設 計 自 動 化 89−12
(1998. 9. 22)

機能メモリを使用した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

寺島 信　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
〒169-8555 東京都新宿区大久保 3-4-1
E-mail: terajima@ohtsuki.comm.waseda.ac.jp

あらまし

本稿では，CAM(一致検索機能を有する機能メモリ) を使用した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を提案する．本システムではC言語で記述されたCAM機能を使用したアプリケーションプログラムを入力とし，CAMとマイクロプロセッサユニットで構成されるCAMプロセッサの論理合成可能なハードウェア記述およびCAMプロセッサ上で動作するバイナリコードを出力する．このシステムによりCAMの並列処理機能を使用したアプリケーションプログラムを高速に実行するハードウェアならびにソフトウェアを短期間で設計可能となる．生成されたCAMプロセッサのハードウェア記述を論理合成し，計算機上で性能を評価した結果を報告する．

キーワード　プロセッサ，CAM，並列処理，ハードウェア/ソフトウェア協調設計

A Hardware/Software Cosynthesis System for Processors with Content Addressable Memory

Makoto TERAJIMA　Nozomu TOGAWA　Masao YANAGISAWA　Tatsuo OHTSUKI

Dept. of Electronics, Information and Communication Engineering
Waseda University
3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan
E-mail: terajima@ohtsuki.comm.waseda.ac.jp

Abstract

This paper proposes a hardware/software cosynthesis system which synthesizes processors with a CAM (Content Addressable Memory) unit. The input of the system is an application program written in C including CAM functions, and its output is hardware descriptions of a synthesized processor and application an application binary code executed on the processor. CAM realizes word-parallel equivalence search and word-parallel writing within one clock cycle. Since the system synthesizes an application specific CAM unit according to the requirements of an input application program, it can execute the application program fast with small amount of processor area. Experimental results demonstrate its efficiency and effectiveness.

Key Words　*processor, CAM, parallel processing, hardware/software codesign*

1 まえがき

機能メモリの一種である CAM (Content Addressable Memory: 連想メモリ) は，通常の RAM が持つアドレスによるデータの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する一致検索機能を持つ．一致検索機能のみでは CAM の用途は辞書的なデータ検索に限られるが，CAM の周辺に簡単なデータ処理回路を付加することで，CAM の各ワードをプロセッシングエレメントとした SIMD (Single Instruction stream, Multiple Data stream) 型プロセッサとして利用できる．現在まで CAM は記憶容量の大容量化と SIMD 型プロセッサ指向の高機能化の方向へ開発が進められてきた．大容量化に関する研究として逐次検索型 CAM [1],[2],[3],[4]，CAM と RAM の複合アーキテクチャ[5] 等が挙げられる．高機能化に関する研究として NTT で開発された 4Kbit，20Kbit CAM [6],[7]，VLSI のレイアウト設計で使用される図形処理を対象とした図形処理用連想メモリチップ [8] 等が挙げられる．しかし，これらの高機能 CAM は一般的なアプリケーションを対象に設計されており CAM を高性能な SIMD 型プロセッサとして使用する際に，アプリケーションが必要とする機能が CAM に存在しない，またはアプリケーションが必要しない機能を CAM が有する等の問題が発生し，プロセッサとしての有効性の低下をもたらす．この問題はアプリケーション毎に必要十分な機能を持つ CAM を設計することにより解決できるが，VLSI 設計コストの問題もあり現在までこのような手法は存在しない．しかし近年の VLSI 設計自動化技術の進歩により VLSI 設計コストが比較的低くなり，アプリケーション毎にそのアプリケーションに最適な CAM アーキテクチャを設計することも不可能ではなくなりつつある．

以上のような背景を踏まえ，本稿では機能メモリを使用した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を提案する．本システムでは C 言語で記述された CAM 機能を使用したアプリケーションプログラムを入力とし，アプリケーションプログラムを実行するプロセッサのハードウェア記述およびプロセッサ上で動作するバイナリコードを出力する．このシステムにより CAM の並列処理機能を使用したアプリケーションを高速に実行するハードウェアおよびソフトウェアを短期間で設計可能となる．

本稿では，まずターゲットアーキテクチャである CAM プロセッサのアーキテクチャを定義し，次に機能メモリを使用した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を提案する．最後に CAM プロセッサを論理合成し提案手法の有効性を評価する．

2 CAM プロセッサ

システムのターゲットアーキテクチャである CAM プロセッサは，既存の一般的なマイクロプロセッサアーキテクチャであるハーバードアーキテクチャ[9] に CAM ユニットを付加した構成をとる．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 およびデータ RAM の 4 要素で構成される (図 1)．マイクロプロセッサユニットはアプリケーションプログラムの実行と CAM ユニットの制御を担当する．CAM ユニットはマイクロプロセッサユニットの補助プロセッサとして動作し，SIMD 型並列処理を実行する．命令 RAM，データ RAM は既存の SRAM を接続して使用すると仮定する．即ちマイクロプロセッサユニットと CAM ユニットを合成対象とする．

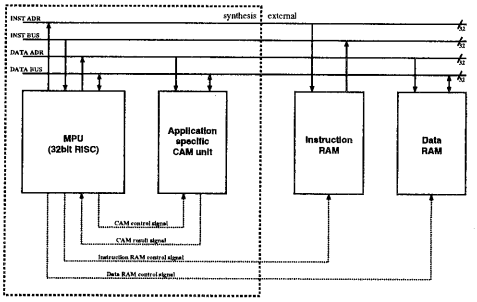

図 1: CAM プロセッサ

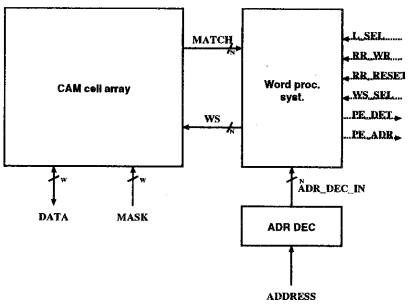

図 2: CAM ユニット

以下，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について説明する．

2.1 CAM ユニット

CAM ユニットはデータの内容を保持し検索を実行する連想メモリセルアレー，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ー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 (図 2)．

以下，構成要素である連想メモリセルアレーおよびワード処理系について説明する．

2.1.1 連想メモリセルアレー

連想メモリセルアレーの構造を図 3 に示す．1 ワードは W 個の 1bit 連想メモリセルで構成され，N 個のワードにより連想メモリセルアレーが構成される．連想メモリセルアレーは通常の RAM が持つワード単位のデータの読みだ

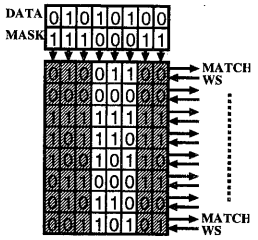

図 3: 連想メモリセルアレー

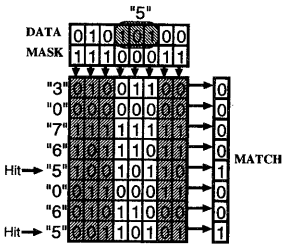

図 4: 一致検索

し，書き込み機能に加え，一致検索，並列書き込み機能を持つ．一致検索は，各ワードに記憶しているデータと外部より与えられた検索データDATAを比較し，一致したデータを記憶しているワードの一致信号線MATCHに'1'を出力する．その際，マスクデータMASKが'1'であるビットは一致判定の対象から除く (図4)．並列書き込みは，複数のワードセレクト線WSを同時に駆動することによって，複数のワードに対しマスクデータMASKで選択されたビットに入力データDATAを並列に書き込む (図5)．CAMプロセッサは連想メモリセルアレーに完全並列検索型を採用する．完全並列検索型連想メモリセルアレーは一致検索と並列書き込みをワード数に依らず1クロックで実行することができる．連想メモリセルアレーを構成する連想メモリセルを図6に示す．図6のDフリップフロップに1bitの情報を保持する．アプリケーションの要求を必要十分に満たすため，連想メモリセルアレーのワードサイズWおよびワード数Nは自由に設定可能とする．

2.1.2 ワード処理系

ワード処理系の構成を図7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1つのワードセレクト線セレクタ (WS_SEL) で構成される．以下，それぞれの構成要素について説明する．

算術論理演算ユニット

算術論理演算ユニット (図8) はワード処理系内の算術論理演算を実行する．算術論理演算ユニットは任

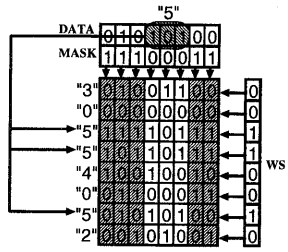

図 5: 並列書き込み

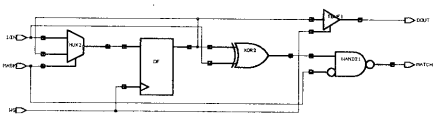

図 6: 連想メモリセル (1bit)

意のユニット数使用可能とする．ユニット数をLとする．算術論理演算ユニットは入力として連想メモリセルアレーの出力である一致信号線MATCH (Nビット)，全てのレスポンスレジスタの出力RR_DATA_X (各Nビット)，全てのプライオリティエンコーダの出力PR_DATA_X (各Nビット) を任意に使用できる．これらの入力を用い，システムによって定められた組み合わせ論理演算を実行して，Nビットの演算結果LOUTを出力する．例えばこのユニットでは検索結果とレスポンスレジスタの値の論理積，論理和演算，加減算演算，レスポンスレジスタの記憶内容のシフトなどの機能を実現する．

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 (図9) は算術論理演算ユニットの演算結果 (Nビット) を格納する．レスポンスレジスタユニットは任意のユニット数使用可能とする．ユニット数をRとする．レスポンスレジスタユニットは算術論理演算ユニットセレクタ (L_SEL_X)，レスポンスレジスタ，プライオリティエンコーダ (PE_X) の3要素で構成される．レスポンスレジスタは，入力として任意の算術論理演算ユニットの出力を使用できる．使用する算術論理演算ユニットを選択するため，各レスポンスレジスタの入力にセレクタを使用する．信号の選択は信号線L_SEL_Xを用いる．レスポンスレ

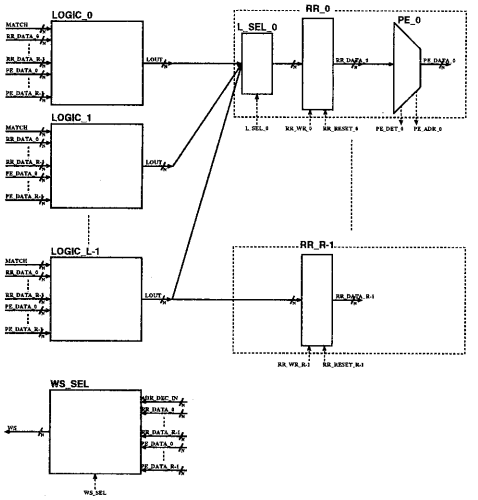

図 7: ワード処理系アーキテクチャ

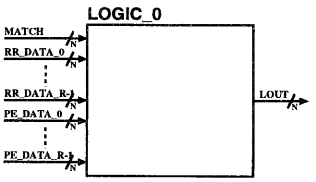

図 8: 算術論理演算ユニット

ジスタへの値の書き込みに信号線RR_WR_X，記憶内容のリセットに信号線RR_RESET_X を用いる．レスポンスレジスタの出力 (N ビット) の中から 1 信号のみを選択したい場合，プライオリティエンコーダを用いる．プライオリティエンコーダはレスポンスレジスタの出力を入力とし，アドレス値の一番小さい 1 信号を選択する．入力信号の有無PE_DET_X (1 ビット)，選択後の信号PE_DATA_X (N ビット)，アドレス値PE_ADR_X を出力する．PE_DATA_X は選択された 1 信号のみ '1' を出力し，他の N-1 本の信号線は '0' を出力する．プライオリティエンコーダはレスポンスレジスタユニット毎に使用/不使用を選択可能とする．

ワードセレクト線セレクタ

ワードセレクト線セレクタ (図 10) はセルアレーのワードセレクト線を駆動する信号を選択する．ワードセレクト線セレクタは，入力としてアドレスデコーダの出力，各レスポンスレジスタ出力，各プライオリティエンコーダ出力を任意に使用できる．これらの入力のうち 1 つを選択，ワードセレクト線と接続する．

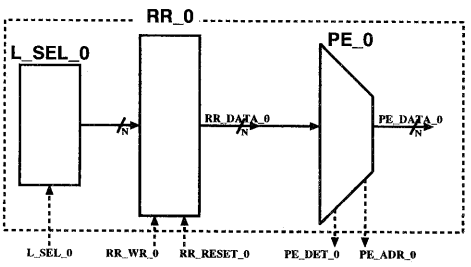

図 9: レスポンスレジスタユニット

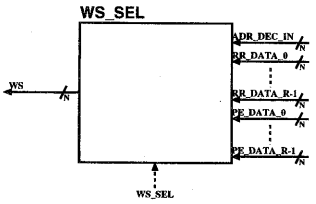

図 10: ワードセレクト線セレクタ

2.2 マイクロプロセッサユニット

マイクロプロセッサユニットは 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．本システムでは 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．マイクロプロセッサユニットの特徴を以下に列挙する．

- 32bit 固定長命令
- 5 ステージパイプライン，フォワーディング機構
- ハーバード アーキテクチャ
- 32bit，32 本，1 入力 2 出力レジスタファイル
- 32bit ALU，32bit バレルシフタ，$32 \times 32 = 64$bit 乗算器
- 基本 31 命令 + CAM 制御 5 命令

マイクロプロセッサユニットと CAM ユニットまたはデータ RAM 間のデータインターフェイスは共通のアドレス/データバスを使用する．マイクロプロセッサユニットのデータアドレス空間を図 11に示す．データアドレス空間の前半部分を RAM 空間としデータ RAM を配置する．後半部分を CAM 空間とし CAM ユニットを配置する．また CAM 空間には CAM ユニットのプライオリティエンコーダの出力 (PE_ADR_X) も配置する．このアドレスにより MPU から CAM ユニット，データ RAM 内のデータ，プライオリティエンコーダの出力にアクセスできる．

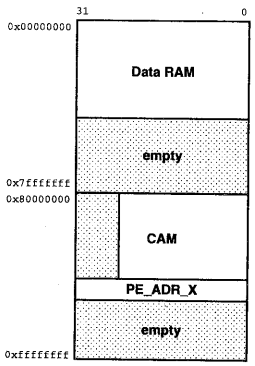

図 11: データアドレス空間

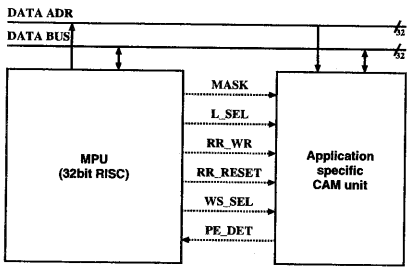

図 12: MPU CAM ユニット インターフェイス

マイクロプロセッサユニットの命令セットは基本命令セットと CAM ユニットを制御する CAM 命令セット (表 1) で構成される．基本命令セットはシステムの入力である C 言語で記述されたアプリケーションを実行するのに十分な命令を備える．基本命令セットは文献 [9] の命令セットを参考にした．CAM 命令セットは CAM ユニットの動作に対応する制御命令で構成される．マイクロプロセッサユニット，CAM ユニット間の信号線を図 12に示す．信号線L_SEL は CAM ユニットの信号線L_SEL_X を束ねた信号線である．同様に信号線RR_WR，RR_RESET，PE_DET は信号線RR_WR_X，RR_RESET_X，PE_DET_X を束ねた信号線である．

以下に CAM ユニットの動作を列挙する．

1. **検索→レスポンスレジスタ書き込み**

   DATA BUS (検索データ)，MASK (マスクデータ)，
   L_SEL (論理演算選択) に値をセットし，
   RR_WR (レスポンスレジスタ書き込み) を駆動することで検索結果をレスポンスレジスタに書き込む．

2. **レスポンスレジスタ→セルアレー書き込み**

   DATA BUS (検索データ)，MASK (マスクデータ) に値をセットし，WS_SEL (ワード書き込み線選択) を駆動することでセルアレーにレスポンスレジスタ値を書き込む．

3. **一致信号検出線の値による分岐**

   プライオリティエンコーダによって検出する一致信号 (PE_DET) の有無により分岐を制御する．

4. **レスポンスレジスタのリセット**

   RR_RESET (レスポンスレジスタリセット) を駆動し，レスポンスレジスタに保持している値を全てクリアする．

次にこれらの動作に対応する CAM 命令を列挙する．

1. **RRW (Response Register Write) 命令**

   DATA BUS にレジスタrs の値，MASK にレジスタrt の値を出力し，信号線の集合 {L_SEL,RR_WR} を値imm で駆動する．

2. **CAW (Cell Array Write) 命令**

   DATA BUS にレジスタrs の値，MASK にレジスタrt の値を出力し，WS_SEL を値imm で駆動する．

3. **BPD (Branch Priority-encoder Detect) 命令，BED (Branch Equal Detect) 命令**

   PE_DET と値imm を比較し，一部でも一致した場合 (BPD)，または完全に一致した場合 (BED)，(PC+1+address) 番地に分岐する．imm の値は '0'，'1'，'X(don't care)' の 3 値論理 (2bit) で表現する．

4. **RRR (Response Register Reset) 命令**

   RR_RESET を値imm で駆動する．

3 機能メモリを使用した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

機能メモリを使用した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は CAM の機能を使用したアプリケーションプログラムを入力とし，マイクロプロセッサユニットと CAM ユニットをハードウェアで実現する論理合成可能なハードウェア記述と，ハードウェア上で実行可能なバイナリコードを出力する．本システムは CAM の持つ並列処理機能を利用しアプリケーションを高速実行できるプロセッサシステムを生成する．システムの概要を図 13に示す．

システムはハードウェア/ソフトウェア分割，ハードウェア生成，ソフトウェア生成の 3 要素で構成される．以下，個々の要素について説明する．

ハードウェア/ソフトウェア分割

本システムでは，CAM 処理に相当する以下のような関数 (CAM 関数) を用意する．

- **word_read(*address*)**
  セルアレーより *address* 番地のワードを読み出す．

表 1: CAM 命令

命令	ニモニック	動作
RRW	rrw rs, rt, imm	DATA BUS = rt; MASK = rs; {L_SEL, RR_WR} = imm
CAW	caw rs, rt, imm	DATA BUS = rt; MASK = rs; WS_SEL = imm
BPD	bpd imm, address	if (imm == PE_DET)any go to PC + 1 + address
BED	bed imm, address	if (imm == PE_DET)all go to PC + 1 + address
RRR	rrr imm	RR_RESET = imm

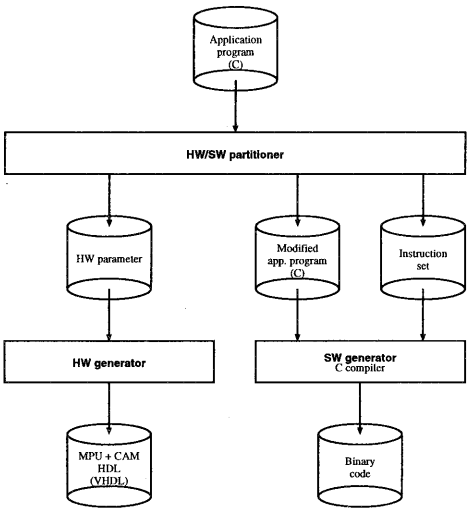

図 13: 協調合成システム

- **word_write**(*address, data, mask*)
  セルアレーの *address* 番地のワードにマスク *mask* の条件の下データ *data* を書き込む.
- **match**(*data, mask*)
  連想メモリセルアレーに対し検索データ *data*, マスク *mask* で一致検索する.
- **rr_drive**(**rr_x, logic_func*)
  一致検索の出力 (関数 match) またはユーザ定義算術論理演算関数 (関数 logic_*) の出力 *logic_func* をレスポンスレジスタユニット *rr_x* に書き込む.
- **ws_drive**(*ws_input, data, mask*)
  ws_input で選択されるワードセレクト線 (WS) を駆動し, セルアレーに対しマスク *mask* の条件の下データ *data* を書き込む.
- **rr_reset**(**rr_x*)
  レスポンスレジスタ *rr_x* の内容をクリアする.

ハードウェア/ソフトウェア分割は CAM 関数を使用したアプリケーションプログラム (図 14) を入力とし, ハードウェア構成を決定するパラメータ, CAM ユニットを用いて実行する CAM 関数を明示したアプリケーションプログラム, CAM ユニット制御命令セットを出力する.

ハードウェア/ソフトウェア分割処理は以下の手順で処理される.

Step 1 アプリケーションプログラム内で使用されている CAM 関数を, CAM ユニット (ハードウェア) で実行する関数とマイクロプロセッサユニット (ソフトウェア) で実行する関数に分ける.

Step 2 ハードウェア化する CAM 関数より CAM ユニットの構成を決定する.

Step 1 は, 現在のシステムではアプリケーションプログラム内の全ての CAM 関数をハードウェア化することでハードウェアとソフトウェアを分割する. 今後システムの拡張に伴いハードウェアの面積/時間制約に基づくハードウェア/ソフトウェア間の交換を考慮する.

Step 2 は次のように実現される. CAM ユニットの構成は, (1) 連想メモリセルアレー, (2) 算術論理演算ユニット, (3) レスポンスレジスタユニット, (4) ワードセレクト線セレクタから構成される. これらの構成を決定すればよい.

(1) のワードサイズ W, ワード数 N はアプリケーションプログラム内の外部変数定義により決定される. 図 14では, 4-5 行目でワードサイズ W を 8, ワード数 N を 8 と決定している.

(2) ではユニット数 L, 各ユニットの入力信号線および演算内容を決定する. ユニットの入力信号線は関数 **rr_drive** の引数である関数 *logic_func* の引数より決定する. 引数は, 一致検索の出力に相当する関数 **match**, レスポンスレジスタの出力に相当する *rr_x.rr*, プライオリティエンコーダの出力に相当する *rr_x.pe* より任意に使用できる. 算術論理演算ユニットの論理演算はユーザ定義算術論理演算関数 (関数 logic_*) より決定する. 図 14では, 32-42 行目で論理和演算を定義している. 入力信号線と論理演算の決定により 1 ユニットが作成できる. 図 14では, 18-19 行目より **match** と *rr_0.rr* を入力信号線とし論理和を演算する算術論理演算ユニットが定義できる. 複数のユニットを作成する過程で, 既に作成したユニットと同一の入力信号線および論理演算をもつユニットは 1 ユニットで代表する. 関数 *logic_func* が一致検索 **match** の場合は, 入力信号線が **match** のみで, 入力をそのまま出力する算術論理演算ユニットを作成する (図 14: 17 行目).

(3) ではユニット数 R，各ユニットの入力信号線およびプライオリティエンコーダの使用/不使用を決定する．ユニットに対応する変数の宣言によりユニット数 R，プライオリティエンコーダの使用/不使用を決定する．図 14では，9 行目でプライオリティエンコーダを使用するレスポンスレジスタユニット **rr_0** を 1 つ作成している．入力信号線は関数 **rr_drive** の引数 *rr_x* と算術論理演算関数 *logic_func* の対応より決定する．図 14では，17–19 行目でレスポンスレジスタユニット **rr_0** に一致検索 **match** と算術論理演算ユニット **logic_or** の出力が入力されている．

(4) ではユニットの入力信号線を決定する．入力信号線は関数 **ws_drive** の引数 *ws_input* で使用される信号線により決定する．図 14では 22 行目でレスポンスレジスタユニットのレスポンスレジスタ出力 **rr_0.rr** を入力信号線としている．

ハードウェア生成

ハードウェア生成はハードウェア構成を決定するパラメータを入力とし，プロセッサを実現する論理合成可能なハードウェア記述を出力する．ターゲットアーキテクチャは CAM プロセッサとする．

ソフトウェア生成

ソフトウェア生成はハードウェア化した CAM 処理を明示したアプリケーションプログラム，CAM 命令セットのフォーマット情報を入力とし，ハードウェア生成で出力するプロセッサシステム上で実行可能なアプリケーションコードを出力する．CAM ユニット制御命令が可変であることを除き，従来の C コンパイラと同様の処理を行う．

```
 1: #include <stdio.h>
 2: #include "cam_lib.h"
 3:
 4: const int W = 8;  /* word width */
 5: const int N = 8;  /* num of word */
 6:
 7: int main(void)
 8: {
 9:   struct RR_X rr_0 = {USE_PE};
10:   int n;
11:
12:   /* cell array write */
13:   for (n = 0; n < N; n++)
14:     word_write(n, n, 0);
15:
16:   /* rr_0 drive */
17:   rr_drive(&rr_0, match(0xff, 0xfe));
18:   rr_drive(&rr_0, logic_or(
19:     match(0xff, 0xfd), rr_0.rr);
20:
21:   /* ws drive */
22:   ws_drive(rr_0.rr, 0xff, 0x8f);
23:
24:   /* cell array read */
25:   for (n = 0; n < N; n++)
26:     printf("word_read(%d): %d\n",
27:       n, word_read(n));
28:
29:   return 0;
30: }
31:
32: struct WS_BUS logic_or(struct WS_BUS match,
33:   struct WS_BUS rr)
34: {
35:   struct WS_BUS lout;
36:   int i;
37:
38:   for (i = 0; i < N; i++)
39:     lout.data[i] = match.data[i] | rr.data[i];
40:
41:   return lout;
42: }
```

図 14: アプリケーション例

4 論理合成結果

提案したシステムによって生成される CAM プロセッサの VHDL 記述を論理合成し性能を評価する．まず，CAM プロセッサの一部である CAM ユニットを既存の CAM と比較する．続いて，3 つのアプリケーションプログラムに対し CAM プロセッサを生成しそれぞれ面積およびクロック周期を比較する．論理合成は Synopsys 社の Design Compiler を用いた．テクノロジライブラリとして VDEC で設計された exd_tbl を使用した．設計ルールは $0.5\mu m$ である．論理合成結果の面積は 2 入力 NAND ゲートを 1 とした値である．

CAM ユニットと既存の CAM LSI である図形処理用連想メモリチップ [8] を比較するため，CAM ユニットのワード処理系機能を表 2 に示す仕様とした．セルアレーのサイズを図形処理用連想メモリチップと同じ 36 ビット × 128 ワードとした．論理合成によって得られた CAM ユニットの性能と図形処理用連想メモリチップの性能を表 3に示す．表 3より CAM ユニットは既存の CAM と同程度の性能を持っているといえる．

3 つの CAM を使用したアプリケーションに対しシステムを適用して得られた CAM プロセッサの論理合成結果を表 4に示す．各アプリケーションの内容および連想メモリセルアレーサイズ (ワードサイズ × ワード数) を以下に示す．

Sort1: バブルソート (32 × 256)
Sort2: バブルソート (32 × 128)
Sim : 並列故障シミュレーション [10] (32 × 128)

結果より，各プロセッサのハードウェア性能差を確認でき，アプリケーションプログラムの実行に必要十分な CAM プロセッサが生成可能であることが示された．

5 むすび

本稿では，機能メモリを使用した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を提案した．システムにより CAM の並列処理機能を使用したアプリケーションに特化したハードウェアおよびソフトウェアが短期間で設計可能となった．

謝辞

本研究に関し御協力いただいた本学修士課程 1 年伊澤義貴氏に感謝致します．本研究の一部は，文部省科学研究費

表 2: CAM ユニットワード処理系仕様例

RR	LOGIC	PE
RR_0 (Main)	Match, And, Or, Not, Tag, Shift up, Shift down, Add(Sum)	○
RR_1 (Sub)	Match, And, Or, Not, Tag, Shift up, Shift down, Add(Sum)	×
RR_2 (Carry)	Add(Carry)	×
RR_3 (Tag)	Match	×

表 3: CAM ユニット論理合成結果

	Ours	[8]
CAM ユニット総面積	66355	79793
セルアレー面積	58069	63206
ワード処理系面積	8118	15853
アドレスデコーダ面積	168	734
CAM ユニット最大遅延 [ns]	16.3	50.0†

†設計ルール $0.8\mu m$

補助金（基盤研究 C(2)，課題番号 10650345 および奨励研究 (A)，課題番号 10750303）の援助を受けた．

参考文献

[1] I. N. Robinson, "Pattern-addressable memory," *IEEE Micro*, pp. 20–30, Dec. 1996.

[2] G. J. Lipovski, "A four megabit dynamic systolic associative memory chip," *J. of VLSI Signal Process.* 4 , pp. 37–51, 1992.

[3] S. Panchanathan and M. Goldberg, "Vector-centerd CAM architecture for image coding using vector quantization," in *Proc. SPIE 1199, Visual Communications and Image Processing IV*, pp. 1084–1094. 1989.

[4] K. J. Schultz and P. G. Gulak, "Architectures for large-capacity CAMs," *Integration*, pp. 151–171, 1995.

[5] K. J. Schultz and P. G. Gulak, "Fully parallel integrated CAM/RAM using preclassification to enable large capacities," *IEEE J. of Solid-State Circuits*, vol. 31, no. 5, pp. 689–699, May 1996.

[6] T. Ogura, S. Yamada and T. Nikaido, "A 4Kbit Associative Memory LSI," *IEEE J. of Solid-State Circuits*, vol. SC-20, no. 6, pp. 1277–1282, 1985.

[7] T. Ogura, S. Yamada and J. Yamada, "A 20Kb CMOS Associative Memory LSI for Artificial Intelligence Machines," in *Proc. of ICCD*, pp. 574–577, 1986.

表 4: CAM プロセッサ論理合成結果

	Sort1	Sort2	Sim
CAM プロセッサ総面積	134405	80932	82461
MPU 面積	27519	27519	27578
CAM ユニット面積	106886	53413	54883
CAM プロセッサ最大遅延 [ns]	19.1	11.3	11.3

[8] 桑原泰雄, 安部正秀, 久保田和人, 佐藤政生, 大附辰夫, "図形処理用連想メモリチップ," 信学技報, SDM92-55, ICD92-54, pp. 9–16, Sept. 1992.

[9] D. A. Patterson and J. L.Hennessy, 成田光彰 訳, コンピュータの構成と設計～ハードウエアとソフトウエアのインターフェイス，日経 BP 社，1996.

[10] N. Ishiura and S. Yajima, "Linear time fault simulation algorithm using a content addressable memory," *IEICE Trans. Fundamentals*, vol .E75-A, no. 3, pp. 41–48, Mar. 1992.