

## 大規模並列プロセッサを用いた論理シミュレーション

木村 功      安部 正秀      桑原 泰雄  
久保田和人      佐藤 政生      大附 辰夫

早稲田大学 理工学部

〒169 新宿区大久保3-4-1

あらまし

論理回路のゲート数に相当するプロセッサ数を持つ論理シミュレータおよびシミュレーション・アルゴリズムについて述べる。提案するシミュレータは、連想メモリの検索機能ならびに並列書き込み機能を用いてすべてのゲートの信号値を並列に計算し、連想メモリの各ワード間に付加されたネットワークによって並列に信号値の転送を行うものである。実験結果より、バスを用いてデータを転送する手法と比較して数十から数百倍の処理性能が得られることが示された。

## Logic Simulation on Massively Parallel Processors

Isao Kimra, Masahide Abe, Yasuo Kuwahara,  
Kazuto Kubota, Masao Sato and Tatsuo Ohtsuki

School of Science and Engineering, Waseda University  
3-4-1 Ohkubo, Shinjuku-ku, Tokyo 169, Japan

Abstract

A Logic Simulator which consists of Content Addressable Memory (CAM) and network is presented. The value of all logic gates are calculated by CAM in parallel and transferred through the network which connects among all the words of CAM. Experimental results show that our method runs tens to hundreds times faster than the method without the network.

## 1. はじめに

論理シミュレーションは、LSI、あるいはコンピュータ・システムを設計する上で必要不可欠な処理である。現在、広く行われている論理シミュレーションの方法は、大型計算機上でソフトウェア的に論理回路をシミュレートするものである。しかし、LSIの高集積化、コンピュータ・システムの大規模化に伴い、その処理時間の増加が切実な問題となっており、各機関で、論理シミュレーションを高速に処理する専用マシンの研究、開発が行われている[1,2]。

これらのマシンのうちの多くのは、プロセッサを数十から数百台使用した並列プロセッサをベースとした論理シミュレータである。シミュレーションの対象となる論理回路は、プロセッサの数に応じておおよかなサブ回路に分割され各プロセッサに割り付けられる。各プロセッサでは並列に論理シミュレーションが実行される。プロセッサを複数用いることで、その台数に応じた処理速度の向上が期待できるが、この場合プロセッサの台数は数十から数百でありそれ以上の処理速度の向上は望めない。また、一般にピーク性能を達成することは難しい。

論理シミュレーションは、本来並列に動作している論理回路をシミュレートするわけであるから、問題そのものは、ゲート数分の並列度を持つことになる。したがって、各ゲート毎にプロセッサを用意し、それらを並列に動作させれば、極めて並列度の高い論理シミュレーションが可能となる。また、LSI技術の進歩はこのような大規模な並列プロセッサの実現を可能にしている。以上の観点から、本稿では対象となる論理回路の規模（ゲート数）に相当するプロセッサを用いた論理シミュレーション手法（大規模並列論理シミュレーション）について述べていく。

これまで発表された大規模並列論理シミュレータには、Simlog社のSuperSim[3]ならびに、京都大学の連想メモリ(CAM)を用いた手法[4]などがある。SuperSimは、2入力1出力の論理ゲートを表現できるプロセッサ(図1.1)を2<sup>2</sup>台納めた専用VLSIを2<sup>13</sup>台接続し、全体で2<sup>20</sup>個のプロセッサを有している。これらのプロセッサは、

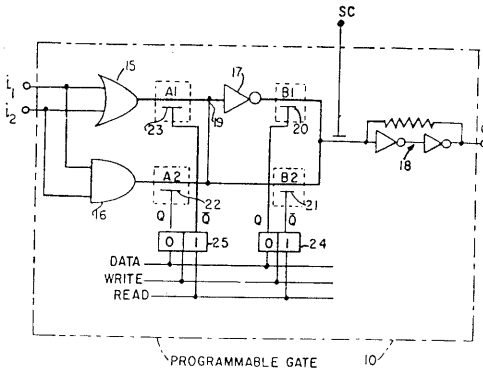


図1.1 Super Sim のプロセッサエレメント [3]

階層的なバスで結合されている。SuperSimでは、ゲートはすべて同時に評価され、結果は各ゲートの出力値が変化しないにかかわらず、その結果を入力とするゲートを表すプロセッサに転送される。すべてのゲートを評価し、すべてのデータを転送するこの方式はイグゾースティブ法と呼ばれる。京都大学から提案されている方法は、CAMをSIMD型の大規模並列プロセッサと見なし、CAMの各ワードに1つの論理ゲートを割当てて論理シミュレーションを行うというものである。CAMの検索機能を用いることにより、各ワードに格納されているゲートの評価を同時に行うことができる。ゲート評価後は、各ゲートでイベントが生じたかどうかを調べ、生じていたならば、そのゲートの出力を入力とするゲートを表すワードに信号値を書き込む。データの転送は、CAMのデータバスを通じて行われるので、CAMを用いた論理シミュレータもバス型のアーキテクチャに分類できる。この手法はゲートの評価については完全に並列であるが、データの転送すなわちイベントの伝搬処理は逐次処理となる。

本稿では、CAMの各ワード間にデータ転送用のネットワークが存在するアーキテクチャを仮定し、そのアーキテクチャを用いた論理シミュレーション手法の提案を行う。提案する手法は、ゲートの評価法としては、基本的に京都大学の方法を採用し、評価後の信号値データの転送は、専用のネットワークを用いてすべてのデータを転送するイグゾースティブ法である。

以下では、2章でCAMを用いたゲート評価法について、3、4章で提案する論理シミュレータのアーキテクチャおよびシミュレーションアルゴリズムを述べる。5章では、MCNCのベンチマーク回路で提案手法の評価を行う。

## 2. 連想メモリを利用したゲート評価法

### 2.1 連想メモリ

CAMは機能メモリ的一种であり、通常のRAMの機能の他に、データの検索機能を持つ。CAMの構成および基本的な機能を図2.1に示す。CAMは、記憶アレイ、インデックス・レジスタ、マスク・レジスタ、レスポンス・レジスタから成り立っている。検索命令により、インデックス・レジスタに格納されたデータ(マスク・レジスタで部分的にマスクされる)と一致するデータのレスポンス・レジスタに1が格納される。この他に、CAMの中には、レスポンス・レジスタで指定されたワードについて、並列にデータ(マスク可能)を書き込む機能を持つものがある[5]。

### 2.2 連想メモリを用いたゲート評価法

連想メモリの検索機能と並列書き込み機能を用いることにより、並列にゲートの評価を行うことができる。ここで述べるゲート評価法は基本的に京都大学で提案されたものと同じものである。図2.2に、複数の2入力NANDゲートを並列に評価する方法を示す。図2.2(a)に示されたNANDゲートの入力1,2および出力の信号値を

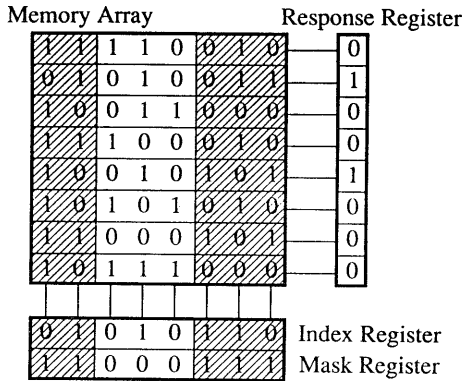


図2.1 CAMの構成と機能

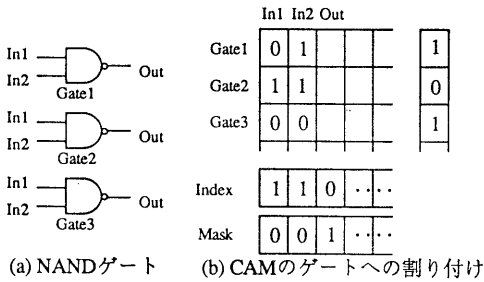


図2.2 CAMを利用した並列ゲート評価

格納するため、CAMのワードに入力1,2(In1,In2)および出力を表すビット(Out)を用意する(同図(b))。2入力NANDゲートの出力が0となるのは、入力1,2が両方も1のときである。CAMの検索命令を使って両方のビットが1であるワードを検索する。検索されたワードについて、出力ビットに0を書き込み、それ以外のワードについて1を書き込めば所望の結果が得られる。このゲート評価の方法は、SuperSimのゲート評価法が2入力1出力のゲートしか扱えないのに対して、多入力ゲート、さらにフリップ/フロップなども扱えるといった点で、柔軟性が高い。

### 3. データ転送ネットワークを持つ大規模論理シミュレータ

ここでは、論理シミュレーションを並列に行う場合、バスでプロセッサが接続されたアーキテクチャを持つシミュレータでは、データ転送がボトルネックとなることを示し、これを解消するようなアーキテクチャを提案する。なお、以下で扱う論理シミュレーションは、2値、ユニット遅延モデル[6]を採用することにする。

#### 3.1 共通バス型アーキテクチャのボトルネック

論理シミュレーションは図3.1に示すように、ゲートの評価と、評価後の信号値の転送からなる。CAMを用いるとゲートの評価が並列に行えることは2節で述べた。しかし、共通バス型のアーキテクチャはデータの通信路を1つしか持たないため、データの転送は逐次処

```
初期設定;
While(1){
  ゲートの評価;
  信号値の転送;
}
```

図3.1 論理シミュレーションの手順

理にならざるを得ない。ゲートの評価によって生じたイベントの数が転送されるべきデータ数であり、この数は最悪の場合ゲート数と等しくなる。したがって、扱う問題によっては、データ転送が処理のボトルネックとなり、高速なゲート評価処理が生かされないことになる。

#### 3.2 データ転送ネットワークの付加

ここで、データ転送を効率化するために、図3.2に示すようなアーキテクチャを持つシミュレータを考える。提案するシミュレータは、ゲート評価を行うCAMの部分とデータ転送を行うネットワークの部分からなる。CAM部は、2節で示したCAMと同等の機能を持つものとし、さらにCAM内のすべてのワードの特定のビットがすべて外部から並列に読み書き可能であるとする。図3.3(a)はCAMの各ワードの3ビット目を並列に読み出した場合であり、図3.3(b)は、CAMの2ビット目に外部から値を書き込んだ例である。ネットワーク部は、n本の入力n本の出力を持ち(nはCAMのワード数と一致するものとする)、入力と出力の間には任意の接続形態が実現できるものとする。

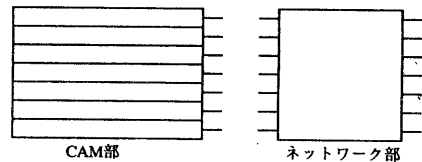
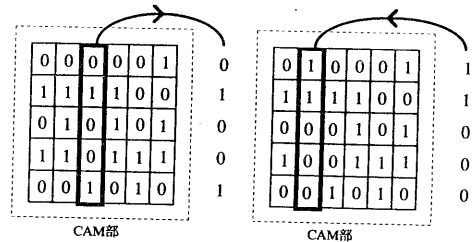


図3.2 提案する論理シミュレータ



(a)ビット並列読み出し (b)ビット並列書き込み

図3.3 CAMの並列読み出しと並列書き込み

#### 3.3 提案したシミュレータ上での論理シミュレーション

提案したシミュレータ上で図3.4の回路をシミュレーションする場合を考える。CAMの各ワードにゲートを割り当てる。CAMの各ビットは、各ゲートの信号値(0か1)を表すことになる。初期値は、図3.5(a)のようになっているとす。まず、各ゲートの評価を並列に行

う。手順は2節で示した通りである。結果は出力ビット(Out)に保持される(図3.5(b))。次に、評価したゲートの出力値をその出力を入力とするゲートへ転送する。これは、In1ビット、In2ビットへのそれぞれ2回の書き込みを用いて行う。まず、In1ビットについて考えると、ゲート1のOutはゲート5のIn1へ、ゲート3のOutはゲート6のIn1へ、ゲート5のOutはゲート7のIn1へそれぞれ転送されるべきであるから、図3.5(c)のようなデータ転送路をネットワーク上に作成する。各ゲートのOutを読み出し、読み出したデータをネットワークの入力へ入れる(図3.5(d))。ここで、ネットワークの出力に得られた結果をIn1ビットに書き戻せば、In1ビットについてデータ転送が全ゲート並列に行われたことになる(ゲート1から4の入力信号は外部入力であるため、信号値の変化は行わないようにする。この処理は、CAMの機能を使えば容易に実現できる)。同様の処理をIn2ビットについて行うと、すべてのデータ転送は完了する(図3.5(e))。ここで示した一連の処理は、1ユニット時間の論理シミュレーションであるが、その処理時間はゲート数に依存しない定数時間となる。また、ネットワークのトポロジは処理の過程で動的に変化することはなく、入力信号の本数分だけのボタンを用意すれば良い。

### 3. 4 ネットワークの構成法

提案したシミュレータでは、入出力間に任意の接続形態が得られるネットワークを使用することを前提としている。要求を満たすネットワークを構成する原始的な方法としては、クロスバスイッチを用いる方法がある。しかし、クロスバスイッチのスイッチ数は入出力線の本数を $n$ とすると $n^2$ であり、 $n$ が大規模になった場合ハードウェア量が現実的でなくなる。

要求を満たすネットワークの構築法としては多段接続ネットワークを用いる方法がある[7](多段ネットワークをプロセッサ間の結合に使用したシミュレータとしてはNECのHALがある[8])。これは、2入力2出力のスイッチング・エレメント(図3.6:入力データがそのまま出力される場合と、反転されて出力される場合がある)を並べたものである。多段接続ネットワークの1つであるバイトニック・ソータ(図3.7)を利用すれば、 $n/4(\log n)(\log n + 1)$ 個のスイッチング・エレメントで、所望のネットワークを構築することができる。したがって、ネットワークによる遅延を考えないとすれば、 $O(n \log^2 n)$ のハードウェアを用いれば、ゲート数によらない定数時間で、1ユニット時間のシミュレーションが行えることになる。

### 4. ネットワークの階層化

前節では、 $O(n \log n)$ 個のハードウェアを用いれば、ゲートの数によらず一定時間で、1ユニット遅延相当の論理シミュレーションが行えることを述べた。しかし、提案したシミュレータをいくつかのLSIチップに分けて実現する場合は、LSIの持つI/Oピンの数の制限によ

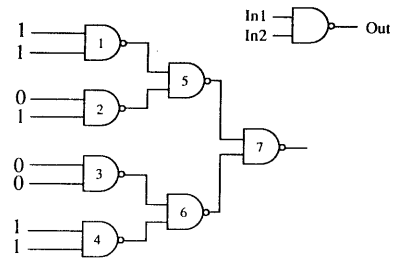
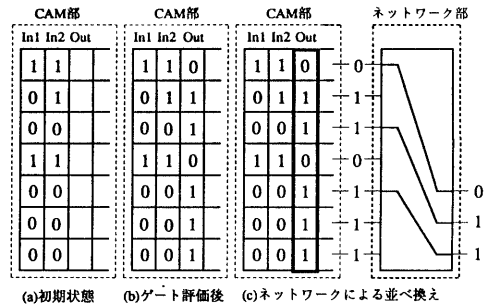
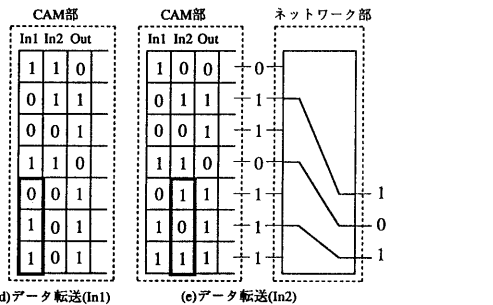


図3.4 対象となる回路



(a)初期状態 (b)ゲート評価後 (c)ネットワークによる並べ換え



(d)データ転送(In1) (e)データ転送(In2)

図3.5 提案したシミュレータ上でのシミュレーション

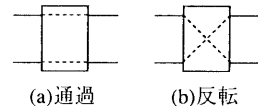


図3.6 スwitching・エレメント

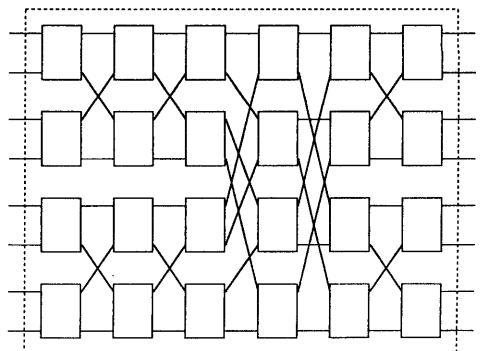


図3.7 バイトニックソータ

て実装が困難であることが予想される。使用するCAMは同一ビットに対する全ワードの同時読み出し、同時書き込みが可能であるとしている。仮に、1024ワードを持つCAMチップの場合は、最低1024本のI/Oピンが必要となる。現在、使われているLSIのパッケージのI/Oピンの数は多くて400本程度であるので、このようなCAMを実現することは難しい。同様なI/Oピン不足は、ネットワーク部分の実現についてもあてはまる。また、ネットワークのハードウェア量はかなり大きなもので、シミュレータの性能を損なうことなくハードウェア量を減らせば、それに越したことはない。

ここでは、このI/Oピン不足およびネットワークのハードウェア量削減を目的として、階層的なネットワーク構造の採用について考えてみる。なお、階層的なネットワークを採用した論理シミュレータの例としては富士通のSP[9]などがある。

階層的なネットワークを採用した、シミュレータを図4.1に示す。各段におけるスイッチは $n+m$ から $n+m$ への任意の接続形態がとれるスイッチとし、 $n$ は下位の階層のスイッチに接続され、 $m$ 本は上位の階層のスイッチに接続されているものとする。このような、シミュレータをいくつかのLSIを用いて実現する場合は、例えば、CAM部分およびCAMに接続されているスイッチをまとめて1つのLSIとすれば（図4.1の波線で囲まれた部分を1つのチップとして実現する）I/Oピン不足の問題は解決できる。

このようなネットワーク構造では、近傍にあるCAMのワード間は結合度が高く、遠いほど弱くなる。したがって、論理シミュレーションを行う場合、遠いワード間のデータ転送が頻発した場合は、著しくシミュレータの性能が低下することが予想される。しかし、論理回路そのものは、ゲートの集まったゲート群、ゲート群の集まったブロック、ブロックの集まったより大きなブロックと階層的に設計されている場合が多いので、このゲート間の階層構造をネットワークの階層構

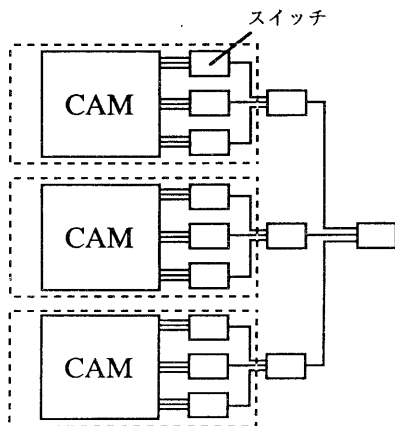


図4.1 階層ネットワーク

造にうまくあてはめることができれば、データ転送による性能の低下をそれほど招かずに済むと考えられる。

階層的なネットワークを採用した場合には、用意しなければいけないネットワークのボタンが、入力信号の本数に限定されない可能性が生じる。提案する方式ではあらかじめスイッチング・エレメントにいくつかのボタンを記憶させておかなければならない。したがって、もし与えられた論理回路に対する接続ボタンが、各スイッチング・エレメントが記憶できるボタン数を越えた場合は、データ転送が行えないゲートが生じることになる。この場合は、残った信号値を逐次的に転送する必要がある。

## 5. 計算機実験によるシミュレータの評価

提案したアーキテクチャを持つ論理シミュレータの性能を、シミュレーション実験で評価した。実験にはMCNCのISCAS85のベンチマーク回路を用いた。これらはすべて組み合わせ回路である。最大のゲート数は3719ゲートであるので、CAMのワード数は4096ワードとした。また、テストデータの中には、9入力や8入力のゲートが存在するが、これらはそのままの形でCAMで評価しており、入力数の小さい論理ゲートに分解することはしていない。各論理回路のゲート評価に要するクロック数、および、さまざまなネットワークの形態をとった場合の、データ転送に要するクロック数を表1に示す。なお、ここでは、各スイッチが記憶することのできる経路接続情報の数には上限がないものとしている。表中の4096とは、ネットワークを1階層で実現した場合、256・32とは、下位に256本、上位に32本の接続をもつスイッチで階層ネットワークを構築した場合のデータ転送に要するクロック数である。バス方式に関しては、イベントの生じたゲート信号のみデータ転送を行った場合のクロック数を測定した。表に示してある値は、10ユニット遅延分のシミュレーションを行った際の平均のイベント値で計算してある。各ゲートの初期値はイベントの発生に大きく影響を与えると考えられるが、ここではすべて初期値0からシミュレーションを開始した。結果を見ると、バス方式ではゲートの評価に要するクロック数に比べて、データ転送にかなりのクロック数を要していることがわかる。C6288のように極端にイベントが発生してしまう回路では、その傾向は顕著である。階層ネットワーク方式では、ゲート評価とデータ転送に要するクロック数はほぼ対等なものとなっている。バス方式と比較すると、一桁から二桁程度、高速にデータを転送できることがわかる。また、データ数が増加しても、データの転送に要するクロック数はそれほど増加していない。

各方式におけるハードウェアの量（スイッチング・エレメントの数）を見ると、4096ワードのCAMに対して約70,000~160,000のスイッチング・エレメントを必要としている（表2）。使用するCAMのワード長を32ビットとすればCAMセルの個数は $4096 \times 32 = 13,1072$ とな

表1 実験結果

回路名	C17	C880	C1355	C1908	C2670	C3540	C5315	C6288	C7552	
ゲート種	1	9	7	13	12	15	15	3	14	
ゲート数	6	383	546	880	1193	1669	2307	2416	3719	
最大入力数	2	4	5	8	5	8	9	2	5	
ゲート評価 (クロック)	24	33	33	39	45	45	48	27	45	
データ転送 (クロック)	4096	8	16	20	32	20	32	36	8	20
	256-32	8	34	38	110	74	104	234	128	260
	128-16	8	64	68	116	188	278	372	188	446
	64-8	8	88	164	308	290	386	654	332	668
	バス方式 (イベント数)	140 (6.67)	3,780 (114.6)	8,557 (223.4)	22,908 (401.9)	11,856 (304)	28,226 (495.2)	37,019 (587.6)	44,062 (2098.2)	45,010 (1154.1)

表2 ハードウェア量

ネット種	ハードウェア量
4096	159,774
256-32	123,120
128-16	95,904
64-8	73,584

り、スイッチング・エレメントの個数と同程度となる。しかし、各スイッチング・エレメントには径路接続情報を記憶するためのRAMを用意しなければならないため、実際のスイッチのハードウェア量は、CAMのハードウェア量をかなり上回る。したがって、ハードウェア量の制限から記憶できる径路接続情報の数に上限が与えられた場合は、なるべく少ないパタン数でデータ転送が行えるような各ゲートのCAMのワードへの割付け方法を考慮したり、8ないし9入力といったゲートへのデータ転送は、バスを用いて行うなどの工夫が必要であらう。

## 6. おわりに

本稿では、すべての論理ゲートを同時に評価し、データ転送も並列に行う論理シミュレータの提案およびその評価を行った。その結果、データ転送において、バスを用いて逐次的にイベントを伝搬する方法と比べ、一桁から二桁ほど処理が高速化されることがわかった。現在は、スイッチのパタン数に上限を設けていないが、上限を設けた場合にデータ転送の効率がどれくらい悪化するかを測定する必要がある。また、なるべく少ない径路パタンでデータ転送が行えるようなゲートのCAMのワードへの割付け方についても考察する必要がある。また、標準遅延や多値論理への対応についても考察して行く必要がある。

謝辞 本研究は財団法人セコム科学技術振興財団の助成のもとに行われたものである。

## 文 献

- [1]T.Blank:"A Survey of Hardware Accelerators Used in Computer-Aided Design",IEEE Design & Test of Computers,Vol.1,No3,pp.21-39(1984).
- [2]小池:"C A Dマシ",オーム社(1989).
- [3]Yoav Lavi:"Hardware Logic Simulator",United States Patent,No.4,pp.466-472,1990.
- [4]安浦,渡辺,左達,田丸:"機能メモリ型並列プロセッサ F M P P 上での論理シミュレーション",信学技報,CPSY90-94,pp.41-48(1990).
- [5]小倉,山田,丹野,石川:"4 k b C M O S 連想メモリ L S I",信学技報,SSD83-78,PP.45-52(1983).
- [6]村井:"ゲートレベル論理シミュレーション",情報処理,VOL.22,No.8,pp.762-769(1981).
- [7]Howard Jay Siegel:"Interconnection Networks for Large-Scale Parallel Processing",Lexington Books(1985).
- [8]小池,大森,佐々木:"論理シミュレーションマシンのハードウェア構成",情報処理学会論文誌,Vol.25,No5,pp.873-881(1984).
- [9]山田,広瀬,新妻,進藤:"シミュレーションプロセッサ S P",電子情報通信学会誌D,Vol.J71-D,No4,pp.644-651(1988).