

## 解説



## 2. 専用 VLSI プロセッサの具体例

2.2 VLSI 論理シミュレーション  
プロセッサ†

広瀬 文保† 山田 博†

## 1. はじめに

論理シミュレータは、デジタル装置の設計を論理的にモデル化し、その動作を模擬するものである。これにより、装置製造前に装置の動作を確認して設計の正しさを検証する。しかし、汎用コンピュータ上にソフトウェアとして実現されたシミュレータでは、シミュレーション速度は、対象が大規模な場合、実機に比して1億倍遅いといわれている<sup>1)</sup>。したがって、設計を十分に検証するための計算量は膨大となる。そこで、論理シミュレーションを超高速に実行する専用ハードウェアが開発・実用化された<sup>2)</sup>。このうち、大容量・超高速のマシンでは、大形コンピュータ上のソフトウェア・シミュレータの1000~10000倍の演算性能を達成しており、大規模装置開発の期間短縮やコストダウンに貢献している。このようなクラスの実用マシンの代表的な例を表-1にまとめる。

これらのマシンは、大容量のメモリを多数使用している。また、たとえば HAL II<sup>3)</sup>ではプロセッサ間通信ネットワークのための LSI が開発されており、SP<sup>4),5)</sup>ではメモリのビット誤り訂正用 LSI を使用している。しかし、本特集のテーマである専用 VLSI プロセッサとしては実現されていない。EVE<sup>6)</sup>や SDE<sup>7)</sup>についても同様のことが推測される。この理由としては、マシンが必要とするロジックについては集積度が比較的少なくて済み、VLSI 化しなくとも大容量メモリと汎用 LSI を用いて実現できたことがあげられる。

しかし、今後一層の高速化を図っていくためには、現アーキテクチャの延長では限界があり、専用 VLSI プロセッサ技術などを用いたブレイクスルーが必要となる。また、論理シミュレーション以外の処理にも現

アーキテクチャの高速性を活かしてゆく柔軟性を備えることも今後の展開方向といえる。

本文では、このような観点から専用 VLSI プロセッサ向きアーキテクチャの方向を位置づける。次に、具体例として、シミュレーション対象の1ゲートを1プロセッサに割りつける専用超並列プロセッサ SuperSim<sup>8)</sup> と、機能分割したシミュレーションの処理のおのおのを実行するハードウェアを従来のように専用化せず、1種類のプログラマブルな VLSI プロセッサとして実現したマシン MARS<sup>9)</sup> を紹介する。また、超並列型アーキテクチャに関連して、プログラマブルゲートアレイの一種である LCA<sup>10)</sup> を紹介する。

2. VLSI 化論理シミュレーション専用  
マシンの技術動向

現在主流を占める、専用 VLSI プロセッサ化していないマシンでは、クロック速度については大形コンピュータより1桁遅いが、プロセッサ1台によるシミュレーション速度は、大形コンピュータ上でのソフトウェア・シミュレータより1桁以上高速である。つまり、これらのマシンのねらいは、シミュレーションの処理を最小のクロック数(1~2クロック)で実行するような専用ハードウェア化にあるといえる。そのための技術は、一般化して述べれば以下の三つに分解できる。

(1) ローカルメモリへのデータ構造の分割によるデータアクセスの独立化。

(2) アクセスしたデータの加工時間の専用ハードウェア化による短縮化と均一化。

(3) ローカルメモリ・アクセスとデータ加工の組をパイプライン状に結合することによる高速化。

このような高速なプロセッサを数10台から数100台用意し、おのおのに回路の部分の部分を割りつけて担当させ、その動作を並列にシミュレーションすることにより、さらに1桁の高速化を達成している。プロセッサ

† Dedicated VLSI Processors for Logic Simulation by Fumiyasu HIROSE and Hiroshi YAMADA (Fujitsu Laboratories LTD).  
† (株)富士通研究所

表-1 論理シミュレーション専用マシンの例

マシン名 企業名		EVE <sup>1)</sup> IBM	HAL II <sup>1)</sup> 日電	SDE-8032 <sup>2)</sup> ZYCAD	SuperSim400 <sup>3)</sup> Simulog	SP <sup>4)</sup> 富士通
アルゴリズム		レベルソート (イグズォースティブ)	レベルソート+ イベント駆動	イベント駆動	イグズォースティブ	イベント駆動
速度		$2.2 \times 10^4$ ゲート評価/秒	$1.9 \times 10^4$ ブロックイベント/秒	$1.0 \times 10^4$ イベント/秒	1~10 kHz	$8.2 \times 10^4$ アクティブ 素子評価/秒
容量	ロジック	$1.7 \times 10^4$ ゲート	$3.0 \times 10^4$ ブロック	$1.1 \times 10^4$ ゲート	$1.0 \times 10^7$ ゲート	$4.2 \times 10^4$ 素子
	メモリ	50 MB	4 MB	64 MB	160 MB	32 MB
演算 単位	入出力数	4 入力 1 出力	128 入力 128 出力	3 入力 1 出力	2 入力 1 出力	4 入力 1 出力
	信号値数	4 値	2 値	3 値	2 値	16 値
ディレーモデル		zero, (unit)	zero	unit, zero	unit	unit, zero
プロセッサ数		256 台	32 台	256 台	$1.0 \times 10^7$ 台	64 台
ネットワーク		クロスバ・スイッチ	多段接続 ネットワーク	階層的 共通バス	階層的 共通バス	階層的 クロスバスイッチ

1 台の担当する回路の規模は、4K~64Kゲート（ないしは評価単位）となっている。回路をモデル化するためのメモリ容量については、マシンにも依存するが、だいたい1ゲート（ないしは1評価単位）当たり数10バイトのメモリ容量が必要であると推測される。したがって、1プロセッサ構築のためのメモリ量は数10K~数Mバイトは必要であるといえる。また、ローカルメモリの数も数個~10数個に及ぶ。これをそのまま1個の専用VLSIプロセッサ・チップとして実現することは、数年前のマシン開発当時のテクノロジーでは「メモリ容量不足」であった。また、ローカルメモリを外部に置くとすれば、「IOピン不足」となる。今後、1プロセッサ当たり数KゲートとしたVLSI論理シミュレーションプロセッサの登場が十分期待できるが、いまだ実現された例はない。

一方、本文で具体例として紹介するマシンでは、VLSIプロセッサ向きアーキテクチャにより、1チップ当たり数万トランジスタ程度の集積度で、VLSI論理シミュレーションプロセッサを実現している。この解説は、シミュレーションのアルゴリズムと切り離しては考えられないので、本章では、まず、三つの代表的なシミュレーション・アルゴリズムを説明し、次にVLSIプロセッサ向きアーキテクチャについて解説する。アルゴリズムについてすでにご存じの読者は2.2にとんでいただきたい。

### 2.1 アルゴリズム

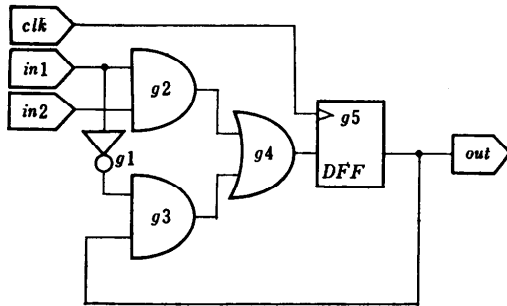
論理シミュレーション専用マシンが採用しているアルゴリズムは、レベルソート法、イグズォースティブ法、イベント法に分類できる。これらのアルゴリズム

の違いはゲートの評価順序にある。レベルソート法とイグズォースティブ法では、評価順序はシミュレーション実行前にあらかじめ決定されている。シミュレーション実行時にはこの静的な順序に従って評価を行う。一方、イベント法では、ゲートの入力に変化（イベント）が発生した場合にそのゲートの評価を動的にスケジュールする。そのイベントがゲートの出力側へ伝播することがゲートの評価により判明した場合にはさらにその先にあるゲートがスケジュールされる。

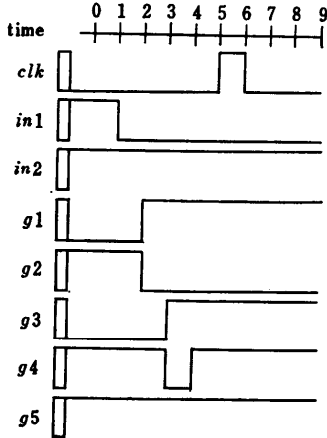
レベルソート法における評価順序の決定は、「ゲートgの評価は、gの入力側にあるゲートの評価を終了した後に進行」というルールの下に行う。レベルソート法は、同期回路の組合せ回路部に関して、回路安定時の最終値を計算することを目的とし、1クロックの間に正確に1回評価する。ハザードなどの途中経過には関知しない。

一方、イグズォースティブ法においては、信号値の変化が入力から出力に伝わる時間を均一に1ユニット時間とし、その1単位時間間にすべてのゲートを正確に1回評価する。このため、レジスタ間のロジックの最大段数をNとしたとき、レベルソート法に比して評価量はN倍と多くなるが、評価の順序に制約はなく、並列処理に際しては、全ゲートを同時に実行してもよい。また、機能的にも、非同期回路のシミュレーションが可能である。

図-1に、同期回路の例について、動作と評価順序を示した。また、各アルゴリズムの比較を表-2にまとめた。なお、シミュレータによっては、このうち、いくつかのアルゴリズムを併用しているものもある。



(a) 同期回路の例



(b) 動作例

時刻	コンパイル法	イグズォースタイプ法	イベント法
1	(in 1, in 2)	全ゲート	in 1
2	→ (g 1, g 2)	全ゲート	g 1, g 2
3	→ (g 3)	全ゲート	g 3, g 4
4	→ (g 4)	全ゲート	g 4, g 5
5	g 5	全ゲート	clk, g 5
6	(in 1, in 2)	全ゲート	clk, g 5
7	→ (g 1, g 2)	全ゲート	g 5
8	→ (g 3)	全ゲート	なし
9	→ (g 4)	全ゲート	なし

(c) 評価順序

図-1 回路動作例と各種アルゴリズムの評価順序

2.2 アーキテクチャ

現在多数派を占めるマシンが専用 VLSI プロセッサ化されない理由が「メモリ不足」ないしは「IO ピン不足」であることはすでに述べた。これらの問題点を解消する方策としての VLSI プロセッサ向きのアーキテクチャを、図-2 に示すように二つの方向に位置づ

表-2 アルゴリズムの比較

	イベント法	レベルソート法	イグズォースタイプ法
ゲートの評価順序	動的にスケジュール	固定的にスケジュール	
1クロック当たりの評価に必要な計算量	15*1	100	3000*2
シミュレーション対象回路	同期, 非同期	同期	同期, 非同期

\*1: イベント率は Blank の比較係数 (文献 2)) に準じて 15% で計算

\*2: レジスタ間の最大ゲート段数を 30 と仮定

けることができる。第1の方向は、メモリ不足を解決するもので、プロセッサ当たりのゲート数を小さくして VLSI に封じこめるものである。この方向の極限状態に位置するのが、1プロセッサの処理する容量を最小数の1ゲートとした超並列型のマシンである。超並列型のアーキテクチャでは、従来のアーキテクチャの高速性のよりどころであるパイプライン処理を使用しない。そのかわり、同時に動作可能なプロセッサ数を最大化することにより高速性を引き出す。超並列型を実現するためのキーポイントは、プロセッサの実行順序の制御法とプロセッサ間の通信方法である。Super-Sim では、アルゴリズムとして、イグズォースタイプ法を採用することより、全ゲートを同時評価可能として実行順序を単純化するとともに、プロセッサ間通信についても、通信制御を単純化した。ネットワークとしては階層型の共通バスを使用し、この上にゲートの接続関係をマッピングすることによりあるゲート(プロセッサ)の出力が接続するゲート(プロセッサ)の入力ヘータを転送する。通信コンフリクトのため同時に転送することが不可能な場合は、時分割的に階層バスを使用して転送を行う。この通信手順はシミュレーション実行前にあらかじめスケジュールされ、1ユニット時刻ごとに繰り返し実行される。

第2の方向は、ローカルメモリをチップの外におくことにより生じる IO ピン不足を解消するものである。このアプローチでは、ローカルメモリを制御するハードウェアを従来のように個別に専用ハードウェア化せず、1種類の標準的な VLSI プロセッサとして実現する。すなわち、ゲートの評価、イベントのスケジュール、ファンアウトゲートの取り出しといったシミュレーションに関する機能の分割方法、および、分割した機能のパイプライン処理による高速化の方法は従来どおりだが、おのおのの機能を遂行するハードを一種のユニバーサルな VLSI プロセッサとして実現す

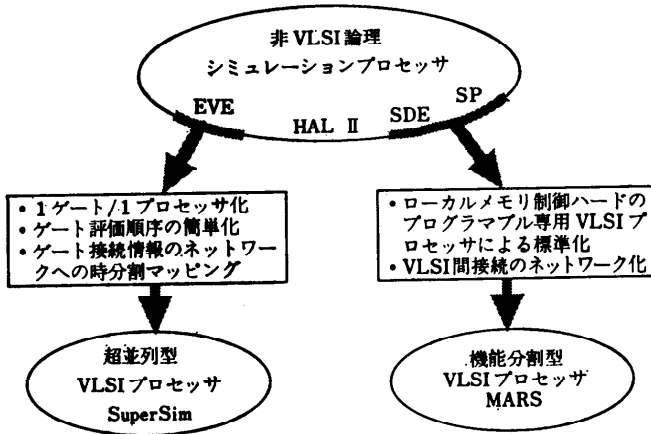


図-2 VLSI 化プロセッサへのアプローチ

る。この機能分割型を実現する際のキーポイントは、従来の専用ハードウェア化されたものに比して、性能をあまり落とさないような VLSI プロセッサの作り方である。MARS では、水平形マイクロ命令形式を採用したプロセッサアーキテクチャをベースに、リストやテーブルのアクセスやビット挿入・抽出を容易にする機能を専用化している。

### 3. VLSI 論理シミュレーションプロセッサ

この章では、超並列型の具体例として SuperSim を、また、超並列型に関連する VLSI として、LCA (Logic Cell Array)<sup>10)</sup> というチップを紹介する。機能分割型の具体例としては、MARS を紹介する。

#### 3.1 VLSI プロセッサの具体例

SuperSim<sup>9)</sup> のプロセッサの構造を図-3 に示す。プ

ロセッサは、プログラマブルゲートと呼ばれる。シミュレーション実行前にレジスタ  $r1$  と  $r2$  に設定 (プログラム) する値により AND, NAND, OR, NOR の機能を実現する。ある時刻の入力値  $i1$  と  $i2$  に対してゲート出力を計算させる。その値は、クロック信号 SC によりパストランジスタ  $pt4$  を開いて出力レジスタ (latch) に蓄えられる。

ネットワーク構造は図-4 に示すように、階層型の共通バスになっている。図中灰色の四角で示したものがシミュレーション対象のゲートの入力値と出力値を保持するレジスタに対応する。現時刻の

回路の状態はプログラマブルゲート (プロセッサ) の出力側のレジスタに保持されている。これをプログラマブルゲートの入力側にあるインプットセクタのレジスタに転送するために、回路の接続状態に準じて階層バスを切り換える。この切り換え方はシミュレーション実行前にスケジュールされ、制御用メモリに蓄えられる。すなわち、任意の回路構造を階層型共通バス上に、時分割形式でマッピングする手順が制御用メモリ (図には示されていない) に蓄えられる。1 ユニット時刻のシミュレーションに要するクロック数は、図-3 のクロック SC を打つための 1 回と、データ転送のためのクロック数を足したものである。

このクロック数が回路規模に依存せず一定な場合は、シミュレーションは回路規模の増大とは無関係に一定の速度で実行することができるため、規模が増大

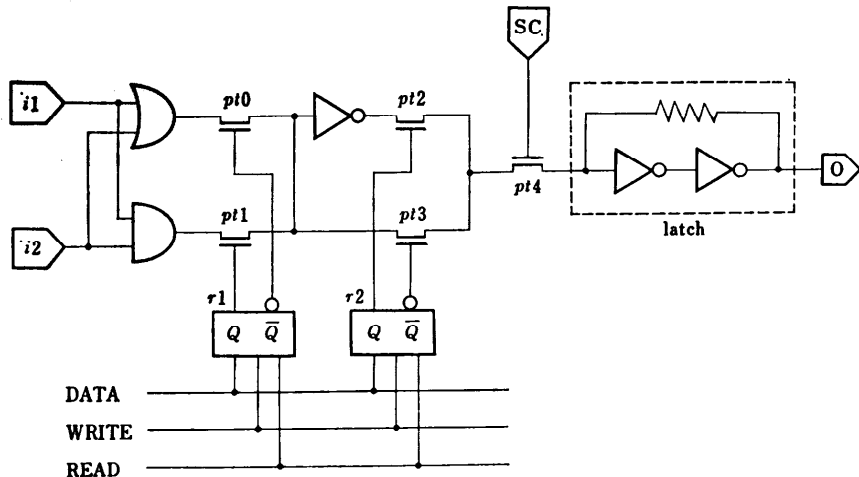


図-3 SuperSim のプログラマブルゲート (文献 8) より引用)

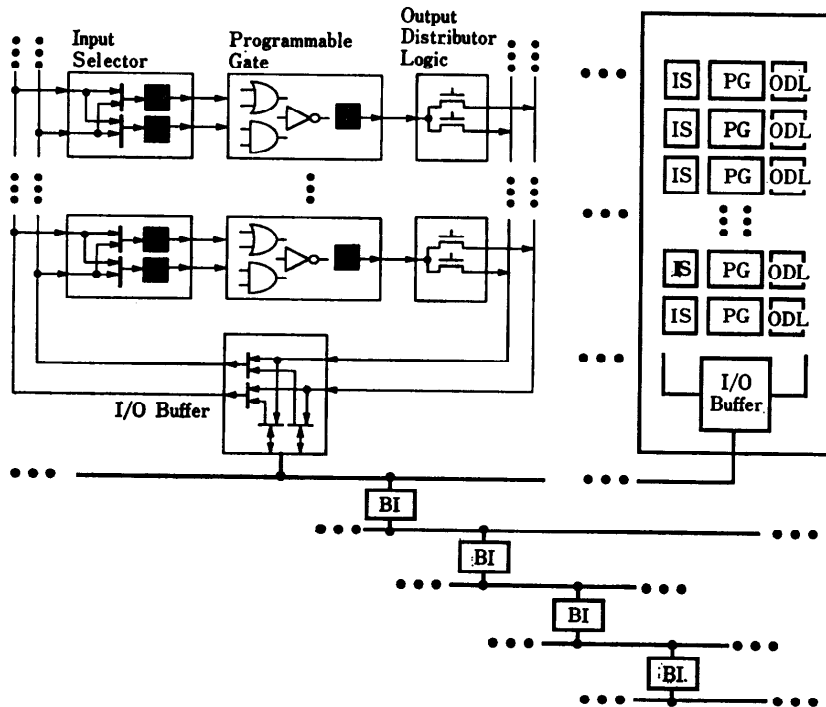


図-4 SuperSim のネットワーク

するほど他のアーキテクチャのマシンに比して相対的に高速になる。SuperSim では、回路規模に依存せず、1 k~10 kHz の速度でシミュレーションが実行可能としている (表-1 も参照)。このマシンは超並列アーキテクチャにより実際に超高速性を達成して示した例としても評価できると考えられる。

図-5 に SuperSim を構成するフルカスタム・チップ GC 2600 の写真を載せる。GC 2600 は 2  $\mu$ m, ダブルメタルの CMOS の製造技術を使用している。1チップ当たり 64 組の Programmable gate, Input selector, Output distributor logic とそれに関連する I/O Buffer, Bus interface (BI) を搭載しており、集積度は約 82,000 トランジスタである。チップのクロックは 10 MHz である。5V の電源で動作し、消費電流は約 30 ミリアンペアである。図-6 は GC 2600 を 128 チップ搭載したボードの写真である。8,192 プログラマブルゲートがこのボードに搭載されている。

LCA<sup>11)</sup> は、本来、ゲートアレイと PLD (Programmable logic device) のギャップを埋めることを目的として開発されたプログラマブルな VLSI である。LCA を用いて論理シミュレーションマシンを構成した例は現状ではないが、以下に述べるように、超並列

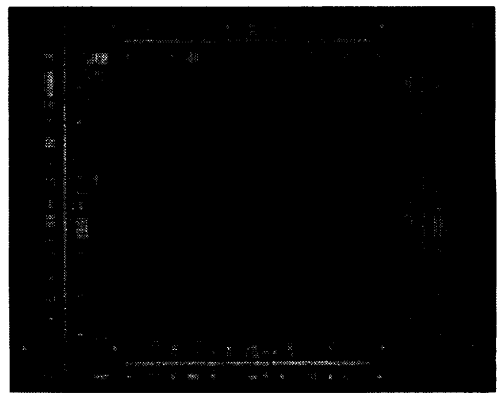


図-5 SuperSim のチップ GC 2600  
(Simulog 社の好意による)

型のアーキテクチャとの関連性が非常に高いので紹介する。LCA では、論理ブロックを設計単位とし、ブロック内論理やブロック間配線を SRAM セルに書いた情報によりプログラムする。論理ブロックは、組合せ回路を定義するための SRAM セルと出力を保持するための FF (フリップフロップ) からなる。FF を使用するかどうかは選択できる。組合せ回路としては、4 入力 1 出力論理、2 組の 3 入力 1 出力などの論理を

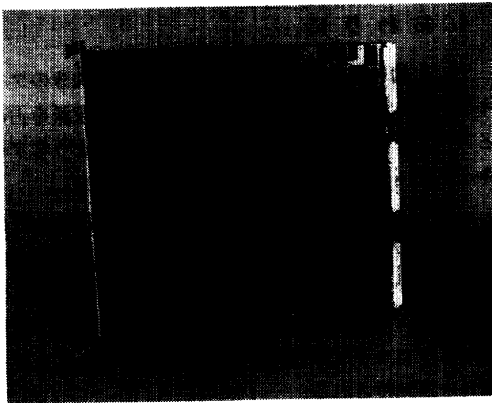


図-6 GC2600 を128 チップ搭載したボード (Simulog 社の好意による)

設計者が定義できる。設計者は論理ブロックを接続して回路図を作成する。この接続に応じて、論理ブロック間をチップ上で配線するが、この配線経路も RAM セルでプログラムできる。チップ上には  $m$  行  $n$  列の論理ブロックが配置されており、ブロックの間を配線チャンネルが縦横に走っている。縦と横の配線チャンネルの交差点にスイッチ・マトリクスというものがあり、マトリクス内の接続の方法を SRAM のセルに値を設定することによりプログラムできる。XC-3090 というチップでは、320 個の論理ブロックを搭載しており、最大トグル周波数は 70 MHz であるとしている。

MARS<sup>9)</sup>における標準化 VLSI は PE と呼ばれる。

PE とローカルメモリの組を 14 対用意し、それをクロスバススイッチにより接続して、論理シミュレーションを実行する 1 台のプロセッサ(クラスタ)を構成する。1 クラスタがシミュレーションできる容量は 64K ゲートである。文献 9) では、クラスタ 1 台を試作したと報告されている。最大仕様は、256 クラスタである。

図-7 に PE のブロック図を示す。マイクロ命令はどれも 3 フェーズで実行される。フェーズ 1 では、データが Register array のレジスタから内部 Bus A, B, C に読みだされる。フェーズ 2 では、二つのファクシオンユニット AAU (Address Arithmetic Unit) と FOU (Field Operation Unit) がバス上のデータについて演算を行う。AAU ではバス A と C 上のデータについて、算術演算、論理演算、シフト演算を施し、結果をバス C へ返す。FOU では、二つのデータから指定したビットフィールドを抽出し、ビット演算を行い、結果をバス A の値に挿入する。フェーズ 3 でその結果がバスに読み出され、それは Register array のレジスタに書き込まれる。水平形マイクロ命令は 64 ビット語長で、Microprogram RAM に格納されている。通常、2~3 の独立した命令が 1 語の水平形マイクロ命令で実行できるとしている。なお、ローカルメモリへのアクセスは、External RAM interface を介して行われる。また、PE 間のデータ通信は、Queue Unit を介して行われる。

イベント法を機能分割してパイプライン処理するた

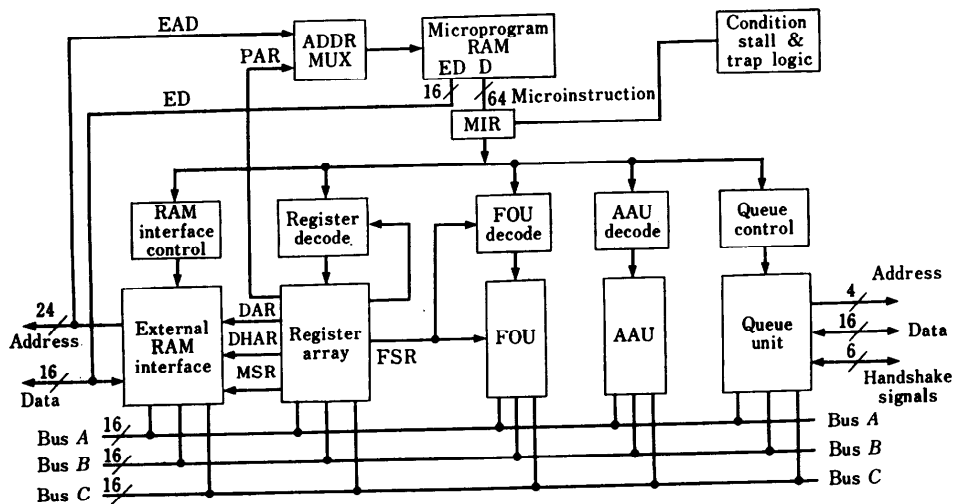


図-7 MARS の PE のブロック図 (文献 9) より引用)

Reprinted with permission from the October 1987 issue of IEEE DESIGN AND TEST OF COMPUTERS, pp. 31-32, copyright 1987 (c) by the Institute of Electrical and Electronics Engineers, Inc.

めに一つの PE が担当する機能には、たとえば、イベント・スケジューリング、ファンアウト先のゲート入力値の更新、ゲート評価などがある。機能分割の詳細に関しては、文献9)を参照されたい。装置のクロックサイクルが 10 MHz、目標性能が 1 Mゲート評価毎秒であることから、1ゲートの処理に平均 10 クロックを要すると推測される。MARS では、PE の機能がプログラムできることと、それらの接続もクロスバ・スイッチで自由に構成されていることにより、論理シミュレーション以外の応用にも柔軟に対応することができる。

PE チップは、1.25  $\mu\text{m}$ 、シングルメタルの CMOS の製造技術を使用している。1 チップの面積は 0.35  $\text{cm}^2$  であり、約 48,000 トランジスタが集積されている。88本の I/O ピンを使用している。チップは 10 MHz で動作する。

### 3.2 今後の VLSI 化課題

超並列型プロセッサおよび機能分割型プロセッサの今後の課題について、それぞれ順に述べる。

超並列型プロセッサの課題は通信のオーバヘッドの削減である。最近の LSI ではチップ内にメモリを含むものが少なくなく、このようなチップを用いた装置では、ロジック部分とメモリ部分を接続する線の本数は膨大になる。ロジック部分を VLSI プロセッサに封じこめたレギュラーな部分と VLSI プロセッサ上にないメモリ部分の間の通信をユニット時刻ごとに行うための通信量は相当なものになると推測される。また、シミュレーション結果をトランザクションファイルとして蓄える場合にも、イグズォースティブ法においてはその通信量は相当なものとなる。メモリ部分やトランザクションファイルを超並列アーキテクチャの中に分散させてゆくなどの展開が必要であろう。

機能分割型プロセッサの課題は、性能を一層向上させるためのアーキテクチャである。具体例として紹介したマシンの 1 プロセッサ (PE 14 台相当) の速度は、マシンサイクル 27.5 ns の大形コンピュータ上ソフトウェアの速度<sup>11)</sup>にほぼ等しい性能に留まっている。一方、汎用の RISC プロセッサではクロック速度が 40 MHz のものも発表されている。このような汎用プロセッサに PE を置換した場合に比して、シミュレーション速度やプログラマビリティによる多機能性において性能的な差を維持してゆけるようなアーキテクチャの研究が今後の課題である。

## 4. おわりに

VLSI 論理シミュレーションプロセッサ向きのアーキテクチャについて解説した。現状では多数派を占める非 VLSI マシンとの対比において、このようなアーキテクチャとアルゴリズムを位置づけた。

論理シミュレーション専用マシンは CAD の分野の専用マシンとしては、技術的成果においても、経済的効果においてももっとも成功している。しかし、装置の大規模・高速化は今後も進んでいくため、専用 VLSI プロセッサなどの技術によるさらなるブレークスルーが期待される。

本稿の執筆に当たり、貴重な資料をご提供いただいた、Simulog 社の Dr. A. Suhami に深謝する。

## 参考文献

- 1) Takasaki, S., Hirose, F. and Yamada, A.: Logic Simulation Engines in Japan, IEEE Design & Test of Computers, Vol. 6, No. 3, pp. 40-49 (1989).
- 2) Blank, T.: A Survey of Hardware Accelerators Used in Computer-Aided Design, IEEE Design & Test of Computers, Vol. 1, No. 3, pp. 21-39 (1984).
- 3) Takasaki, S. et al.: HAL II: A Mixed Level Logic Simulation System, Proc. of ACM IEEE 23th Design Automation Conference, pp. 581-587 (1986).
- 4) Hirose, F. et al.: Simulation Processor "SP", Proc. of IEEE International Conference on Computer Aided Design, pp. 484-487 (1987).
- 5) 山田 博, 広瀬文保, 新妻潤一, 進藤達也: シミュレーションプロセッサ "SP", 信学論 (D), J71-D, 4, pp. 644-651 (1988).
- 6) Beece, D. K. et al.: The IBM Engineering Verification Engine, Proc. of ACM IEEE 25th Design Automation Conference, pp. 218-224 (1988).
- 7) Erickson, B.: Automation and Simulation in Large System Design, VLSI SYSTEM DESIGN, pp. 42-48 (1986).
- 8) Yoav Lavi: Hardware Logic Simulator, United States Patent, No. 4, 697, 141 (1987).
- 9) Agrawal, P. et al.: MARS: A Multiprocessor-Based Programmable Accelerator, IEEE Design & Test of Computers, Vol. 4, No. 5, pp. 28-36 (1987).
- 10) XILINKS, Inc.: The Programmable Gate Array Data Book, XILINKS, P/N0010048 01 (1988).
- 11) Krohn, H. E.: Vector Coding Techniques for High Speed Digital Simulation, Proc. of ACM IEEE 18th Design Automation Conference, pp. 525-529 (1981). (平成元年 12 月 4 日受付)