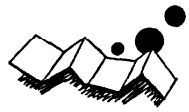
解　説



5. 最近の CAD システムの話題

5.3 CAD 専用処理装置†

大森健児††　小池誠彦††

0. はじめに

1980年代に入って論理装置の CAD を対象とした専用処理装置の発表が相次いでいる．これは，LSI 化と論理規模の増大にともないより高い設計品質とより高い生産性への要求に答えるために現われてきた傾向である．電子産業の分野においては，この傾向は相当に長い期間続くものと予想され，装置 CAD 用の専用処理装置に対する需要はますます強まるものと推察される．そこで，すでに発表されている専用処理装置について述べ，今後の装置 CAD 用専用処理装置を考える上での助けとしたい．

1. 専用処理装置の必要な分野

装置 CAD にとって，必要なアプリケーションプログラムは，回路図入力，回路シミュレーション，論理シミュレーション，タイミングベリフィケーション，ルーティング，レイアウト，デザインルールチェックなどであり，**表-1** に示すような特徴を持っている．これらの中には，満足のゆく情報を得るためには，現在の計算機の処理能力より二桁，三桁，あるいは，四桁上の高い処理能力を必要とするものがある．論理シミュレーション，回路シミュレーション，ルーティングなどがそうである．一方で，インタラクティブ処理という，より高度な使用法を指向して，別の面からの処理能力を要求しているものもある．例えば，回路図入力，部分的な論理シミュレーション，タイミングベリフィケーションなどである．前者は，今の計算機あるいは既知のアルゴリズムではその要求に答えることができないものであり，新しい方法が求められている．後者は，計算機を独占すれば要求に答えられるものであり，専用ワークステーションという形で解決されつつある．

専用処理装置を必要とする分野の特質を以下に簡単に述べる．

イ）論理シミュレーション

論理シミュレーションは回路の論理検証に使われるもので，LSI の内部の検証から装置全体の回路の検証まで，その用途は広い．シミュレーションの単位は MOS 回路でのトランスファゲートを扱うスイッチレベル，論理の最小単位を構成するゲートレベル，機能単位を中心としたブロックレベルなどいくつかがある．しかし，装置 CAD においてはシミュレーションできる規模に重点がおかれるため，シミュレーションの単位はゲートレベルあるいはブロックレベルとなっている．装置の大規模化にともない，100 万ゲート以上もの回路で構成される装置，例えば最近の大型計算機，も出現してきており，通常のソフトウェアシミュレーションで装置の論理シミュレーションを行うと，必要な時間は数百日から数十年にも及ぶ．そのため，この分野に対する専用処理装置への要求は非常に高く，また，専用装置化が最も進んでいる．

ロ）ルーティング

LSI やプリント板の自動配線を行うためのものである．基本的には2点間の最短経路を求める問題に帰着する．デバイス技術の進歩により配線密度が高くなり 1 mm 当り数百本ものトラック数になると1チップで 100万点もの格子点を扱う必要があり，高速処理が必要になってきている．いくつかの実験的な専用マシンが開発されてきている．

ハ）回路シミュレーション

LSI を構成するトランジスタ回路の動作シミュレーションを行うものである．抵抗，コンデンサ，スイッチング回路などの素子で構成されたネットワークの回路方程式を解き，各所の信号波形を得るためのシミュレーションである．一般には，大規模なスパースマトリクスとなる微分方程式を解くことになる．ニュート

† CAD Machines by Kenji OHMORI and Nobuhiko KOIKE (C & C Systems Labs., NEC Corporation).

†† 日本電気(株) C & C システム研究所

表-1 CAD の分野と処理形態

	問題及び問題の限定	アルゴリズム	並列性	処理形態	結合方式
論理シミュレーション	○ 同期型/非同期型 ○ 論理値 0.1 多値 ○ 遅延 0.1 可変 ○ 論理単位 スイッチ, ゲート, ファンクション	○ ランクオーダ ○ イベント駆動 ○ タイムホイール	○ 素子数分だけの並列度 ○ 機能分割	○ 並列 ○ パイプライン	○ クロスバ ○ 結合ネット ○ 共通バス
回路シミュレーション	○ 繰り返し分の省略 ○ 空きマトリクスの圧縮 ○ 波形近似	○ レベル分け ○ イベント駆動	○ 素子数分だけの並列度		
ルーティング	○ セル位置の固定 ○ エリアの限定 ○ 目的方向のサーチ	○ Lee Moore の手法	○ 2次元同時処理可能	○ 2次元配列プロセッサ	○ 4近傍接続
デザインルールチェック	○ パターン間隔の検査 ○ 短絡 ○ 層間のパターンの重なり	○ ラスタスキャン法 ○ ビットマップ処理	○ 2次元同時処理 ○ ベクトル演算	○ 並列 ○ パイプライン	○ 専用接続 ○ 4近傍接続
自動合成	○ 論理合成 ○ マスクパターン生成 ○ 設計ルール検証	○ プロダクションシステム			
図面入力 (エンジニアリングワークステーション)	○ ビットマップ処理 ○ 設計ルール検査 ○ 高機能ユーザインタフェース ○ インタラクティブ論理シミュレーション	(回路/論理シミュレーションと同じ)	○ 機能分割	○ 表示プロセッサ ○ 言語プロセッサ ○ 付加シミュレーションエンジン	○ メモリ共有 ○ 共通バス

ン法などにより細かい時間きざみごとに解を求める．ノード数をNとすると計算量は$O(N^{1.2}\sim N^{1.5})$程度である．専用処理装置はまだ現われていないが，回路の高精度な遅延計算をはじめとして，その需要は高い．

2. 高速化の一般的な手法

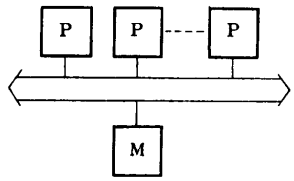
高速化を図る方法として，新しいアルゴリズムを発見して処理の高速化を図る方法と，現在使われているアルゴリズムそのもの，あるいはその変形を高速に実行する専用処理装置を考え出して処理の高速化を図る方法の2つが考えられる．前者が漸進的であるのに対し後者は革新的である．CAD 専用処理装置は，後者の立場をとるものである．

専用処理装置による高速化の方法として，並列処理を利用する，アルゴリズムをハードウェアで実現する，あるいはこれらを併用するなどが考えられる．

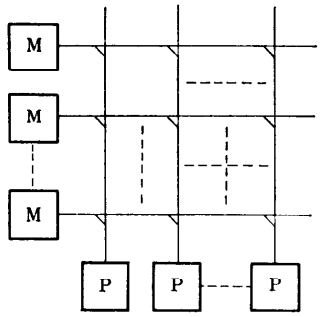
イ) 並列処理

数十台あるいは数百台のプロセッサで並列にしかも同時に処理を行うことができれば，理想的には台数分だけの性能増加をもたらすことができる．しかも，装置 CAD の中では，並列に処理可能なものが多く，この並列処理は極めて有効である．例えば，論理シミュレーションを考えると，回路の一つ一つが動作しているものと考えることができるのでこの動作をシミュレーションの対象とする限り，並列度は極めて高い．また，ルーティングの場合においても，二次元配列上での各格子での処理が主体となるため，並列度は高いといえる．

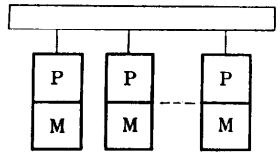
このような特色を活かしたいわゆる並列アーキテクチャには，図-1 に示すようにメモリ共有型，クロスバ型，ネットワーク型，プロセッサアレイなどのアーキテクチャがある．メモリ共有型は柔軟性に富み，価格も低いという特徴を持つが，プロセッサ台数が限られるために，数十台以上の並列アーキテクチャには有効でなく，このため CAD 用の専用処理装置のアーキテクチャにはあまり利用されていない．クロスバ型は拡張性に劣るが，プロセッサ間でのデータ転送時間が短いため，短時間のうちにプロセッサ間でデータを送



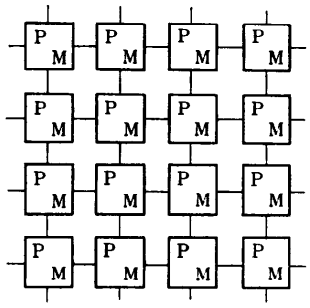
(a) メモリ共有型



(b) クロスバ型



(c) ネットワーク型



(d) プロセッサアレイ

図-1 並列アーキテクチャ

りたいという処理には有効である．このため，論理シミュレーション用の専用処理装置のアーキテクチャとして利用されている．ネットワーク型は，多数のプロセッサを接続するとき極めて有効であり，やはり論理シミュレーション用の専用処理装置のアーキテクチャの中で用いられている．プロセッサアレイは隣接するプロセッサとの間で交信ができれば済むような処理に対して極めて有効で，ルーティング用の専用処理装置のアーキテクチャの中で利用されている．

ロ） アルゴリズムのハードウェア化

ソフトウェアで行われている処理の中で，処理そのものは簡単であるが，処理時間が相当にかかるものはハードウェア化するとかなり高速化することができる．例えば，ゲートのシミュレーションをする場合には，ソフトウェアシミュレーションでは一つのゲートに対して少なくとも数命令は必要である．このため，これだけで数マイクロ秒から数百マイクロ秒かかる．ところが，これをハードウェア化すれば，数十ナノ秒あるいは数百ナノ秒に短縮することができる．このため，論理シミュレーション用の専用処理装置においてはシミュレーションの単位となっているものに対し，その機能のハードウェア化が，また，ルーティング用の専用処理装置においては Lee アルゴリズムの LSI 化が検討されている．

3. 論理シミュレータ

論理シミュレーションの役割は，装置を作り始める前にすべての論理を正しておくということにある．しかし，LSI の論理規模の増大と新規設計カスタム LSI の多用という流れのなかでは，ソフトウェアでシミュレーションできる範囲は限定されたものになりつつある．

装置の論理シミュレーションの最も手っ取り早い方法は，装置をテストするプログラムを流すことである．回路あるいはファームウェアに誤りがある場合には正常に走らないので，このテストプログラムを用いれば，装置の論理検証を簡単に行うことができる．ところで，大型計算機を対象にする場合には，実際には数分しかかからないテストプログラムも，シミュレーション上で行うと何十年という歳月を必要とする．

このため，装置の論理シミュレーションを主目的とした専用処理装置が表-2 に示すようにいくつか現われてきている．テストプログラムを二週間程度で実行できる専用処理装置も作られており，装置製造前に装置全体の論理を，テストプログラムにより検証することが可能になってきている．

装置 CAD 用では，ゲートレベルシミュレーションと，ブロックレベルシミュレーションがある．ゲートレベルシミュレーションは，AND, OR, NAND, NOR, NOT などのゲートをシミュレーションの単位とするものである．ブロックレベルシミュレーションは，IC そのもの，あるいは，それを構成するブロ

表-2 論理シミュレーションマシンの発表例

	YSE IBM	HAL 日電	LE-1002 ZYCAD 社	LSM IBM	Bell 研-1 Abramovici et al.	Bell 研-2 Levendel et al.
容量	10^6 Gate	1.5×10^4 Block (10^6 Gate)	6.4×10^4 Gate	6.4×10^4 Gate		
速度	2×10^9 ゲート/秒	2.6×10^7 ブロックイベント/秒	1×10^6 Gate イベント/秒	6.4×10^8 ゲート/秒	ソフトウェアシミュレータの10～60倍	同左
シミュレーション方式	ゲート(4入力)	ブロック (16入力)	ゲート(3入力)	ゲート(5入力)	ゲート+ファンクション	ゲート+ファンクション
ディレ	Unit, Zero	Zero	可変	可変	可変	可変
実行	ランクオーダ	レベルオーダ イベント駆動	タイムホイール イベント駆動	(タイムホイール)	タイムホイール イベント駆動	タイムホイール イベント駆動
RAM/ROM	△	○ (4 MB)	○	○ (2.3 MB)		
アーキテクチャ	並列/パイプライン	並列/パイプライン	並列	並列/パイプライン	機能分散	並列/パイプライン
	≤256 台	≤32 台	≤16 台	≤64 台		
特徴	クロスバスイッチネットワーク	多段接続ネットワーク ブロックレベルシミュレーション	共通バス 小型，フォールトシミュレーション	クロスバスイッチネットワーク		クロスバ+共通バス
現状，その他	試作終了 (1982 年発表)	32 台構成稼動中 (1982 年発表)	製品化済 (1982 年発表)	64台構成を2セット～80年より社内使用 (1983 年発表)	ペーパマシン (1982 年発表)	ペーパマシン (1983 年発表)

ックをシミュレーションの単位とするものである．ゲートレベルシミュレーションは，より精度の高いシミュレーションを目的としている．専用処理装置として，IBM の YSE (Yorktown Simulation Engine)[1]，LSM (Losgatos Simulation Machine)[2]，あるいは，Bell 研のペーパマシン[3),4)] などをあげることができる．ブロックレベルシミュレーションは，より高速なシミュレーションを目的としている．専用処理装置として，日本電気の HAL[5),6)] をあげることができる．

イ） ゲートレベルシミュレータ

YSE は，図-2 に示すように，256 台までの専用プロセッサを接続することが可能な負荷分散型の並列処理装置である．YSE は，ゲートをシミュレーションの単位とし，信号の伝播順序に従って全ゲートのシミュレーションを行うランクオーダ方式をとっている．YSE はシミュレーションを実行する本体と，シミュレーションの制御を行うコントロールプロセッサと，シミュレーションのための命令を作り出すためにコンパイルを行うホストマシンとで構成される．本体はランダムロジックのシミュレーションを実行するロジックプロセッサと，メモリのシミュレーションを実行するアレイプロセッサと，これらプロセッサを結合するためのスイッチとで構成される．ロジックプロセッサにはゲートをシミュレーションする機能とシミュレーションの結果得たゲートの出力を他のプロセッサに伝える機能とがある．ゲートの出力信号を伝播させる機能はゲートの出力での新たな論理値と，その出力につながっているゲートの入力での論理値を一致させるために，入力側のゲートのシミュレーションを分担しているロジックプロセッサに新しい論理値を送り，それを更新させるためのものである．

ロジックプロセッサは，命令メモリ，データメモリ，ファンクションユニットなどからなる．命令メモリの各命令は，ゲートのシミュレーションと，ゲートの出力の転送を指示する．命令メモリは 8 k 語の容量をもち，命令は固定長で一語で記述される．このため，ロジックプロセッサは最大で 8 千個のゲートをシミュレーションすることができる．命令メモリにはランクオーダにもとづいて，シミュレーションの実行順序に従って並べられたゲートに対応する命令が並べられている．データメモリには各ゲートの入出力値が記憶され，命令により読み出し，書き込みが行われる．また，ファンクションユニットはゲートの種類と入力値が与えられたとき，そのゲートに対する新しい出力値を出力する．LSM は YSE の前身で同様の構成をとる．容量は YSE より少ないが，遅延シミュレー

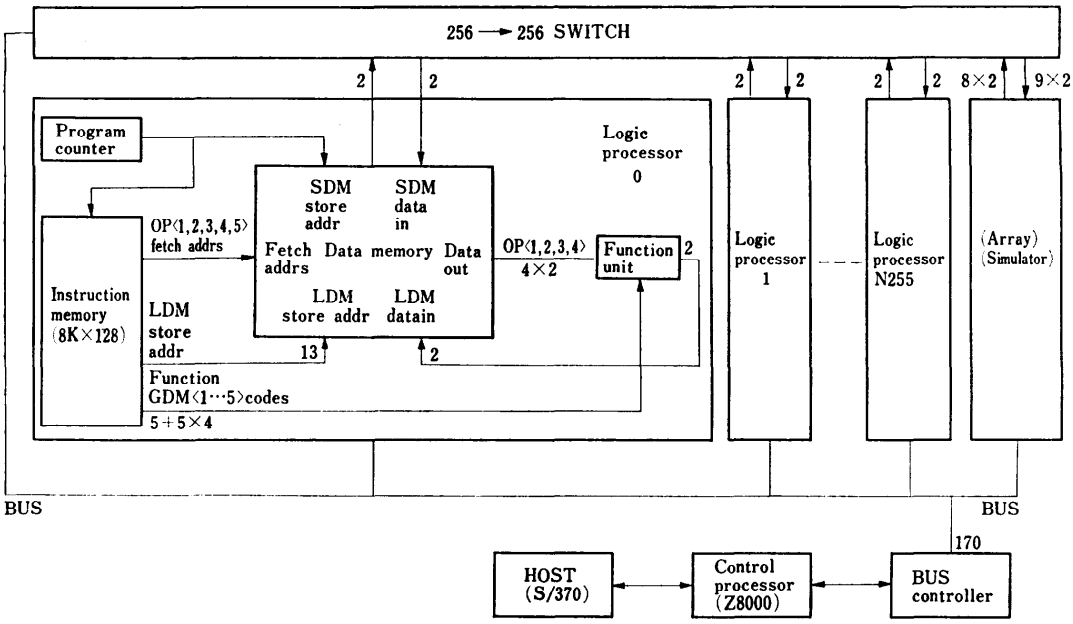


図-2 YSE の 構 成

ションができるのが特徴である．64台のロジックプロセッサを並列に用いることにより 6.4×10^8 logic expression/秒 の処理性能を得ることができる．

ロジックエバリュエータ LE-1002 は最初の商用機である．汎用計算機に接続しLSIの検査に使用することが目的である．2台の専用プロセッサでソフトウェアシミュレーションの30～100倍の性能を得ることができる．シミュレーションの方式はタイムホイールにもとづくイベント駆動方式である．3入力1出力のゲートの入出力それぞれに遅延を持たせることができる．

Bell 研からは2つのシステムが発表されている．その一つはこれまで述べた負荷分散型と異なり，図-3 のような機能分散型のマルチプロセッサシステムである．ゲートに対する論理シミュレーションの処理過程をいくつかの独立な単位に分け，それらを同時に処理しようとするものである．汎用計算機で動くソフトウェアシミュレータに比べ10～60倍のスピードとなる．

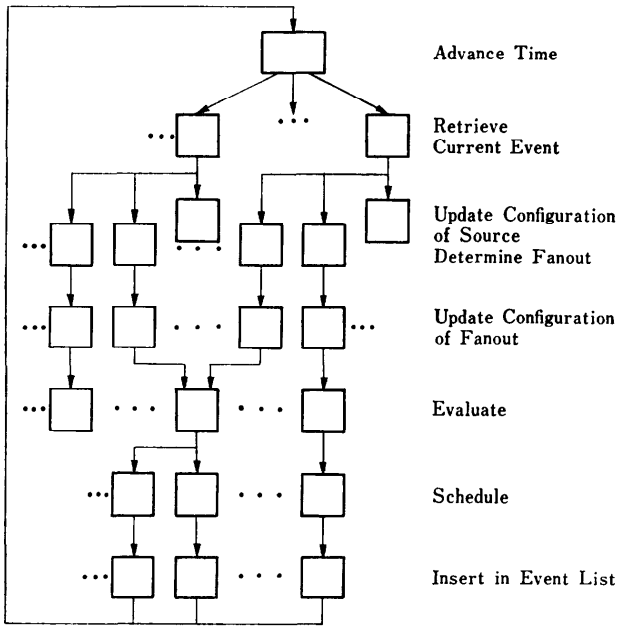


図-3 機能分散型論理シミュレーションマシン

もう一つの Bell 研のシステムは，単純なゲート類の処理と，より複雑なファンクションブロックの処理とに分けたものである．単純なゲート類の処理プロセッサをクロスバスイッチで結合し，ファンクション系のプロセッサを並列バスで結合した階層構造のアーキ

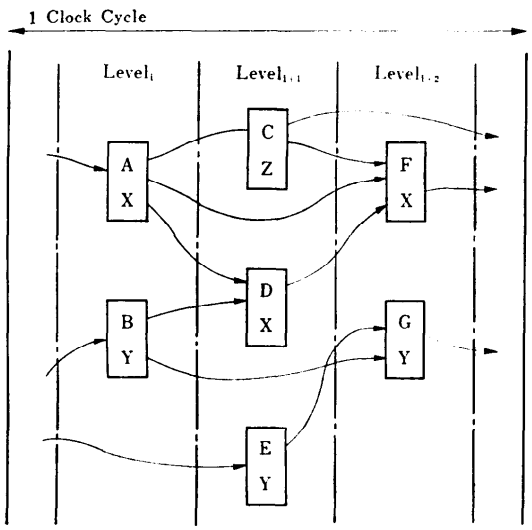


図-4 ブロックのレベルづけ

テクチャになっている.

ロ) ブロックレベルシミュレータ

HAL はゲートの集まりであるブロックをシミュレーションの単位とし，クロック同期式回路からなる論理装置をシミュレーション対象としている．ブロックとして，IC そのもの，あるいは LSI の中の機能モジュールが選択されるが，このブロックは信号の伝播順序をもとにして並べられる．これをブロックのレベルづけと呼ぶが，その例を図-4 に示す．このように並べられたブロックについて，クロックごとに次のようにシミュレーションを行う．最も小さなレベル番号をもつブロックの中で入力に変化のあったものすべてについてシミュレーションを行う．シミュレーションを行った結果，出力に変化があったならば，接続先のブロックの入力を新しい出力値で置き換える．このレベルのブロックについてのシミュレーションが終わったとき次の番号のレベルに対して同じような方法でブロックのシミュレーションを行う．すべてのレベル番号についてシミュレーションを終了したとき一つのクロックに対してのシミュレーションが終了する．次のクロックに対して最も小さな番号のレベルからシミュレーションを始める．

HAL は，上に述べたシミュレーション手順の中に含まれる並列性を利用して高速化を図ったものである．

同一レベル番号のブロックについてはいずれを先にシミュレーションしても構わないので，HAL では，複数のプロセッサがこれらを同時にシミュレーション

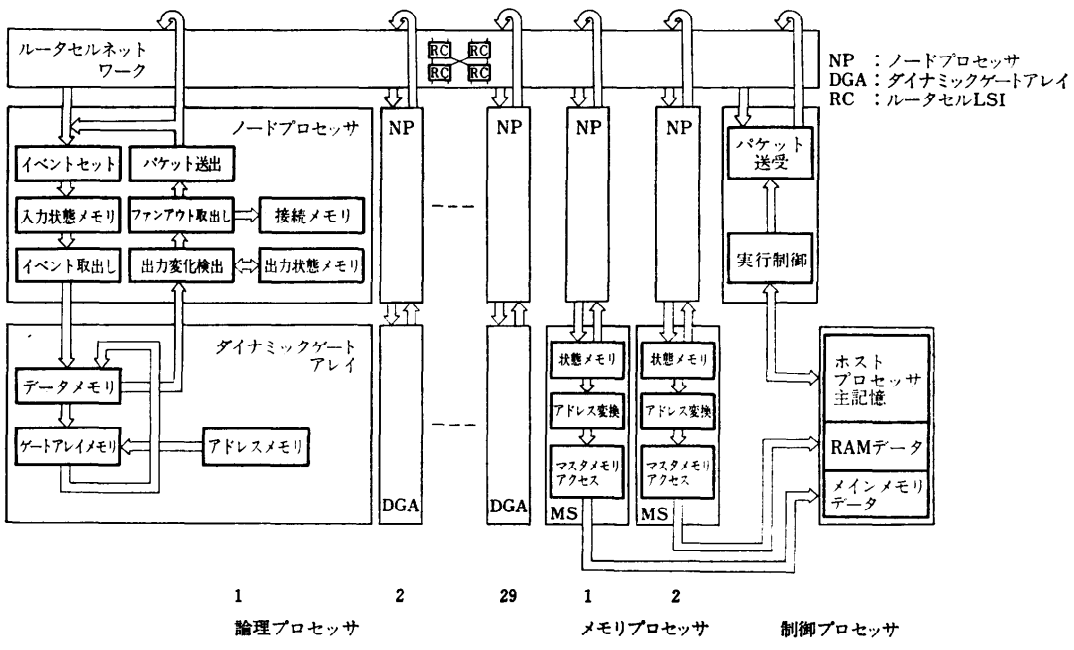


図-5 HAL の構成

するようにした．また，ブロックのシミュレーションをハードウェア化することによりブロック自体のシミュレーションの高速化を図った．さらにシステム全体がパイプライン処理となるようにアーキテクチャ上の考慮を図った．これにより従来の大型計算機上でのソフトウェアによるシミュレーションと比較して 10^4 倍の処理速度となった．

HAL の構成を図-5 に示す．HAL は 32 台のプロセッサ構成である．このうち 29 台はランダムロジックをシミュレーションするための論理プロセッサ，2 台はメモリをシミュレーションするためのメモリプロセッサ，残りの 1 台はホスト計算機とのインタフェースを行うためのマスタコントロールプロセッサである．これらプロセッサはルータセルネットワークとよばれるもので結ばれている．シミュレーションの対象となる装置の回路は，ブロックの種類で整理され，各プロセッサに分配される．各プロセッサは，共通のメモリをもたず，ブロックのシミュレーションに必要なデータはすべて自身のプロセッサの中にもつ．また，出力の変化は接続先のブロックの入力値を変えるという動作を引き起こすが，伝達先は同一のプロセッサ内とは限らない．異プロセッサである場合にはこの情報はルータセルネットワークを介して伝える．

論理プロセッサの部分は二つに分かれていてシミュレーションの制御をするためのノードプロセッサとブロックのシミュレーションをするためのダイナミックゲートアレイからなる．ノードプロセッサは ①出力側に変化があった場合，その情報を受けて入力値を変える（イベントセット），②入力値に変化のあったブロックを取り出し（イベント取り出し）ダイナミックゲートアレイにこのブロックのシミュレーションを行わさせる，③出力に変化があるかどうかを調べる（出力変化検出），④出力に変化があった場合その相手先を調べる（ファンアウト取り出し），⑤相手先に変化を伝える（パケット送出）などを行う．ダイナミックゲートアレイはゲートアレイメモリに書き込まれたブロックの論理パターンをもとにしてシミュレーションを行う．

4. ハードウェアルータ

自動配線においては，チップ内に詰め込む素子の数が増えてチップの中が密になってくると配線できない個所が表われてくることがある．未配線の部分については，人手に頼ることになるが，多くの時間を費やすことも稀ではない．そこで，計算機側からくる制約をなるべく小さくし，未配線部分が残る割合を少なくしようという立場から，現在利用されているアルゴリズムを高速に実行するような専用処理装置[7] がいくつか現われてきている．

イ） Lee アルゴリズム

Lee アルゴリズム[8] を簡易化して図-6 で述べる．Lee アルゴリズムではルーティングの対象となる平面を格子状に分割する．この格子に沿って，始点から終点へと配線を行うが，すでに配線されている格子は使わないようにする．アルゴリズムは次のようになる．始点に隣接している格子の中に終点の格子がなければ，配線に使われていない隣接の格子すべてに対して 1 というラベルを与える．次にラベル 1 の格子に隣接している格子の中に終点の格子がなければ，配線に使われておらずかつラベルづけされてもいない隣接の格子すべてに対して 2 というラベルを与える．このように新しくラベルづけされた格子に対し，その隣接する格子に終点のものがない限り，配線に使われておらずかつラベルづけされてもいない隣接の格子に 1 加えたラベルを与える．終点にぶつかったときは，この操作は終わりである．このとき，終点からラベルにつけられた番号を逆順しながらその番号に対応する格子を一つ選び，それを配線されたものとして始点に至るまでたどる．配線されたものとして選ばれなかった格子のラベルを消去して配線を終了する．ラベルづけをしているとき，新しくラベルづけされた格子のすべてにおいて，その隣接するブロックがすでにラベルづけされた格子であるかあるいは配線ずみの格子であるということが起こる．このような場合には，この始点と終点の間の配線は行えない．これは未配線部分として残

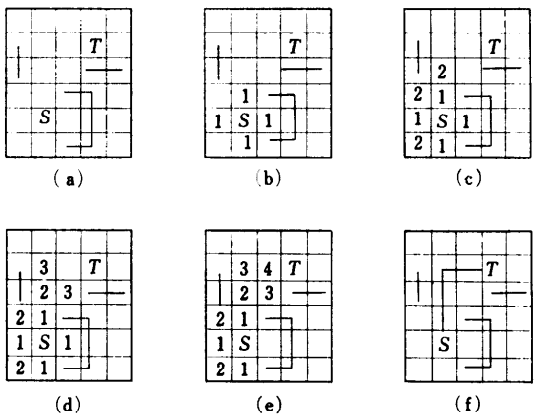


図-6 Lee アルゴリズム

し，後で人手により配線処理をする．

ロ） ハードウェアルータの例

L-マシン[9]は，図-7に示すように，コントロールユニット，プロセッサユニットからなり，コントロールユニットは全体の制御を，プロセッサユニットはLeeアルゴリズムを実行する．プロセッサユニットはN×Nの L-セルからなる．L-セルは隣りあう四つの L-セルと交信できる．各 L-セルは次に示すいずれかの状態をもつ．1) ブランクである．2) ラベルづけされている．3) ブロックしている．4) 終点である．5) 始点である．プロセッサユニットでの処理は，与えられた始点と終点の間を結んでいるブランクの L-セルの列の中で最短のものを見つけることである．これは，コントロールユニットから次のようなコマンドにより行われる．1) 初期化せよ→すべてのセルをブランクの L-セルとする．2) ブロックの L-セルの書き込みを行え→x, y のアドレスで指定されたセルをブロックの L-セルとする．3) 始点の L-セルの書き込みを行え→x, y のアドレスで指定された L-セルを始点の L-セルとする．4) 終点の L-セルの書き込みを行え→x, y のアドレスで指定された L-セルを終点の L-セルとする．5) ラベルづけを行え→始点の L-セルまたはラベルづけされた L-セルに隣りあう四つの L-セルの中でブランクのものはラベルづけされた L-セルとする．もし，隣りあう L-セルに終点の L-セルがあればラベルづけを終了する．ラベルづけをするとき，どの方向の L-セルによってラベルがつけられたかを明らかにするために矢印により方向づけをする．二つ以上の L-セルがラベルづけをしようとすることがあるので，この時は，上，左，右，下の順序により選ぶ．6) 選ばれたパスを外部に伝達せよ→終点の L-セルからラベルづけされた L-セルの中の矢印に従って L-セルを追ってゆく．

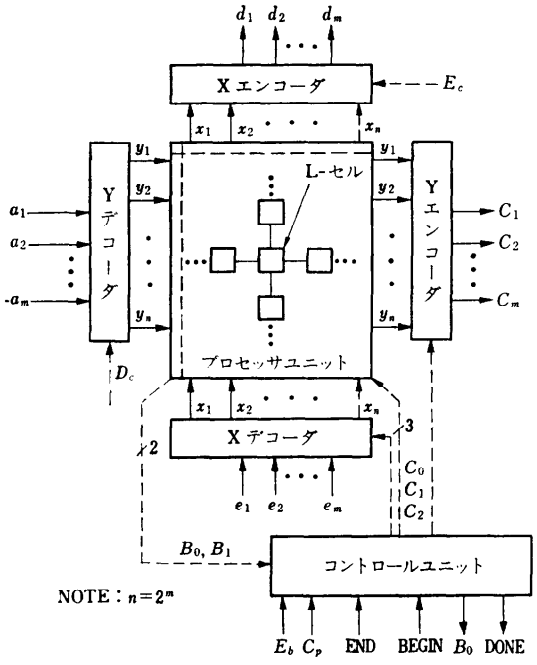


図-7 L-マシンの構成

L-マシンは実際には作られていないが，L-セルは75ゲート程度で実現でき，VLSI 化すれば 64×64 あるいは 256×256 規模の L-マシンが実現できると予想されている．始点と終点の距離を p としたとき，Lee アルゴリズムでは処理量は $O(p^2)$ であるが，L-マシンでは $O(p)$ となる．ただし，L-マシンの難点は格子の数が L-セルの数を越える場合については処理することができない点にある．

これに対して，一つのプロセッサにいくつもの格子を分担させる折りたたみ法により，物理量を越えるものについても可能にしたのがSAM-マシン[10]，AAP[11]である．折りたたみ法は物理量を減らす上で有効であり，また，ハードウェア化による高速化も十分に図ることができる．例えば，1000×1000 格子のルーティングに対し，普通の計算機で行えば，5時間のところが，1000×1000 のセルを用いた SAM-マシンでは 0.4秒＋ロード時間，32×32 のセルを用いた折りたたみの SAM-マシンでは 15秒＋ロードの時間となる．

ハ） 評価計算をともなうルーティングマシン

Lee アルゴリズムでは，最短の配線路を選ぶことを基準としていたが，これ以外のものを基準とした方がよい場合がある．マスタスライスチップでは，格子の縦と横の方向にチャネルが存在し，チャネルには，配線に使われるトラックがいくつか存在するということを条件にして配線を行う．このような場合，割り当てをしていないトラックが各所で平均化するように配線をしてゆくことが，配線できないという局面を少なくする．そこで，割り当ての済んだトラック数などをコストとして，このコストが最小となるような配線路を選択するのがよいと考えられる．例えば，マスタスライスチップでは，評価計算によりチャネルを選び（グローバル配線），チャネルの中のトラックを後で選ぶ（埋め込み配線）という方法がある．

IBM のワイヤルーティングマシン（WRM[12),13)]）

は，コストにもとづく配線を行うものである．チップの中の素子数が多くなると，コスト最小の配線路を見つけることに膨大な時間を必要とする．そこで，限られた時間の中で処理できる配線の数を多くするために，WRM では迂回という方法がとられる．

WRM は，市販のマイクロプロセッサを用い 8×8 のプロセッサアレイとして実現され，折りたたみ法によりルーティングを行う．19×23 のセルに対するルーティングにおいて，WRM で1分の CPU 時間に対し，IBM 3033 で 45 秒の走行時間になったと報告されている．したがって，プロセッサアレイを VLSI により実現することにより高速なルーティングマシンを得ることが期待できる．

5. その他の専用マシン

ボードやチップのレイアウトを決める際に，モジュールの配置はルーティングと並んで重要な問題である．モジュールの配置は，総配線長最小のようなコスト計算にもとづいて決められる．コスト最小の配置を求めるアルゴリズムとしてモジュール入れ換え法があるが，これを利用したのがモジュール入れ換え配置マシン[14]である．

その他に，設計ルール検証用のマシン[15]がある．このマシンは，レイアウトの設計ルールを検証するためのもので，専用のハードウェアがラスタ走査法を用いて幅や角の検証を行う．300 ミル (7.6 mm) 角のレイアウトに対して1分半で処理する．

6. おわりに

論理シミュレーションとルーティングを中心に装置 CAD 用の専用処理装置について述べた．筆者は HAL の開発に携わってきたが，この専用処理装置は実際の現場ですでに使われ，かなりの実績をあげている．専用処理装置の出現はある面では従来の設計開発手順を変えつつある．専用処理装置を用いれば，手順は次のようになる．装置開発において最初に開発すべきものはテストプログラムであり，この後で回路図レベルまでのハードウェア設計が続く．この段階が終了すると先のテストプログラムを用い専用処理装置で回路の論理検証さらにはファームウェアの論理検証まで進められる．この検証に合格した後で，製造の段階に入り，従来の工程をたどる．しかし，製造後のデバッグはアセンブリミスを見つけるという程度にとどまり従来とは様相を異にする．専用処理装置によるデバッグでは，当たり前のことではあるが，実機によるデバッグよりもずっと必要な情報を得やすいということであった．このようなことから，今後各種の CAD 用専用処理装置が作られ，活発に利用されることになると考えられる．

参考文献

1) Denneau, M. M.: The Yorktown Simulation Engine, 19th DA Conf., pp. 55-59 (1982).
2) Howard, J. K., Malm, R. L. and Warren, L. M.: Introduction to the IBM Los Gatos Logic Simulation Machine, Proc. IEEE Conf. on Computer Design, pp. 580-583 (1983).
3) Abramovici, M., Levendel, Y. H. and Menon, P. R.: A Logic Simulation Machine, 19th DA Conf., pp. 65-73 (1982).
4) Levendel, Y. H., Menon, P. R. and Patel, S. H.: Special-Purpose Computer for Logic Simulation Using Distributed Processing, BSTJ, Vol. 61, No. 10, pp. 2873-2909 (1982).
5) Koike, N., Ohmori, K., Kondo, H. and Sasaki, T.: A High Speed Logic Simulation Machine, Compcon '83 Spring., pp. 150-156 (1983).
6) Sasaki, T., Koike, N., Ohmori, K. and Tomita K.: HAL; A Block Level Hardware Logic Simulator, 20th DA Conf., pp. 150-156 (1983).
7) Hong, S. J. and Nair, R.: Wire-Routing Machines, Proc. IEEE, Vol. 71, No. 1, pp. 57-65 (1983).
8) Lee, C. Y.: An Algorithm for Path Connections and Its Applications, IRE Trans. on Elect. Computers, Vol. EC-10, pp. 346-365 (1961).
9) Breuer, M. A. and Shamsa, K.: A Hardware Router, J. Digital Syst., Vol. 4, No. 4, pp. 393-408 (1980).
10) Blank, T., Stefik, M. and Cleemput, W.: A Parallel Bit Map Processor Architecture for DA Algorithms, Proc. 18th DA Conf., pp. 837-845 (1981).
11) 渡辺琢美，北沢仁志，杉山 吉：LSI 配線並列処理の一手法，情報処理学会第 27 回全国大会論文集，pp. 1383-1384 (1983).
12) Hong, S. J., Nair, R. and Shapiro, E.: A Physical Design Machine, Proc. VLSI 81, pp. 257-266 (1981).
13) Nair, R., Hong, S. J., Liles, S. and Villani, R.: Global Wiring on a Wire Routing Machine, 19th DA Conf., pp. 224-231 (1982).
14) Iosupovici, A.: A Module Interchange Placement Machine, 20th DA Conf., pp. 171-174 (1983).
15) Seiler, L.: A Hardware Assisted Design Rule Check Architecture, 19th DA Conf., pp. 232-238 (1982).

(昭和 59 年 5 月 15 日受付)