論理シミュレーションマシンのハードウェア構成†

小池誠彦†† 大森健児†† 佐々木徹†††

本論文は大規模な装置の論理シミュレーションを高速に実行する専用マシンのハードウェア構成と設計思想について論じている．専用マシンは3種のプロセッサからなる複合体で構成され，プロセッサ間を専用に開発したLSIを用いた接続ネットワークで結合する．32台の専用プロセッサ群によって論理シミュレーション処理が並列に実行される．そして，各プロセッサはそこで必要な処理を高速に行うためにすべてハードウェアで専用化しパイプライン式に処理を行う．本論理シミュレーションは製作を完了し数千ブロックからなるテストモデルによる動作確認を終え，従来のソフトウェアによるシミュレータより数千倍以上の速度で実行されることが明らかになった．

1. はじめに

大規模なシステムの論理シミュレーションを高速に実行する専用マシン（超高速論理シミュレータ）を開発した[1]が，ここでは超高速論理シミュレータのハードウェア構成について論じる．

いままでにいくつかの論理シミュレーションマシンの開発あるいは構想が発表されているが[6),7)]，これらはゲートをシミュレーション単位としている．そのため，ハードウェアの規模が大きくなりすぎたり，あるいはシミュレーション容量が小さすぎたりという欠点をもっている．さらに，大型計算機等をシミュレーションするのに不可欠なキャッシュ，ファームウェア，メインメモリの動作をシミュレーションできないという欠点をもっている．このため，大型計算機の開発者にとっては，これら論理シミュレータは満足のゆくものではない．

そこで，これらの問題を解決するためにブロックと呼ぶ数十から数百個のゲートで構成された一つの論理的にまとまったゲートの集合体を一括して取り扱う効率のよいシミュレーション方式を考えた．本マシンはパイプライン式に論理シミュレーションを行う専用のプロセッサを並列に動作させることにより，これまでの論理シミュレーションマシンに比べて大容量化・高速化を図るとともに低価格化を可能とする実用性の高いマシンである．本マシンは組合せ回路のブロックを扱う論理プロセッサ群とメモリ系のブロックを扱うメモリプロセッサ群と全体を制御するコントロールプロセッサからなる並列複合体で構成される．内部処理を高速化するためにすべてハードワイアードな制御回路群で各プロセッサを実現した．プロセッサを結ぶネットワークは専用のLSIを用いて実現し小型大容量化を図った．本マシンはホストとなるミニコンピュータに接続し，その主記憶の一部をメモリデータ領域に使用することにより大容量のメモリデータを収容することとした．

2. シミュレーションマシンの特徴

論理シミュレーションマシンの特徴をまとめると次のとおりである．

（1）専用のハードウェアによるパイプライン処理

シミュレーション処理過程を複数のステージに分け，各ステージの処理をオーバラップさせ，パイプライン式に実行する方式を用いる．各ステージの処理をすべて専用のハードウェアで実現することで高速化を図る．

（2）並列プロセッサ構成による高速化および大容量化

専用のプロセッサを並列に用いることでほぼ線形な処理性能の向上および処理容量の増加を図る．1台のプロセッサは1,024個のブロックまで収容可能とし数十台のプロセッサを用いて約1万ブロックあるいはゲートに換算すれば100万ゲート程度の規模のシステムまでシミュレーション可能とする．

（3）モジュール機能

シミュレーションの対象によって処理速度あるいは処理容量を変更可能とするためにマシンのモジュール化を図りプロセッサの追加・削除を容易とする．使用

† Hardware Organization of the High Speed Logic Simulation Machine by NOBUHIKO KOIKE, KENJI OHMORI (C & C Systems Research Laboratories, Nippon Electric Co., Ltd.) and TOHRU SASAKI (Computer Engineering Division, Nippon Electric Co., Ltd.).

†† 日本電気(株) C & C システム研究所
††† 日本電気(株)コンピュータ技術本部

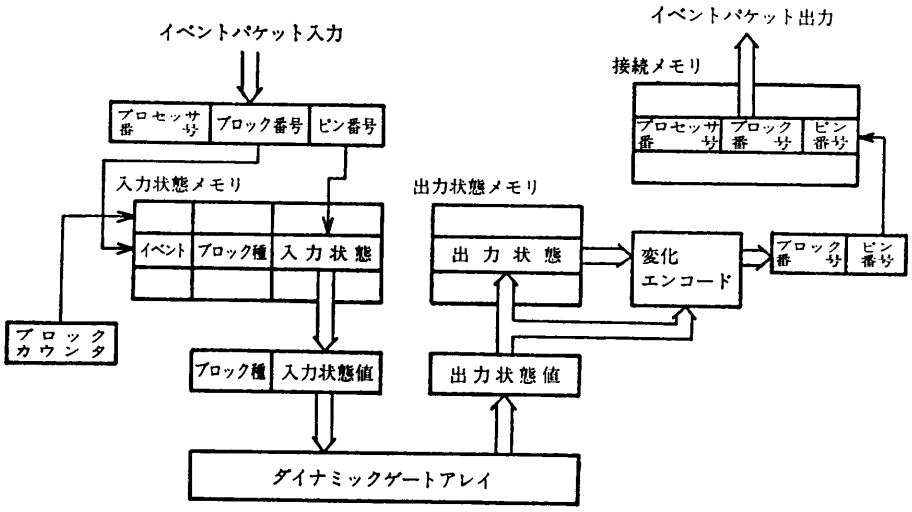

図 1 シミュレーション処理
Fig. 1 Block level simulation mechanism.

する基板は各プロセッサで共通化し必要な基板の種類の減少を図った．実装上モジュール間の結線数を減少させるために大型マルチワイヤボードを用いてモジュールのワンボード化につとめた．さらに LSI化に向く部分はカスタム LSI で実現し小型・モジュール化を図った．

3. シミュレーション方式

論理シミュレーションマシンではイベント駆動方式によりシミュレーションを行う．シミュレーションを行う単位は従来のようなゲートではなく，その集合体であるブロックを単位とする．ブロックは数十から数百のゲートで構成される組合せ回路である．ブロックは多くの場合 IC そのものあるいは LSI のビルディングブロックにあたる．

シミュレーション対象の装置は信号の伝播順序に従ってブロックがレベル分けされる．シミュレーションはレベル順に，入力ピンの信号値が変化したブロックについてのみ行う．同一レベルに属する複数のブロックは同時にシミュレーションしてもかまわない．このため複数のプロセッサをもつ論理シミュレーションマシンでは同時にシミュレーションされるブロックの数が大きくなるように各プロセッサにブロックが分配される．シミュレーションは概略，次のように行われる．それぞれのプロセッサは分担するブロックの入力と出力さらに出力の接続先を記憶するために入力状態メモリ，出力状態メモリおよび接続メモリの三つのメモリをもつ．シミュレーションが開始されるとブロックカウンタを用いて入力状態メモリを順次アクセスしイベントを発見する．イベントの存在するブロックの入力状態値とブロック種を取り出す．取り出された入力状態値から，ダイナミックゲートアレイと呼ぶハードウェアを用いて，ブロックの新しい出力状態値を求める．新しい出力状態値を出力状態メモリにセットし，同時に以前の値と比較する．もし出力値に変化があるときは変化ピン番号を一つずつ調べ，ブロック番号と変化ピン番号を用いて接続メモリをアクセスし接続先を得る．接続先の情報はプロセッサ/ブロック/ピン番号を含むイベントパケットの形をもち，ネットワークに送出される．イベントパケットを入力するとブロック番号を用いて入力状態メモリをアクセスしピン番号で指定されたビット位置を反転させ，同時にイベントフラグをセットする．

以上述べたシミュレーション処理を図式化して表したのが**図 1** である．

4. システム構成

図 2 は超高速論理シミュレータのシステム構成を示す．組合せ回路系のブロックを担当する 29 台の論理プロセッサ，メモリ系のブロックを担当する 2 台のメモリプロセッサと全体の制御を行うコントロールプロセッサとからなる．これら 32 台の専用プロセッサをルータセルネットワークが結合する．ルータセルネットワークは 2 入力 2 出力をもつルータセルを多段に接続したものであり，パケットを入力しパケット内のプロセッサ番号で指定された出口ポートに転送する．論

理プロセッサはイベント処理を行うノードプロセッサとブロックの評価を行うダイナミックゲートアレイからなる．メモリプロセッサはノードプロセッサとメモリシミュレータからなる．メモリシミュレータはホスト計算機としてのミニコンピュータの主記憶のバスに直接インタフェースをもち，主記憶の一部を共有しメモリデータを記憶する．

5. 論理プロセッサ

論理プロセッサにおいて一つのブロックをシミュレーションするための処理過程をステージに分けると次の七つとなる．

（1）イベントフェッチ：入力状態メモリを順次アクセスし，イベントを取り出す．

（2）ブロック論理演算：取り出した入力状態値，ブロック種を用いて出力状態値を得る．

（3）出力状態値更新：出力状態メモリからシミュレーション前の出力状態値を取り出しこれをテンポラリなレジスタにセーブした後，新しい出力状態値と入れ替える．

（4）変化エンコード：出力状態値の変化を調べ，変化のあったピンごとに位置番号にエンコードする．

（5）接続メモリアクセス：変化のあったブロックのピンに対する接続先を接続メモリより取り出す．

（6）パケット送出：変化イベントをパケットの形で送出する．

（7）イベントセット：パケットを入力し，パケット内の情報にもとづいて入力状態メモリを更新し，イベントをセットする．

七つの各ステージはパイプライン式に並列処理可能である．しかし，次に述べる点を考慮してパイプライン化を進める必要がある．(a)すべてのブロックに対する処理がこの七つのステージをすべて経由するわけでない．(b)各ステージの処理時間は入力データによってまちまちである．(c)複数のステージ間で同一のメモリに対し，アクセスの競合が起こる．

このような問題があることから通常の計算機で行われている同期式のパイプライン処理は本マシンには有効でない．そこで，非同期パイプライン処理方式を採用することとした．各パイプラインステージはデータの到着と同時に処理にとりかかり，結果を次段に送る．さらにパイプラインステージの間に FIFO あるいはバッファをおくことにより各ステージの処理時間の不均衡によって生じる待合せをできる限り少なくするようにつとめる．**図3**に一つのプロセッサにおけるパイプライン処理を示す．プロセッサでは一つのブロックの処理におけるステージ間の並列性とブロック間の並列性の両方を利用することができる．図3においてボックス中の番号はそれぞれ処理中のブロック番号を示す．ブロック3の処理は出力ピンに変化がないため，ステージ4で終了している．ステージ 4, 5, 6, 7 の各処理は前のステージが完了する前から動作することが可能である．これはブロック内のピンごとの処理の間での並列性に着目し，一つ

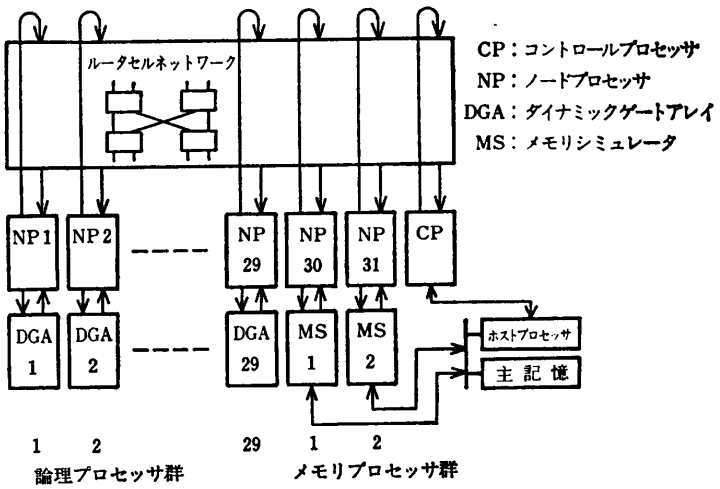

図2 超高速論理シミュレータ
Fig. 2 High speed logic simulation machine blockdiagram.

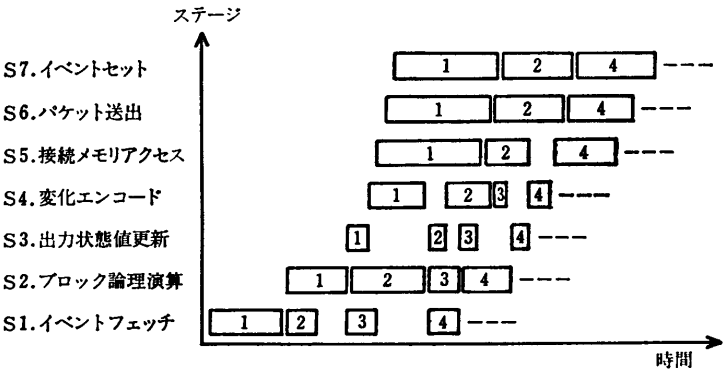

図3 パイプライン処理
Fig. 3 Pipeline processing in a node processor.

の変化ピンをエンコードしている間に前にエンコードしたピンについては接続メモリアクセス以後の処理を開始することができることによる.

6. ノードプロセッサ

ノードプロセッサはブロックの入力状態メモリ，出力状態メモリ，接続メモリを内蔵しイベント処理を行う．**図4**はノードプロセッサの構成を示したものである．前章で述べた七つのステージの処理をさらに細かい処理ステップに分け並列性をさらに増大させた．ノードプロセッサは10個の独立した制御回路をもち，イベントをパイプライン式に処理する．

ノードプロセッサはおもに次に述べる処理を行う．

（1）イベントセット/イベント取出し処理

ここで二つの処理を独立に行う．そのうちの一つは入力ピン変化イベントをパケットの形で受け付け，入力状態メモリを更新し，イベントをセットする．残りの一つは，ブロックカウンタによって順次イベントフラグを調べ，もしイベントがセットされていれば入力状態値とブロック種類を取り出す．ここでは四つの制御回路が相互に関連して動作するので次の点に考慮して設計を行った．

・ルータセル経由のイベントパケット入力部と自プロセッサ内からのイベントパケット入力部がパケット

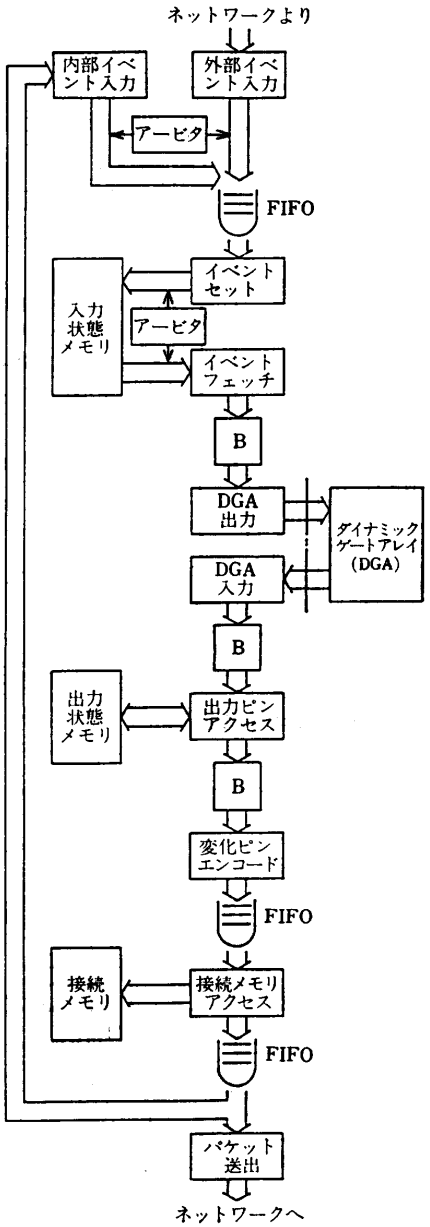

図4 ノードプロセッサの構成
Fig. 4 Event flow in a node processor.

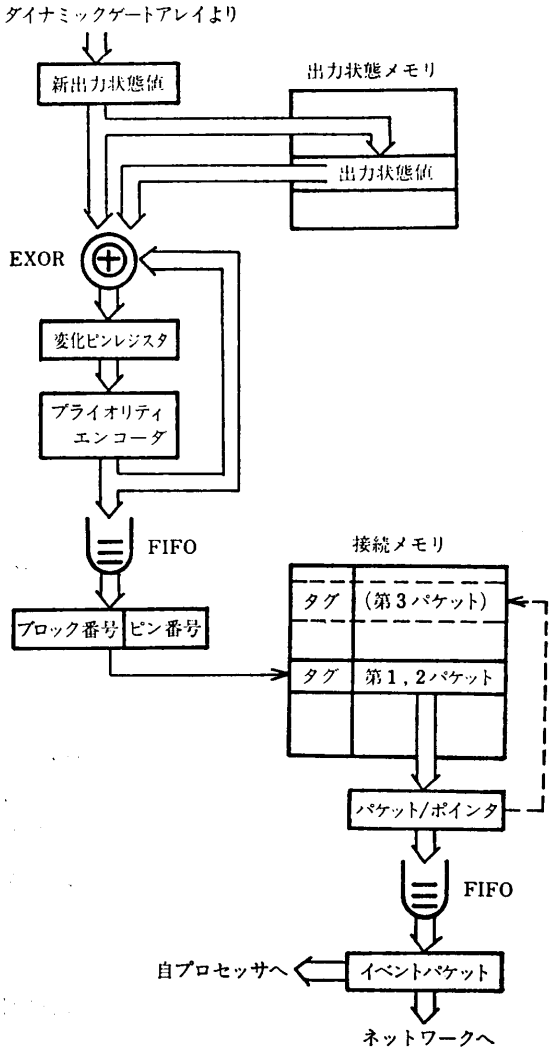

図5 出力ピン変化検出/接続メモリアクセス
Fig. 5 Update and fanout processing.

FIFO の使用で起こる競合をアービタで処理する.

・イベント取出し部とイベントセット部が入力状態メモリアクセスで起こる競合をアービタで処理する. また, メモリ更新中はメモリを占有し同時更新を防止する.

・パケット入力部とイベントセット部に FIFO をおき処理時間の変動を吸収する.

(2) 出力ピン変化検出/接続メモリアクセス

図5に処理の流れを示すように, ここでの処理はまずダイナミックゲートアレイから新しい出力状態値を得て出力状態メモリを更新する. このとき以前の出力状態値をあらかじめ変化ビットレジスタにセーブしておく. 次に新しい出力状態値と排他論理和をとり変化した出力ピン群を得る. プライオリティエンコーダを用いて1個ずつ変化ピン番号にエンコードしていく. ブロック番号と変化ピン番号を結合し接続メモリのアドレスを生成し FIFO を介し接続メモリへ送る. 接続メモリには接続情報がイベントパケットの形で記憶されていて, 一度に2パケット分読み出せる. パケットの読出しの制御は接続メモリに含まれるタグフィールドによって行う. もし, ファンアウトが3以上のときはアクセスしたデータの第2パケット目に相当するフィールドがタグの指定によって次の接続メモリの読出しアドレスとなる. 二つの FIFO をおき, リスト処理に伴う処理時間の変動を吸収することとした.

7. ダイナミックゲートアレイ

ダイナミックゲートアレイはブロックの論理演算を効率よく行うために新たに開発したハードウェアである. ダイナミックゲートアレイはブロックの入力状態値とブロックの種類を入力し, ブロックの出力状態値を出力する. ブロックは多数のゲートで構成されているが, これらのゲートは信号伝播順に従い, いくつかのグループに分けられる. 信号伝播順に従ってグループごとに論理演算処理を行いながら最終的にブロックの出力値を求める. このとき, ダイナミックゲートアレイは同一グループ内のゲートの論理演算を同時に行う.

ダイナミックゲートアレイの設計にあたり次の点を考慮した.

・ブロックの入力数は32, 出力数は32を標準とするが, 多入力/多出力も扱えるようにする.

・任意の入力数をもったゲートを扱えるようにする.

・一つのダイナミックゲートアレイで数十ブロック種まで取扱い可能とする.

・シミュレーション対象によって容易に変更可能とする.

ダイナミックゲートアレイは**図6**に構成を示すようにおもに次の二つの部分から構成される.

(1) ゲート論理演算部

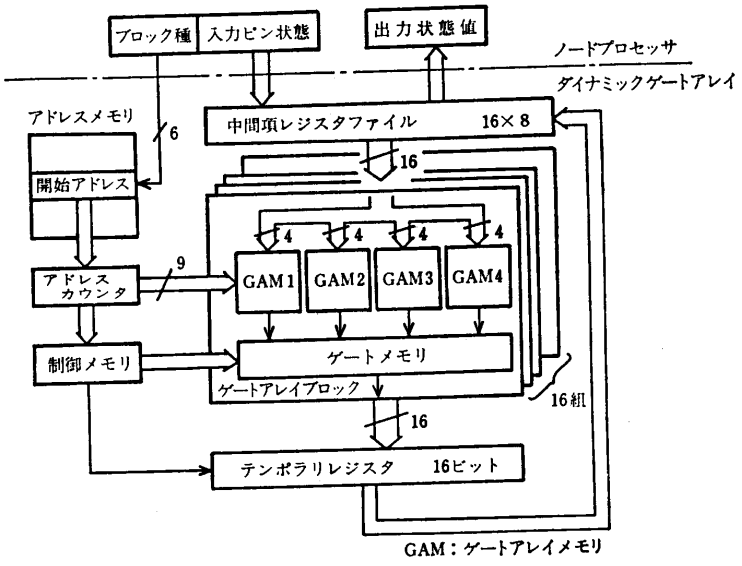

図6 ダイナミックゲートアレイの構成
Fig. 6 Dynamic gate array blockdiagram.

ゲート論理演算部では，ゲートアレイブロックによってブロックを構成するゲートの論理演算を行い，演算結果を中間項レジスタファイルに収容する．ゲートアレイブロックは16入力をもつゲートの論理演算を行う．ゲートの論理機能はあらかじめゲートアレイブロック内のメモリにビットパターンとして展開しておき，ゲートの入力状態値をメモリのアドレスとして与え，メモリの読出しデータをゲートの出力値とする．本マシンでは，メモリを2段用いてゲートの論理機能を実現し，メモリ容量を減少させることを考えた．図6に示すように初段のゲートアレイメモリは4個ずつのゲートの入力を分担する．初段の4個のゲートアレイメモリの各出力を2段目のゲートメモリの入力とすることによりゲートの論理機能を実現する．この方式によれば，各ゲートアレイメモリのアドレス線を4本ずつしか必要としない．そのため，市販の 8KRAM (アドレス13本) を用いても他のアドレスが9本残る．これを用いて512個のゲート機能を収めることができる．

高速化を図るため，ゲートアレイブロックを16組並列に用いて16個のゲートの論理演算を同時に行えるようにした．ゲートの論理演算結果は16ビット×8ワードの中間項レジスタファイルに収容する．これは，次のゲート論理演算のための入力となる．レジスタファイルには最大128ビットまで入る．このため，128本の入出力ピン数まで処理可能である．入力数が16を越えるゲートについては16個ずつの複数のゲートに分割して処理する．分割した各ゲートの演算結果をテンポラリレジスタに集計し多入力ゲートを実現した．

(2) ブロック選択制御部

ブロック選択制御部はブロック種ごとのゲート論理演算の順序制御を行う．入力したブロック種番号からアドレスメモリを参照し，ゲートアレイメモリのアクセス開始位置を得てアドレスカウンタにセットする．アドレスカウンタを順次進めながらそれぞれの評価すべきゲート番号の指定を行い，同時に制御メモリを参照しゲート種の選択，入力幅の指定および論理演算終了の判定を行う．ブロック番号に6ビットを用いて最大64種類まで収容可能とした．一つのダイナミックゲートアレイで最大512×16個のゲート機能を収めることができる．アドレスカウンタを用いて任意のアクセス開始位置を指定することができるので，ゲートアレイメモリの領域をつめて複数のブロック種を収容可能とした．

8. メモリシミュレータ

メモリ系のブロックを組合せ回路系のブロックと同様にダイナミックゲートアレイで対処すると大量の状態メモリが必要となる．さらにアドレス線や制御信号線のように共通に使用される信号もシミュレータ上ではそれぞれ別個のイベントとして処理されるので大量のイベントが発生することになり，全体のシステムの大幅な性能低下の要因となってしまう．そこでメモリ系のブロックを効率よく処理するメモリシミュレータを考えた．

設計にあたり次の点を考慮した．

・スタティック形 RAM，ダイナミック形 RAMの両者を取り扱う．

・メモリチップ単体および複数のメモリチップを集成した集合メモリブロックの両方を取り扱う．

・メモリ容量は最大 2MB まで収容する．

・ダイナミックゲートアレイと同一のインタフェースを守る．

図7がメモリシミュレータの構成を示したものでおもに次の二つの処理を行う．

(1) アクセス状態の判定

メモリブロックはメモリアドレス，書込みデータおよびアクセス制御信号群を入力し，読出しデータを出力とする．メモリ系のブロックは以前のアクセス制御信号等の状態と現在の入力状態とで次の状態が定まる点が組合せ系のブロックと異なる．したがって，各メモリブロックごとに以前の状態を記憶しておくためのアクセス状態メモリと，新しい入力状態値を入力したときにアクセスを検出しアクセス状態メモリを更新するアクセス制御回路を設けた．アクセス制御回路はメモリブロックの種類に応じてリード/ライトサイクル，あるいは他の特殊サイクル（リードモディファイライト，部分アクセス等）の所定の動作をシミュレートする．

(2) ホストメモリアクセス

メモリシミュレータの基板上に実際のメモリブロックのデータを記憶させることは容量的に制約があること，記憶データのセーブ/リストア等のシステム運用面で不都合が大きいことがわかった．そこで本シミュレータではホストとなるミニコンピュータとメモリシミュレータとのメモリ共有方式を採用し，ホストの主記憶の一部をメモリブロックの記憶域に使用すること

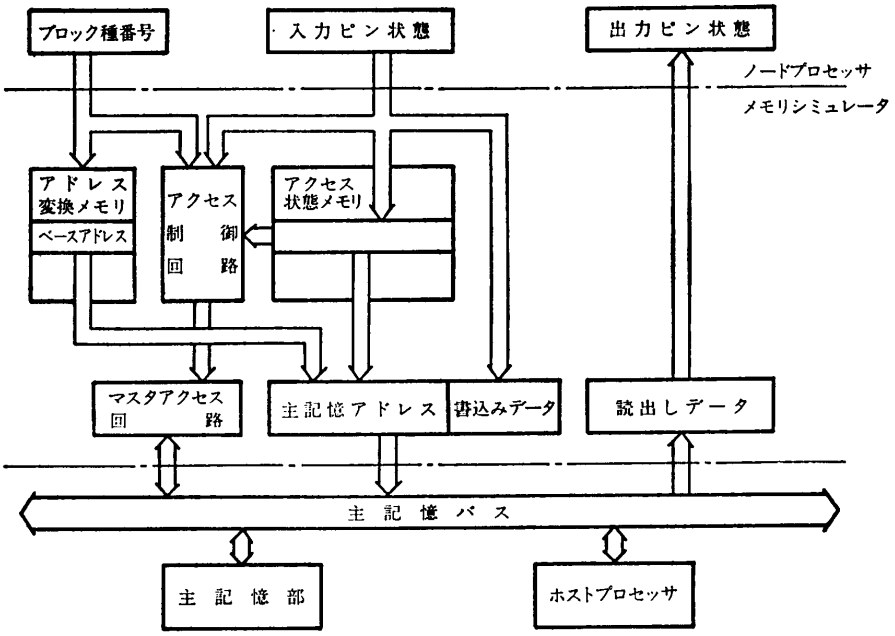

図 7　メモリシミュレータ
Fig. 7　Memory simulator blockdiagram.

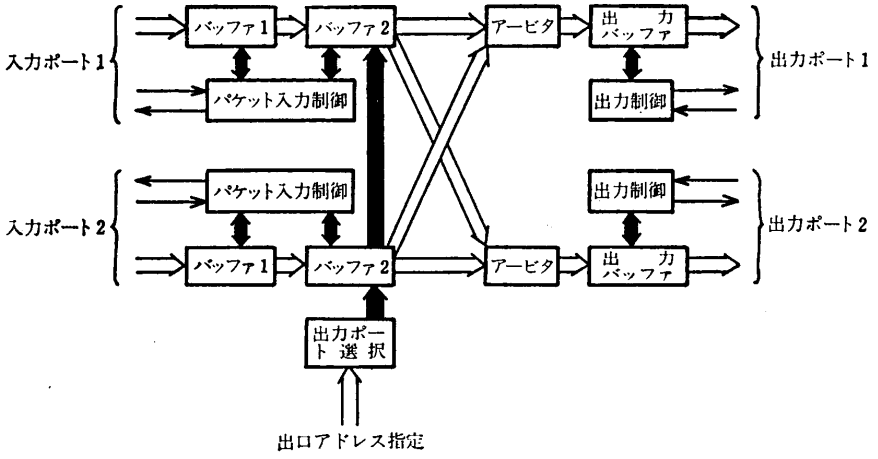

図 8　ルータセル LSI の構成
Fig. 8　Router cell LSI blockdiagram.

とした．メモリシミュレータでは各ブロックごとに主記憶上の領域を対応させるためのアドレス変換部と，ホストのバス上で直接メモリアクセスするためのマスタアクセス部をもたせることとした．ホストのバス上では16ビット単位にまとめてメモリアクセスを行うことによりバス上のトラフィックを軽減することができる．

メモリアクセスが検出されるとアドレス変換メモリを参照し，ブロックに対応するアドレスデータを読出しブロック内アドレス部と結合し，主記憶アドレスを生成する．マスタアクセス回路はアクセス制御回路の指令によって主記憶バス上でメモリアクセスを行う．メモリシミュレータはホストのメモリバスの最大アドレス幅まで使用可能である．実際には2MBをこのシミュレータ用のメモリ領域にあてる．

9. ルータセルネットワーク

24ビットのイベントパケットをプロセッサ間で転送するために新たな結合ネットワークとして蓄積交換形の多段接続ネットワークを考えた．さらにネット

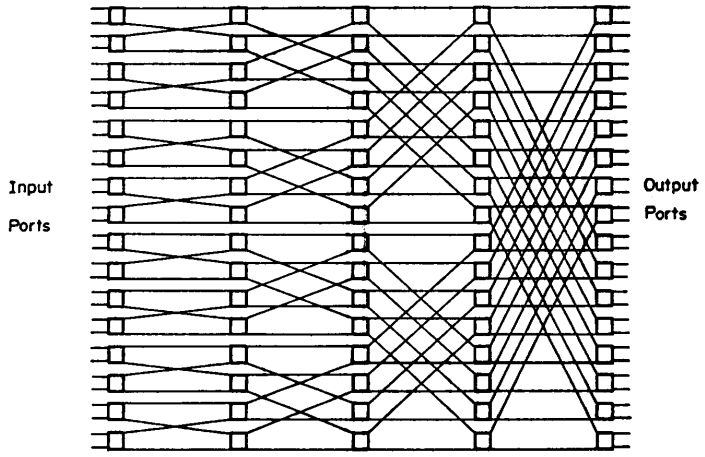

図 9 32入力32出力ルータセルネットワークの構成
Fig. 9 32 input 32 output router cell network.

ワークをモジュール式に構成するのに適したルータセルを LSI 化し，これを用いることとした．ルータセルを製作するにあたり次の点を考慮した．

・自動アドレシング方式としてルータセルごとにパケット内の特定の1ビットで出口ポートの指定を行う．

・セル間のパケット転送は非同期式とする．

・高速化を図る．

・入出力ポートにパケットのバッファをおき，パケットの衝突を防止する．

ルータセルは MOS マスタスライス LSI を用いて製作した．2入力2出力のルータセルを約1,000ゲートで実現し，64ピンのパッケージに収容した．ルータセルの構成を図8に示す．一つのパケットを8ビットずつ並列に3回に分けて転送する．ルータセルには各入口に2パケット分と各出口に1パケット分のバッファをもたせてある．1回のパケット転送に 400 ns を要する．出口ポートの指定はパケットの先頭8ビット内のあらかじめ設定されたビット位置のデータで判定する．もし，二つのパケットが同時に同じ出口ポートを指定すると競合が起こる．出口ポートにあるアービタが早いもの勝ち式に処理する．

このルータセルを80個用いて32入力32出力ネットワークを構成したのが図9である．1列16個の LSI を5段に接続したものであり，並列かつパイプライン式にパケットの転送を行う．

10. コントロールプロセッサ

コントロールプロセッサはホストのミニコンピュータの指令によってシミュレーションの実行制御を行う．

コントロールプロセッサを設計するにあたり次の点を考慮した．

・シミュレーション性能を低下させないような制御方式を用いる．

・他のプロセッサと同様のルータセルネットワークのインタフェースをもたせる．

・ホストマシンの負荷を軽減する．

コントロールプロセッサはおもに次の処理を行う．

(1) シミュレーション非実行時の処理

本シミュレータは実行モードと非実行モードの二つのモードがある．非実行モードではルータセルネットワークを各プロセッサの状態メモリ等のデータ転送のために用いる．コントロールプロセッサはホストから DMA でデータを受け取り，パケットの形にして所定のプロセッサに転送する．また，反対にそれぞれのプロセッサからくるパケットを分解し，データをホストに DMA で転送する．

(2) シミュレーション実行時の処理

実行モードではシミュレーションの実行あるいは停止，ならびにレベルとクロックの同期処理を行う．同期処理のためにレベルカウンタとクロックカウンタおよびレベル上限レジスタを用いる．クロックカウンタはシミュレーションすべき所定のクロック数を，レベル上限レジスタには1クロックの最大レベルをあらかじめセットしておく．ホストよりシミュレーション実行の指令がくるとコントロールプロセッサはレベル開始指令を全プロセッサに通報する．このとき，レベルカウンタを一つ進める．それぞれのプロセッサは現在のレベルに属するすべてのブロックのシミュレーション処理を終了するとレベル終了信号を出す．すべてのプロセッサがレベル終了状態になるとコントロールプロセッサは次のレベル開始を再び通報する．このとき，もしレベルカウンタがレベル上限レジスタと等しいときはレベル開始を通報する前にクロック終了を全プロセッサに通報し，次のクロックのシミュレーションを開始する．各プロセッサはクロック終了を受けるとブロックカウンタを初期化し，最初のレベルからイベント取出しを開始する．コントロールプロセッサはクロックカウンタを一つずつ減じ，カウンタの値が0になるまでシミュレーション実行制御動作を繰り返す．

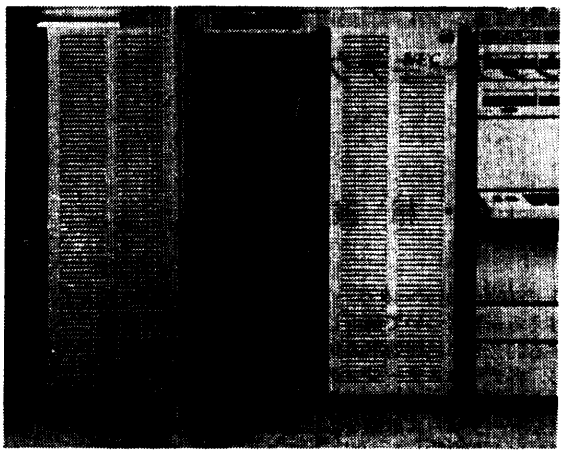

図 10 超高速シミュレータの外観
Fig. 10 An overview of the high speed logic simulation machine.

実行中のコントロールプロセッサのもう一つの役割はシミュレーション実行中に起こる例外事象の処理がある．各プロセッサで実行中に誤りや例外事象が発生するとトラップが生じ，これに対応するパケットがコントロールプロセッサ宛に送られる．コントロールプロセッサはトラップのパケットを受けるとホストに割込みを発生し，例外処理の依頼をする．

図10が本シミュレータのプロトタイプとして製作した超高速論理シミュレータの外観を示す．システムはルータセル LSI を除きすべて標準の IC を用いて構成した．プロセッサ部は30×35インチのマルチワイヤボードで製作した．ラック3本に本体を収容し，システム全体で約25,000個の IC を使用した．

11. むすび

本論理シミュレーションマシンはプロトタイプ1号機のハードウェアを完成し，支援ソフトウェアを開発中である．現在数千ブロックからなるテストモデルによる動作確認を終え，従来のソフトウェアによるシミュレータより数千倍以上の速度で実行されることが明らかになった．とくに高性能が実現できた理由として

・専用ハードウェア群による並列パイプライン方式をとったこと．

・ブロックレベルシミュレーション方式を実現し，ブロックの論理演算を行う専用のダイナミックゲートアレイを用いたこと．

・ルータセル LSI を開発し，高速かつ大容量なネットワークを用いたこと．

等をあげることができる．

なお，本シミュレータを実際の装置開発用ツールとして大規模システムへ適用したときの分析評価結果についての報告は別の機会にゆずることとする．

謝辞 最後に研究の機会を与えてくださった，日本電気（株）C&C システム研究所三上所長代理，箱崎部長，山本課長，コンピュータ技術本部斉藤本部長，桑田部長，情報処理製造・装置システム事業部山田技師長，病院情報システム事業部北野事業部長代理に深謝します．数々の提案をいただいた第二 OA 装置事業部赤井主任，伝送通信事業部富田主任に感謝します．また，ハードウェアの設計・製作を担当してくださった三野輪氏，近藤氏，幅田氏，ソフトウェアサポートシステムの製作を担当してくださった堀田主任，野水主任，伊藤氏，田中氏に感謝します．

参考文献

1) 小池，大森，佐々木：論理シミュレーションマシンのアーキテクチャ，情報処理学会論文誌，Vol. 25, No. 5, pp. 864-872 (1984).
2) 小池，大森，佐々木：論理シミュレーションマシン，信学研資，EC 82-42 (1982).
3) 大森，小池，佐々木：論理の動的可変なダイナミックロジックアレイ，信学研資，EC 82-12 (1982).
4) Koike, N., Ohmori, K., Kondo, H. and Sasaki, T.: A High Speed Logic Simulation Machine, Compcon 83 Spring, pp. 446-451 (1983).
5) Sasaki, T., Koike, N., Ohmori, K. and Tomita, K.: HAL: A Block Level Hardware Logic Simulation, 20th DA Conf., pp. 150-156 (1983).
6) Pfister, G.F. : The Yorktown Simulation Engine; Introduction, 19th DA Conf., pp. 51-54 (1982).
7) Abramovici, M. et al.: A Logic Simulation Machine, 19th DA Conf., pp. 65-73 (1982).

(昭和59年2月13日受付)
(昭和59年4月17日採録)