

## SOUNDコンピュータの試作

梶岡秀朗, 籠屋健\*, 笹尾和宏\*, 高橋雅哉\*, 中村維男\*  
株式会社フレンドリーシステムズ  
東北大学\*

SOUNDコンピュータは、ニューラルネットワークをプログラム可能にした全く新しい概念のコンピュータである。従来のノイマン型とは、手順の与え方、データの流れ方において異なる。その構成は、コネクシオンネットワーク、コネクシオンレジスタ、演算ユニットから成り、各ユニットは、コネクシオンネットワークを通して繋がっている。その動作は、まず、コネクシオンレジスタの内容に基づいてコネクシオンネットワークを繋ぎかえて、データフローグラフをそのままのトポロジーでハードウェア上に実現し、そして、そこに同期式にデータを流して並列処理を行う。その応用分野としては、データの圧縮、伸長、認識や三次元グラフィクスなどのマルチメディア処理や科学技術計算の高速化が挙げられる。

## Concept and Construction of the SOUND Computer

H.Tsukioka, T.Komoriya\*, K.Sasao\*, M.Takahashi\*, T.Nakamura\*  
Friendly Systems Co., Ltd.  
Tohoku University\*

The SOUND computer is a novel non-von Neumann computer. Its differences from von Neumann computer are the ways of giving order and flowing data. The SOUND computer consists of a connection network, a connection register, and operation units. Each unit is connected through a connection network. In its action, firstly, implement a dataflow graph directly on a hardware in the same topology, then, execute the graph by flowing data synchronously. The SOUND computer is aiming at the mathematically sound computer on its hardware and software. And its purpose is to accelerate the scientific calculation and the multi-media processing such as data compressing, extracting, recognition, and 3-dimensional graphics.

### 1. 序論

現在普及しているコンピュータは、ノイマン型と呼ばれる処理機構で動作している。ここでは、ノイマン型を広く解釈し、命令の順序実行によりプログラムを実行するコンピュータの総称とする。これに対して、多くの人々が、新しい処理パラダイムによるコンピュータのブレイクスルーを目指して、非ノイマン型コンピュータの研究開発を行っている。現在存在する非ノイマン型コンピュータとしては、データフロー型コンピュータとニューラルネットワークがある。この様子を表1に示す。

データフロー型コンピュータはノードと呼ばれる非同期に動作する多数の処理エレメントから成り、ノイマン型コンピュータの処理形態の枠を打ち破った全く新しい処理パラダイムを目指して開発された。スタンフォード大学のD. A. Adams<sup>[1]</sup>とマサチューセッツ工科大学のJ. E. Rodriguez<sup>[2]</sup>により考案され、マサチューセッツ工科大学のJ. B. Dennis<sup>[3-5]</sup>により発展させられた。しかし、そのハードウェアの実現においては、命令駆動となっており、結局、広い意味でのノイマン型に属する。

ニューラルネットワーク<sup>[6]</sup>は、脳の神経細胞が形成するネットワークであり、完全に非ノイマン型コンピュータである。ニューラルネットワークの特徴として、非常に簡単な構造、非常にきめ細かい並列性、処理系が均一でどこでも同じ構成、学習により処理手順を得ておりプログラムが不必要の4つがある。そして、沢山の神経細胞が協調して一つの仕事をっており、その処理手順は、神経細胞の間の結合の強さとして蓄えられている。しかし、処理手順を学習により得ているため、学習に時間がかかり、処理手順の冗長性が高く、無駄が多い。

SOUNDコンピュータは、従来型コンピュータとニューラルネットワークの中間に位置する全く新しい処理原理のコンピュータである。その従来型との違いは、処理手順の与え方、データの流れ方の2点であり、そのデータフロー型との違いは、ノイマン型と非ノイマン型の入れ子構造が逆なことである。その名称のSOUNDは、そのアーキテクチャを表すSynchronously Operating Units for Neural Dataflowの頭文字を取ったものある。本報告書では、このSOUNDコンピュータの概念およびその試作・評価について述べる。第2章で処理概念について、第3章でそのアーキテクチャについて、第4章でその試作機について、第5章でその評価について述べる。

	従来型	データフロー型	SOUND	ニューラルネットワーク
並列度	小	小	中	非常に大
構造	複雑	複雑	簡単	簡単
処理手順の指示	プログラム	データフロー図 (プログラム)	データフロー図 (プログラム)	学習 (非プログラム)
データの流れ	水滴の落ちる様	水滴の落ちる様	滝水の落ちる様	滝水の落ちる様
分類	ノイマン型  完全にノイマン型	非ノイマン型 ノイマン型  外見は非ノイマン型 中身はノイマン型	ノイマン型 非ノイマン型  非ノイマン型を プログラムで制御	非ノイマン型  完全に非ノイマン型

表 1 コンピューター一覧

## 2. 処理概念

SOUNDの処理概念は、プログラム可能なニューラルネットワークである。ニューラルネットワークは、

神経細胞間の結合の強さによりデータの処理の仕方が決められている。その様子を図1に示す。この結合の強さは、通常、学習により得るが、この値を外部から与えることにより、学習をせずにその能力を得ることができる。この事は、結合の強さがプログラムと見なせることを示している。SOUNDコンピュータは、この概念を数値演算を行う通常のコンピュータに適用したものである。すなわち、演算ユニットの機能および演算ユニット間の結合の仕方を指定することにより、処理手順を与えている。次に、このことについて、もう少し詳しく説明する。

SOUNDコンピュータは、沢山の演算ユニットを持っていて、それらをプログラムにより任意に結合し、処理環境であるデータフローグラフをそのままのトポロジーでハードウェア上に作り、そこに同期式にデータを流して処理を行う。命令を逐次与えて処理を行うのではなく、決められた結線の中にデータを流して、全体として一つの処理を行うのである。その結果、極めて簡単な構造、自然な並列性で高効率に処理ができる。同期式データフローグラフの例を図2に示す。この例では、8個の演算ユニットを繋ぎ合わせてパイプラインを構成し、 $\sin(x)$ を級数展開して並列計算している。ここで、Dは遅延素子を表している。

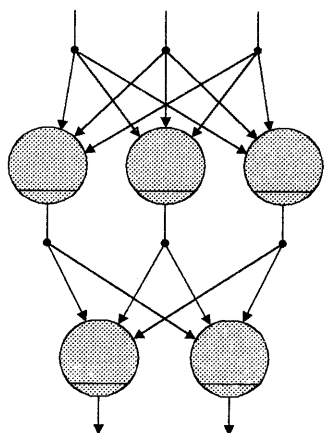


図1 ニューラルネットワーク

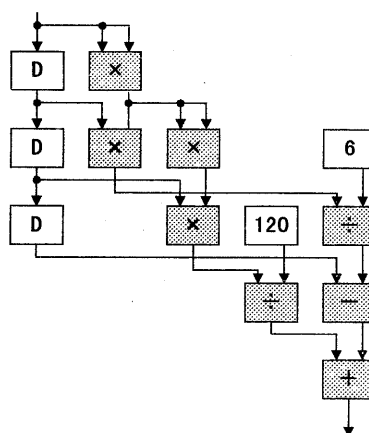


図2 同期式データフロー

### 3. アーキテクチャ

SOUNDコンピュータは、コネクションネットワーク、コネクションレジスタ、演算ユニットで構成されており、全てのユニットは、コネクションネットワークを通して繋がっている。その様子を図3に示す。そして、その動作は、フェッチフェイズで、データフローグラフをそのままのトポロジーでハードウェア上に実現し、データフローフェイズで、同期式データフローコンピュータとして並列処理を行う。その様子を図4に示す。本章では、上記のハードウェアによる任意のデータフローグラフの実現の仕方、および、データフローグラフにおける同期式並列処理の特徴について述べる。

データフローグラフをハードウェア上に実現するということは、コネクションネットワークを特別な結線に固定し、演算ユニットの機能を特定することにより、高い対称性を持つハードウェアの対称性を下げることである。その具体的な様子を次に示す。任意のデータフローグラフは、コネクションネットワークの結線、演算ユニットの機能の2つの情報で表現される。フェッチフェイズにおいて、SOUNDコンピュータは、この情報をコネクションレジスタに読み込む。すると、交換機であるコネクションネットワークは、この情報に

基づいて、ある結線を実現し、また、演算ユニットは、ある機能を実現する。その結果、任意のデータフローグラフがハードウェア上に形作られる。

データフローフェイズにおいて、SOUNDコンピュータは、データフローグラフに同期式にデータを流し、並列処理を行う。SOUNDコンピュータにおける同期式並列処理の特徴として、次の4つを挙げる事ができる。第一に、命令が流れず、データのみが流れる。そのため、データ転送にメモリのバンド幅をフルに使える。第二に、同期式であるため、制御が非常に簡単であり、制御のためのハードウェアが節減できる。第三に、ユニット間のデータ転送を、流れが一方方向で時間的に経路が変化しない専用線を通して行うため、転送待ちなく、進行波として連続してデータを送れる。このことは、半導体デバイスの動作周波数が上がってくるにつれて、大きな意味を持つてくる。これに対し、従来のバス型アーキテクチャは、データの流れが双方向の共有バスを時分割で用いているため、制御の高速化が大変であり、輻輳が起り易い。第四に、分岐を制御の流れでなく、データの流れるに負わせるため、条件分岐によりパイプラインが乱れることがない。

以上、ニューラルネットワークとの類似として展開してきたが、別の見方をしてみると、SOUNDコンピュータは、最小ロジックを演算ユニットにしたPGA (Programable Gate Array)、あるいは、命令を供給しないでデータだけで燃焼するパイプライン<sup>[7]</sup>と見做すこともできる。

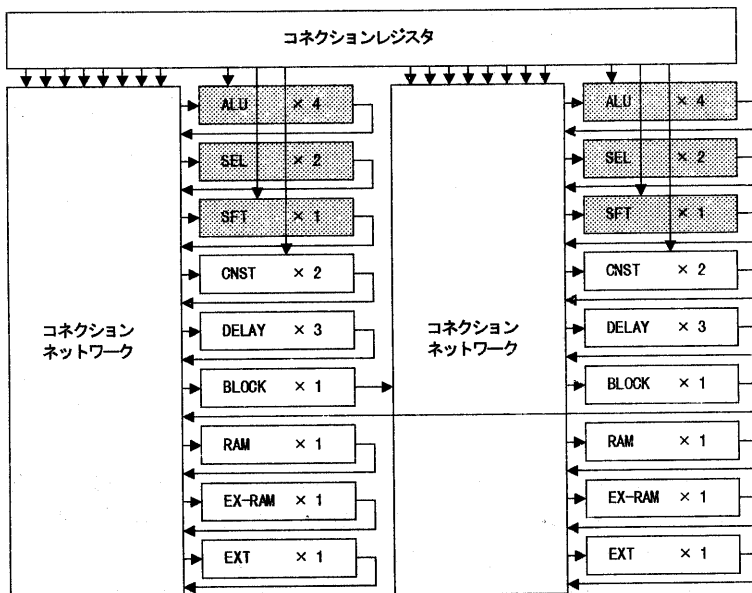


図3 SOUND試作機のアーキテクチャ

#### 4. 試作機

試作機は、ALU, SEL, SFT, CNST, DELAY, BLOCK, RAM, EX-RAM, EXTの9種類、計32個のユニットから成っている。コネクションレジスタの長さは174ビットであり、1バイト毎にアドレスが割り振っており、外部からデータフローグラフの情報をここに書き込むことができる。ユニットは2つのブロックに分けられており、ブロック内は完全結合、ブロック間は一対の経路で結線されている。また、できるだけ

多くの演算ユニットを配置するために、シリアルでデータ処理を行っている。ALUは論理演算／加減演算、SELは選択演算、SFTはビットシフト演算を行う。CNSTは定数の生成を行い、DELAYはデータを遅延させデータ間のタイミングを取る。そして、BLOCKはブロック間のデータ交換、RAMは無競合共有メモリ入出力、EX-RAMは外部メモリ入出力、EXTは外部入出力を行う。

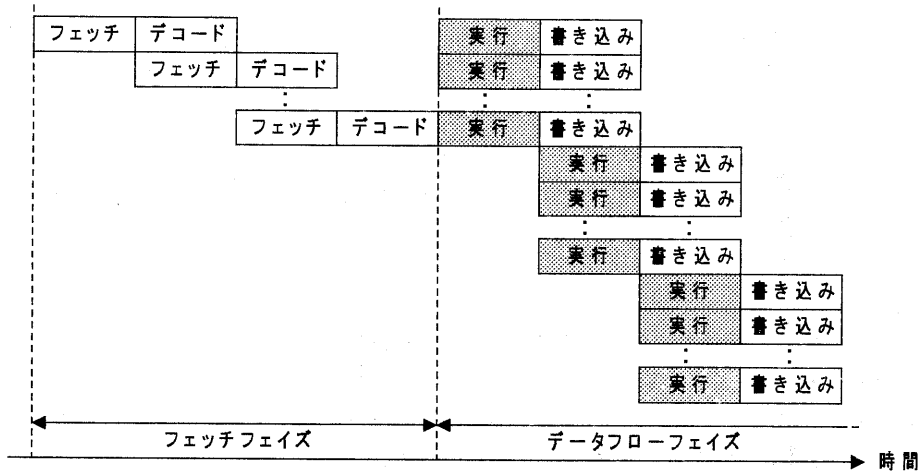


図4 SOUNDの動作

### 5. 試作機の評価

試作機の基本性能を表2に示す。試作機は、14個の演算ユニットを1MHzで動作させており、1秒間に14 M ビットの情報を処理できる。ここでは、マルチメディア処理の例として、式により表現されている情報を画像に展開してみる。

試作機の評価のために、乱数を用いて三角形の中心を逆三角形に切り抜いた形のフラクタルである有限 Sierpinski ガスケットを計算するプログラム<sup>[8]</sup>を実行させた。そのプログラムを図5に示す。プログラムでは左側半分で乱数を生成しており、右側半分で描画点の位置を計算している。ここで、XORのループは0, 1を交互に生成している。プログラムは、9個の演算ユニットがほぼフルタイムに並列動作しており、4000点の座標を64ミリ秒で計算している。これは、毎秒112万演算に相当している。

演算ユニットの数	14個
クロック	1MHz
処理能力	14 × 1MHz = 14Mビット/秒
回路規模	約5万ゲート

表2 SOUND 試作機の基本性能

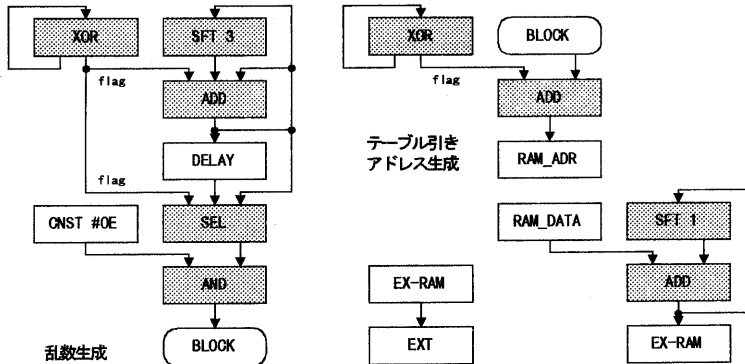


図5 フラクタル計算プログラム

## 6. 結論

我々は、簡単な構造で基礎のしっかりしたコンピュータを目指して、ニューラルネットワークにより近い非ノイマン型コンピュータであるSOUNDコンピュータの処理概念を提唱した。そして、SOUNDコンピュータの試作機を実際に製作し、マルチメディア処理のプログラムを走らせて評価を行った。その結果、我々が開発したSOUNDコンピュータは、対称性が高く、構造が簡単であるため、ハードウェア、ソフトウェアの開発が容易であり、自然な並列性で高効率に並列処理が行えることがわかった。

## 謝辞

本研究を進めるに当たりご指導、ご支援していただきました島津泰正氏に深く感謝いたします。

## 参考文献

- [1] Adams, D.A., "A computation model with data flow sequencing.", Computer Science Department, School of Humanities and Sciences, Stanford University, Technical Report CS-117, pp.130 (1968)
- [2] Rodrigues, J.E., "A graph model for parallel computation", Project MAC report Nos. ESL-R-398, MAC-TR-56, MIT (1969)
- [3] Dennis, J.B., and Fossen, J.B., "Data flow schemata", Lecture notes in Computer Science, 5, pp.187, Springer-Verlag (1969)
- [4] Dennis, J.B., "First version of data flow procedure language", Lecture notes in Computer Science, 19, pp.362, Springer-Verlag (1974)
- [5] Dennis, J.B., "Data flow supercomputers", Computer, Vol.13, No.11, 48 (1980)
- [6] Hassoun, M.H., "Fundamentals of artificial neural networks", MIT (1995)
- [7] Katahira, M., Shen, H., Kobayashi, H., Nakamura, T., "Jetpipeline: A Hybrid Pipeline Architecture for Instruction-Level Parallelism", Proceedings of High Performance Computing Conference '94, pp.317 (1994)
- [8] Gulick, D., "Encounters with chaos", McGraw-Hill (1992)