

CAD高速化のための Thread Processor TP5000

下郡慎太郎 高山浩一郎 松岡英俊 平原貴音 広瀬文保
(株)富士通研究所

概要 データパイプライン処理に適したハードウェアアクセラレータTP5000を開発した。LSI-CAD のように大量のデータに同質の処理を繰返し実行するようなアプリケーションにおいては、処理を複数のステージに分割し、その上をデータが流れるようにしたデータパイプライン処理が適している。TP5000は問題に合わせてパイプラインを再構成可能なデータパイプラインマシンである。パイプライン1ステージを実現するPE(MPE)では、ALU、ローカルメモリなどの複数のファシリティを同時にコントロールすることで、高速性と汎用性を実現している。また、MPE間を静的な設定と動的な切り換えが可能な高速ネットワークで接続することにより高いスループットと再構成自由度を達成している。80MPEを使用して、8パイプラインを並列に動作させた場合には、VHDLシミュレータでソフトウェアの数百～千倍、min-cut法による配置プログラムで2～3百倍の性能が出ることが見積もられた。TP5000では最大5120MPEまで接続できるので、最大構成時にはさらに1桁以上の高速化が期待できる。

Abstract We have developed a hardware accelerator TP5000 which uses the data-pipeline architecture. The data pipeline processing is suitable for the applications that treat the large amount of data such as LSI-CAD. In the data pipeline architecture, the whole processing will be partitioned into many small stages, and the data will flow on the pipeline successively. TP5000 is a data-pipeline hardware that can reconfigure the pipeline structure according to the problems it solve. The MPE, which executes one pipeline stage, can control all facilities, such as local memories and ALUs, concurrently, freely, and quickly. The inter-MPE network, which utilizes static setup and dynamic routing, has enough throughput and reconfigurability to realize many kinds of pipeline. The VHDL simulator is estimated to be at several hundred to one thousand times as fast as the software program when using 80 MPEs or 10 parallel pipelines. The min-cut partitioning program is estimated to be at a few hundred times as fast as the software. As TP5000 is scalable up to 5120 MPEs, the maximum performance is expected to be more than ten times faster than the speed estimated above.

1. はじめに

最近のLSIの高密度化と高速化には目ざましいものがある。それに伴って、大量のデータを高速に扱う要求は大きくなる一方である。例えばLSI-CADの分野では、1LSI 100万ゲート規模の時代がすぐ目の

前まで来ている。シミュレーション・配置・配線などのLSI-CADアプリケーションではこのような大規模な問題を扱う必要が出てきている。CADアプリケーションにおいては、処理時間がデータの大きさに比例することは希であり、多くの場合、それ以上の割合で処理時間が増大していく。そのためソフトウェアによる実行は限界に近付いており、その解決手段として種々のハードウェアアクセラレータが開発されている[1, 2]。ところが、これらのアクセラレータは特定のアプリケーションのみを高速化するもの

Thread Processor TP5000 for CAD Acceleration

Shintaro SHIMOGORI, Koichiro TAKAYAMA,
Hidetoshi MATSUOKA, Kioto HIRAHARA,
and Fumiyasu HIROSE

FUJITSU LABORATORIES LTD.

であり、LSI開発全体のスループットの向上は十分とは言えない。また、単機能であるが故に導入コストが割高であり、広く使われるまでには至っていない。

そのため、汎用並列計算機を使用した高速化[3]や、プログラマブルなアクセラレータによる高速化[4]も試みられている。しかしながら、専用アクセラレータを用いた場合ほどに効果を上げるまでには至っていない。

そこで、問題の特徴をうまく生かした汎用的なアーキテクチャを用いることにより、複数のアプリケーションを高速化できれば、全体のスループット向上が期待できると同時に、アプリケーション毎にアクセラレータを開発・使用するよりも安価となるはずである。筆者らはLSI-CADアプリケーションが共通に持つ性質を調査した上で、これらのアプリケーションを高速処理するアーキテクチャを検討し、Thread Processor TP5000を開発した。

本論文の第2章では、LSI-CADアプリケーションの性質を検討し、データパイプライン処理が最適な方法であることを示す。第3章では、TP5000のアーキテクチャを説明する。第4章では、TP5000の性能見積りを述べる。

2. CADの性質とデータパイプライン処理

2. 1 LSI-CAD問題に共通な性質

LSI-CADでは、非常に大規模なデータを扱う。その性質をいくつかのアプリケーションを例にとって考えてみよう。まず、図1にイベントドリブン論理シミュレーションの処理の流れを示す。信号の変化を表すイベントはイベントキューに蓄えられ、管理される。ある時刻になったとき、その時刻のイベントが取り出されて、そのファンアウトゲートの評価を行う。このとき、ゲートの種類により異なる評価を行う。出力ピンに変化があればそれを新たなイベントとしてイベントキューに登録する。各イベントは独立に発生するため、これらは並列に処理できる。

次に、min-cut法による配置プログラムの処理の流れを示す。min-cut法は、回路全体を2分割したときに分割されるネット数(cut数)を最小にする問題である。その中でも有名な Fiduccia と Mattheyses により提案された方法[5] (以下FM法) では、図2のような処理を行う。初期分割 (たとえばランダム分割) により回路を二つの領域に分けた状態から始める。あるセルを反対側の領域に動かしたときのcut数の増減をgainと呼び、gainの大きなセルから順に相手側に動かしていく。全セルを動かしたとき、cut数が最も少なかった組合せを次のステップの初期解として再びgainの大きいものから順に移動させていく。この処理を改善が行われなくなるまで繰り返す。図3にFM法の処理フローを示す。FM法ではまず、全セルを管理しているゲインリストから、最も大きなgainを持つセルを取り出す。次にそのセルに接続しているネット、そのネットに接続しているセルを順に取り出して行き、最初のセルが動いた場合のゲイ

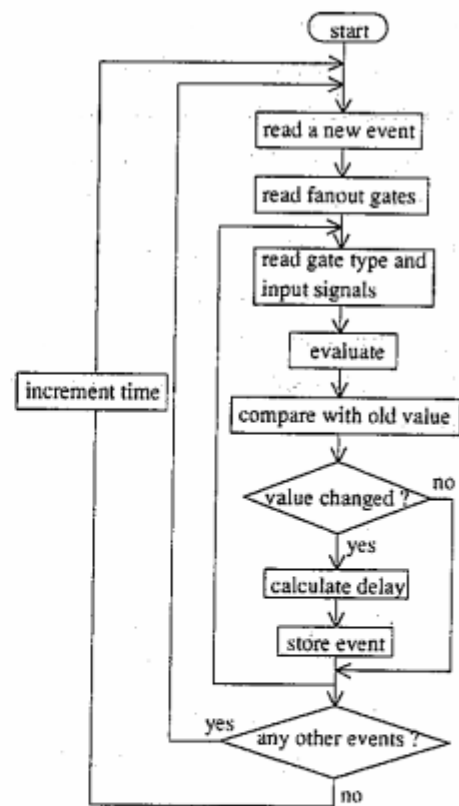


図1 イベントドリブン論理シミュレーションの処理フロー

ンの変化をそれぞれ計算する。1個のセルを動かした場合、他のセルのゲインが変わり得るので、移動するセルの選択は独立に行うことはできない。しかし、ネットとセルの取り出し、およびゲインの更新は独立に処理できる。

これらの問題を見ると、一定の条件のもとで各データの処理は独立であり、各処理(図1, 3の箱)をパイプラインステージと捉えてその上をデータが流れるようにするデータパイプライン処理が適していると考えられる。この場合、以下の共通点がある。

- 1) パイプラインはメインとなるthread(パイプライン全体)と、各パイプラインステージをさらにパイプライン処理するサブthreadとに分けて考えられる。
- 2) 多くのパイプラインステージではメモリアクセスを必要とする。また、アクセスされるメモリ内容は多くの場合ステージ毎に独立にすることができる。
- 3) 各ステージで処理すべき内容は非常に単純なものである。
- 4) データによって、ステージ内(サブthread)の処理内容に少しずつ違いが生じる得る。しかし、パイプライン全体の流れはあまりデータによらない。

これらの性質を考慮した上で、データパイプライン処理に適したハードウェアとはどのようなものか検討してみる。

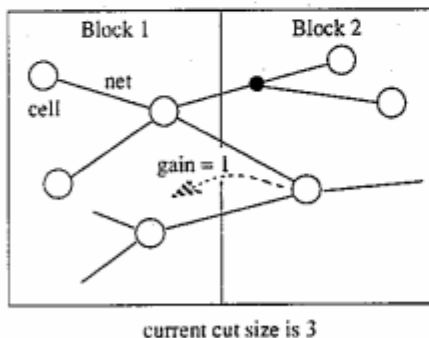


図2 FM法による回路分割

2. 2 データパイプラインアーキテクチャ

イベントドリブン論理シミュレーションとFM法による配置をデータパイプライン処理で構成してみた例をそれぞれ図4, 図5に示す。これらの例では、パイプラインの各ステージに独立なローカルメモリ(LM)を持たせている。前のステージから渡されたゲート番号等を元にローカルメモリから信号値やゲートタイプを読み出したり、新しい信号値を書き込んだりする。また、信号値の比較やイベントキューの管理等の処理も行う。

また、パイプラインのある1ステージの処理の例として、論理シミュレーションのファンアウト取り出しステージでの処理を図6に示す。

これらの例からデータパイプライン処理に最適な

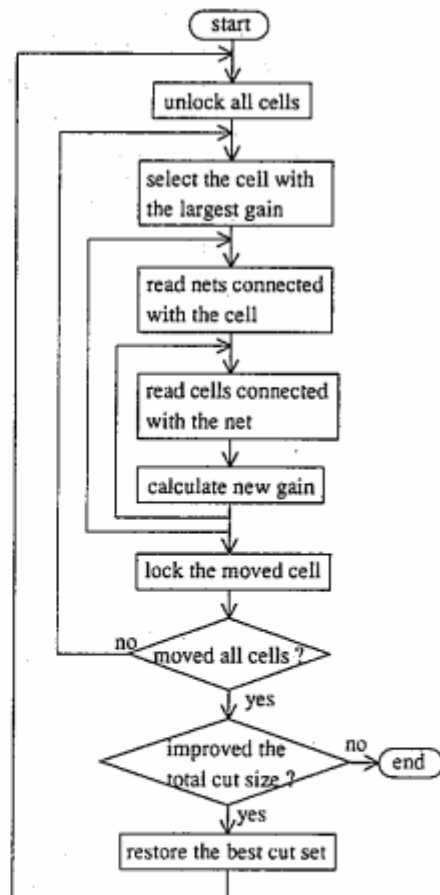


図3 FM法の処理フロー

ハードウェアアーキテクチャは次のようなものであることがわかる。

確保するためには高速な通信を実現するネットワークが必要である (3.2を参照)。

1) パイプラインの1ステージを単位として見た場合、その接続には自由度よりも高速なデータ通信のほうが重要。

2) パイプラインの1ステージ (sub thread) の実現には大きな自由度が必要。

図4, 5からわかるように、パイプラインの構成は比較的シンプルであり、データはほぼ一方向に流れる。従って、MARS[4]の完全結合のような密な接続は必要なく、ある程度のパイプライン再構成自由度があれば良い。パイプラインの高スループットを

パイプライン1ステージ内の処理 (図6) については、データによって処理を動的に変える必要があること、処理の流れがフィードバックループを形成するような複雑なものになり得ることを考慮して、演算器間が密に結合しており、組み替えの自由度が高いほうが良い。さらに、パイプラインの実行速度

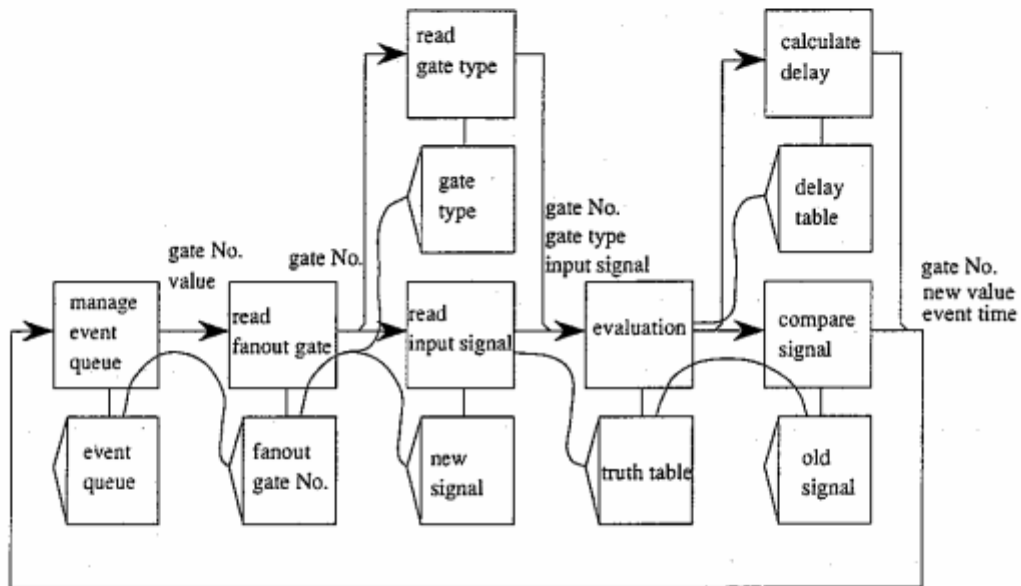


図4 データパイプラインによる論理シミュレーションの実現

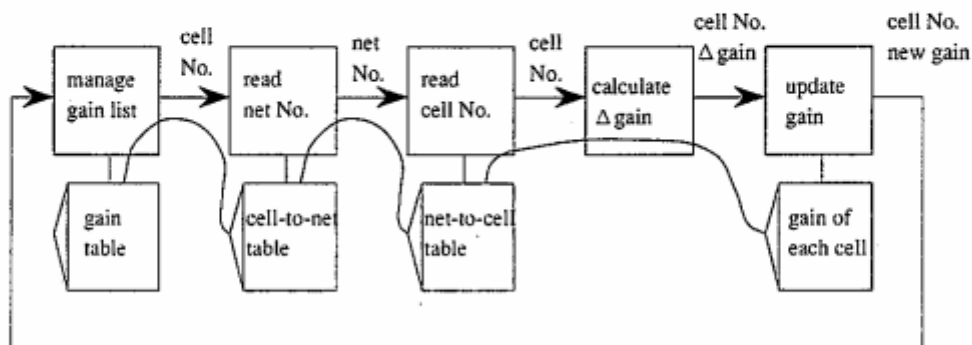


図5 データパイプラインによるFM法の実現

は最も遅いステージの実行速度によって決るため、できる限り少ないクロック数で多くの処理をできるようにする必要がある(3.1を参照)。

3) 各パイプラインステージには高速なローカルメモリを持たせる。

CADアプリケーションでは一般に非常に大規模なデータを扱う。さらにメモリアクセスの大部分がランダムなアドレスへのアクセスとなるため、キャッシュの効果が出にくいという問題がある。データパイプラインアーキテクチャとしてはデータをローカルメモリに分散させて持たせることによりメモリバンド幅を広げるとともに、高速RAMを使用してランダムなアドレスへのアクセスにも対応できるようにするのが望ましい(3.1を参照)。

3. Thread Processor TP5000

2章で説明したような、CADアプリケーションに共通の性質とデータパイプラインに最適なアーキテクチャを検討し、Thread Processor TP5000を開発した。

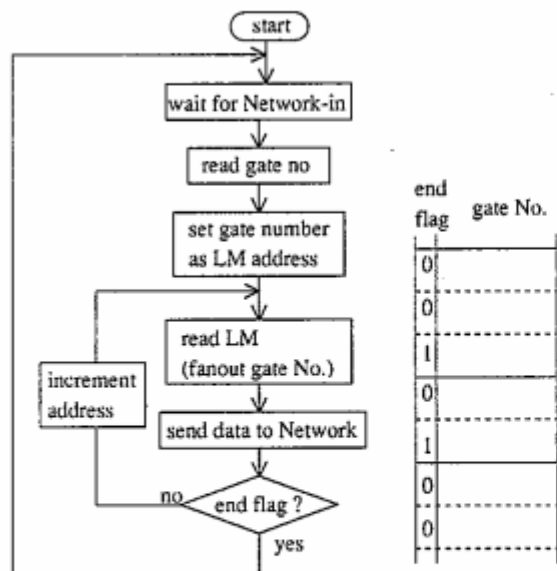


図6 論理シミュレーション「ファンアウト取り出し」ステージの処理フロー

TP5000は、パイプライン1ステージを処理するためのMPE、それを10個接続して1パイプラインを構成するための processor group (PG)を基本構成とする。PGは最大512個(5120MPE)まで接続できる。

3.1 TP5000におけるパイプラインステージ (サブthread) プロセッサ (MPE)

図7にMPEの構成を示す。パイプライン1ステージで、できるかぎり多くのファシリティを同時に動作させ、かつその組合せを柔軟にするため、複数のファシリティをクロスバスイッチで結合し、その制御をVLIW型マイクロコードにより行っている。その特徴は以下の通りである。

- 1) 2個のローカルメモリ、2個のALU、デュアルポートメモリ、ネットワーク等のファシリティを同時に動作させることができる。
- 2) 全ファシリティはクロスバスイッチにより結合されているため、その組み替えを柔軟に行える。
- 3) 各PEは高速SRAMによる2個のローカルメモリを持っているため、メモリアクセスは非常に高速である。
- 4) 同時に4アドレスへの分岐条件を処理できる。
- 5) 256bit幅VLIW型水平マイクロコードにより、上記制御を動的かつ高速に行える。

図8に、論理シミュレーションのファンアウトゲート取り出しステージをMPEのマイクロコードで実現した例を示す。このように、1クロックで多数のファシリティを動作させることが可能であり、最も頻繁に処理されるコード(図6の内側のループ)は1クロックで実行している(図8のアドレス2)。これはRISCコード5命令分の処理に相当する。

また、MPEのネットワーク部分を使用し、MPEとピン互換のLSIを開発することにより、アプリケーションに固有かつ重い処理を専用LSI化することが可能である。現時点では、論理シミュレーション専用のLSI(WPE)が用意されている。

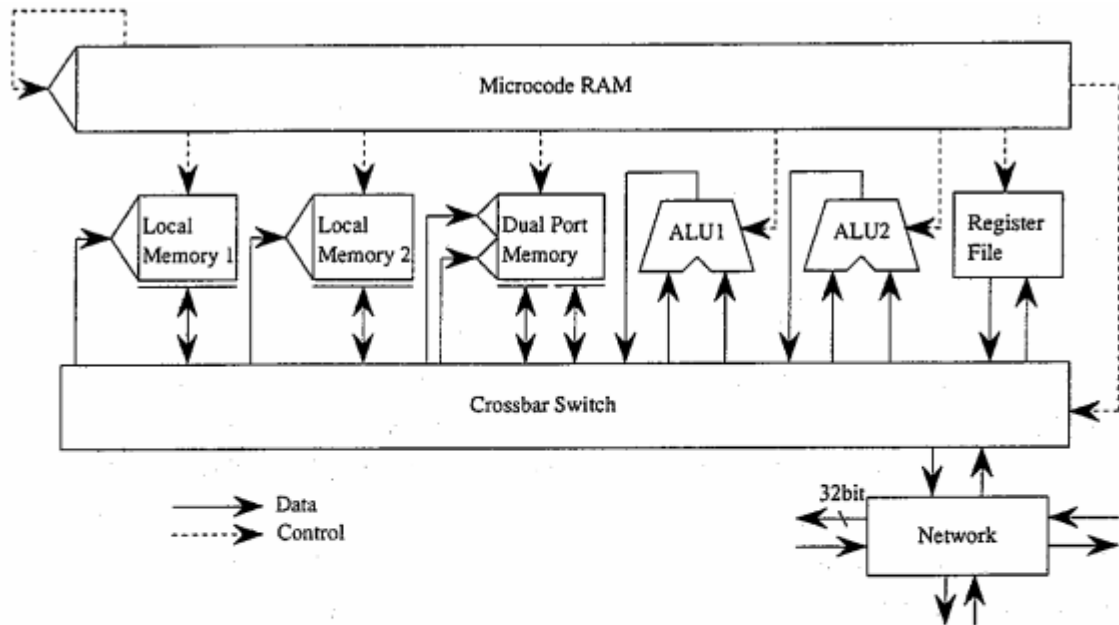


図7 sub-thread processor MPE のブロックダイアグラム

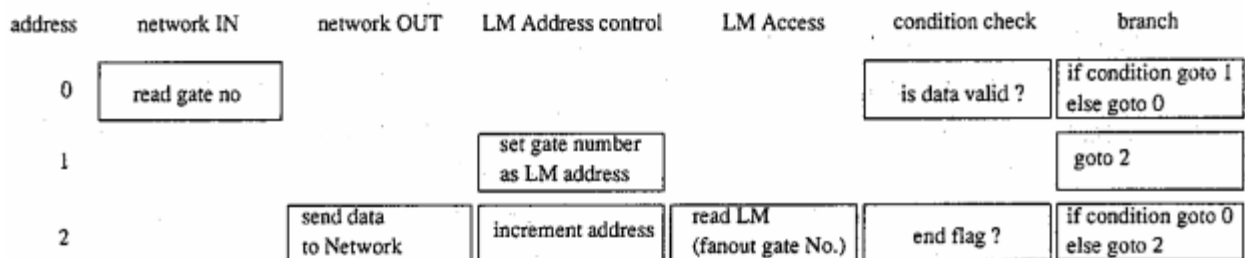


図8 論理シミュレーション「ファンアウト取り出し」ステージのMPEでの実現

3. 2 TP5000におけるパイプラインの構成

2. 2で述べたように、パイプラインの構成において最も重要なのは、PE間の転送速度である。TP5000では、各MPEに32bit幅・往復のネットワークを3対持たせている。転送速度は1ワード/クロックノポートであり、非常に高速である。また、パイプラインの組み替えについては静的設定と動的な切り換えが可能である。

MPE内部とは、上記3対のネットワークとは別の1往復のネットワークにより接続している。入力と出力とは同時に実行可能である。

TP5000では、パイプライン用に密に結合されたMPE群のことをプロセッサグループ(PG)と呼んでいる。1PGは10個のMPEにより構成されている(図9)。パイプラインの実現に10MPE以上必要な場合は、PG間のネットワークを使用してさらにMPE数を多くとることも可能である。

1PGは1枚のプリント板で実現されている。1ボードにはMPE用の10個のLSIの他にPG間ネットワーク用3個、制御用2個の合計15個のLSIを搭載している。

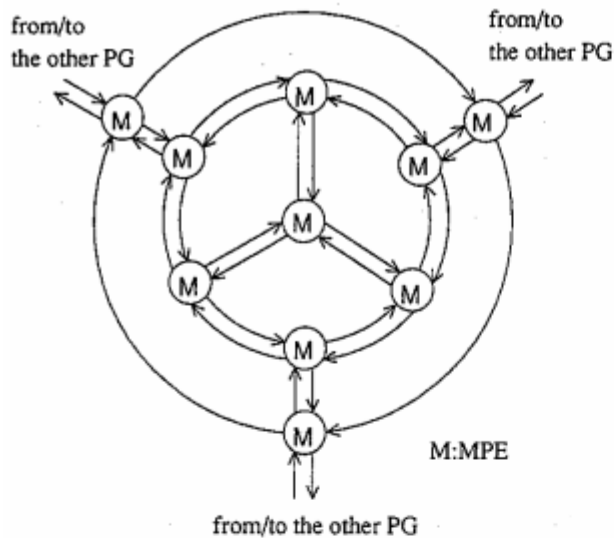


図9 MPE間 (1PG) の接続

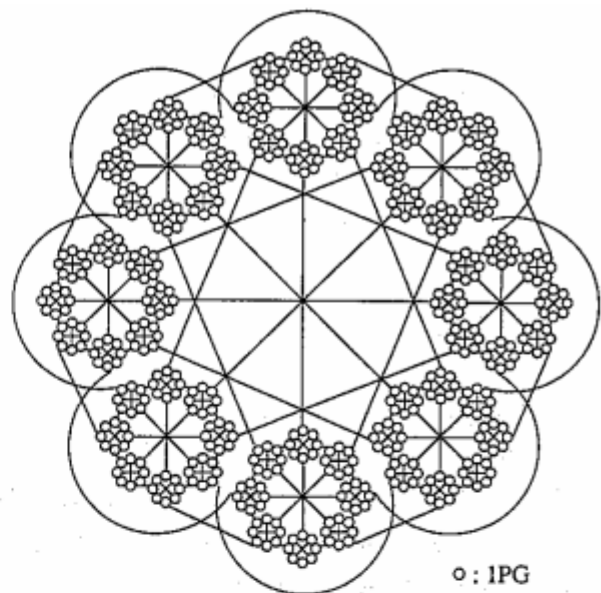


図10 512PGの場合のToF-Net構成例

3. 3 ネットワークの再構成

TP5000では、目的のデータを分割し、複数のPGに与えることでデータ並列処理も可能である。

PG間の接続には、対象とする問題により再構成が可能なネットワーク ToF-Net (Topologically Flexible Network) [6]を使用している。ToF-Net は、階層アドレスの使用により、アプリケーションのデータのローカルティに応じてトポロジーを組み替えることのできるネットワークである。LSI-CAD のようにローカルティの高いアプリケーションをターゲットとして考えた場合、図10のようなトポロジーが有効である。なお、現在評価しているのは8PG構成のため、図11のような接続としている。各PGには32bit幅往復のネットワークが3対あり、隣接PG間では非常に高速な通信が可能のため、ローカルティが高い問題には特に有効である。

また、放送機能により、アプリケーション実行用のマイクロコードプログラムなどはホスト計算機から高速にロードできる。

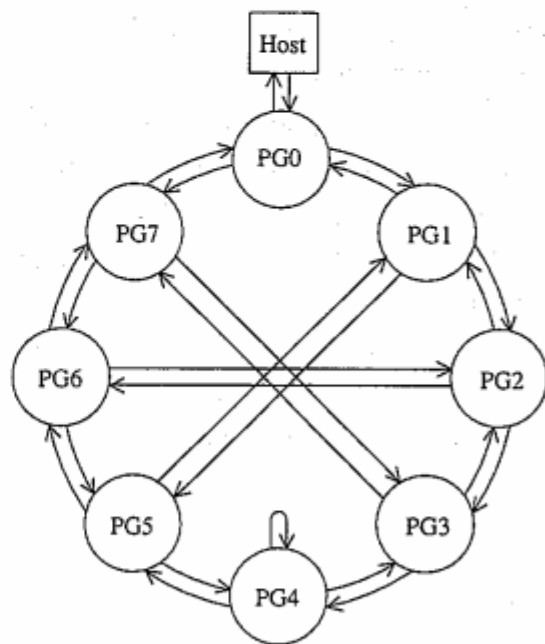


図11 8PGの場合のToF-Net 構成例

3. 4 Scalability

ToF-Net では階層アドレスによる転送を実現しており、今回開発したLSIでは $8 \times 8 \times 8$ の最大512PG (5120MPE)まで接続することが可能である。

また、コントロール系についても8個単位での階層コントロールにより、任意個のMPEを接続することができる。

4. 性能見積り

TP5000は、8PG構成での試作機が完成し、現在性能評価を行っている。TP5000用のアプリケーションは、ホスト計算機上でデータの作成・結果の表示等を行うホスト計算機側のプログラムと、TP5000上でMPEを制御するためのマイクロコードプログラムとからなる。現在、VHDLシミュレータ[7]、min-cut配置プログラム、詳細遅延付き論理シミュレータの各アプリケーションが開発中である。

これらのうち、VHDLシミュレータについては、論理シミュレーション用特殊LSI(WPE)を用いることにより、8PG構成のTP5000で、ワークステーション上のソフトウェアによるVHDLシミュレータの数百～千倍の速度を見込んでいる。現在マイクロコードのチューニングと実データによる速度の測定を行っている。なお、パイプラインの1ステージは2クロックで実現している。

一方、min-cut配置プログラムでは、2百～3百倍の速度を見込んでいる。この数値は、ソフトウェアでの処理回数と、TP5000での処理クロック数から見積もったものである。パイプライン1ステージは4クロックで実現できる予定である。

5. まとめ

本論文では、LSI-CADのように大量のデータを処理するアプリケーションにはデータパイプラインアーキテクチャが向いていることを指摘した。そして、データパイプラインアーキテクチャに基づいて開発した Thread Processor TP5000について、パイプ

ライン1ステージを担当するMPEと、MPE10個を接続してパイプラインを構成するPGの説明を行った。

8PGよりなる試作機でもソフトウェアの数百～千倍の速度を達成できる見通しである。

今後は、アプリケーションのさらなるチューニングと評価を行う一方で、他のアプリケーションへの応用も考えていく予定である。また、マイクロコードプログラムの自動コンパイルについても研究を行う予定である。

参考文献

- [1] T.Blank, "A survey of hardware accelerators used computer-aided design", IEEE Design and Test of Computers, Vol. 1, pp.21-39, Aug. 1984.
- [2] F.Hirose et al., "Simulation Processor SP", Proc. of Int. Conf. on Computer Aided Design, pp.484-487, Nov. 1987.
- [3] R.B.Mueller-Thuns et al., "VLSI Logic and Fault Simulation on General-Purpose Parallel Computers," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, pp.446-460, Mar. 1993.
- [4] P.Agrawal et al., "MARS: A Multiprocessor-Based Programmable Accelerator," IEEE Design and Test of Computers, pp.28-36, Oct. 1987
- [5] C.M. Fiduccia et al., "A linear-time heuristic for improving network partitions," Proc. 19th Design Automation Conf., pp. 175-181, 1982.
- [6] H. Matsuoka et al., "Topologically Flexible and Highspeed Network (ToF-Net)," Proc. of the Seventh Annual IEEE Int. ASIC Conf. and Exhibit., pp.300-303, Sept. 1994.
- [7] M.Shoji et al., "VHDL compiler of behavioral descriptions for ultrahigh-speed simulation," Proc. of Second Asian Pacific Conference on Hardware Description Languages, pp.85-88,1994. Sept. 1994.