ターゲットビットを0に戻し、検索データをさらに下位へと走査する。次に発見した0を新たなターゲットビットとし、3.へ。最下位ビットまで走査し、0が無ければ終了する。

less than 検索

1. レスポンスレジスタを0にクリアする。
2. 検索データを上位から下位へと走査し、最初の1のビットをターゲットビットとする。
3. ターゲットビットを0にして新しい検索データを作る。
4. ターゲットビットより下位のビットをマスクして一致検索する。その結果とレスポンスレジスタの論理和を取り、レスポンスレジスタに格納する。
5. ターゲットビットを1に戻し、検索データをさらに下位へと走査する。次に発見した1を新たなターゲットビットとし、3へ。最下位ビットまで走査し、1が無ければ終了する。

以上検索、以下検索は、それぞれ greater than 検索、less than 検索後に検索データで一致検索し、その結果とレスポンスレジスタの論理和をとることで実現できる。

3.2 RAMへの置き換えた時の検索方法

CAMプロセッサの面積を削減する方法として、CAMセルアレイの一部をRAMに置き換える手法がある。1ワード当たりのCAMセルアレイの面積は1ワード当たりのRAMの面積より大きい。よって、CAMセルアレイの一部をRAMに置き換える事でCAMプロセッサの面積を削減できる。

図4にCAMセルアレイの一部をRAMに置き換えた時の一致検索の様子を示す。図4にはCAMに500ワード、RAMに500ワードのデータが格納されている。CAMに格納されている500ワードのデータは並列に一致検索する。RAMに格納されている500個のデータは1つずつ直列に一致検索する。よって、CAMをRAMに置き換える量を増やすと、CAMプロセッサの面積は減少するが、一致検索にかかる時間は増加する。

3.3 置き換えワード数の導出

本システムは、CAMをRAMに置き換える際に、置き換えるワード数を面積見積もり式を参照に導出する[9]。CAMプロセッサ全体の面積 A_{CAM} の面積見積もり式を以下に示す。

$$A_{CAM} = A_{MPU} + A_{ARRAY} + (A_{LOGIC_0} + A_{LOGIC_1} + \dots) + (A_{L_SEL0} + A_{L_SEL1} + \dots) + R \cdot A_{RR} + P \cdot A_{PE} + A_{WS_SEL} + A_{ADR}$$

A_{MPU}	マイクロプロセッサの面積
A_{ARRAY}	CAMセルアレイの面積
A_{LOGIC_X}	各算術論理演算ユニットの面積
R	レスポンスレジスタの個数
A_{L_SELX}	各算術論理演算ユニットセクタの面積
A_{RR}	レスポンスレジスタ1個の面積
P	プライオリティエンコードの個数
A_{PE}	プライオリティエンコード1個の面積
A_{WS_SEL}	ワードセクタ線セクタの面積
A_{ADR}	アドレッシングユニットの面積

プロセッサを構成する各ユニットの面積見積もり式は、ほとんどがCAMセルアレイのワード数 N の1次関数の式で表すことができるが、算術論理演算ユニットの面積 A_{LOGIC_X} の見積もり式のみ N の1次関数の式で表すことができない。 A_{LOGIC_X} の見積もり式を N の1次関数の式で表すことができれば、 A_{CAM} の式の右辺は全て N の1次関数の式となる。よって A_{CAM} の式は N の1元1次方程式と考えることができる。RAMへ置き換えるワード数を W_x として、上記の式の A_{CAM} に面積制約値を与え、 W_x の方程式として解けば、最適な W_x を求めることができる。

3.3.1 算術論理演算ユニットの面積

算術論理演算ユニットは、MATCH信号とレスポンスレジスタユニットの内容を入力として(いずれもビット幅 N) 論理積などの論理演算をビット毎に実行し、出力するユニットである。例として、 $N = 500$ の時、全ての入力ビットに対して and を実行するユニットの面積見積もり式を以下に示す[9]。

$$A_{LOGIC_X}(W_x) = 1.32W_x + 10.5 \quad (0 \leq W_x \leq 500)$$

全ての入力ビットに対して同一の論理演算を実行するユニットの面積見積もり式はRAMへ置き換えるワード数の1次関数の式で表すことができる。

算術論理演算ユニットは、入力ビットに対して異なる演算を実行する場合がある。1番目から300番目までの入力ビットと、301番目から500番目までの入力ビットで異なる演算を実行する時、ユニットの面積見積もり式は以下のように表せる。

$$A_{LOGIC_X}(W_x) = \begin{cases} f(W_x) & (0 \leq W_x \leq 300) \\ g(W_x - 300) + f(300) & (301 \leq W_x \leq 500) \end{cases}$$

この場合はRAMへ置き換えるワード数の1次関数の式で表すことが出来ない。この式をグラフにしたものを図5に示す。算術論理演算ユニットの面積 A_{LOGIC_X} の見積もり式をRAMへ置き換えるワード数の1次関数の式で表すた

表 2 計算機実験結果

面積制約値 [gate count]	CAM セルアレイ	CAMプロセッサの面積 [gate count]	実行時間 [ms]	CAM [ワード数]	RAM [ワード数]
制約なし	CAM7	573075	0.31	1000	0
570000	CAM4	560509	0.35	1000	0
560000	CAM5	559997	7.60	995	5
550000	CAM1	479773	8.91	1000	0
470000	CAM7	469777	210.14	820	180
300000	CAM7	299886	550.73	479	521
200000	CAM7	199735	752.08	295	705
100000	CAM7	99583	952.78	97	903

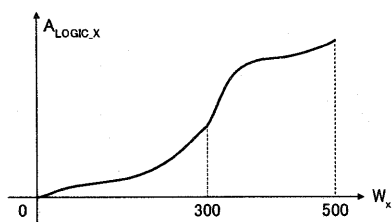


図 5 A_{LOGIC_X} と W_x の関係

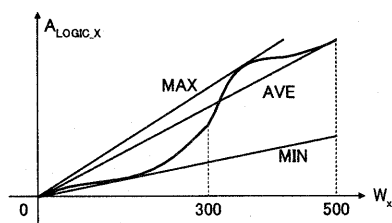


図 6 3通りのAの表現

めに、図5のユニットを図6のように考える。 A_{LOGIC_X} を3通りの1次関数で表す。原点から上方向と下方向へのそれぞれの接線をMAX, MIN, A_{LOGIC_X} を入力ビット幅で割ったものをAVEとする。

A_{LOGIC_X} の見積もり式を3通りの1次関数の式で表すことによって、CAMプロセッサ全体の面積 A_{CAM} の見積もり式を3通りの1次関数の式で表せる。ただし、3通りの A_{CAM} の見積もり式は近似値であるので正確でない。面積制約値から、RAMへ置き換えるワード数 W_x を正確に求めるために A_{CAM} の式を正確に計算する補正処理が必要となる。

3.3.2 補正処理

入力となるアプリケーションを実行するために必要な全ての算術論理演算ユニットが、全ての入力ビットに対して同一の論理演算を実行する時、すなわち面積見積もり式が W_x の1次関数で表せる時、前述した方法に従うと、MAX, AVE, MINは全て同一になる。言い換えると、 A_{CAM} の見積もり式は1通りで表せる。この場合の補正処理の内容は以下である。

1. W_x の1次関数に見立てた A_{CAM} の見積もり式と面積制約値によって W_x を算出する。
 2. 算出した W_x と A_{CAM} の見積もり式によって A_{CAM} を再計算する。
- 3(a).再計算した A_{CAM} が面積制約値より大きければ、 W_x を1減らして A_{CAM} を再計算する。これを繰り返して

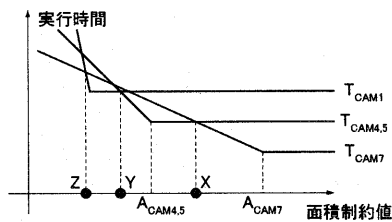


図 7 面積制約値と T_{CAMn} の関係

A_{CAM} が面積制約値より小さくなれば終了。

- 3(b).再計算した A_{CAM} が面積制約値より小さければ、 W_x を1増やして A_{CAM} を再計算する。これを繰り返して A_{CAM} が面積制約値より大きくなれば終了。

算術論理演算ユニットの面積見積もり式が W_x の1次関数で表せない時、MAX, AVE, MINはそれぞれ違う値をとる。それぞれの場合で求めた置き換えワード数を W_{MAX} , W_{AVE} , W_{MIN} とする。 A_{CAM} は W_x に対して単調増加であり、最適なCAMワード制限値 W_x は W_{MAX} と W_{MIN} の間にある。よって、区間 $[W_{MAX}, W_{MIN}]$, W_{AVE} を初期値として二分探索法を用いて W_x を求める。

3.4 提案手法の利点

提案手法による面積制約は、CAMセルアレイの一部をRAMへ置き換える手法とCAMセルアレイの種類を選択する手法の組み合わせである。

CAMセルアレイの種類を選択する手法のみの場合、面積制約値を超えたCAMセルアレイは分割候補でなくなるが、CAMワード数を制限することで面積制約値に関わらず全てのCAMセルアレイを分割候補にできる。この2つの手法を組み合わせる事でより多くの分割候補を考慮に入れることが可能となる。

4. 計算機実験結果

入力するアプリケーションとして1000個のデータをCAMセルアレイに格納し、一致検索、greater than 検索、less than 検索を500回ずつ繰り返すアプリケーションを使用した。計算機実験の結果を表2に示す。

CAMnにおけるアプリケーションの実行時間を T_{CAMn} とする。面積制約値が十分に大きい時、CAMセルアレイのワードをRAMに置き換える必要がないため、CAMプロセッサの面積は面積制約をしない場合と同様になる。面積制約をしない場合のCAMプロセッサの面積を A_{CAMn} とすると、面積制約値が A_{CAMn} 以下の時にRAMへの置き換えが起こるので、アプリケーションの実行時間は増加する。面積制約値と T_{CAMn} の関係を図7に示す。面積制約値を

m とした時、 $m \leq X$ ではCAM7のセルアレイを使用したCAMプロセッサがシステムの出力となる。 $Y \leq m \leq X$ の時はCAM4(またはCAM5)のセルアレイを使用したCAMプロセッサが、 $Z \leq m \leq Y$ の時はCAM1のセルアレイを使用したCAMプロセッサが出力となる。 $m \geq Z$ の時はCAM7の場合がシステムの出力となる。

5. むすび

本稿では、CAMプロセッサを対象としたハードウェア/ソフトウェア協調合成システムに面積制約機能を付加した。計算機実験によって、面積制約を満たす範囲でアプリケーションを実行時間最速で実行するCAMプロセッサのハードウェア構成が得られる事を確認した。

今後の課題として、ワード数を削減した時の実行時間の増加量を減少させる事が考えられる。

文 献

- [1] S. Azgomi, "Using content-addressable memory for networking applications," *Communication System Design*, vol.5, no.11, 1999.
- [2] A. Belenkiy and N. Uzun, "Deterministic IP table lookup at wire speed," in *Proc. INET'99*, 1999.
- [3] T. Ikenaga and T. Ogura, "A fully-parallel 1Mb CAM LSI for realtime pixel-parallel image processing," in *Proc. IEEE International Solid-State Circuits Conference*, 1999.
- [4] T. Ogura and M. Nakanishi, "CAM-based highly-parallel image processing hardware," *IEICE Trans. Electron.*, vol.E80-C, no.7, pp.868-874, 1997.
- [5] N. Togawa, T. Wakui, T. Yoden, M. Terajima, M. Yanagisawa, and T. Ohtsuki, "CAM processor synthesis based on behavioral descriptions," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E83-A, no.12, pp.2464-2473, 2000.
- [6] 戸塚崇夫, 石川裕一郎, 戸川望, 柳澤政生, 大附辰夫, "閾値検索機能を持つCAMプロセッサの自動合成システム," 信学技報, VLD2002-113, ICD2002-157, DC2002-65, 2002.
- [7] 戸塚崇夫, 石川裕一郎, 戸川望, 柳澤政生, 大附辰夫, "閾値検索機能付きCAMプロセッサの最適化手法," 信学技報, VLD2002-158, ICD2002-223, 2003.
- [8] 涌井達彦, 戸川望, 柳澤政生, 大附辰夫, "CAMプロセッサを対象とするハードウェア/ソフトウェア協調合成システム," 信学技報, VLD2000-141, pp.89-94, 2000.
- [9] 余傳達彦, 戸川望, 柳澤政生, 大附辰夫, "機能メモリを使用したプロセッサの面積/遅延見積もり手法," 信学技報, VLD2000-83, 2000.
- [10] M. Zhang and K. Asanović, "Fine-grain CAM-tag cache resizing using miss tags," in *Proc. the 2002 international symposium on Low power electronics and design*, pp.130-135, 2002.