1　はじめに

共有メモリ型のマルチプロセッサは、並列プログラムの実装、特に並列化コンパイラにより生成されるコードの実装に大きな利点を持っている。共有バスとスヌープキャッシュを用いた共有メモリ型並列計算機はようやく商品化の段階に達したが、一方で既にその限界が見えはじめている。この型のマルチプロセッサの本質的な問題は、共有バス、共有メモリをアクセスできるプロセッサが一時に一つに制限される点にある。最近のプロセッサの性能向上はめざましく、いかにスヌープキャッシュの構成を工夫し、バスの高速化を行なっても、結合できるプロセッサ数は4程度に制限され、しかも各プロセッサの性能を十分に発揮できない場合が多い。

そこで、大規模な共有メモリ型マルチプロセッサを構築するためには、共有バスとスヌープキャッシュで結合されたプロセッサをクラスタとし、これらを高速なリンク結合ネットワークで結ぶ構成が必要になる。このような構成に基づく大規模分散共有型メモリシステムは、コンシステンシィモデル、レイテンシィの隠蔽、キャッシュの構造等、様々なレベルでの研究、実装が行なわれている[1][2][3][4]。

しかし、これらの共有バスを基にした分散共有型マルチプロセッサは、クラスタ内外の交信形態が全く異なり、クラスタ間の交信に時間のかかるものが多い。さらに、多くの科学技術計算では、同一プロセッサでのデータ再利用が少ないため、結局、共有バス、メモリの混雑で性能の向上が制限される。この問題はクラスタ間の交信についてはさらに大きくなる。また、並列化コンパイラ、静的スケジューラの扱う対象としては、4プロセッサから成るクラスタのサイズは小さ過ぎる。しかし、より大きなプロセッサ数に対してはクラスタ間交信が入るのでスケジュールが難しくなる。

これらの問題は、接続できるプロセッサ数の限界が小さい共有バスと、本来メッセージ転送用に開発された高速ネットワークを利用して大規模分散メモリシステムを実現しようと試みている点から生じている。将来の実装技術から考えると、16-128程度のプロセッサは1ボードに実装可能であり、この程度のプロセッサ数でクラスタを構成するのが有利である。そこで、我々はクラスタ内外を結合する結合網として、多段結合網(Multistage Interconnection Network :MIN)に着目した。MINはプロセッサ–メモリ間の結合方式として長い間研究が行なわれたにもかかわらず、広く一般的に用いられるに至っていない。これは、多数のスイッチングエレメントを必要とすることによる実装の難しさとコストの大きさに加え、hot spotによる性能低下を回避するのが難しいのが主な原因である。

我々はこの点を解決する安価で高性能なMINのアーキテクチャとしてSSS(Simple Serial Synchronized)型MIN[11][18]を提案した。SSS型MINは、単純なエレメントから構成される高い通過率を持つ結合網により、同期してビットシリアルで入力されるパケットを交換する。エレメントの構造と動作の単純化により高い周波数での動作を可能にすると共に、LSIのピン数制約の問題を解決し高密度実装を可能とする。また、SSS型MINの動作上の特徴を生かすことにより、簡単にMessage Combineの実装が可能である[12]。さらに、SSS型MINはきわめて規則的な動作を行なうため、全プロセッサを同期して動作させることでマルチプロセッサシステム全体を規則的に動作させることができる。この動作の規則性は通常のプログラミングに際して高いデバッガビリティを提供するだけではなく、インストラクションレベルでのスケジューリングを容易にする。

我々は、SSS型MIN用のプロトタイプチップを実装し[13]、これを用いたマルチプロセッサプロトタイプSNAIL(SSS Network Architecture ImpLementation) [14][17]を開発している。現在、16プロセッサのシステムを調整中であるが、4プロセッサを用いた構成で簡単なアプリケーションプログラムでの評価を行なうことができたので、この結果を報告し検討する。

2　SNAILの構成

2.1　プロトタイプの概観

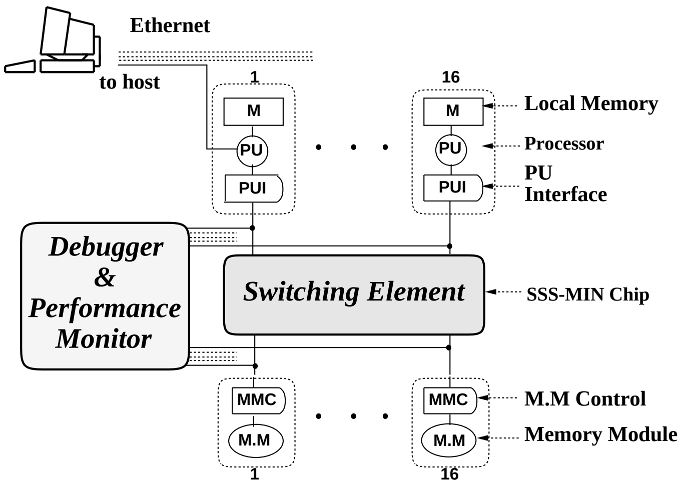

図1: SNAILの構成

SNAILの構成は、典型的なMIN結合型のマルチプロセッサ[6][7]であり、図1に示すように、プロセッサ(MC68040)とローカルメモリから成るサ16個のPUがSSS型MINを介してメモリモジュール16個と接続されている。プロセッサのうち4個に1個はEthernetインタフェースを持ち、ホストのワークステーションと結合されている。モニタ/デバッグ用ハードウェアは、別に簡単な16bit CPU(MC68000)を持ち、MINの両側を監視しながら、本体とは独立に動作する。全システムは、単一のシステムクロック(50MHz)と、フレームクロック(50/16=3.125MHz)を共有し、同期動作する。

SNAILの開発目的を以下に示す。

- SSS型MINの性能と特性を実際のアプリケーションに基づいて評価する。特にメッセージ結合の効果に関しては、今まで実際的な報告がなかったことから興味深い。

- プリフェッチ操作とローカルメモリの積極利用により、共有メモリへのアクセスレイテンシィによる性能低下がどこまで防げるか評価する。

- コンパイラ、スケジューラによるMIN自体のスケジューリング、プリフェッチ、ブロック転送等がどこまで可能か評価する。

- SSS型MINの動作の規則性を利用し、ハードウェアにサポートされたデバッガを構築する。

これらの目的を満足するため、SNAILは絶対性能よりも柔軟性を重視した設計になっており、SSS型MIN本体以外はFPGA(Xilinx XC3090:トグル動作125MHz)で構成されている。従って、構成をある程度変更して実験を行なうことが可能である。

2.2 共有メモリのアクセス

プロセッサの共有メモリへのアクセスは、パケットの形でフレーム信号に同期して MIN に対して入力される。書き込み、Prefetch 等のアクセス (ノンブロッキングアクセス) を行なう場合、プロセッサは、パケットバッファ(2 ワード) が空いていれば、ウエィトなしに次の命令に移ることができる。これに対し、読み出し、同期操作等のアクセス (ブロッキングアクセス) を行なう場合、プロセッサはデータの到着まで待たされる。いずれの場合も、アクセスを発生したフレームを i とすると、i+1 のフレーム信号に同期してアドレスパケットの転送が行なわれ、プロセッサからメモリモジュールへの経路 (SSS 型 MIN ではトレースと呼ぶ) が設定される。データの転送は、このトレースに従って i+2 フレームで行なわれ、32bit データの場合、その前半で読み出し可能となる。パケットバッファにパケットは入っている場合、これらの操作は全てフレーム単位でパイプライン的に実行される (パイプライン化サーキットスイッチング)。MIN 内で衝突が発生すると、転送が成功するフレームまでアクセスが引き延ばされる。

共有メモリは、4 ワード (128byte) 単位に 16 モジュールにインタリーブされており、以下に示すアクセスが可能である。

- 読み出し: プロセッサは読み出しを完了するまで、待ち状態になる。Combine 可能である。
- Prefetch: プロセッサはパケットバッファ(2 パケット分) が空いていれば、次の命令に進むことができる。データが必要になった時、プリフェッチバッファ(2 ワード) に対し読み出しを行なう。この時データがまだ到着していなければ、到着するまでプロセッサは待ち状態になる。Combine 可能である。
- 書き込み: プロセッサはパケットバッファ(2 ワード) が空いていれば、次の命令に進むことができる。
- ブロック転送: プロセッサのオンチップキャッシュの 1 ブロック (4 ワード) 分を高速に読み込む。ローカルメモリと共有メモリ間のデータのブロック転送に有効である。
- 同期操作: Test&Set と Fetch&Dec が可能である。Test&Set は SSS 型 MIN 内で Combine されるが、Fetch&Dec はメモリインタフェース内でのみ行なわれ、Combine されない。

同一メモリアドレスに対し、読み出し/書き込み, Test&Set, Fetch&Dec, Prefetch の 4 つのアドレス空間が割り付けられており、上記の操作はアドレスによって識別される。ブロック転送は共有メモリに対し、オンチップデータキャッシュを ON にし、その領域に対して読み出しを行なうことにより起動される。

SNAIL では、後に示す MIN とメモリの構成から、フレーム周期はクロック周期の 8 倍になるため、最大周波数動作時の共有メモリのアクセス時間の最小値は表 1に示すようになる。

表 1: 共有メモリのアクセス時間

フレーム時間	320nsec
32bit data 読み出し/同期	380nsec
1 ラインブロック転送	800nsec

以降、これらの操作を実行するための各部の構成を簡単に解説する。

2.3 プロセッサ部

プロセッサ部の構成を図 2に示す。PU は MC68040(25MHz) で、ローカルメモリは SRAM で 512Kbyte、ウェイトなしにアクセス可能である。プロセッサの周辺回路と SSS 型 MIN に対するインタフェースは Xilinx 社の FPGA (XC3090) 2 個で構成されている。このインタフェースはプロセッサ / MIN 間のパラレル/シリアル変換、入力パケットバッファ2、Prefetch バッファ2 ワードを含んでいる。

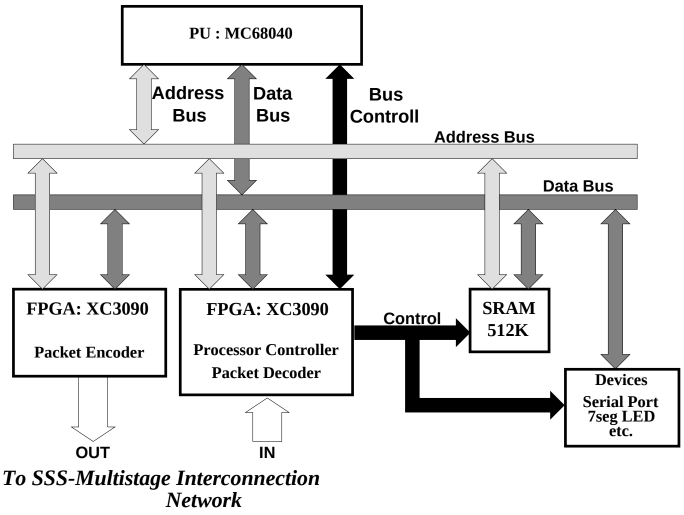

図 2: プロセッサ部の構成

プロセッサ部は 4 プロセッサが一枚のボードに実装されている (図 3)。各ボード内のプロセッサのうち一個がマスタプロセッサで、Ethernet 経由でホストワークステーションと接続されている。マスタプロセッサ上の ROM にはモニタプログラムが搭載されており、プログラムをホストから転送し、共有メモリを経由して、各プロセッサのローカルメモリに配った後各プロセッサを起動することができる。

図 3: SNAIL の PU ボード

2.4 MIN と共有メモリ

SSS 型 MIN は MIN 内にバッファを持たないため、衝突による性能低下が大きい。このため、われわれは、同一宛先への複数パケットの通過を許す MIN に着目し、オメガネットワークを直列に複数接続した TBSF(Tandem Banyan Switching Fabrics)[8][10]、三次元的に構成する PBSF(Piled Banyan Switching Fabrics)[16] について検討を行なってきた。今回は構成の簡単な TBSF 用のチップを実装し [13]、これを利用した。

2.4.1 TBSF チップ

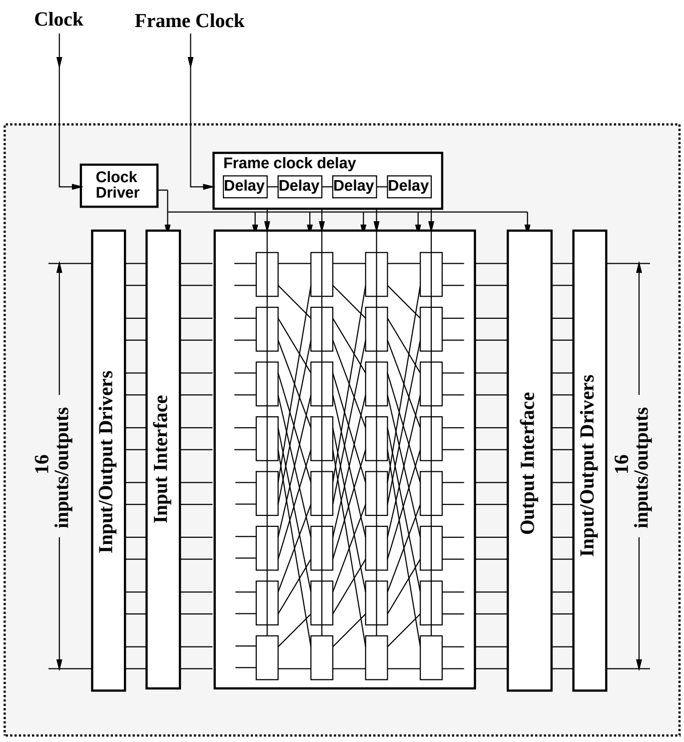

図 4: SSS-TBSF chip の構成

図 5: SSS-TBSF chip

このチップは、図4に示すように、16入出力、エレメント数32、クロック50MHzで動作する[1]。アドレス3bit、データ2bit (双方向) パラレルで各入力当たりの最大転送容量250Mbit/sec、全体で500Mbyte/secである。詳細な仕様を表2に示す。このチップは、複数チップを縦列接続することにより任意の網数のTBSFを構成することができる一方、ビットスライスに複数個用いることにより、レイテインシィを改善することができる。また、このチップを32個用いることにより、256入出力のオメガネットワークを構築することが可能である。16入出力、32エレメントが実装されているにもかかわらず、全体として9838セル (簡単なゲートならば1ゲート1セル、フリップフロップ等は4セルに当たる) で収まっている。エレメント1つは約200セルで実現されている。セル利用の内訳を分析するとメッセージ結合に要するハードウェアは全体の20%に過ぎない。

2.4.2 パケットの形式と転送法

表 2: SSS-TBSF チップの仕様

Clock: 50MHz	Bit width: 3bit(address)2bit(data)
Size: 16-in/out	Max bandwidth: 250$Mbits/sec \times 16$
セル仕様数: 9838	セル仕様率: 49%
信号ピン数: 180	テクノロジ: 1.0μm CMOS sea-of-gates

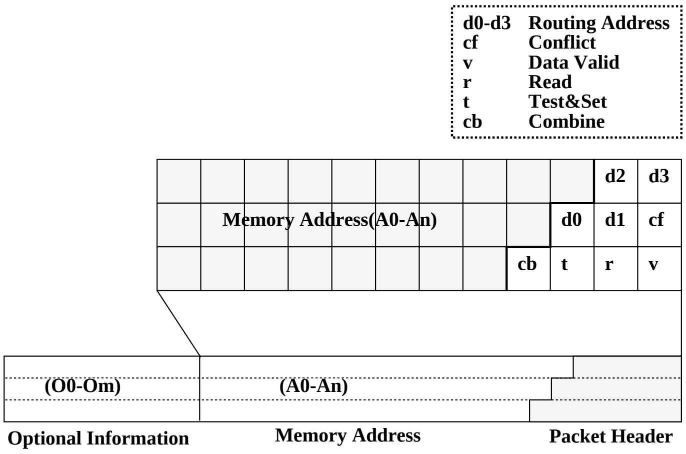

図 6: Address packet format

アドレスパケットの構成を図6に示す。上述にようにアドレスパケット用の線は3bit幅である。ヘッダの各bitは以下の通りである。

- cf: Conflict bit。パケットの衝突により希望の方向以外にルーティングされたことを示す。このbitがセットされているパケットは、以後セットされていないパケットの進路を妨害しない。
- $d_3 \ldots d_0$: 出力線の番号 (この場合はモジュール番号) を示す。各エレメントでこの番号に従いディスティネーションルーティングが行なわれ、エレメントの状態が決定する。
- v: Data valid。そのパケットが有効であるかどうかを示す。
- t: Test&Set。そのパケットがTest&Setを行なうことを示す。Test&SetをCombineする場合、各エレメントからプロセッサに対して1を返す特殊な処理を行なう。これを識別するためのbitである。
- r: Read。TBSFチップでは、データは双方向であり、この方向を切替えるためのbitである。
- cb: Combine。Combine可能かどうか示すbit。このbitが立っているTest&SetとRead要求のパケットはCombineされる。

アドレスパケットのヘッダ部以外はフリーフォーマットで、パケット長はフレーム信号によって任意の長さに設定することが可能である。アドレスパケットは図7に示すように、フレーム信号 (FRMCLK) に同期して入力される。フレーム信号がHレベルの間に転送される部分がアドレスと判断され、エレメント内でCombineを行なう場合の比較の対象となる。フレーム信号がLになり、次に立ち上がるまでの間、オプションのデータを送ることができる。実装したTBSFチップでは、トレースの設定は5クロックで行なわれ、その後パケットは各エレメントで蓄えられることなく、1クロックの遅延で転送される。

---

[1] このチップは川崎製鉄株式会社の協力のもとに作成された。

データパケットは完全にフリーフォーマットである。図7に示すように、i 番目のアドレスパケットで設定された経路 (トレース) に従って FRMCLK の間任意の長さのデータを交換することができる。データ幅は 2bit で、転送方向はアドレスパケットの Read bit で設定される。トレースは既に設定されているので、MIN 内での遅れは 1 クロックである。

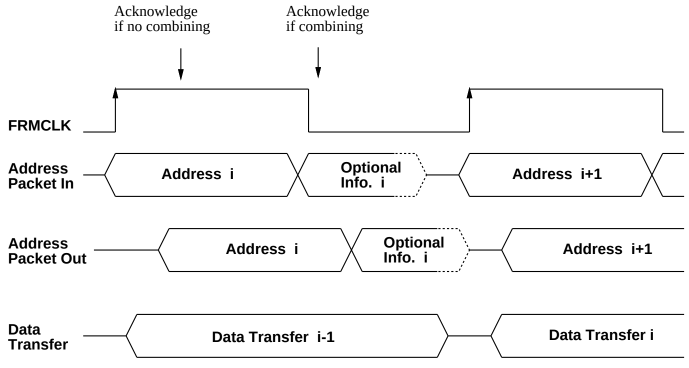

図 7: TBSF チップの動作

2.4.3 SNAIL における TBSF チップの利用

今回実装した TBSF チップは 16 プロセッサ、16 メモリモジュールを 1 チップで接続可能である。しかし 50MHz という SSS 型にとっては低い周波数のクロックを用いるため、チップをいくつかビットスライスに用いて、共有メモリアクセスのレイテンシィを改善する必要がある。SNAIL ではこのチップ 4 個を用いてビットパラレル転送で、TBSF 網の banyan(Omega) 網一段を構成する。したがって、SNAIL のネットワークは、アドレスのパスは 12bit, データは 8bit の構成となる。トレース設定は全チップで同一の動作を行ない 5 クロックを要するので、これ以上の個数チップを並列接続しても、性能の向上は頭打ちになる。

次に、TBSF 網では、網をいくつ縦列接続するかが問題となる。[18] のシミュレーションの結果では、256 入力のシステムでの最適網数は 2 または 3 になる。そこで、我々は、SNAIL で用いる TBSF チップの仕様、メモリの動作速度を基に、ATTEMPT-0 上で開発された並列アプリケーション [15] のアクセス確率を用いてシミュレーションを行なった。その結果、SNAIL における TBSF の接続網数は 1 または 2 が最適との結論を得た [17]。網数を大きくしても効果が上がらない理由は、まず、16 プロセッサでは、256 入力に比べ MIN 内での衝突は激しくないことが挙げられる。また、SNAIL では一般の特に高速でない DRAM を共有メモリモジュールとして用いている。共有メモリモジュールの動作速度が遅い場合、TBSF 網の網数を大きくしても効果は少ない。そこで、今回の実装では、TBSF の使用網数は 1 とした。この場合、トポロジは通常のオメガ網と同じで、通過率は制限されるが、メモリインタフェースの構成が簡単になる。

このような構成を取ることで SNAIL では、フレームは 8 クロック (プロセッサの 25MHz クロック。すなわち 50MHz のメインクロックでは 16) となり、表 1に示すアクセス時間を実現する。書き込みアクセスを高速に行なうために、書き込みデータはデータパスを用いず、アドレスパケットのオプションデータの部分を用いて転送している。このことにより、書き込みと読み出しでほぼ同様のサイクルでメモリアクセスが行なわれる。

2.4.4 共有メモリ

共有メモリは各モジュールが 4Mbyte の DRAM で、全体で 64Mbyte であり、Xilinx 社の FPGA1 個を用いたメモリインタフェースによって制御される。前述のようにキャッシュブロックに合わせた 4 ワード (16byte) 単位にインタリーブされるが、この構成は FPGA のコンフィグレーションの変更により、任意のサイズに変更することができる。メモリコントローラは、MIN とメモリ間のパラレル / シリアル変換を行ない、加算器により Fetch&Dec を実現すると共に、DRAM の制御、リフレッシュ等も行なう。リフレッシュは通常は使用されないフレームを利用して行なわれ、メモリの連続利用のためにこれが不可能な場合は、全体の動きを 1 フレーム中断して行なう。このため、リフレッシュにより、プロセッサの動作のタイミングが狂うことはない。

共有メモリ 16 モジュールとネットワークは合わせて 1 つのボードに実装されている。このボードに 4 プロセッサを含むボード (図 3) を差すことにより、16 プロセッサの SNAIL が構成される。

2.5 システムモニタ / デバッガ

一般的に、MIN を用いたマルチプロセッサは、共有バス型マルチプロセッサに比べると、メモリモジュールが多数あるので、交信の全体を把握するのが困難である。しかし、SSS 型 MIN は基本的にビットシリアル通信で、しかも全パケットが同期して入出力されるため、MIN を用いた他のシステムに比べ、システムモニタの実装が容易である。SNAIL に装備したモニタ / デバッガシステムの構成を図 8に示す。このシステムは、簡単な CPU(MC68000) と RS232C を装備することにより、SNAIL 本体とは独立にホストと交信することができる。

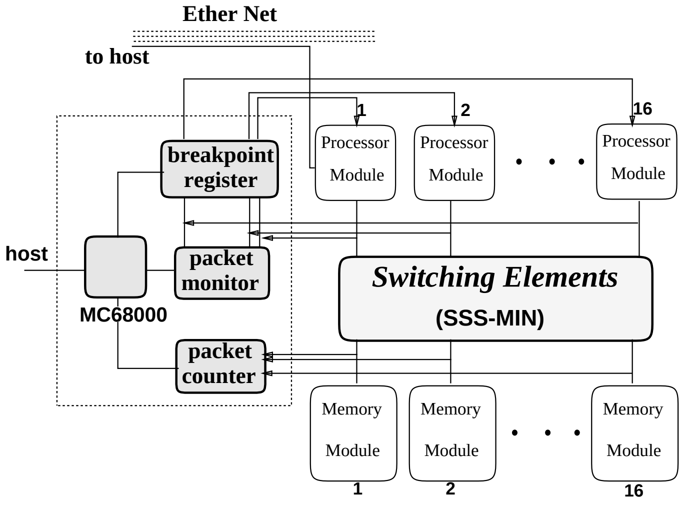

図 8: モニタ / デバッガの構造と接続法

モニタ部では、Xilinx 社の FPGA XC3090 5 個を用いて、MIN のメモリモジュール側からパケットをモニタする。モニタには、加算器付きのカウンタ (図 8 の Packet Counter) が用いられ、メモリモジュールに対する 16 本のラインからのパケットに関する情報を集計する。

プロセッサの動作をトレースするためには、プロセッサ側の情報も必要であるため、MIN のプロセッサ側からもモニタを行なっている。これも専用のカウンタ (図 8:Packet Monitor/Event Counter) を用いて集計している。しかし、16 プロセッサの全てを同時にモニタすることは実装面積の点で困難であるため、4 プロセッサを選んで測定する構造を取っている。

以上のモニタにより、総パケット数、衝突による転送失敗パケット数、プロセッサからの同期操作の回数、Combine の成功した回数等の情報を得ることができる。

さらに、このモニタ能力を利用し、共有メモリに対するアクセスによって全プロセッサに割り込みを発生する共有メモリブレークポイントや、1 フレーム単位で割り込みをかける 1 フレーム実行等のデバッグ機能もサポートされる。

3 評価

3.1 評価用アプリケーション

以下の 4 種類のアプリケーションを用いて評価を行なった。

- Four coloring
  隣接する領域が同色とならないように各領域を塗り分ける問題をニューラルネットワークを用いたアルゴリズムにより解くプログラムである。本評価ではアメリカ合州国の 48 州を用いた。

- Jacobi
  多元連立一次方程式をヤコビ法を用いて解くプログラムである。本評価では 200 元の問題を用いた。

- Knight's tours
  チェスのナイトが盤上の同じマスを通らずに全てのマスを通り元の位置に戻る経路を決定する問題をニューラルネットワークを用いたアルゴリズムにより解くプログラムである。本評価では 8×8 のチェス版を用いた。

- N-Queen
  $n \times n$ のチェス盤上に n 個のクィーンを互いに取り合うことのない位置に配置する問題をニューラルネットワークを用いたアルゴリズムにより解くプログラムである。本評価では $n = 32$ を用いた。

これらアプリケーションではコード・ローカルデータ・初期化後変更されないデータ等を各プロセッサのローカルメモリに配置した。

3.2 実行時間の評価

3.2.1 台数効果

各アプリケーションの、プロセッサ数の増加による性能の向上を図 9に示す。この図は 1 プロセッサで並列プログラムを実行した場合の実行時間を基準にしているが、同じ問題を逐次処理で解いた場合の実行時間 (serial) も併せて図中に示している。この逐次プログラムは、全てのデータをローカルメモリ上におくと共に、同期操作の部分等並列化のロスを省いている。

この図によると、Knight's tours 以外は、逐次処理で解いた場合の実行時間と比べてもかなりの性能の向上が実現されている。Knight's tours は、探索アルゴリズムの並列化の部分のロスが大きいため、逐次実行に比べ並列処理での性能の向上は少ない。

このグラフから、これらのアプリケーションでは、結合網のレイテンシィや同期操作のロスはこの程度のプロセッサ数ではさほど大きく現れていないことがわかる。

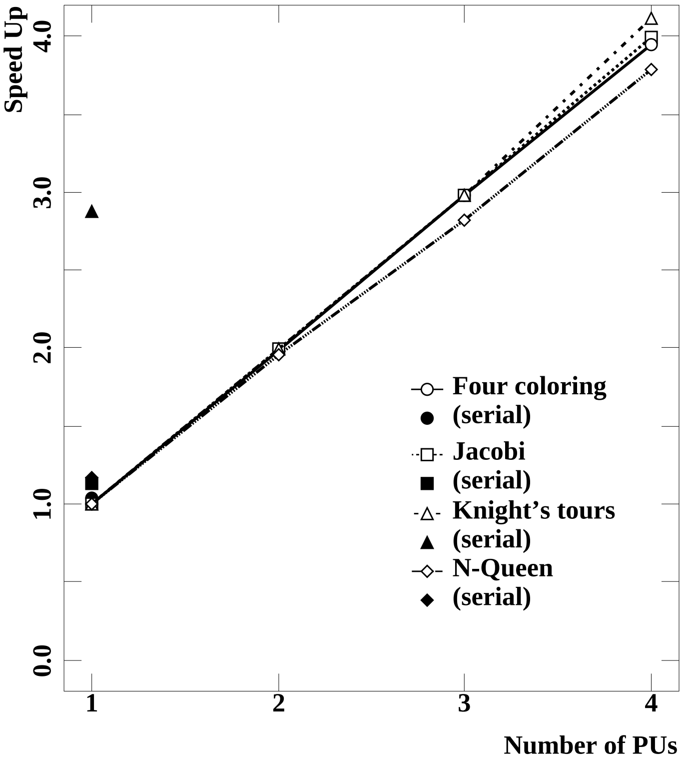

図 9: 台数効果

3.2.2 Prefetch の効果

N-Queen に対して共有メモリへのアクセスを prefetch 操作により行なった場合の台数効果を図 10に示す。prefetch 命令は、各ループで次のループで必要になるデータを prefetch するように人手により挿入してある。図によると prefetch 命令の挿入はかなりの効果を上げていることがわかる[2]。この結果は、prefetch を実行するためにアセンブラレベルでの最適化が行なわれているため、性能は純粋に prefetch のみの効果よりもやや大きめに出ているが、prefetch 命令の効果は期待できることがわかる。

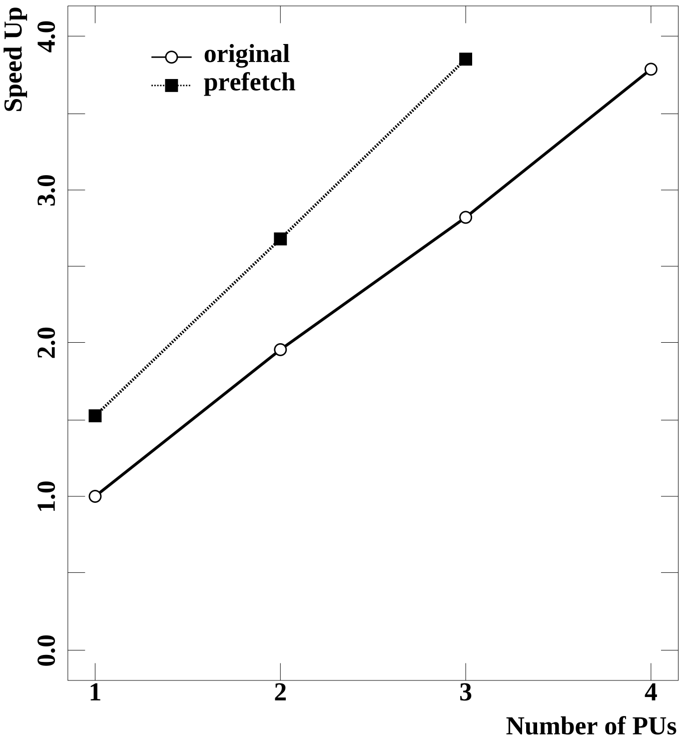

図 10: プリフェッチの効果 (N-Queen)

---

[2] 現在のシステムのバグのため 4 プロセッサでのデータが取れていない。

3.2.3 Combine の効果

各アプリケーションについて、4 プロセッサでメッセージ Combine を行なった場合の実行時間 (秒) を表 3に示す。今回実装を行なったアプリケーションは、Combine の行なわれない Fetch&Dec を用いて同期操作が行なわれているため、実際に行なわれる Combine は同一の共有データに対して行なわれた読み出しアクセスに限られる。また後に示すように、今回のアプリケーションは負荷率が小さいため、衝突するパケット自体が少ない。以上の点から、実行時間の改善はごくわずかである。

表 3: Combine の効果 (4PU)

	実行時間 (秒)	
	Combine なし	Combine あり
Four coloring	5.203	5.203
Knight's tours	14.861	14.854
N-Queen	81.586	81.584

3.3 ネットワークに対するアクセスの評価

SNAIL では、システムモニタを用いることにより、MIN に対するパケットのトラフィックを計測することができる。以下、この結果を示す。

3.3.1 フレーム使用率

各アプリケーションにおけるフレーム使用率を図 11に示す。ここで、フレーム使用率とは、ある MIN の入力について、フレームがパケットにより使用される割合いを示し、MIN に対する負荷率に相当する。

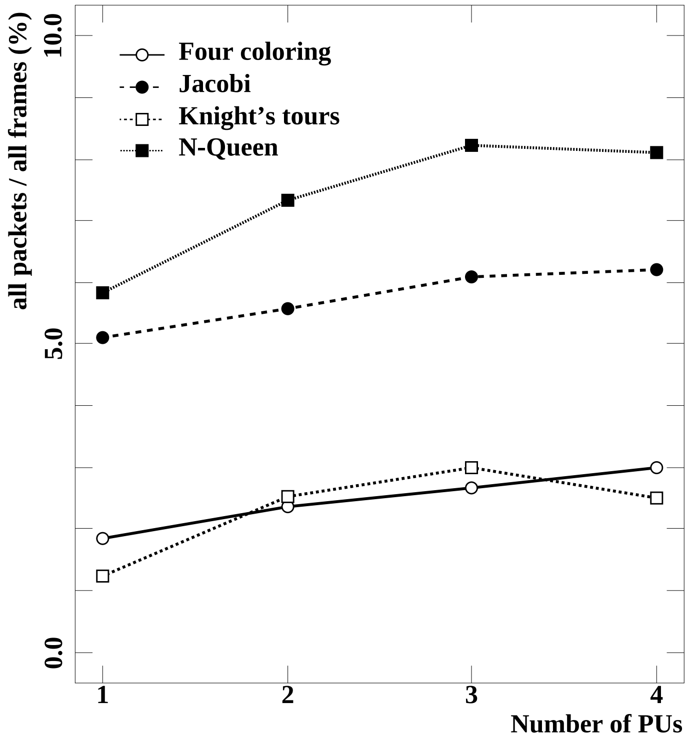

図 11: フレーム使用率

図によると、アプリケーションにより若干の相違はあるが、フレーム使用率は 10%以内である。プロセッサ数の増加により若干の増加が見られるが、これは、プロセッサ数の増加により増加した衝突パケットが再入力されるためである。しかし、その増加はさほど大きくなく、この程度の負荷であれば、MIN による損失は少なく、これが図 9において直線的な台数効果が得られる原因となる。

3.3.2 パケットの衝突

図 12に全パケットに対する衝突により再送されるパケットの割合を示す。当然のことながら、フレーム使用率の大きいアプリケーションでは衝突の割合も大きいが、最大でも 3%程度である。この程度の衝突率であれば、アプリケーションに与える影響は少ない。今回の測定で Combine の効果が少ないのも衝突自体の影響が小さいためである。

しかし、プロセッサの増加に対する増加率はかなり大きいので、16 プロセッサで実行すれば無視できない性能低下の原因になり得る。

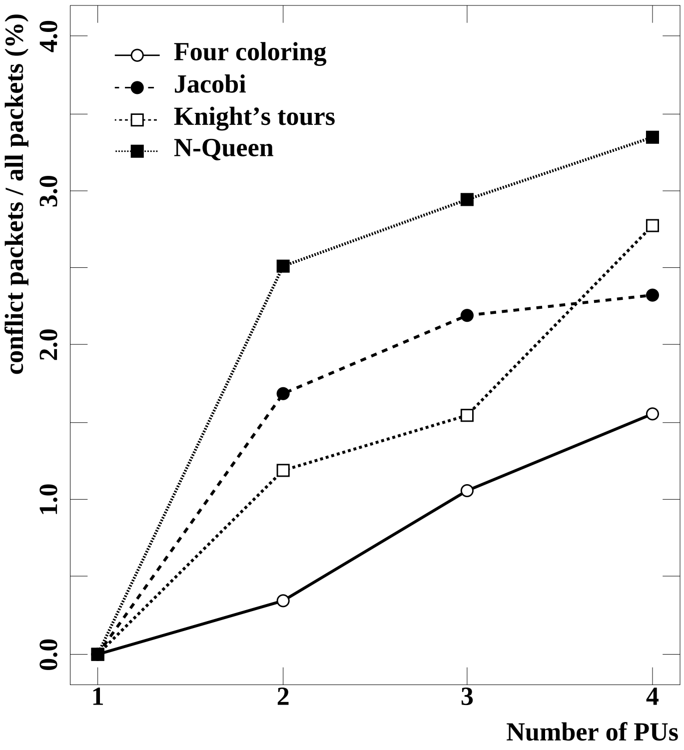

図 12: パケットの衝突率

4 おわりに

本論文の評価により、SNAIL の MIN が十分な性能を持っていることが示された。しかし、現在、ネットワーク／共有メモリボードのデバッグが終了していないため、SNAIL は 4 プロセッサで稼働しているにすぎない。また、プロセッサボードにもバグが残され、アプリケーションプログラムも比較的 MIN に対する負荷の少ないもので、かつ最適化も不十分である。従って、本論文で示した結果は中間結果報告であり、プロセッサ数とアプリケーションの種類を増やし、各部を最適化すれば、かなり異なった結果が得られる可能性がある。

謝辞

SSS 型 TBSF チップは川崎製鉄株式会社 LSI 事業部の御協力により開発されました。深く感謝いたします。また、同事業部のカスタマサポート部の成石正明氏には特にご尽力いただき、深く感謝いたします。

また、SNAIL の設計、実装に当たってお世話になった、慶應義塾大学計算機科学専攻の小椋 里氏、塙 敏博氏、アリュメ株式会社に深く感謝します。この研究の一部は文部省科学研究費 (重点領域研究 (1) 超並列ハードウェア・アーキテクチャの研究」) によります。

参考文献

[1] Wilson Jr. A.W. : *Hierarchical Cache/Bus Architecture for Shared Memory Multiprocessors*, Proc. of Int. Symposium on Computer Architecture, pp.244-252, (1987).

[2] Lenoski,D. Laudon,J. Gharachorloo,K. Weber,W.F. Gupta,A. Hennessy,J. Horowitz,M. Lam, M.S. : *The Stanford Dash Multiprocessor* Computer, Vol.25 No.3, pp.63-79, Mar. (1992).

[3] Cheriton,D.R. Goosen,H.A. Boyle,P.D. : *Paradigm: A Highly Scalable Shared-Memory Multicomputer Architecture* Computer, Vol.24 No.2, pp.33-46, Feb. (1992).

[4] 松本、平木 : Memory-Based Processor による分散共有メモリ, 並列処理シンポジウム JSPP93 予稿集, pp.245-252 (1993).

[5] Broomell, G. and Heath, J.R. : *Classification Categories and Historical Development of Circuit Switching Topologies*, ACM Computing Surveys, Vol.15, No.2, (1983).

[6] Gottlieb, A. Grishman, R. Kruskal, C.P. Mcauliffe, K.P. Rudolf, L. and Snir, M. : *The NYU Ultracomputer - Designing an MIMD Shared Memory Parallel Computer*, IEEE Trans.on Comput. vol. c-32, No.2, (1983).

[7] Phister, G.F. et al.: *The IBM Research Parallel Processor Prototype (RP3): Introduction and architecture*, Proc. of 1985 Int. Conf. Parallel Processing, (1985).

[8] 坂元, 荒川, 正木, 井上, 天野: 自己ルーチングスイッチの構成とその評価, 信学技報 ISSE88-30 8, (1988).

[9] Konicek, J. et al,: *The Organization of the Cedar System*, Proc. of 1991 Int. Conf. Parallel Processing, (1991).

[10] Tobagi, F.A. Kwok, T.:*The Tandem Banyan Switching Fabric: a Simple High-Performance Fast Packet Switch*, Proc. of INFOCOM91, (1991).

[11] Amano, H. Zhou, L. Gaye, K.:*SSS (Simple Serial Synchoronized)-MIN: A novel multi stage interconnection architecture for multiprocessors*, Proc. of IFIP Congress 92, Sept. pp. 571-577, (1992).

[12] Gaye, K. Amano, H.:*A Batcher Double Omega network with Combining*, IEICE Trans. on information and systems, Vol.E75-D No.3 (1992).

[13] 周洛, 天野英晴: SSS-MIN 用プロトタイプチップの実装, 情報処理学会第 44 回全国大会論文集,6-103,104 (1992).

[14] 周, 天野, 笹原, 寺田, 小椋: SSS-MIN に基づくマルチプロセッサプロトタイプ SNAIL, 信学報 CPSY92-27(SWoPP92), (1992).

[15] 鳥居, 竹本, 天野, 小椋: バス結合型並列計算機の交信用メモリの性能評価, 情報処理学会論文誌 Vol.33 No.3 (1992).

[16] 天野英晴, 藤川義文:マルチステージネットワーク PBSF, 情報処理学会計算機アーキテクチャ研究報告 No.94-5, (1992)

[17] 笹原, 寺田, 小椋, 周, 天野: SSS-MIN アーキテクチャに基づくマルチプロセッサプロトタイプ SNAIL, 情報処理学会第 46 回全国大会論文集,6-99,100 (1993).

[18] 天野, 周, 藤川: SSS (Simple Serial Synchronized) 型マルチステージネットワーク, 情報処理学会論文誌 Vol.34, No.5, pp. 1134-1143, (1993).