アドレスルックアップ仮想ページング方式による ファジィ推論チップ†

有 川 晴 彦*・廣 田 薫**

Fuzzy Inference Chip by Address-look-up and Virtual Paging Method

Haruhiko ARIKAWA* and Kaoru HIROTA**

A new type of fuzzy inference chip adopting address-look-up and virtual paging method, i. e. sophisticated use of common digital memory devices, has been successfully developed. The characteristics of the inference chip are summarized first and then its theoretical background is introduced.

This inference chip possessing robust noise immunity executes any kind of fuzzy operations and variables using a limitless number of fuzzy production rules with unlimited fuzzy linguistic labels and free-shaped membership functions in continuous valued mode at high-speed (more than 60 MFLIPS) and low-cost. It can be easily linked to the commonly used digital computer system and furthermore helps to pave the way for harmonizing fuzzy system with crisp system to construct the next generation computer system.

Key Words: digital memory, address-look-up, virtual paging, access time, number of rules

1. ま え が き

ファジィ理論応用を目的としたハードウェアの集積回路については、いくつかのものがすでに提案され、または開発されている^{1)~3)}。山川¹⁾の開発したファジィチップは、アナログ回路の連続値モードによるファジィ演算を可能とし、戸貝²⁾や渡辺³⁾による離散値モードで演算を実行する専用 LSI は、ディジタル型ファジィチップの集積化の道を拓いた。

それらはファジィ推論を汎用的に処理することを目的としたものであるが、ハードウェアの制約上ファジィ推論の合成規則演算方式を max/min 法などに、また逆ファジィ化の手法を重心法などに限定したりせねばならず、さらにたとえばルール数の増加に対してもハードウェアの構成を追加するか、処理速度の低下を招かざるをえなかった。

本研究ではそれら諸問題の解決を目的として、演算処理上の制約をより少なくし、かつより小規模の電子回路でより高速により安価な推論チップの実現を目指した。その結果、本論文で提唱するアドレスルックアップ仮想ページング方式に基づいて汎用のディジタル記憶素子を使用すれば、その目標を達成できることが判明した。

その有効性を確認するために試作機を作成したところ、前述の制約例などから解放されるとともに、ファジィ推論結果を每秒 6 百万回得ることができる処理速度を実現し、さらに每秒 6 千万回以上のファジィ推論への見通しを与えた。

2. アドレスルックアップ仮想ページング方式によるファジィチップの特徴

ファジィ推論チップはすでにいくつかのものが報告されている^{1)~3)}。それらのチップと本提案のチップとの特性一覧表を、Table 1 に示す。

以下ではその概要を述べる。

2.1 推 論 速 度

試作機においては每秒 6 百万回のファジィ推論を実現したが、現在開発されているもので最も高速な素子を使用すれば每秒 6 千万回以上のファジィ推論を達成することができる。さらに並列分散処理を併用するとその整数倍の高速化を計ることも可能である。この高速性の理由は、基本的に記憶素子単体またはそのネットワークのみでハードウェアが構成されているので、

† ファジィシステムの応用に関する国際ワークショップで発表 (1988・8)

* アーク・テクノリサーチ(株) 京都市伏見区新町 5-498-7 (投稿時 マイコム(株))

** 法政大学工学部 小金井市梶野町 3-7-2

* Ark Techno-Research Inc., Kyoto (Mycom Inc. When Received)

** Faculty of Engineering, Hosei University, Koganei (Received November 21, 1988)
(Revised August 7, 1989)

Table 1 Fuzzy inference engine on chip

	Yamakawa ¹⁾	Togai ²⁾	Watanabe ³⁾	Presented
Technique	Analogue	Digital and software	Digital	Software and digital memory
Compositional rule of inf.	max/min	max/min (many)	max/min	Any
Defuzzification	C. G.	C. G. (many)	C. G.	Any
Max speed [MFLIPS]	0.2	0.25 (0.4)	0.625	60
Number of rules/chip	1	16 (700)	102	Limitless
Number of items for antecedent	3	2 or 4 (256)	2 or 4	Variable, 2-6 and more
Number of items for conclusion	1	2 (256)	2	Variable, 1-6 and more
Membership value [0, 1]	[0, 5] V	4, 6, 8 bits	4 bits	Floating point

Note C. G.: center of gravity method

(): to be developed

推論速度は構成記憶素子の読み取り時間（現有最高速のもので 15 ns）だけに依存するからである。

2.2 推論演算の任意性

メンバーシップ関数の形状をまったく自由に定義することができるうえ、ファジィ推論の合成規則に関する演算方式も max/min 法などに限定されずいかなる種類の演算をも実行できる。逆ファジィ化の手法も重心法などに限定されず自由に選択できる。またルールの後件部に数式モデルを採用することや、非線形処理を任意の演算部で実行させることも可能である。

2.3 ルール数・ラベル数

同一のハードウェア構成でルール数やラベル数を無制限に増加させたり、または自由に減少させることができる。しかも処理速度がそれによって影響を受けることはないので、観測値をフィードバックさせて制御を行っている場合などに、その同期タイミングを変更する必要性が生じることもない。

2.4 メンバーシップ関数値・台集合の表現法

デジタル記憶素子を使用しているのにもかかわらず、メンバーシップ関数値および後件部の台集合を離散化しないで浮動小数点数値の意味での連続値として扱える。これは3章で詳述するように、仮想ページ上で電算機などにより別途計算を行うため、この表現法を仮想連続値表現と呼称することとする。

ただし前件部の台集合および確定出力は、入出力ピンがデジタル信号であることから離散化される。

2.5 台集合と前件部後件部の項目数

本提案において推論処理上の唯一の制約をこの節で述べる。構成デジタル記憶素子の入出力情報の台集合が離散化されていることから、観測対象や確定出力の台集合も離散化される。また入(出)力ピン総数 P が有限であることから、前(後)件部の項目数 N により各

台集合の離散要素数 U_j は次式の制約を受ける。

$$U_j = 2^{P_j} \quad (j=1, 2, \sim, n) \quad (1)$$

ここに各項目への割当ピン数を P_j として、

$$P = P_1 + P_2 + \sim + P_j + \sim + P_n \quad (2)$$

$$2^P = U_1 \cdot U_2 \cdot \sim \cdot U_j \cdot \sim \cdot U_n \quad (3)$$

ただし以上の制約条件に対しても、記憶素子を複数個用いた構成にするか、補助記憶装置による仮想記憶空間を構成することで対処することができる。また階層的に観測対象を把握することも有効な手段である。

2.6 耐ノイズ性・信頼性

デジタル記憶素子のみを用いたシンプル（試作機ではただ1個）な基本構成である上に、たとえばメンバーシップ関数値を仮想連続表現とするなど、ハードウェア上では実際に演算処理を実行しないので、その過程でノイズが混入することはない。またホスト CPU や命令コードを必要としないため、クロックを誤読してシステムが暴走に至ることもない。加えてデジタルシステムが本来有する耐ノイズ性とを考慮すると、きわめて高い信頼性が実現できたものと考ええる。

2.7 外部機器との整合性

デジタル記憶素子を基本構成としていることにより、従来のデジタル型コンピュータシステムとのリンクは容易である。たとえば試作機においては、入出力の信号自体が一般的に用いられている TTL レベルのものである上、チップ選択ピンを有しているので、本提案の推論チップの出力を直接にシステムのデータバスなどに接続しても、ホストコンピュータからのチップ選択信号により、適宜フローティング状態を保持させることが可能である。

一方アナログ値による入出力への対応には、AD/DA 変換器を必要とする。

2.8 回路規模・経済性・実用性

試作機では汎用のデジタル記憶素子 (512 K タイプの EPROM) をただ 1 個のみ使用した. 28 ピン DIP に収まったこの記憶素子の前後には, 20 ピンの LS 374 をサンプル & ホールドとグリッチ除去の目的でおのおの 1 個用いて, 計 3 個の集積回路による回路規模となっている. これらはすべて汎用的なもののばかりなので, そのコストはきわめて低い.

この構成でたとえば観測対象を 256 要素離散台集合上の連続値メンバーシップ関数で表現される 2 入力として, 結論部を同じく 256 要素離散台集合上の連続値メンバーシップ関数で表現されるファジィ集合の 1 出力として得ることができる.

また観測対象を 3 入力とした場合は, たとえば 64 要素離散台集合上の連続値メンバーシップ関数の 1 入力と 32 要素離散台集合上の連続値メンバーシップ関数の 2 入力に割り当てることができる. したがってすでに実用に供されている山川のファジィコントローラ⁴⁾が 25 要素離散台集合上の連続値メンバーシップ関数の 3 入力であることなどと比較しても, 実用上十分な構成になっている.

2.9 拡張性

記憶素子を複数個組み合わせることによって, ハードウェア上での階層処理を行うことができる. また随時書換え可能な記憶素子 (特にデュアルポート RAM) を採用することによって, ハードウェアによる学習機能を付加することも可能である.

さらにファジィ推論が複数の入力に対して重み付けを行っている点に着目すれば, 随時書換え可能な記憶素子にその重みを学習させることによって, ニューラルネットワークを構成させることも困難ではない.

また記憶素子の出力をラッチ回路を通して帰還を掛けることにより, 前向き推論による組合せ回路だけでなく, 後向き推論による順序回路を構成することも可能である.

3. アドレスルックアップ仮想ページング方式

3.1 アドレスルックアップ方式の基本概念

Fig. 1 に基本的なアドレスルックアップ方式の概念を示す. この例では $2K (=256 \times 8)$ ビットの記憶素子を使用して, メンバーシップ関数値および台集合は, とともに 8 ビットの 256 離散関数値で表現している. この方式において測定器から得た観測値は直接に記憶素子のアドレスピンに入力される. その 8 ビット

Fig. 1 Concept of address-look-up method

た台集合の要素に対応させて考える. そしてその観測値に対応する 8 ビット 1 ワードのメンバーシップ関数値をあらかじめその該当番地に格納しておけば, 求めるべき関数値がデータピンを通じて出力されることになる.

$$y = \mu(u) \quad (4)$$

ここに,

y : 8 ビットの出力メンバーシップ関数値

u : 8 ビットアドレスへの入力観測値

μ : ファジィ集合のメンバーシップ関数 (以下同様)
一つのメンバーシップ関数を表現するに当たって, 以上のごとく対応する記憶素子の記憶空間を, 1 ページという単位で呼称することとする.

アドレスルックアップ方式では, 格納するメンバーシップ関数値および台集合を離散表現 (Fig. 1 の例ではおのおの 256 要素) とする制約範囲内で, このページ上にメンバーシップ関数の形状を自由に定義できるという大きな利点がある.

なおここで特に注意を払うべき点は, アドレスピンを番地検索の目的ではなく, データ入力ピンとして使用している点である.

3.2 ページング方式の基本概念

基本的なページング方式の概念を示すため, 例として Fig. 2 のように $512K (=64K \times 8)$ ビットの記憶素子を使用する. ここでは上位 8 ビットのアドレスをページ番号 (0~255) と考えて, 下位 8 ビットのアドレスを 3.1 節のごとく 256 値に離散化された台集合の要素に対応させて考える. これにより上位 8 ビットのページ番号が確定すれば, 下位 8 ビットにより 3.1 節で定義した 1 ページに相当する記憶空間を確保することができるので, あらかじめ各ページに任意のメンバ

Fig. 2 Concept of paging method

ーシップ関数を定義しておくと、下位 8 ビットの値によって、次式のごとくアドレスルックアップを適用することが可能となる。

$$y = \mu_p(u) \quad (5)$$

ここに、

y : 8 ビットの出力メンバーシップ関数値

μ_p : ページ番号 p (上位 8 ビットアドレスへの入力値 0~255) のページに定義されたメンバーシップ関数

u : 下位 8 ビットアドレスへの入力観測値

このように記憶素子の全記憶空間を分割してページに対応させ、任意のページを選択してアドレスルックアップを行う方式を、ページング方式と呼称することにする。

3.3 ページング方式の階層化その 1

Fig. 3 にページング方式を階層的に発展させた概念を示す。ここでは Fig. 3(a)のごとく、前段に 2K (=256×8) ビットの記憶素子を、後段に 512 K(=64 K×8) ビットの記憶素子を各 1 個使用する。そこで前段の記憶素子の出力ピンを後段の記憶素子の上位 8 ビットのアドレスピンに直接に導き入力する。このときピン接続の順番は、各ビットの重みの順とする。

なお前段の記憶素子の記憶空間には 3.1 節で述べたと同様に次式のメンバーシップ関数を定義し、上位ページとする。

$$x = \mu_A(a) \quad (6)$$

ここに、

x : 前段の記憶素子の 8 ビットの出力メンバーシップ関数値かつ下位ページのページ番号

a : 前段の記憶素子の 8 ビットアドレスへの入力観測値

一方後段の記憶素子の上位 8 ビットのアドレスには、前段の記憶素子の出力 x が直接に入力されている。そこで 3.2 節と同様にその記憶空間を、上位 8 ビ

(a) Hierarchical paging (1)

(b) Min operation

(c) Max operation

Fig. 3 Hierarchical paging

ットのページ番号 x (0~255) に対応して 256 ページに分割し、下位ページとする。

この構成において Fig. 3(b)のごとく 2 入力 1 出力ファジィ min 演算 (8) 式を実行させるには (9) 式を満たすように、また Fig. 3(c)のごとく 2 入力 1 出力のファジィ max 演算 (10) 式を実行させるには (11) 式を満たすように、ページ番号 x の下位ページおのの (12) 式のメンバーシップ関数を定義する。

$$y = \mu_A(a) \wedge \mu_B(b) \quad (8)$$

$$y \leq x \quad (9)$$

$$y = \mu_A(a) \vee \mu_B(b) \quad (10)$$

$$y \geq x \quad (11)$$

$$y = \mu_{xB}(b) \quad (12)$$

ここに、

y : 後段の記憶素子からの 8 ビット出力値でかつ求める演算結果

b : 後段の記憶素子の下位 8 ビットアドレスへの入力観測値

μ_{xB} : x をしきい値としてメンバーシップ関数 μ_B を修正したもの

以上の階層ページ構造を事前に準備しておけば、2 種の観測値 (a, b) が同時に入力されることで、演算結果 y が出力される。これは下位ページに対してアドレスルックアップを行う際に、 x の値をしきい値とし

てメンバーシップ関数が描かれているからである。

なお下位ページのメンバーシップ関数の修正は、上の例に制約される必要はなく、実行するファジィ演算に従って自由に行えばよい。

つまりページング方式階層化の大きな利点の一つとして、実行するファジィ演算が論理積や論理和のみでなく、代数積や代数和、限界積や限界和などをはじめとするいかなるファジィ論理演算でも実行できる点である。

一般に M 個の記憶素子を連鎖状に並べ、前段の出力を次段のアドレスにその観測値と同時に入力するように構成し、かつ記憶素子の 1 ワードを N ワードにビット分割すれば、その最終段の出力に M 入力 N 出力のファジィ演算結果を得ることができる。

3.4 ページング方式の階層化その 2

ページング方式の階層化を 3.3 節の例では 2 個の記憶素子を用いて行った。ここでは 512 K (= 64 K × 8) ビットの記憶素子 1 個のみを用いることで、任意のファジィ演算を実行できることを、2 入力 1 出力の例で説明する。

3.3 節の例と対比させて述べると、3.3 節で用いた前段の記憶素子は省略すると考え、観測値 a を Fig. 4 で示すように直接に記憶素子の上位アドレスに導く。

この構成で、ページ番号 a の下位ページには (6) 式で得る x をしきい値として修正される (12) 式のメンバーシップ関数を定義する。これにより観測値 a が確定

Fig. 4 Hierarchical paging <2>

すれば下位ページも確定するので、観測値 b によりアドレスルックアップを行えば、次式の演算結果 y を得る。

$$y = \mu_R(a, b) \quad (13)$$

ここに、

y : 出力すべき 8 ビットの演算結果

a : 上位 8 ビットアドレスへの入力観測値

b : 下位 8 ビットアドレスへの入力観測値

ただしこの例において、実際の記憶空間に格納されるのは、2 層の階層ページ構造をもつ複合アドレス空間により規定された下位ページである。

また一般に、1 個の記憶素子の入力アドレスピンを M 個の複合アドレス空間に、かつ出力データピンを N ワードに分割すれば、 M 入力 N 出力の任意のファジィ演算を行うことができる。

3.5 仮想ページング方式その 1

ページング方式の階層化に加えて、仮想の階層ページにページングを行うことで、 k 個のファジィプロダクションルールの max/min 合成演算が行えることを示す。ここでは Fig. 5 のように、512 K (64 K × 8) ビットの記憶素子 1 個のみを用いて、前件部の項目数を 2、後件部の項目数を 1 の、2 入力 1 出力として説明する。

ルール 1: IF $A = A_1(a)$ & $B = B_1(b)$

THEN $C_1 = \min \{A_1(a), B_1(b)\}$

Fig. 5 Concept of virtual paging <1>

ルール 2: IF $A=A_2(a) \ \& \ B=B_2(b)$
 THEN $C_2=\min \{A_2(a), B_2(b)\}$
 $\vdots$
 ルール k : IF $A=A_k(a) \ \& \ B=B_k(b)$
 THEN $C_k=\min \{A_k(a), B_k(b)\}$
 $\rightarrow c=\max \{C_1, C_2, \dots, C_k\}$ (14)

ここに、

a : 上位 8 ビットアドレスへの入力観測値

b : 下位 8 ビットアドレスへの入力観測値

c : 出力すべき 8 ビットの合成演算結果

3.4 節に従えば各ルールに各 1 個の記憶素子が必要とするので、 k 個のルールの演算には k 個の記憶素子が必要とする。しかしながら 1 個の記憶素子において (14) 式の $\max/\min$ 合成演算を行う場合、観測値 a を上位ページとした下位ページは、 $1 \sim k$ の各ルールに対応した k 枚の仮想ページが重ね合されたものとして考える。

まずこの仮想ページのおおののに、3.4 節に従って得られる階層ページ構造を適用する。すなわち観測値 b に対応するべきメンバーシップ関数が、観測値 a により得られるメンバーシップ関数値を最大値として修正され、各ルールにおける $\min$ 演算の結果である $C_1 \sim C_k$ の値を得ることが可能となる。

これらの値の $\max$ 演算を行うためには、 k 枚の仮想ページに描かれたおおのののメンバーシップ関数を、重なる部分が大いほうの値をとるように、元の 1 枚の下位ページに重ね合せて描く。

こうした仮想ページによる階層構造をあらかじめ用意しておくことにより、観測値 a および b が形成する複合アドレス空間上において、上位ページに対応する a および下位ページの台集合に対応する b の値が確定すれば、任意の合成ルールによる演算結果 c を得ることができる。

また一般に、1 個の記憶素子の入力アドレスピンを M 個の複合アドレス空間に、かつ出力データピンを N ワードに分割すれば、前件部の項目数が M 、後件部の項目数が N の、任意の合成規則によるファジィ演算を行うことができる。

こうした仮想ページング方式による最大の利点は、同一のハードウェア構成で、しかも演算速度に影響を与えず、ルール数 k を任意に増減させることができる点である。またラベルの定義においても、仮想ページの台集合上にメンバーシップ関数が任意の位置を占めるので、自由に行うことができる。さらにメンバーシップ関数値に関しても、最終出力を除く仮想ページ上での演算はこれを離散化する必要がなく、浮動小数

点実数値の意味での連続値として、別途計算機などで計算して処理することが可能となる。

3.6 仮想ページング方式その 2

3.5 節よりさらに階層化された仮想ページに仮想的なページングを行うことで、逆ファジィ化までの一般的なファジィ推論の全過程を、Fig. 6 のように 512 K (=64 K×8) ビットの記憶素子 1 個のみを用いて得られることを示す。ここでは簡単のためルール前件部の項目数を 2、後件部の項目数を 1、ルール数を 2、合成規則に関する演算方式を $\max/\min$ 法、逆ファジィ化の手法は重心法を採用するとして説明する。

ルール 1: IF $A=A_1(a) \ \& \ B=B_1(b)$

THEN $C=C_1(c)$

ルール 2: IF $A=A_2(a) \ \& \ B=B_2(b)$

THEN $C=C_2(c)$ (15)

各ルール前件部の $\min$ 演算結果は、

$$AB_1=A_1(a) \wedge B_1(b) \quad (16)$$

$$AB_2=A_2(a) \wedge B_2(b) \quad (17)$$

3.5 節ではこれより $\max \{AB_1, AB_2\}$ を求め出力した。ここでは後件部のメンバーシップ関数である $C_1(c)$ および $C_2(c)$ を表現するために、さらに下位の階層に仮想ページを考えて、後件部のメンバーシップ関数を以下のように修正する。

$$C'_1(c)=AB_1 \wedge C_1(c) \quad (18)$$

$$C'_2(c)=AB_2 \wedge C_2(c) \quad (19)$$

これらの処理過程はすべて仮想ページ上のことであ

Fig. 6 Concept of virtual paging (2)

るから、後件部の台集合に関しても離散化する必要は生じず、別途計算機などで浮動小数点実数値の連続値として計算することができる。

さらにこれらの仮想ページに描かれたおのおののメンバーシップ関数を、重なる部分が大きいほうの値をとるように、1枚の仮想ページに重ね合わせて max 合成を行うことで、ファジィ出力としてのメンバーシップ関数 $C(c)$ を得る。

$$C(c) = C'_1(c) \vee C'_2(c) \quad (20)$$

こうした仮想ページによる階層構造をあらかじめ用意しておけば、観測値 a および b が複合的に形成する仮想的な階層アドレス空間上において、 a および b の値が確定すれば、(20)式を表現する最終的な仮想ページも確定するので、次式を別途計算機などで計算することにより、確定出力 C を得る。

$$C = \int C(c) \cdot c \cdot dc / \int C(c) \cdot dc \quad (21)$$

確定出力を得るための逆ファジィ化の手法においても別途計算機などで計算を行うので、(21)式の重心法に限定されず、面積法・高き法など、いずれの方式をも採用することができる。

しかるに上の例において、アドレス $(a \times 2^8 + b)$ の実際の記憶空間に格納されるのは、最終出力結果である C である。

また一般に、前件部の項目数が M 、後件部の項目数が N の任意のファジィ演算を行う場合においては、2.5 節で既述した制約を受ける。

3.7 仮想ページング方式まとめおよび応用例

たとえば次式を演算するに当たって、アドレスルックアップ方式により汎用の記憶素子を応用する方法は、しばしば用いられてきた。

$$b_1 = f_1(a_1, a_2) \quad (22)$$

ここで以下の例のごとく、複数の入力から複数の演算方法を経て複数の演算結果を得、さらに複数の階層的な演算処理する場合について考えてみる。

$$\begin{aligned} b_1 &= f_1(a_1, a_2) \\ b_2 &= f_2(a_1, a_2) \\ b_3 &= f_3(a_1, a_2, a_3) \\ c_1 &= f_4(b_1, b_2) \\ c_2 &= f_5(b_2, b_3) \\ c_3 &= f_1(a_2, b_1, b_2) \\ d_1 &= f_6(c_1, c_2, c_3) \\ d_2 &= f_7(a_1, c_1, c_2, c_3) \end{aligned} \quad (23)$$

ところが、これらの式は次式に総括して考えることが可能である。

$$(d_1, d_2) = F(a_1, a_2, a_3) \quad (24)$$

Fig. 7 Hierarchical operation

このことは換言すれば、一定の入力条件に対し一義的に出力結果を得る場合において、その中間段階に存在するいかなる種類の演算に対しても、別途計算機などでその演算を行ってその演算結果を記憶素子に格納しておけば、実際の制御に当たってはアドレスルックアップを適用して高速に任意の演算結果を得られることを意味する。

仮想ページング方式によるファジィ推論チップは、この原理を応用したものであり、仮想ページは中間の演算を実行するのに必要な仮想上の階層的なワーク空間である。

しかるに 2.5 節の制約条件は、換言すればデジタルの記憶素子をもつ制約であり、最大の制約原因はその記憶容量に帰結する。ゆえに入出力の各項目に同一の分解能を与えると仮定した場合において、出力の項目数が増加するのに必要な記憶容量は 1 次関数的に増加させれば良いのに対して、入力項目数が増加するのに必要な記憶容量は指数関数的に増加させなければならない。

必要記憶容量の増加に対しては、記憶素子を複数個使用するか、大容量の外部記憶装置を利用することが考えられる。ただし入力対象を階層的に把握することにより、必要記憶容量の増加を大幅に抑えられる場合があるので、その 1 例を以下に示す。

$$\text{IF } A1 \ \& \ A2 \ \& \ A3 \ \& \ A4 \ \text{THEN } C \quad (25)$$

上式のようなルールに対してすべての項目に 8 ビットの分解能が要求され、かつアドレスルックアップによるデジタル記憶素子を用いると仮定すれば、必要記憶容量は $2^{32} \times 8$ ビットになる。そこで (25) 式をつぎのように階層化する場合を考える。

$$\begin{aligned} \text{IF } A1 \ \& \ A2 \ \text{THEN } B1 \\ \text{IF } A3 \ \& \ A4 \ \text{THEN } B2 \\ \text{IF } B1 \ \& \ B2 \ \text{THEN } C \end{aligned} \quad (26)$$

上式では必要記憶容量を $2^{16} \times 8 \times 3$ ビットとすることができ、Fig. 7 に示すように 512 K (= $64 \text{ K} \times 8$) ビットの記憶素子を 3 個用いることで足りる。

4. おわりに

アドレスルックアップ仮想ページング方式によるフ

ファジィ推論チップの特徴を挙げて、動作原理について記述した。

本方式の採用によって、自由形状のメンバーシップ関数と任意の数のラベルを使った任意の数のファジィルールに基づく任意のファジィ演算を連続値モードで高速に実行させることが可能となった。

また命令コード不要などを理由とする耐ノイズ性と、汎用ディジタル素子を利用したゆえの経済性および現行コンピュータシステムとの整合性を論じるなど、その他総合的な実用性を確認した。さらに階層推論や後向き推論への応用および学習機能や神経回路網への展望を示した。

アドレスルックアップの基本概念として、アドレスピンの利用目的を番地検索でなくデータ入力とする点を強調し、複合的な階層ページ構造をもつ仮想ページを台集合に対応させ、仮想ページングによるファジィ推論方式の多くの利点を説明した。

具体的な回路構成は本論文の主題ではないので詳述しないが、最も簡単な例では、記憶素子単体の前段にサンプル & ホールド回路、後段に出力確定用ラッチ回路を設けて、同相のクロックエッジによりタイミングを取る。これにより推論処理速度を可変とでき、ノイズ除去にも役立つ。

今後の研究として、高性能コンピュータを援用したファジィ推論チップ開発支援システムの構築、推論チップを搭載した PC 用拡張ボードなどによる実時間エキスパートシステムへの応用、大規模容量の記憶素子と論理回路や変換回路との混成集積回路の開発、学習機能を有するコントローラの開発、それを拡張した神経回路網への応用、ファジィ理論による画像認識や記憶方式の確立と、これらを総合したファジィとクリスプシステムの融合を目指す相互補完型コンピュータへの発展が考えられる。

参 考 文 献

- 1) T. Yamakawa: Fuzzy Microprocessors-Rule Chip and Defuzzifier Chip, Proc. International Workshop on Fuzzy System Applications, Iizuka, 51/52 (1988)
- 2) M. Togai and H. Watanabe: A VLSI Implementation of a Fuzzy Inference Engine: Toward an Expert System on a Chip, Information Sciences, 38, 147/163 (1986)
- 3) H. Watanabe and W. Dettloff: Reconfigurable Fuzzy Logic Processor a Full Custom Digital VLSI, Proc. International Workshop on Fuzzy System Applications, Iizuka, 49/50 (1988)
- 4) T. Yamakawa: A Simple Fuzzy Computer Hardware System Employing MIN and Max Operations, Proc. 2nd Congress of IFSA, Tokyo, 827/830 (1987)