

SSS 型 MIN に基づくマルチプロセッサ SNAIL

笹原 正 司[†] 寺 田 純[†] 大 和 純 一^{††}
 埜 敏 博[†] 天 野 英 晴[†]

SSS (Simple Serial Synchronized) 型 MIN (Multistage Interconnection Network) は、数百プロセッサ程度の中規模の並列計算機に適したプロセッサ・メモリ結合網である。SSS 型 MIN は、基本的にシリアルな同期通信を行うことでエレメントの構造と制御の簡単化を行った。これにより実装が容易で高速かつ効率の良い動作が可能であり、メッセージコンバインの実装も容易である。一方でシリアル転送と同期通信によるレイテンシの増大、エレメント中にバッファを持たないことによる性能の低下が問題である。マルチプロセッサ SNAIL (SSS Network Architecture ImpLementation) は、この SSS 型 MIN の評価を行うために実装中のプロトタイプであり、現在 16 プロセッサ・16 メモリモジュールのフルスケールで稼働中である。本論文では、SNAIL 上に並列プログラムを実装し、結合網の性能評価を行った。結果として、SSS 型 MIN におけるレイテンシと同期のオーバヘッドは性能向上に大きな影響はなく、バンド幅も十分であることが明らかになった。また、メッセージコンバインの性能向上への効果は見られなかったが、かなりの割合でコンバイン自体が実行されていることは明らかになり、SNAIL のスケジューラビリティを生かして効果的に活用することで、さらなる性能向上が得られると考えられる。

SNAIL: A Multiprocessor Based on Simple Serial Synchronized Multi-stage Interconnection Network

MASASHI SASAHARA,[†] JUN TERADA,[†] JUN-ICHI YAMATO,^{††}
 TOSHIHIRO HANAWA[†] and HIDEHARU AMANO[†]

The SSS (Simple Serial Synchronized)-MIN (Multistage Interconnection Network) is a network for connecting between processors and memory modules of middle scale parallel machines. In the SSS-MIN, packets are transferred in the serial and synchronized manner. Synchronized bit-serial communication simplifies the structure/control, and also solves the pin-limitation problem. Moreover, the message combining can be implemented with a small increase of hardware using a technique called *bit-serial message combining*. In order to demonstrate the performance of the SSS-MIN, the overhead caused by the serial transfer and synchronous communication must be evaluated. The performance degradation caused by packet re-insertion must be also evaluated. The multiprocessor SNAIL (SSS Network Architecture ImpLementation) is a prototype machine to evaluate such factors of the SSS-MIN, and now a full scale system with 16 processors/16 memory modules is operational. In this paper, the small overhead of the SSS-MIN is demonstrated with some parallel benchmark programs. As a result, the SSS-MIN achieves comparable or superior performance to the conventional ones with much smaller hardware cost. Though the message combining does not improve the performance with the current machine scale and applications, it is obvious that message combining itself operates, so more improvement can be achieved with the best use of the scalability of SNAIL.

1. はじめに

多段結合網 (Multistage Interconnection Network : MIN) は中規模 (数十から数百プロセッサ) の並列計

算機におけるプロセッサとメモリ間の接続に多く用いられる結合方式であり、現在までに様々な研究がなされてきた¹⁾。これらの MIN では、メモリに対するアクセスはパケットの形で行われ、パケットはネットワーク内のスイッチングエレメント間をビットパラレル (8~64 bit) で転送される。結合網のトポロジは Omega 網を代表とするブロッキング網で、スイッチングエレメント内で衝突が起きた場合、片方のパケットはエレ

[†] 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

^{††} 日本電気 (株) C&C 研究所コンピュータシステム研究部
 Computer System Research Laboratory C&C Research
 Laboratories, NEC Corporation

メント内のバッファに格納される。

このような従来型の MIN には高速、高密度実装に困難な点が多く、これを用いたマルチプロセッサシステムの実装の遅れ、規模縮小^{2),3)}の原因の1つになってきた。まず、パケットが Store&forward 方式で転送されるため、各スイッチにパケットバッファやその制御機構を設けなくてはならず、エレメントの構造が複雑になる。また、ビットパラレル転送を行う場合にはプロセッサ当たり 8~64 bit の入出力線が必要となり、これを 1 チップに収めるため、LSI チップ中に実装できるエレメント数がピン数によって制限されてしまう。このため、全体として用いるチップ数、基板面積が増大してしまい、高速なクロックの利用も困難になる。

これらの問題点を解決するため、我々は SSS (Simple Serial Synchronized) 型 MIN を提案し⁴⁾、プロトタイプチップの実装を行ってきた⁵⁾。SSS 型 MIN では、パケット転送を基本的にビットシリアルとし、同期転送にすることによってエレメントの制御機構を単純化し、高速なクロックを利用可能にすると共に、LSI のピン数制約の問題を解決し、高密度実装を可能にしている。

また、動作上の特徴を生かすことにより、少ないハードウェアでメッセージコンバインの実装が可能である。更に、SSS 型 MIN は極めて規則的な動作を行うため、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グビリティを提供するだけでなく、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。

本論文では、この SSS 型 MIN チップを用いた並列計算機 SNAIL のアーキテクチャと、SSS 型 MIN および SNAIL の評価について述べる。

2. SSS 型 MIN

2.1 SSS 型 MIN の基本構造

SSS 型 MIN の基本動作を図 1 に示す。プロセッサからのアクセスは MIN との間にあるバッファでシリアル化され、ISDN 等で用いられている ATM 用交換システム同様、フレームクロックと呼ばれるクロックに同期してネットワークに入力される。スイッチングエレメント内にはバッファは設けられておらず、各エレメントは 1 bit 分の記憶しか行わないため、ネットワークは全体としてスイッチング機能のあるシフトレジスタのように動作する。そして全ステージを通過する分の遅延の後、パケットの先頭が出口から出力される。

SSS 型 MIN はパケットバッファを持たないため、

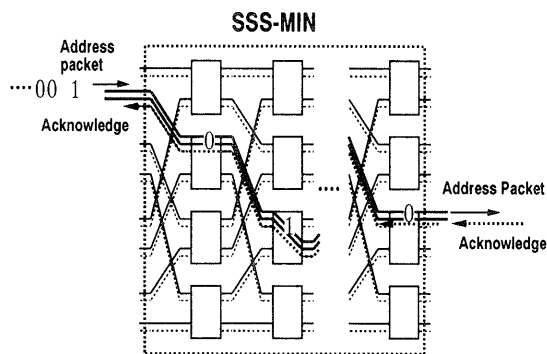


図 1 SSS-MIN の構造

Fig. 1 Structure of the SSS-MIN.

スイッチに入力された 2 つのパケットが同一出力に向かった場合、一方のパケットは正しい方向に進むが、他方は正しい方向に転送されない。正しく転送されなかったパケットは、コンフリクトビットがセットされ、以降このパケットはデッドパケットとみなされる。スイッチにデッドパケットが入力された場合、そのパケットのルーティングタグは無視され、他方のパケットのタグにしたがってスイッチの状態が決定されるため、このパケットは他のパケットの進行を妨げない。

パケットのアドレスの先頭が出口に到達すると、ネットワークのすべてのエレメントの状態が確定し、入力から出力への論理的なパス(これをトレースと呼ぶ)が設定される。トレースが設定されると、パケットのコンフリクトビットを MIN の出口で参照してパケットが正しく転送されたかを調べ、ACK (ACKnowledgement) もしくは NAK (NotAcKnowledgegement) がプロセッサ側に返される。このアクノリッジ信号は、MIN の各段でレジスタにストアされることはなく、マルチプレクサとチップ内の配線による遅延のみでプロセッサ側に到達する。プロセッサ側が NAK を検出した場合、プロセッサは次のフレームでパケットを再送しなくてはならない。また、パケットを受けとるメモリ側は、パケット中のコンフリクトビットを参照し、衝突したパケットである場合はそのパケットを無視する。

2.2 高速化手法

SSS 型 MIN では動作の高速化のため、以下のような工夫がなされている。

2.2.1 パイプライン化サーキットスイッチング

SSS 型 MIN ではアドレス、データ、アクノリッジの転送路は独立しており、アドレス転送によってトレースが設定されると、アクノリッジおよびデータパケット転送のトレースも同時に設定され、これらの転送は、

図2に示すとおりオーバーラップして行われる。図中、フレーム i でアドレスが転送され、トレースが形成されると、同時に各入力に対してアクリッジ信号 (ACK/NAK) が返される。正しく転送が行われ、ACK が返送された場合は、このトレースを利用し、フレーム $i+1$ でデータの転送が行われる。これをパイプライン化サーキットスイッチングと呼ぶ。

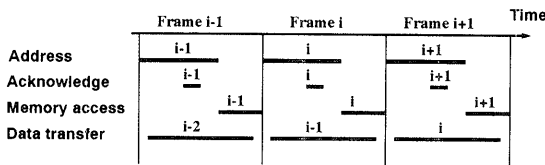
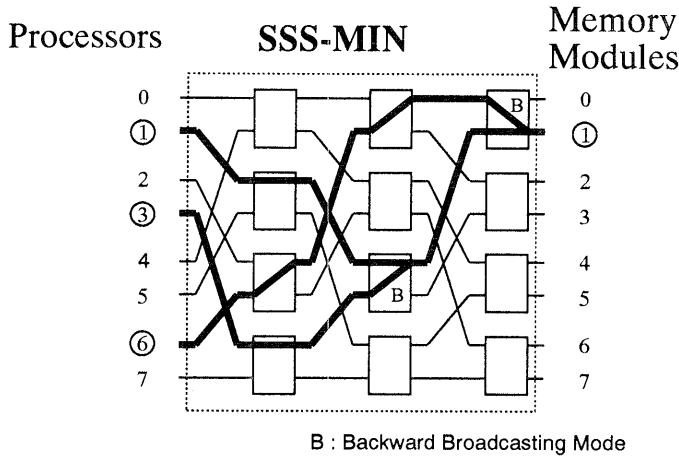


図2 SSS-MINでのパケット・信号のタイミング
Fig.2 The timing chart of packet transfer in SSS-MIN.

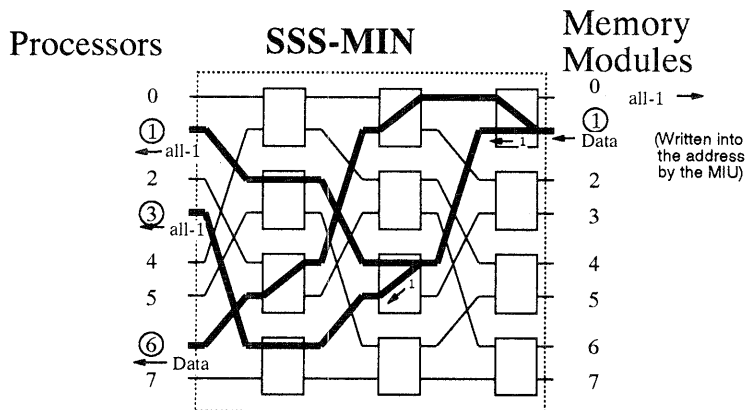
2.2.2 メッセージコンバイン

メッセージコンバイン機能は同一宛先への複数パケットをエレメント内で結合し、1つのパケットとして扱うことにより、Hot spot へのアクセス集中による Tree saturation を回避する方法である⁶⁾。この機能の効果はシミュレーション等で確認されているが、従来型の MIN でこの機能を実現するには通常の6倍から32倍のハードウェアを要するため⁷⁾、実現が困難であった。

しかし、SSS型MINではシリアル転送の特性を生かして簡単に実現することができる。図3(a)に読みだしアクセス同士の結合の実現例を示す。各エレメントはアドレスパケット通過時に入力された2つのパケットの全アドレスフィールドを比較する。そして、両方のアドレスが完全に等しければ、データ転送路のマルチプレクサを Backward Broadcasting モード (図



(a) Combining of read references



(b) Combining of Test&Set references

図3 メッセージコンバイン
Fig.3 Message combining.

中 (B)) にセットし、全体として Tree 状のトレースを形成する。読み出されたデータはこのトレースに従ってマルチキャストされ、同時に各プロセッサに到着する。この方法では、比較とマルチキャストがビットシリアルに行われるため、付加回路が簡単で、動作速度に与える影響も少ない。

さらに、同様にして Test&Set の結合も容易に実現可能である (図 3 (b)). アドレス通過時には、図 3 (a) 同様にエレメント間に Tree 状のトレースが形成されるが、データの転送時に片方のデータ転送路に強制的に H レベル (all-1) が挿入される。このため、ただ一つのプロセッサだけがメモリからのデータを受け取り、残りのプロセッサは all-1 のデータを受け取る。これと同時に、メモリコントローラによってメモリの値は all-1 とされ、全体として Test&Set 操作の結合が実現される。

2.2.3 ステージの飛び越し

アドレスパケットのルーティングタグにより一度トレースが設定されてしまえば、アドレスパケットの残りデータパケットは、既に設定された経路をたどっていけばよい。このため、各ステージで記憶を行う必要はなく、応答確認信号 (ACK/NAK) 同様に、動作速度に応じていくつかのステージを飛び越して転送する。

2.3 SSS 型 MIN の問題点

SSS 型 MIN の問題点は、シリアル転送によるレイテンシの増大と、同期転送に起因するロスがある。これについては、高速な動作クロックを利用し、さらに LSI のピン数を考慮した上で数 bit のパラレル転送にすることである程度克服することができる。

次に、SSS 型 MIN では、衝突が起きた場合、次のフレームで入力バッファから再送する必要があり、再送による共有メモリアクセスの遅延が性能低下を起こす可能性がある。

このように、SSS 型 MIN は従来型の MIN とは異なった性能低下の要因を持ち、性能低下に対する高速化技法の効果について、確率モデルによる評価が行われているが⁹⁾、より実際の評価のためにはプロトタイプ上での実アプリケーションによる評価が必要である。そこで我々は SSS 型 MIN と SSS 型 MIN を用いたマルチプロセッサ SNAIL の実装を行っている。

3. マルチプロセッサプロトタイプ SNAIL

SNAIL は、SSS 型 MIN の性能を評価し、さらに SSS 型 MIN の規則性を利用したスケジューリングの実験を行うために実装されたプロトタイプである。

3.1 SNAIL の概要

SNAIL の構成は図 4 のようになっており、プロセッサとローカルメモリから成る 16 個の PU と 16 個の共有メモリモジュールが SSS 型 MIN を介して接続されている。プロセッサのうちの 1 つは Ethernet インタフェースを持ち、ホストのワークステーションに接続されている。また、パフォーマンスモニタとデバッグシステムが実装されており、SNAIL と独立に MIN の入出力を監視している。プロセッサおよびメモリの動作クロック (50 MHz) とフレームクロック (50/16 = 3.125 MHz) は単一のクロックから生成され、システム全体で共有されている。

SNAIL の主な目的は SSS 型 MIN の性能評価にある。特に、SNAIL は 10 以上のプロセッサを持つマル

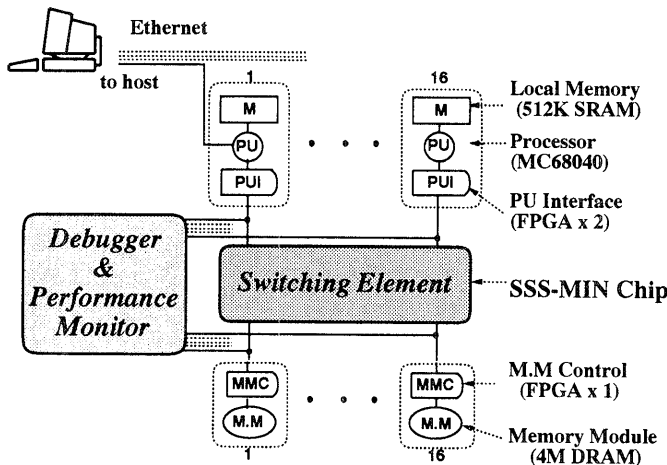


図 4 SNAIL の構成

Fig. 4 Diagram of a multiprocessor SNAIL.

チッププロセッサの中で、メッセージコンバイン機能を実装した初めてのマシンである。また、SSS 型 MIN は動作が単純で、転送が同期して行われるため、動作の再現性が極めて高く、並列化コンパイラや最適化コンパイラが進歩すれば、ネットワークのレイテンシの隠蔽、インストラクションレベルスケジューリングによる同期操作の排除などを行うことができる。

このため、SNAIL では以下のような機能が提供されている。

- 全プロセッサは SSS 型 MIN で用いているクロックに同期して動作する。またすべての共有メモリモジュールのリフレッシュと I/O の動作も SSS 型 MIN のフレームクロックに同期している。SSS 型 MIN ではパケットの転送はフレームに同期しており優先度は原則として固定であるため、プロセッサ間の動作のタイミングが狂うことはなく、インストラクションレベルのスケジューリングにより、ほとんどの同期のコードを省略することができる。しかも、並列プログラムは常に全く同じ動作をする。しかし共有メモリアクセスのスケジューリングを行わない場合、優先度の固定化によりスタベーションが発生する可能性があるため、16 個の共有メモリモジュールをインタリーブして用い、アクセスの集中が起きにくいようにしている。
- 共有メモリに対しては、プリフェッチ、ライトバッファ、共有メモリとオンチップキャッシュ間のブロック転送をサポートしている。これらの機能と最適化コンパイラを併用することで、共有メモリアクセスのレイテンシを隠蔽することができる。ブロック転送はキャッシュラインサイズ単位 (16 byte) でのみ行われる。また、プリフェッチバッファ、ライトバッファは各 2 ワードずつ用意されている。
- 従来型 MIN を用いたマルチプロセッサでは、ハードウェアモニタを用いて共有メモリへのアクセスを解析する場合、非常に多くの線をモニタしなくてはならず、ほぼ不可能であった。しかし SSS 型 MIN では、パケットが通過する線が少ないため、これらをモニタしてトラフィックを解析することができる。この機構は、スケジューリングされたコードの性能解析とデバッグに有効である。また、このモニタ機構により、共有メモリブレイクポイントが実装されており、プロセッサがある共有メモリアドレスにアクセスすると、ネットワークの動作が停止し、すべてのプロセッサに割り

込みがかかる。

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いる。PU-MIN インタフェースとメモリコントローラは Xilinx 社の FPGA XC 3090 を用いて実装している。FPGA の動作速度は速くはないが、コンフィグレーションデータを書き換えることにより、容易にハードウェアを変更することができる。

以下、SSS 型 MIN、SNAIL のハードウェアの詳細について述べる。

3.1.1 SSS-MIN チップ

SSS 型 MIN では先に述べたパケット再送のロスの問題を解決するため、従来型 MIN で用いられるプロッキング網よりも高い通過率を持つ結合網を用いなくてはならない。我々はこのような結合網の構成として、B-ISDN (Broadband Integrated Services Digital Network) で用いられるパケット交換用に提案した TBSF (Tandem Banyan Switching Fabrics)^{9),10)} を検討し⁹⁾、更にこれを改良した PBSF (Piled Banyan Switching Fabrics)¹¹⁾ を提案した。

我々が実装した SSS-MIN ゲートアレイチップ^{*}は、この TBSF を構成できるように設計されており、1 チップで 16 入出力のオメガ¹²⁾網を持つ。各入出力はピン数と集積度を考慮してアドレス線 3 bit、データ線 2 bit とし、ネットワーク全体がパイプライン動作する。さらに先に述べたステージ飛び越し (4 ステージ) を利用し、データ転送およびトレース確定後のアドレス転送を高速化している。

このチップの仕様を表 1 に示す。

図 5 のように、チップは入力インタフェース、ネットワーク、出力インタフェースで構成される。入力インタフェースは、トレースをセットする間、アドレスパケットの先頭を保持するためのバッファを持っている。ネットワークは 2 入出力のスイッチングエレメントにより 4 段のオメガ網が構成されている。出力インタフェースはパケットが届いているかをチェックし、ACK/NAK 信号を生成する。出力インタフェースは

表 1 SSS-TBSF チップの仕様
Table 1 The specification of the SSS-TBSF chip.

Clock : 50 MHz	Bit width : 3 bit (address), 2 bit (data)
Size : 16-in/out	Max bandwidth : 250 Mbits/sec × 16
セル使用数 : 9,838	セル使用率 : 49%
信号ピン数 : 180	テクノロジ : 1.0 μm CMOS sea-of-gates

* 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。

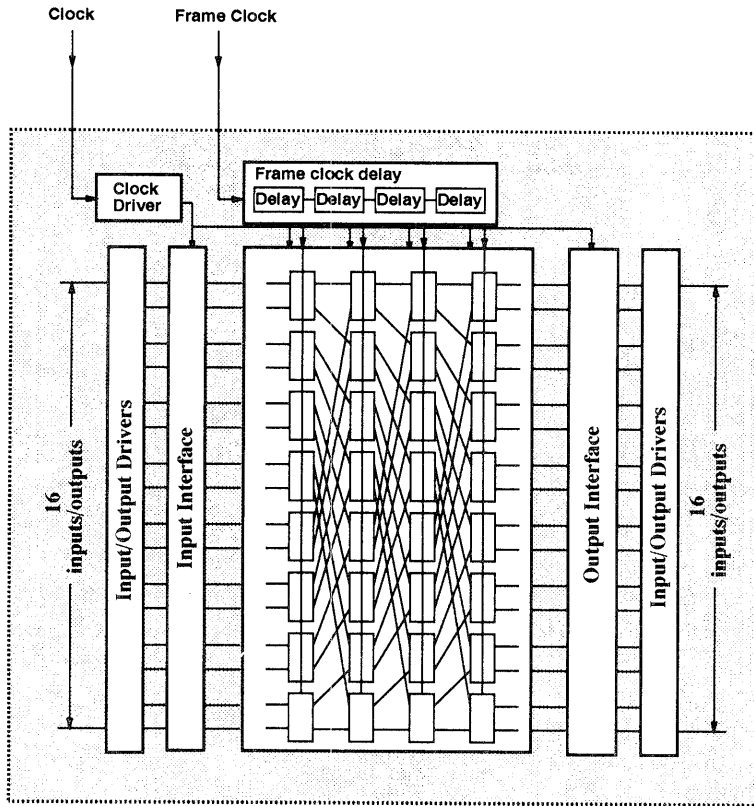


図5 SSS-TBSF チップの構成
 Fig.5 The structure of SSS-TBSF chip.

出力する 1 ビットだけを保持する。

全セル数 9,838 のうち、メッセージコンバイン機能に使用したセルは約 20% である。これより、ビットシリアルメッセージコンバイン機能が低コストで実現されていることがわかる。

1 つのスイッチングエレメントは 192 セルで構成され、70% 以上のセルはアドレス線のコントローラに使用されている。

3.1.2 SNAIL のハードウェア

以下に SNAIL のハードウェアの詳細を示す。

● PU

プロセッサに Motorola 社の MC 68040 を使用しており、各プロセッサはローカルメモリとして 0.5 Mbyte の SRAM を持つ。プロセッサ周辺回路、SSS-MIN チップ間のパラレル/シリアル変換をする回路および SSS 型 MIN に対するインタフェースは FPGA 2 チップで実装されている。

● ネットワーク

SSS 型 MIN は従来型 MIN と異なり、チップを並列に使用してパフォーマンスを改善することが

大変容易である。SNAIL では SSS-MIN チップを 4 個並列に使用しており、パケットはアドレス 12 bit、データ 8 bit パラレルで転送されている。また SSS-MIN チップは TBSF を構成できるようになっているが、シミュレーションを行った結果から 16 プロセッサ程度では TBSF 1 段で十分と判断されたため¹³⁾、ネットワークは TBSF 1 段、すなわち通常のおメガネットワークである。

● メモリモジュール

共有メモリは各モジュールが 4 Mbyte の DRAM で、全体で 64 Mbyte 実装されており、オンチップキャッシュのブロックサイズ (4 ワード) でインターリーブされている。

メモリコントローラは、パラレル/シリアル変換と同期操作、ブロック転送、メモリリフレッシュ等の制御をしている。また、全共有メモリ領域に対して Test&Set と Fetch&Dec の 2 つの同期機構が用意されている。

各メモリインタフェースは FPGA 1 個で構成されている。

4. 性能評価

現在, SNAIL は 16 プロセッサ・16 メモリモジュールのフルスケールで稼働している. 図 6 に示すように, SSS-MIN チップとメモリモジュールを搭載しているネットワークボードに 4 枚のプロセッサボードが繋がっており, プロセッサボードにはそれぞれ 4 プロセッサが搭載されている.

ここでは, 次に示すアプリケーションプログラムを用いて SNAIL の評価を行うことにより, SSS 型 MIN の問題点であるシリアル化や再送によるレイテンシの影響, および SSS 型 MIN に実装されたメッセージコンバインの効果について検討する.

- **Four Coloring**: Hopfield モデルのニューラルネットワークを用いてアメリカ合衆国 48 州の 4 色問題を解くプログラム¹⁴⁾.
- **Nqueen**: 同様のニューラルネットワークを用いて N-Queen 問題を解くプログラム. 問題のサイズは 64×64 ¹⁴⁾.
- **Jacobi**: ヤコビ法により 200 元の三重対角行列を係数を持つ線形一次方程式を解くプログラム. これらの並列アプリケーションはローカルメモリに関して最適化がなされている. つまり, ローカルで使

用されるインストラクションやデータはローカルメモリ上に置かれる. プログラムは C 言語で各プロセッサの動作を直接記述しており, OS のオーバヘッドはない.

4.1 SSS 型 MIN のレイテンシ

MIN 内での衝突がない場合の共有メモリへのアクセス時間を表 2 に示す. ローカルメモリは 125 nsec でアクセス可能なので, 共有メモリへのアクセスはローカルの約 8 倍である.

図 7 は 1 プロセッサにおける実行時間を基準とした時の並列プログラム処理のスピードアップ率である. 並列実行でのオーバヘッドを示すために, 逐次実行での処理結果を合わせて示してある (図中の Serial). 逐次実行の場合, すべての変数やインストラクションをローカルメモリに置き, 同期操作を行わない.

この図によると, 1 プロセッサでは逐次処理は並列処理に比べ 10% から 60% 程度高速であることが分か

表 2 メモリアクセス時間

Table 2 Shared Memory access latency of SNAIL.

Frame period	320 nsec
32 bit data read/synchronization	360 nsec
1 line block transfer	800 nsec

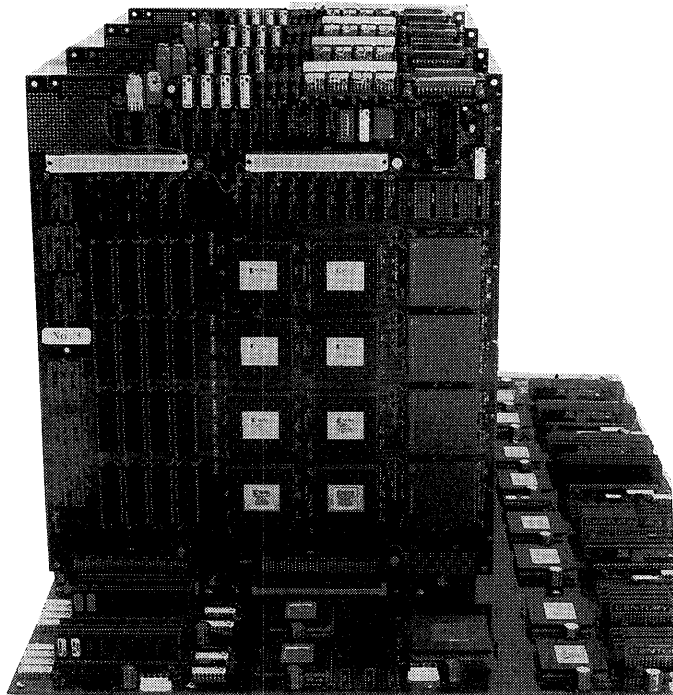


図 6 SNAIL の概観

Fig. 6 The picture of a multiprocessor SNAIL.

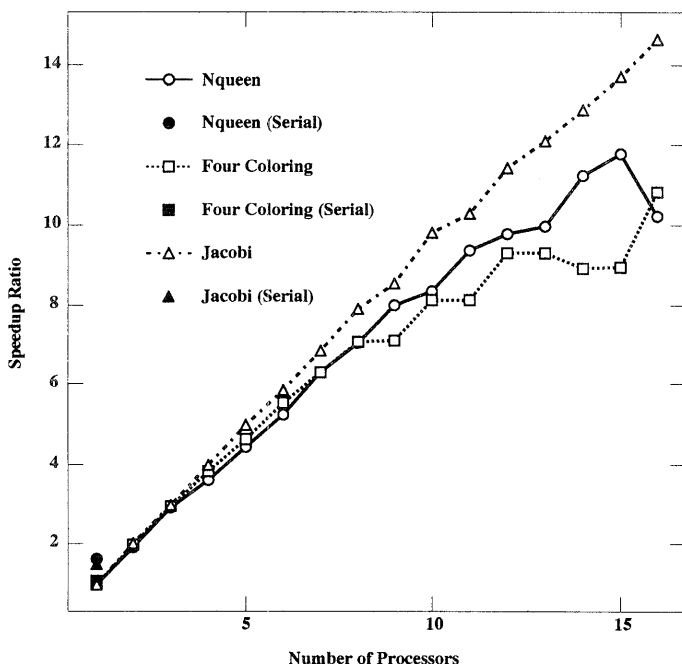


図7 スピードアップ率

Fig. 7 Speed up ratio of applications.

る。1プロセッサでは同期によるオーバーヘッドがないため、実行時間が増えた分はSSS型MINのレイテンシによるものである。共有メモリへのアクセス時間がローカルメモリの約8倍であることを考えると、この程度のオーバーヘッドは妥当である。

Four Coloring については並列性の限界により、12プロセッサ以上では性能の向上が見られない。これに対し **Jacobi** は、浮動小数点演算が多いので、ネットワークアクセスよりも計算時間が長くなる。そのため、ネットワークのアクセスやパケットの衝突によるロスが相対的に小さくなり、直線的に向上する。**Nqueen** では、16プロセッサでの値が減少している。これは、次節で述べるとおりネットワークへの負荷が大きくなり、衝突が増えたためと考えられる。

しかし、全体としては高い台数効果が得られており、このことから共有メモリアクセスによるレイテンシは、SNAILのパフォーマンスを著しく低下させてはいないことがわかる。

4.2 SSS型MINの混雑

上記のアプリケーションにおけるSSS型MINの混雑の程度をパフォーマンスモニタを用いて測定した。図8では、全プロセッサが送出した全パケットの衝突の割合を示している。

この図をみると、すべてのアプリケーションで、プ

ロセッサの増加に伴い、衝突の率も上昇していることが分かる。特に **Nqueen** では、16プロセッサの時には、45%程度とほぼ半分のパケットが衝突している。このことにより、16プロセッサでは実行時間が低下したものと考えられる。

また、メモリモジュール数を8つに減少させ、共有メモリへのアクセスがより集中するようにし、同一のプログラムを実行した際の衝突率を図9に示す。グラフは16メモリモジュールの場合とほとんど同一であり、SSS型MINの高速性により、アプリケーション実行時のアクセス集中による性能低下がかなり抑えられていることがわかる。また、先に述べたようにSSS型MINではスタベーションが発生する可能性があり、共有メモリへのアクセスを集中させた場合、優先度の低いプロセッサのパケット再送回数が増加する。そのため衝突率が増加することが予想されるが、今回の評価ではこの現象は見られず、アプリケーション実行時のスタベーションによる影響は小さいと思われる。

図10は、プロセッサ台数対フレーム利用率を示している。フレーム利用率とは、あるフレームでプロセッサがパケット転送を行う割合をさす。

この図によると、フレーム利用率はアプリケーションにより異なっていることがわかるが、**Jacobi** と **FourColoring** では台数が増えてもあまり増加しな

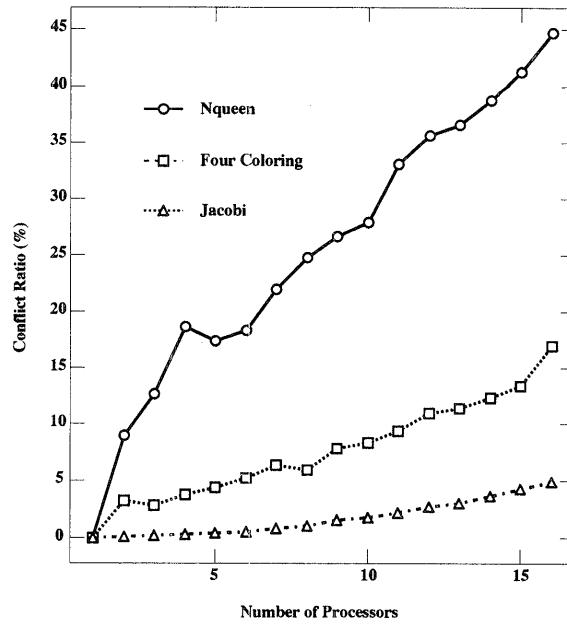


図8 パケット衝突率 (16 メモリモジュール)

Fig. 8 The ratio of the conflicting packets to all packets (16 memory modules).

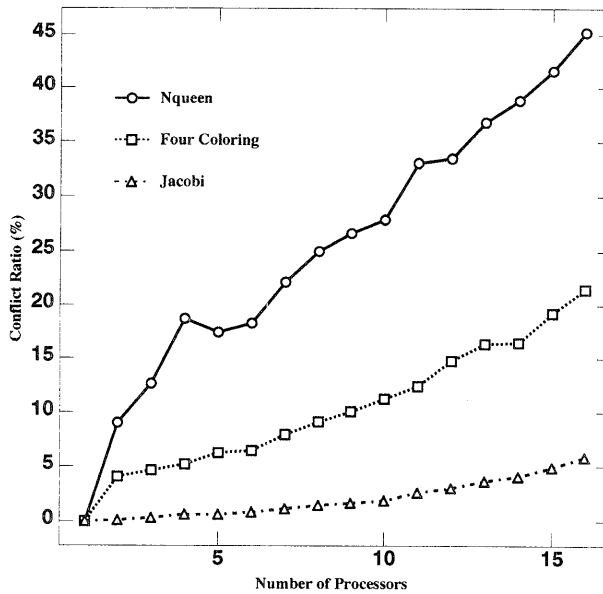


図9 パケット衝突率 (8 メモリモジュール)

Fig. 9 The ratio of the conflicting packets to all packets (8 memory modules).

い。それに対し **Nqueen** では、台数の増加と共に利用率も増えていることがわかる。これは **Nqueen** はもともとネットワークの利用率が高い上に衝突の頻度も高く、パケットの再送が頻繁に行われるためである。

4.3 メッセージコンバインの効果

従来型の MIN における Hot Spot による性能の低下やメッセージコンバインの効果は、様々な論文で解析、評価が行われている^{(6),(7),(15)}。SSS 型 MIN は衝突に

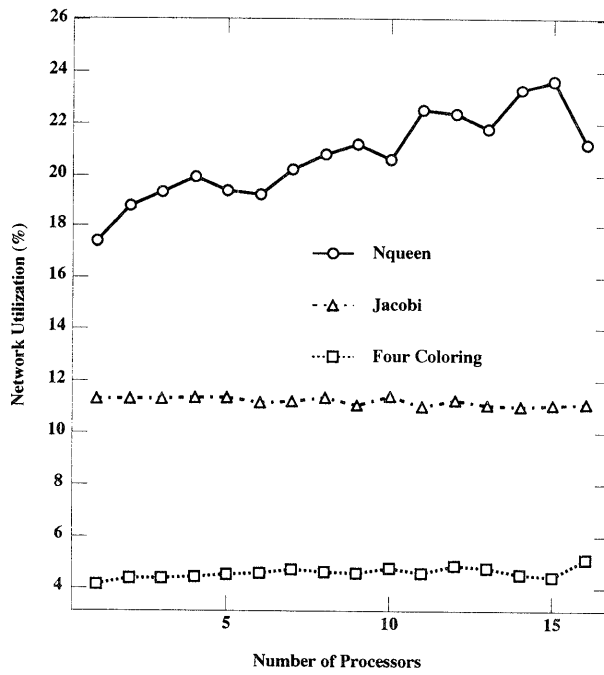


図 10 フレーム利用率

Fig. 10 The ratio of the frame utilization for a processor.

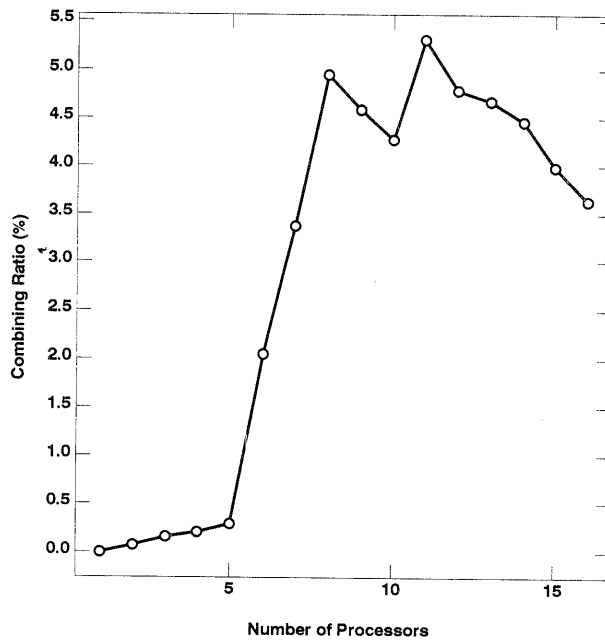


図 11 メッセージコンバイン率

Fig. 11 The ratio of combining packets to all packets.

よってパケットの再送が行われる点で, Wu¹⁶⁾らによる Circuit Switching 転送の Dropped のケースに似ている. このモデルに基づき, 確率モデルとシミュレー

ションによって SSS 型 MIN における Hot Spot とコンバインの解析を行った結果, パケットの再送が Tree Saturation によるネットワークの麻痺状態を防ぐた

め、SSS型MINは従来型MINに比べ、Hot Spotに対する破滅的な性能低下を起こさないことがわかった¹⁷⁾。さらに、コンバインの効果を評価したところ、16プロセッサでフレーム使用率を35%、全アクセスのうち5%が同一のHot Spotへのアクセスと仮定した場合、コンバインによって約5%の性能向上が予想されている¹⁷⁾。

さて、実際のアプリケーションプログラムで、メッセージコンバインの効果を調べたが、全アプリケーションで16プロセッサシステムにおいてできえ実行時間改善には至らなかった。

図11に示すとおり、全パケットに対するコンバインされたパケットは数%程度であった。単純計算でもネットワーク利用率が20%、コンバイン率が数%であることより、実行時間改善に至らなかったのは、コンバインされたパケットの多くは、同期の際のビジーウェイトのパケットであるためと考えられる。

SNAILでは、フレームクロックに同期してパケット転送をするため、同一アドレスへのアクセスが、多少ずれたタイミングで起きても、同一タイミングで転送されコンバインされるが、ネットワークにバッファを持たないため、異なるフレームで転送された場合にはコンバインすることができない。このためSSS型MINでは、データ参照でのコンバインの効果を発揮させるには、スケジューリングによって同一アドレス参照を同一フレームに合わせる事が不可欠である。そこで、ハードウェアによる同期機構を設けて同期の解放をクロックレベルで行えるようにし、読み出しコンバインのブロードキャスト能力をより効果的に活用する方法について現在実験中である。また、この機構を用いて最低限度の同期を行い、小さな同期をスケジューリングによりはずしてしまう方向も考えられる。この場合同期コストの削減に加えてコンバインによるブロードキャストを有効に活用でき、現在この方法による性能向上の評価が行われている。

5. 結 論

SSS型MINの性能を評価するため、我々はマルチプロセッサSNAILを実装し、評価を行った。SNAILは、現在16プロセッサ・16メモリモジュールのフルスケールで稼働中である。SNAILでは、16プロセッサからのパケットが1 μ COS SSS-オメガゲートアレイによって50MHzのクロックでスイッチングされる。また、わずかに20%のハードウェアを付加するだけでメッセージコンバインの実装を可能にしている。

SNAIL上で数種のアプリケーションを用いて行っ

た評価の結果、SSS型MINは十分なバンド幅を持っており、アーキテクチャ上の問題であるレイテンシと再送による影響はそれほど大きくないことが確認された。またSNAIL自身も高い台数効果を持つことが分かった。さらに、ビットシリアルメッセージコンバインについては、今回の評価では十分な効果は確認されなかったが、プログラムの最適化・スケジューリングにより効果が引き出せると期待される。

現在、スケジューリング支援のためのフレーム単位のバリア同期機構をデバッグ中である。また現在のアプリケーションの最適化と、SPLASHベンチマークを実装中である。さらに今後、スケジューリングによる同期操作の除去、この場合のコンバインの有効な利用法についても検討を加える予定である。

参 考 文 献

- 1) Broomell, G. and Heath, J. R.: Classification Categories and Historical Development of Circuit Switching Topologies, *ACM Comput. Surv.*, Vol. 15, No. 2, pp. 95-132 (1983).
- 2) Pfister, G. F. et al.: The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture, *Proc. of 1985 Int. Conf. Parallel Processing*, pp. 764-771 (1985).
- 3) Konicek, J. et al.: The Organization of the Cedar System, *Proc. 1991 Int. Conf. Parallel Processing*, pp. I-49-56 (Aug. 1991).
- 4) Amano, H., Zhou, L. and Gaye, K.: SSS (Simple Serial Synchronized)-MIN: A Novel Multi Stage Interconnection Architecture for Multiprocessors, *Proc. of the IFIP 12th World Computer Congress*, Vol. I, pp. 571-577 (Sept. 1992).
- 5) 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).
- 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. Comput.*, Vol. C-32, No. 2, pp. 175-189 (1983).
- 7) Phister, G. F. and Norton, V. A.: Hot Spot Contention and Combining in Multistage Interconnection Networks, *IEEE Trans. Comput.*, Vol. C-34, No. 10, pp. 943-948 (1985).
- 8) Hanawa, T. and Amano, H.: Multistage Interconnection Networks with Multiple Outlets, *Proc. of 1994 Int. Conf. Parallel Processing*, Vol. 1, pp. I-1-8 (Aug. 1994).
- 9) 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).

- 10) Tobagi, F. A. and Kwok, T.: The Tandem Banyan Switching Fabric: a Simple High-Performance Fast Packet Switch, *Proc. IN-FOCOM91*, 3.3.1-3.3.6 (Apr. 1991).
- 11) 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会報告, No. 94-5 pp. 33-40 (1992).
- 12) Lawrie, D. H.: Access and Alignment of Data in an Array Processor, *IEEE Trans. Comput.*, Vol. C-24, No. 12, pp. 173-183 (1975).
- 13) 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, CPSY 92-27 (SWoPP 92), pp. 57-64 (1992).
- 14) 大和純一, 鈴木響太郎, 天野英晴, 武藤佳恭: ニューラルネットワークを用いた最適化アルゴリズムの並列計算機への実装と評価, 信学報, CPSY 92-1-8 (SWoPP 92), pp.17-24 (1992).
- 15) Lee, G. H., Kruskal, C. P. and Kuck, D. J.: The Effectiveness of Combining in Shared Memory Parallel Computers in the Presence of 'Hot Spots', *Proc. Int. Conf. Parallel Processing*, pp. 35-41 (1986).
- 16) Wu, C. L. and Lee, M.: Performance Analysis of Multistage Interconnection Network Configurations and Operations, *IEEE Trans. Comput.*, Vol. 41, No. 1, pp. 18-27 (1992).
- 17) Gaye, K., 埴 敏博, 天野英晴: SSS 型 MIN における hot spot の影響とメッセージ結合の効果, 電子情報通信学会論文誌 D-I, Vol. J-77-D-I, No.5, pp. 354-363 (1994).

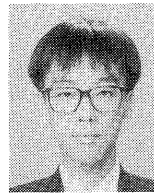
(平成 6 年 10 月 7 日受付)

(平成 7 年 4 月 14 日採録)



笹原 正司

平成 5 年慶應義塾大学工学部電気卒業。平成 7 年同大学大学院計算機科学専攻修士課程修了。現在株式会社東芝に勤務。並列計算機のアーキテクチャに興味を持つ。



寺田 純

平成 5 年慶應義塾大学工学部電気卒業。平成 7 年同大学大学院計算機科学専攻修士課程修了。現在日本電信電話株式会社に勤務。並列計算機のアーキテクチャに興味を持つ。



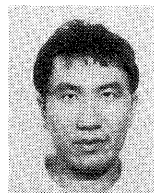
大和 純一

平成 4 年慶應義塾大学工学部電気卒業。平成 6 年同大学大学院計算機科学専攻修士課程修了。現在 NEC C&C 研究所に勤務。並列計算機のミドルウェアに興味を持つ。



埴 敏博

平成 5 年慶應義塾大学工学部電気卒業。平成 7 年同大学大学院計算機科学専攻修士課程修了。現在同大学大学院博士課程在学中。並列計算機の相互結合網の性能解析に興味を持つ。



天野 英晴 (正会員)

昭和 56 年慶應義塾大学工学部電気卒業。昭和 61 年同大学大学院博士課程修了。工学博士。並列計算機の研究に従事。現在慶應義塾大学工学部電気工学科助教授。