

解説

2. 専用 VLSI プロセッサの具体例



2.6 VLSI ニューロプロセッサ†

村岡 洋 一†

1. ま え が き

ニューロネットワークの研究の方向を大きく分けると、次の二つになる。

(1) 人間 (も含めて生物) の脳の働きを解明する。

これが達成できれば、その知見を利用することによって、今までは目的ごとに実現していた各種の人工知能技術を、汎用を実現することが可能になる。

(2) 形式ニューロンを使ったネットワークを工学的に利用する。

ニューロンのモデル (形式ニューロン) を多数個並べたネットワークを構成して工学的に利用する。このような研究の代表例が、バックプロパゲーションネットワーク (Back Propagation Network: BPN) やホップフィールドネットワーク (Hopfield Network) である¹⁾。このための VLSI ニューロプロセッサを開発する。

本小文では(2)に関する動向について概観してみよう。

ネットワークの規模は記憶させるパターン数などによるが、たとえば記憶パターン数の約1桁上のニューロンが必要となるとされている。このために、実用的なネットワークを考えると、その規模が非常に大きくなる。このような背景のもとに、上記の計算に要する時間は以下のように長大化する傾向にある。

(a) 収束時間

ホップフィールドネットワークなどのようにフィードバックのあるネットワークでは、その収束に時間がかかる (図-1)。

(b) 学習時間

バックプロパゲーションネットワークなどにおける学習では、たとえ1回ずつの計算時間はそれほど大き

くなくとも、学習回数が大きくなれば全体の所要時間は非常に大きくなる。英文の文-音声変換ネットワークの学習に数週間を要したという例もある。

このような計算には、もちろんスーパーコンピュータに代表される大型汎用コンピュータを使ってもよい。しかし、後に述べるようにニューロンの計算モデルは比較的簡単であるので、経済的かつ高速に行える専用アーキテクチャの開発の余地が大きい。

ニューロネットワーク用専用アーキテクチャを大別すると、以下ようになる。

(a) 仮想アーキテクチャ

1個のプロセッサに複数個のニューロンの動作のシミュレーションを分担させる。代表例は、いま盛んに製品化の進んでいるパーソナルコンピュータ用アクセラレータである。

(b) 並列アーキテクチャ

1個のニューロンに対応させて1個のニューロプロセッサを設ける。大規模なニューロネットワークの計算には、たくさんのプロセッサを使った並列コンピュータが不可欠になる。

以下では、並列アーキテクチャの構成要素となる専用 VLSI ニューロプロセッサ (なお、現実には最先端の VLSI 技術によるニューロプロセッサの実現例は

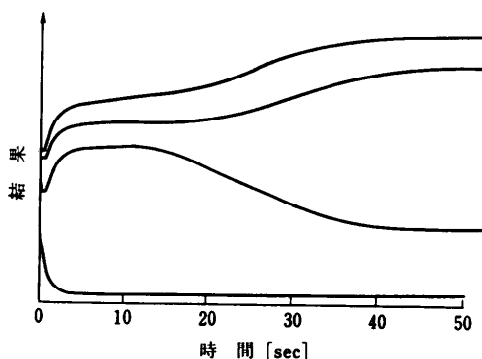


図-1 ネットワークの収束時間の例

† VLSI Neuro Processor by Yoichi MURAOKA (Waseda University).

† 早稲田大学理工学部

ないし、アナログ技術を使った実現例はプロセッサというよりもチップやデバイスと呼んだほうがふさわしいものが多い。しかし、本小文ではこの特集のタイトルを尊重してあえてすべてを VLSI ニューロプロセッサと呼称する) をとりあげる。

2. ニューロネットワークのモデル

基本的なニューロンのモデル (形式ニューロン) を以下に示す (図-2)²⁾。

(1) 他ニューロンの出力 (o_i) またはネットワーク外部からの信号が、シナプスを介してニューロン j に与えられる (なお、ネットワーク外部からの信号は基本的には時間変化のない固定値をとるモデルが多数であるが、これを受けて動作したニューロンの各信号値は、時間的に変化する値である)。

(2) それらの入力それぞれに重み (W) をかけて総和をとったもの (積和) が、入力値 net_j となる。

$$net_j = \sum_{i=1}^n o_i W_{ij} - \theta$$

θ は入力値にバイアスを与えるためのしきい値である。

(3) 入力値を元にニューロン j の内部状態 $a_j(t)$

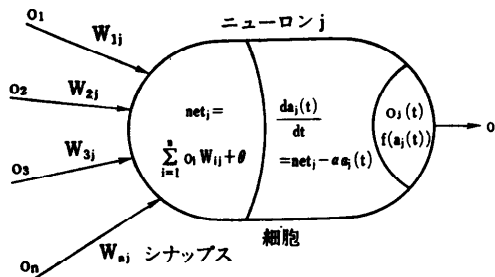


図-2 形式ニューロン

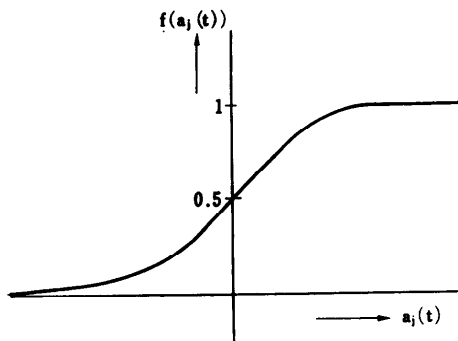


図-3 ロジステック関数

が変化する。

$$da_j(t)/dt = net_j - \alpha a_j(t)$$

バックプロパゲーションネットワークなどの1方向ネットワークでは、内部状態は単に入力値に等しい、すなわち、

$$a_j(t) = net_j$$

であってもよいが、ホップフィールドネットワークなどのような帰環型ネットワークでは、時定数が不可欠である。

(4) ニューロン j からの出力値 o_j は、内部状態の関数である。

$$o_j(t) = f(a_j(t))$$

ここで、関数 f はシグモイド関数などと呼ばれる出力関数であり、代表例は、

$$f(a_j(t)) = 1 / (1 + \exp(-a_j(t)))$$

で与えられるロジステック関数である (図-3)。簡単なモデルでは、シグモイド関数が、

$$o_j(t) = \begin{cases} 1 & a_j(t) \geq 0 \\ 0 & a_j(t) < 0 \end{cases}$$

というスレッシュホールド関数となる。この場合には、信号は当然 0/1 の2値となる。

3. VLSI ニューロプロセッサの実現技術と課題

VLSI ニューロプロセッサの実現技術を大きく分類すると、以下ようになる。

(1) 信号 (入力および出力)

ニューロンの信号の表現法には以下の2通りがある (表-1)。

・普通のコンピュータのように直接数値で表現 (デジタル表現) する。信号の時間変化はこの数値列となる。または、デジタルパルスの密度で表現する方

表-1 アナログ回路とデジタル回路の比較

比較項目	アナログ回路	デジタル回路
動作速度	◎ 実時間性がある	△ 一定の速度である
基本素子 (回路エレメント) のエリアの大きさ	○ 小さい (アンプなど)	◎ 非常に小さい (ゲート)
ニューロン回路の大きさ	◎ 小さい	○ 大きい
重み (シナプス) の可変性の実現	△ 困難	◎ 容易
消費電力	回路による	△~○
耐雑音 (発振など)	△	○

法もある。

・数値以外の表現（アナログ表現）を使う。信号の大きさの表現には、アナログ信号の振幅値（連続値）を対応させる。

形式ニューロンの実現を考えるかぎり、アナログの表現を採用する数学的な必然性はない（生物のニューロンのシミュレーションについては別であるが）。したがって、工学的な優劣の比較が大きなポイントとなる。

(2) 重 み

可変／固定の2種がある。可変のほうが利用価値は大きい。アナログ信号を採用した場合などには作りやすさから固定にならざるをえないことがある。アナログ信号で固定値でよければ、抵抗によって簡単に重みを実現することができる。重み値の記憶／付加にもアナログ／デジタルの2種がある。VLSI 記憶技術などとの親和性からはデジタルが勝る。

(3) 入力 の 積 和 と 内 部 状 態 の 計 算

アナログ信号（振幅値表現）を採用する場合には、電流を信号に使用して抵抗によって重みを表し電流の重ね合わせをすることによって、積和处理は簡単に実現

できる。デジタルな数値表現においては、演算回路を使うことになる。一般に n 入力の積和演算をそのための専用回路を作るか、または2入力演算回路を繰り返して使うかの判断となる。

(4) シグモイド関数による出力の計算

アナログ信号の場合には、シグモイド関数の実現法として、オペアンプやインバータなどを使う方法がある。

デジタル信号の場合には、シグモイド関数を級数展開で近似しておいて、入力値から出力値を計算するのがオーソドックスな実現法である。高速で計算するためには、関数計算をするかわりに表を用意しておいてこれを引く方法などもある。

(5) ニューロン間の接続

ニューロネットワーク実現時の最大の問題といってもよいのは、多数のニューロン間の接続技術である。

これまでに研究・開発された主な VLSI ニューロプロセッサを表-2 にまとめた。次章以下にこれらの詳細について説明することにしよう。

4. デジタル VLSI

ニューロプロセッサ

デジタル VLSI ニューロプロセッサの機能は図-4 のようである⁹⁾。信号をすべて完全にデジタル数値化して、デジタル演算技術を使って実現すれば、自由度が大きくなりいろいろなモデルのシミュレーションに柔軟に対応できる汎用プロセッサが実現できることになる。しかも、これまで培った VLSI 技術の恩恵をそのままこうむることもできる。しかし、技術的に従来技術の延長であるということから面白味が多くないということもあって、このアプローチはアーキテクチャやデバイス技術者からはそれほど重視されていない。だが、ニューロネットワークの研究を加速するツールとしての役割から考えると、その柔軟性は大切であり、もっと本腰を入れられてし

表-2 主な VLSI ニューロプロセッサ

	デジタル回路		アナログ回路	
	可変重み	固定重み	可変重み	固定重み
シナプス部 (重 み)	(a) デジタル数値信号 ・2値—デコーダ ・広大 ¹⁾ ・多値—乗算器 ・日立中研 ⁴⁾ (b) パルス列信号 ・レートマルチプライア ・筑波大 ^{6),7)}	(a) アモルファス抵抗 ・ベル研 ⁹⁾ (300 k Ω) ・JPL ¹⁰⁾ (1 M Ω) (b) MOS 抵抗 ・JPL (200 k Ω)	(a) 2 値 ・ベル研 ¹⁰⁾ (b) 多値 ・MOS 抵抗 ・アリゾナ州立大 ¹¹⁾ ・MIT ・薄膜抵抗 ・JPL ¹⁰⁾ ・スイッチドレジスタ回路 ・慶大 ⁹⁾ ・乗算型 D-A 変換器 ・富士通 ¹¹⁾	
細胞部 (シグモイド関数)	(a) デジタル数値信号 ・テーパーバックアップ ・広大 ¹⁾ ・関数回路 ・日立中研 ⁴⁾ (b) パルス列信号 ・カウンタ ・筑波大 ^{6),7)}		(a) インバータ (b) オペアンプ	

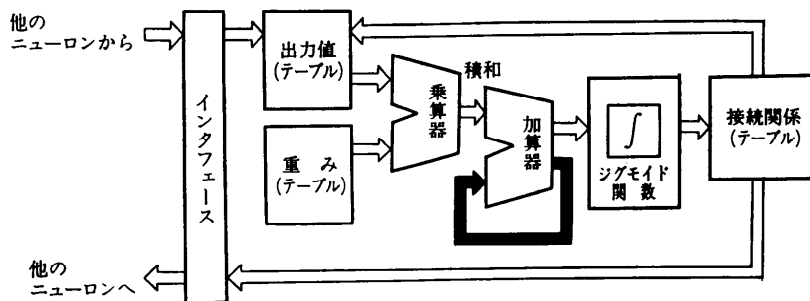


図-4 デジタル VLSI ニューロプロセッサ

かるべきものであろう。また、アーキテクチャ的にも工夫のしどころはまだ以下のようにある。

(1) 積和回路

n 入力回路の実現、またはパイプライン的に動作できる積和回路の実現。たとえばログサム (logsum) 法などの採用がある。ただし、シグモイド関数がスレッシュホールド関数で信号が2値となる場合には、後述するように簡単なテーブルルックアップ回路で実現できる。

(2) シグモイド関数

テーブルルックアップなどによる実現。

(3) パイプラインネットワーク

ネットワーク全体（ニューロン間のデータ転送およびニューロン内の演算）の動作をすべて、パイプライン化する。その場合、ビットシリアル・ビット単位パイプライン化の可能性も追求すべきである。この実現には、レシデュアルナンバ表現などの採用が考えられる。もしこれが可能となれば、ニューロン数の増大が容易になる。

以上のような完全デジタル回路の泣きどころは、VLSI チップ内の回路エリアが大きくなるところである。

デジタル VLSI ニューロプロセッサの例として、日立中央研究所の山田らが開発したものがある⁴⁾。基本的構成は図-4 に同等である。この例では、約 750 ゲートで実現できた。重み値はメモリにあり、結合相手先アドレスで検索するしくみになっている。積和は、乗算器と加算器を使って実現している。ニューロン間の接続にはバスが使われており、信号伝送はパイプライン化されている。これにより、経済的で性能のよい接続を実現した。576 ニューロンからなるチップが試作された。

広島大学の阿江らは、ホップフィールド型ニューロネットワークを従来のコンピュータのメモリとみなして、最適化問題のための専用アーキテクチャを開発し

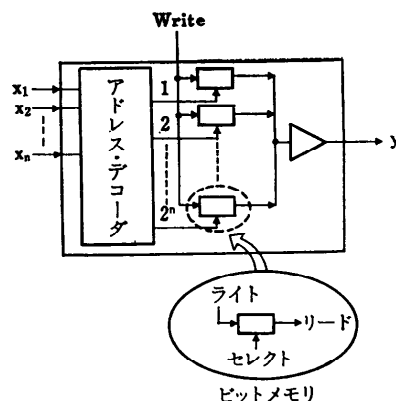


図-5 RAM ニューロン

ている⁵⁾。本回路では、シグモイド関数の代わりにスレッシュホールド関数を採用している。信号は 0/1 の 2 値になる。 n 入力から m 出力へのブーリアン関数は、一般にはメモリを使ってテーブルルックアップで実現できるので、積和とスレッシュホールド関数は図-5 のように、メモリとアンプおよび積分回路を組み合わせた回路で実現している。

これに対して、パルス密度で信号値を表す回路が筑波大学の平井らによって提案されている（図-6）。

この回路では、シグモイド関数として、

$$\phi[a] = \begin{cases} a & a > 0 \\ 0 & a \leq 0 \end{cases}$$

というスレッシュホールド関数を使うモデルを実現している。本回路の動作原理は以下のようなものである。

(1) 積和回路

重み付けはレートマルチプライア回路で実現している。レートマルチプライアは、重み値（1 以下）に比例して、入力パルス列からパルスを間引く回路である。

パルス列の OR をとることで総和を実現している。

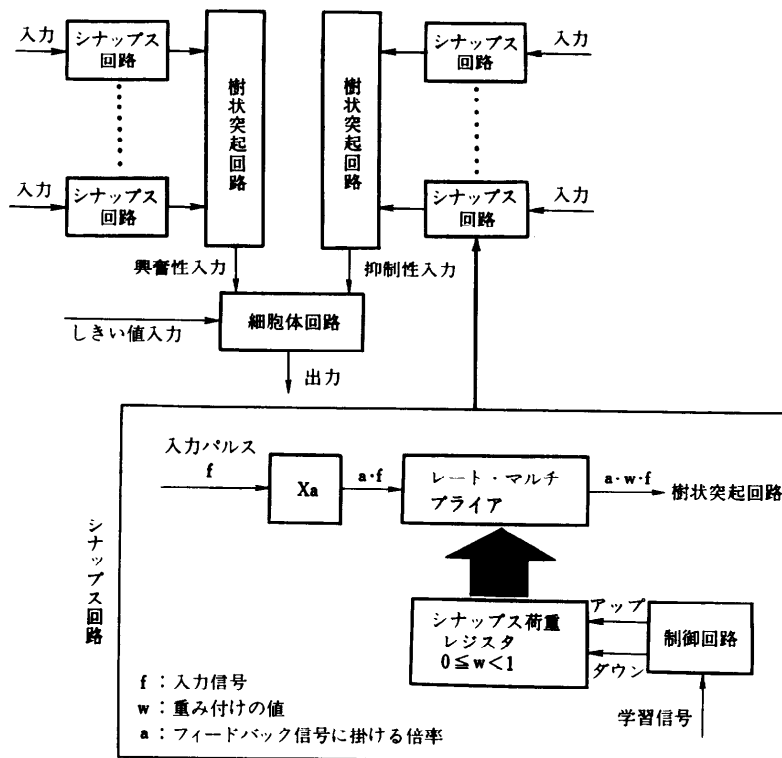


図-6 パルス列を信号とするニューロン回路

パルスが重なっても1パルスにしかカウントされないという問題があるが、それぞれのニューロンのクロックをずらすなどの工夫をこらして対処する。

(2) 出力回路

ニューロンの内部状態値を保持するアップダウンカウンタ（正負の値をとることができる）を設ける。アップダウンカウンタに内部状態を積分する。レートマルチプライアによって、このカウンタの値に比例したパルス列をクロックから生成する。ただし、内部状態が負ならば、出力値は0となる。

現在のところ、6ニューロンからなるチップが試作されている。

5. アナログ VLSI ニューロプロセッサ

生体ニューロンの動作は本質的にはアナログであるので、VLSI ニューロプロセッサもアナログのものを設計したいという願望がある。また、形式ニューロンのネットワークでも、実時間性を重視するような応用ではアナログが有利になってくる。

アナログ回路を使ったニューロンの実現では、入力信号と重みとの積処理（シナプス）部を抵抗回路

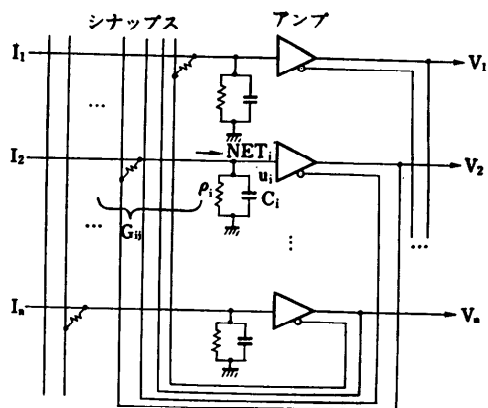


図-7 アナログニューロン回路の例

に、それらの信号の和をとる処理部をコンデンサに、そしてシグモイド関数をインバータに、それぞれ対応させるのが自然である（図-7）が、この実現には以下のような問題がある。

(a) シナプス部の接続の重みを可変にするには、可変抵抗器を VLSI 上に実現しなければならないが、これは一般には困難である。

(b) 大容量コンデンサも VLSI にはなじまない。
 (c) アナログ回路自身、消費電力が大きく、VLSI 化には不利である。

(d) 大規模なネットワークの実現では、多数のシナプスが並列接続されることになるので、シナプス抵抗値を大きくしておかないと大電流が流れてしまう。これを避けるために非常に高抵抗値を実現しなければならない。

以下にさらに詳しくみてみよう。

(1) 重み

連想メモリなどのように、重みの値をあらかじめ設計時に決定できる応用では、アモルファスシリコンなどの固定抵抗が利用できる。ベル研究所で試作した例では、ポリイミドを絶縁体に使いその間にアモルファスシリコンを埋め込んで、約 300 k オーム (ばらつき 5%、ラインスペース 2 μm) の抵抗を実現した⁸⁾。

可変重みの実現には、次のような手法が提案されている。

(a) トランジスタの抵抗の利用 (図-8)

ゲート電圧を制御してチャネル抵抗を変える。

(b) CR 充電回路の応用

ホップフィールドネットワークを実現する電気回路は図-7 のようであった。シナプス (接続部) の重み W_{ij} はコンダクタンス G_{ij} で、ニューロンはオペアンプで、それぞれ実現されており、ニューロンの出力値はオペアンプの出力電圧で表されている。オペアンプ i に流れ込む電流の総和 NET_i は、

$$\text{NET}_i = \sum G_{ij} V_j + I_i$$

であり、ニューロンの入力値 net の式と比較すれば、 NET が net に、 G_{ij} が W_{ij} に、 V_j が o_j に、そして I_i が θ_i に、それぞれ対応することが分かる。また、オペアンプの入力電圧 u_i がニューロンの内部状態 a_i に対応する。オペアンプの出力電圧 V_i が、ニューロンの出力に対応する。

シナプス接続の重みを可変にするには、コンダクタンスを可変にすることが必要であるが、これを CR の充電回路 (スイッチドレジスタ: Switched Resistor) を使って実現する。 u_i の式は次のようである。

$$\begin{aligned} du_i/dt &= -u_i/\tau + \text{NET}_i/C_i \\ &= -u_i/\tau + (G_{ij} V_j + I_i)/C_i \end{aligned}$$

ただし、

$$\begin{aligned} \tau &= R_i C_i \\ &= C_i / \{1/\rho_i + \sum G_{ij}\} \end{aligned}$$

これから、重みに対応する G_{ij} の値を変化させる代わりに時定数 C_i/G_{ij} の値を変化させても、その役割りは十分に果たすということが分かる。このような時定数の値を変化することのできる回路の一つが、スイッチドレジスタ回路である⁹⁾。

スイッチドレジスタ回路の原理は、図-9 のようであり、スイッチの開閉を同図に示すようなデューティ比 ($d = t_{\text{on}}/(t_{\text{on}} + t_{\text{off}})$) で行くと、時定数 τ は、 $\tau = RC/d$ となる。実際にデューティ比 d の開閉をするようなスイッチの回路の例は、図-10 のようである。

(2) 積和回路

信号に電流を使えば、信号の加算は電流加算で簡単に実現できる。

(3) シグモイド関数

CMOS インバータの入出力特性はシグモイド関数

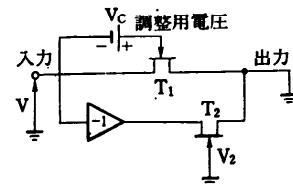


図-8 トランジスタの抵抗の利用

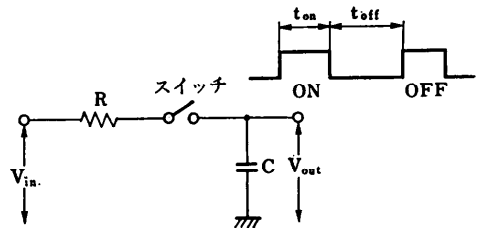


図-9 スwitchドレジスタ回路

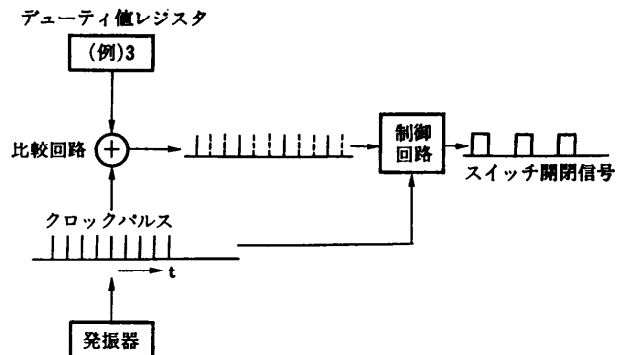


図-10 デューティ比設定回路

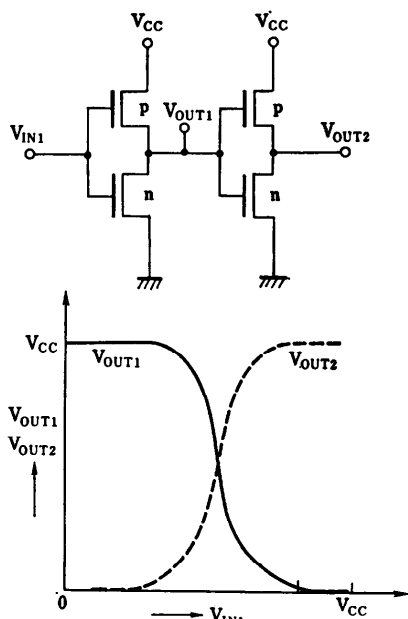


図-11 CMOS インバータ回路

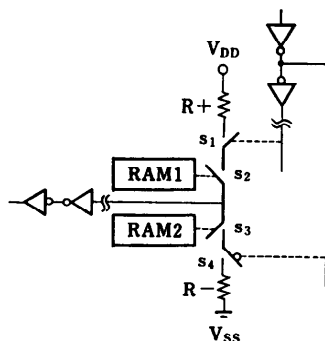


図-12 ATT のニューロン回路のシナプス部

に似ているのでこれを利用すれば、小さいエリア面積で簡単に実現できる(図-11)。ただ、関数形に制限があるという欠点がある。これを解決するためには、オペアンプや差動増幅器などを使えばよい。

初期のアナログ VLSI ニューロプロセッサの代表例は、A. T. T. ベル研究所の Graf らのプログラマブル連想メモリである¹⁰⁾。ホップフィールドネットワークのチップであり、ニューロンにはインバータを使用している。シナプス部は図-12 のようであり、2 値のうちのいずれかを選択できるようになっている。2.5 μm ルールで 54 ニューロン/チップを実現した。

また、図-13 は富士通の土屋らが試作した VLSI

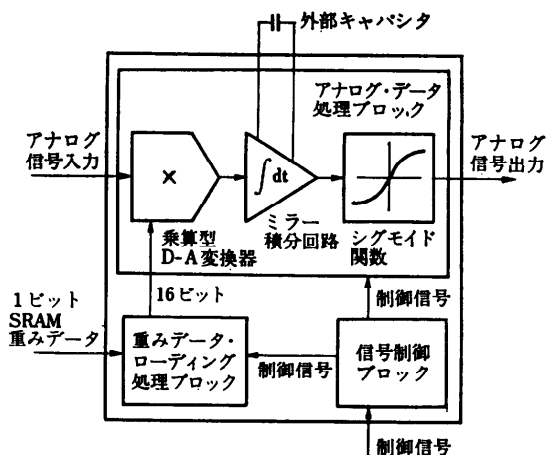


図-13 富士通の ANP

ニューロプロセッサ (ANP) である¹¹⁾。ニューロン間をつなぐ接続路は時分割で使用する。入力信号はアナログとして重みはデジタル (数値) として、これを乗算型 D-A 変換器を使って掛け算する。ミラー積分回路でその和を取り、結果をコンデンサに蓄える。最後のスレッショルド関数は、6 個の差動アンプを組み合わせた折れ線近似回路で実現する。それぞれのアンプの特性を適当に調整することにより、近似積分を構成する。いまのところ、10 mm 角チップ上に 32 ニューロン程度が実現できると予測されている。

6. おわりに

以上に駆け足ではあるが、VLSI ニューロプロセッサの技術について紹介した。本小文では触れる余裕がなかったが、生体ニューロンの VLSI 化については、Mead らの目の網膜のニューロンの研究もある¹²⁾。ただ、形式ニューロンにしる生体ニューロンにしる、まだまだそれらの定説はないといってよい。そういった状況では、考えたニューロンモデルを気楽に (VLSI) チップとして実現できる環境が欲しい。アメリカでは MOSIS がある。日本には何があるだろうか。VLSI ニューロプロセッサの研究状況を眺めると、その研究の内容もさることながら、研究環境で大幅にアメリカに差をつけられていることを痛感する今日この頃である。

参考文献

- 1) McClelland, J. L., Rumelhart, D. E. and PDP Research Group: Parallel Distributed Processing, Vol. I and Vol. II, MIT Press, 1986 (甘

- 利監訳：PDP モデル—認知科学とニューロン回路網の探索，産業図書（1989）。
- 2) 甘利俊一：神経回路網の数理，産業図書（1978）。
- 3) 阿江 忠：ニューロチップ，bit，Vol. 21，No. 12，pp. 54-62（1989）。
- 4) 山田 稔他：大規模ニューラルネットワーク向けハードウェア方式，情報処理学会第 38 回全国大会，3 U-2，pp. 1526-1527（1989. 9）。
- 5) 久長 穰他：アルゴリズム駆動ニューロコンピュータ AN 1 のハードウェア，情報処理学会計算機アーキテクチャ研究会，74-2（1989. 1）。
- 6) 鎌田勝浩他：ディジタル神経回路モデル，電子情報通信学会技術研究報告，MBE 87-157，pp. 415-422（1988. 3）。
- 7) 平井有三他：ディジタルニューロチップの開発，情報処理学会第 38 回全国大会，3 U-1，pp. 1524-1525（1989. 9）。
- 8) Hubbard, W. et al.: An Associative Memory Based on an Electronic Neural Network Architecture, IEEE Trans. Electron Devices, Vol. ED-34, No. 7, pp. 1553-1556（1987. 7）。
- 9) 秋山 泰：ガウシアンマシンとそのアナログ/ディジタル専用アーキテクチャ，電子情報通信学会技術研究報告，CPSY 88-16，pp. 69-76（1988. 7）。
- 10) Graf, H. et al.: VLSI Implimentation of a Neural Network Model, IEEE Computer, Vol. 21, No. 3, pp. 41-49（1988）。
- 11) 土屋主税他：世界で初めて商品化されるニューロチップ，日経マイクロデバイス，1989 年 3 月号 pp. 123-129（1989）。
- 12) Thakoor, A. et al.: Binary Synaptic Connections Based on Memory Switching in a-Si: H, AIP Conf. Proc. pp. 151-158（1986）。
- 13) Walker, M. R. and Akers, L. A.: A Neural Approach to Adaptive Digital Circuitry, IEEE Proc. of Seventh Annual International Phoenix Conference on Computers and Communications, pp. 19-23（1988）。
- 14) Daud, T. et al.: Neural Network Based Feed-Forward High Density Associative Memory, 1987 IEEE International Electron Devices Meeting Technical Digest, pp. 107-110（1987）。
- 15) Mead, C.: Analog VLSI and Neural Systems, Addison-Wesley（1989）。

（平成元年 12 月 4 日受付）