計算機アーキテクチャ 120－2
（1996．10．31）

SCIを用いた並列プロセッサの試作と評価

宮田裕行† 西方茂樹††
山崎弘巳†† 青山和弘††

並列プロセッサを構成する高速バスとして、近年米国を中心にして注目を浴びているバスにSCI(Scalable Coherent Interface)と呼ばれるIEEE規格のインタフェースがある。SCIは、従来のバス構成を越える高い拡張性と共有メモリプログラミングパラダイムの両者を実現する目的を持つ。筆者らは、これまでにその基本結合網の検討、動的負荷分散方式の検討を行ってきた。本稿では、それらの実証とSCI自身の性能を評価することを目的として設計した試作機の概要とその評価結果について述べる。

A Prototype Design and an Evaluation of SCI-based Multi Processor

HIROYUKI MIYATA,† SHIGEKI NISHIKATA,††
HIROMI YAMAZAKI†† and KAZUHIRO AOYAMA††

SCI (Scalable Cohenret Interface) is an IEEE standard interface which can be used for constructing a multi processor. The standardization has been discussed in USA in the last few years. SCI can be an enabling technology to develop scalable multi processor and give a shared memory programming paradigm. We have been doing research about SCI network topologies and dynamic load balancing on it. In this paper, we describe an abstract of a prototype machine of SCI-based multi processor and the result of the evaluation.

1. はじめに

ユーザに対してプログラムし易い共有メモリプログラミングパラダイムを提供するという要件と、性能面で高水準の拡張性を備えるという要件を、一つの並列計算機において同時に実現することは、一般に難しい。しかし、この2要件を同時に満たす並列計算機を、従来を越える水準で実現することに対する潜在的需要は依然大きく、この二要件を実現するための実装技術が研究されている。この研究領域で、近年重視されているのが、IEEEが策定したSCI (Scalable Coherent Interface) と呼ばれる要素技術規格[1]である。

SCIは、従来のバス構成を越える高い拡張性と、共有メモリプログラミングパラダイムの同時に実現という目的に徹したインタフェース規格である。米国において1980年代後半からその規格化が始まり、1991年に最初の1.00バージョンが策定され、現在もリアルタイム性を重視する要素等を含めた形で更新が続いている[2]-[5]。SCIの特徴とその将来性については、その3文字(SCI)の各々の元の単語(Scalable, Coherent, Interface)が語るように、次の3点が考えられている。

第一に、SCIはpoint-to-point接続のリングベース規格であるため、ノード数の増加に際して、共有バスで見られるような競合の集中が生じないばかりか、共有バスでは不可能であったデータ並行転送ができる。これは、1ノードあたりのバンド幅が、バス構成に比べて飛躍的に向上される結果を生む。1 Gbyte/secという高速なデータ転送速度も、上記の特徴と相乗的に働き、SCIの将来性を高めている。第二に、SCIではプログラマビリティの要請が視野に入っているため、ディレクトリベースのキャッシュ整合性制御方式が規定されている。これは、キャッシュ整合性制御のハードウエア実装への障壁を低くするものである。この制御をハードウエア実装し制御オーバーヘッドを縮小することで、システム性能劣化の主要因のひとつを回避できる。この特徴によりSCIの実際的価値は極めて高いものとなった。第三に、SCIはIEEE策定の標準規格であるため、ベンダー・サードパーティ各社の市場参入により、市場全体としての開発コストが低減され、さらにそれに伴なってSCI方式の採用の活性化という好循環が期待されている。

筆者らは、これまでに、SCIの種々の接続形態に対する理論的な評価を行い、ノード数により最適となるアーキテクチャを示した[6]。また、リアルタイムの信号処理をSCIを用いた並列プロセッサで構築する場合に問題

† 三菱電機(株) 情報技術総合研究所
Information Technology R&D Center,
Mitsubishi Electric Corporation
†† 三菱電機(株) 鎌倉製作所
Kamakura Works, Mitsubishi Electric Corporation

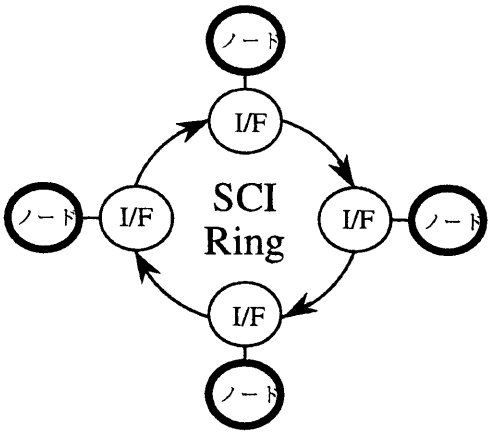

図1　試作機の全体構成

CPU	DEC 社製 Alpha 21064@166MHz
1次キャッシュ	16KB（CPU に内蔵）
2次キャッシュ	256KB（非同期 SRAM）
メインメモリ	8MB（高速ページモード DRAM）
インターコネクト	SCI

表1　プロトタイプの諸元

となるタスクの動的負荷分散に対し、シミュレーションによる評価を行ってきた[7]-[8]。これにより、SCI を使用したリアルタイム信号処理の問題点がクリアになり、さらにその実証のために、プロトタイプの開発を進めている。

本論文では今回試作したプロトタイプの概要を紹介すると共に、これまでに行ったハードウェアの性能評価について報告する。

以下、2章では、試作したプロトタイプの諸元を述べ、3章でハードウェアの性能評価を、4章で評価に対する考察を述べ、5章をあとがきにあてる。

2. プロトタイプの諸元

2.1 概　　要

今回試作したプロトタイプは、4つのノードが単一方向の1重リングからなる構成である（図1参照）。その諸元を表1に、試作したボードの写真を図2に、ブロック図を図3に示す。ノードは DEC 社製のアルファ CPU を用い、1次キャッシュはチップ内蔵の 16KB, 2次キャッシュは外づけで 256KB、メモリは 8MB の容量とし、試作のための必要最低限の量とした。また、メモリコントローラは、FPGA により設計した。SCI 用のリンクには、Dolphin 社の Link Controller-1（以下 LC-1) チップを使用した。LC-1 回路のプロセッサ側のインタフェースは、Blink と呼ばれるバスによって接続されるため、この Blink と CPU のローカルバスとの接続にはデュアルポートメモリ（以下 DPM）を介して接続した。また、この制御を行うバスブリッジ回路も FPGA を2石用いて設計した。このバスブリッジ回路については、次に示す。

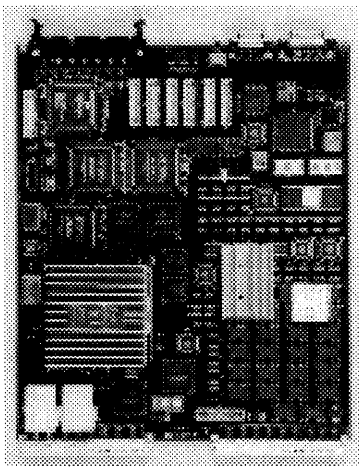

図2　プロトタイプのボード写真

2.2 バスブリッジ回路

LC-1 チップは、SCI と Blink と呼ばれる CPU 側との接続を仲介する回路である。そのため、図4に示すように、Blink と CPU ローカルバスとのプロトコルを変換するためのブリッジ回路が必須となる。すなわち、CPU のローカルバスはアドレス、データ、制御信号がそれぞれ専用線を使って同時に出力されるが、Blink はアドレス、データ、ステータス信号をパケットに埋め込んで単一のデータ線で転送する。このため、両者のデータ転送プロトコルをブリッジ回路により変換する。

ブリッジ回路は信号線の種類が多くなり、1 LSI のピンでは不足するため、2個の FPGA に分割して構成した。分割は信号の区切りのつけやすさから、CPU のローカルバス側の DPM 制御部と、Blink 側の DPM 制御部とを分離し、この2石のインタフェースは専用のハンドシェイク信号により行った。ブリッジ回路を構成するにあたり、CPU バス操作と SCI のサブアクション [1] とのすり合せが必要になる。今回のプロトタイプでは、共有メモリ方式を採用するため、メモリ操作に直接対応できるサブアクションに限定した。すなわち、グローバルメモリの読み出しには、SCI の nread64 トランザクションを、グローバルメモリの書き込みには、SCI の dmove64 トランザクションを対応させた。

2.3 メモリマップ

図5には、プロトタイプでのメモリマップを示す。SCI のメモリ構成に合わせた共有メモリ方式を採用しており、各ノード単位にアドレスをアサインした。このため、ノード n から全アドレスマップを参照すると図5の

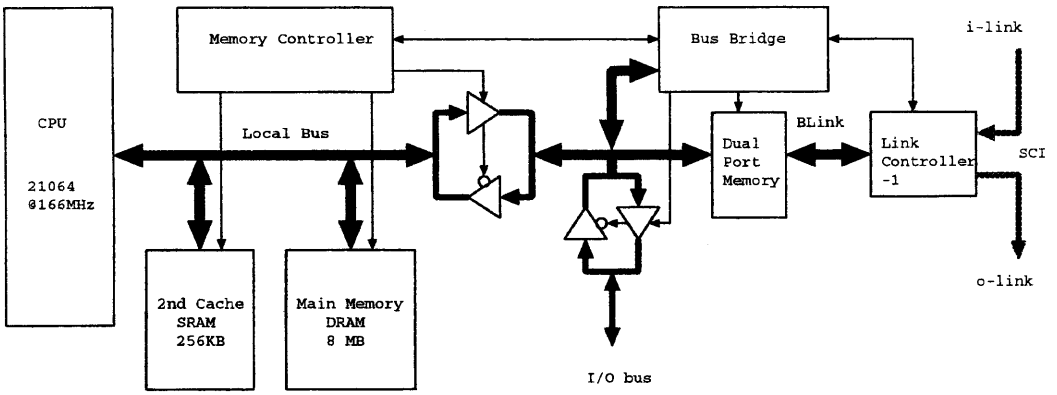

図3　ボードの主要ブロック図

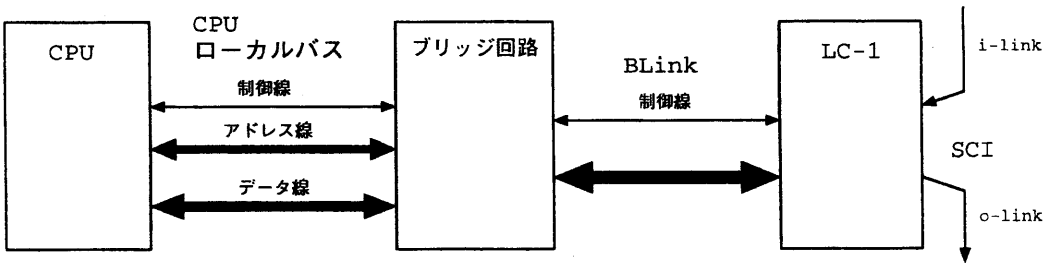

図4　LC-1 と CPU との接続

ようになる。最もアドレスの低い所はローカルメモリとし、各ノードに与えられたメモリと同一のものとした。すなわち、各ノードにおいて、メモリ上に同一の自身のメモリ領域が2つ存在することになる。プロセッサはローカルメモリに対しては、ローカルバス経由のアクセスであるが、他のノードのグローバルメモリに対してはSCIを経由してアクセスする。

また、ローカルメモリのアクセス時には、キャッシュを介在させ、高速なメモリ操作が可能であるが、グローバルメモリに関し、SCIのキャッシュコヒーレントプロトコルを使用したキャッシュシステムはLC-1チップ以外にかなりの規模の外づけ回路が必要であるため、今回は実装しなかった。

2.4 デュアルポートメモリ (DPM)

DPMは、グローバルメモリアクセスのためのデータ管理に使用する。CPUでのデータアクセスは一度に8バイトまでしかできないため、SCIでの64バイト単位転送との間で無駄が生じる。そのため、グローバルメモリの読み出しにおいては、一旦、リードデータをDPMに保存し、残りの56バイトのデータアクセスについては、自身のバッファをリードすることとした。また、DPMでは、読みだし、書き込み用に、各々8個のバッファを設けることとし、他のノードから自ボードへのアクセスは一度に8個まで受信可能とした。これを超えるアクセスはビジーを出力して拒否するが、今回は4ノードであるため、この制限を超えることはない。また、CPUから他のノードへのアクセスは他ノードからのパケットが受信バッファに溜まっている間でも可能であり、スプリットトランザクションの利点を活かすことができる。

3. 性能評価

本プロトタイプによりSCIを使用したことが、従来のバス結合によるマルチプロセシングシステムに比べて、どの程度の優位性を有するのかを調べるために性能評価を行った。

3.1 評価項目

本報告における測定項目を以下に示す。

(1) メモリアクセス時間（グローバル、ローカル）
(2) スケーラビリティ
(3) SCI上のノード通過時間

3.2 評価条件

測定条件を以下に示す。

(1) CPU動作周波数：166MHz
(2) システム周波数：20.75 MHz（CPUのクロックの1/8）
(3) Blink周波数：10.375 MHz（システム周波数の1/2）
(4) SCI周波数：50 MHz

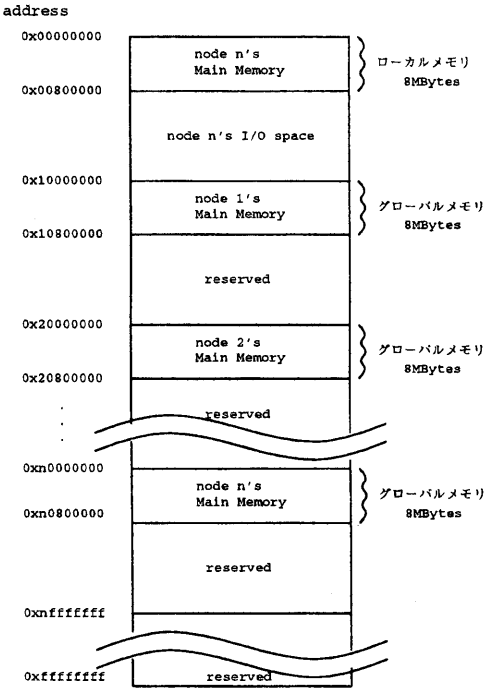

図5　メモリマップ

(5)　4台ノード構成

なお、SCIの論理的な最大転送レートはクロックの立ち上がりと立ち下がりの両方でデータを転送するため、50MHz x 2 x 16bits = 200Mbytes/s である。システムの最大スループットはこれらのラインが4本並列に動作可能なため、800Mbytes/s となる。ただし、これはすべてのノードがリング方向に隣接するノードにデータを送り続けるときにのみ出る値である。

また、評価においては、測定誤差を抑えるため、すべての計測値は5回測定し、最大値、最小値を除いて平均値を求めた。

3.2.1　メモリアクセス時間

メモリアクセス時間の評価結果を表2に示す。単位としては、ローカルメモリの読み出し時間を1.0とした場合の相対時間で表した。対象としたのは、SCIを使用した本システムのグローバルメモリアクセス、ローカルメモリアクセス、および同条件下でのVMEでのバス結合によるマルチプロセッサのグローバルメモリアクセス、DPM上のバッファへのアクセスである。ただし、メインメモリアクセスに関し、キャッシュの影響はなくす用にして行った。この考察については、次章でまとめて行う。

3.2.2　スケーラビリティ

システムのスループットは、リング上に最大負荷のトラフィックを発生させ、転送が滞りなく行われるかどうかを確認すればよい。この計測のために、図6に示すようにリング上の4つのノードにそれぞれから最も遠いノードのメモリに対してライト操作を行った。この結果、図中のすべてのノードにこのシステムでの最大トラフィックが発生する。もし、このトラフィックを処理できない場合、いずれかのノードでビジーが発生し、再送ができなくなれば、ハードウェアエラーが発生する。しかし、計測の結果、ビジーは発生したものの、再送処理で正常にライトされるため、ハードウェアエラーで停止することはなかった。このことからSCIのリンクにはまだ余裕があることがわかり、この時点でのシステムでのスループットを算出すると64Mbyte/sであることが判明した。このスループットは、評価条件として

		転送データ量	相対時間
ローカル	読み出し	32B	1.0
	書き込み	32B	2.0
グローバル	読み出し	8B	7.4
(VME)	書き込み	8B	7.4
グローバル	読み出し	32B	13.4
(SCI)	書き込み	32B	4.0
バッファ	読み出し	8B	2.2

表2　メモリアクセス時間

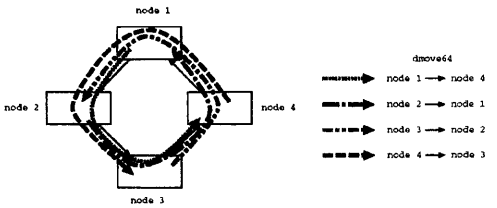

図6　スループット計測

求めたSCIの1ラインあたりの最大データ転送レートの200Mbyte/sに比べて、1/3程度であるため、SCIの負荷的には余裕のあるシステムと言える。逆算すると、(200 Mbyte/s/64Mbyte/s) x 4ノード = 12.5となり、理想的なノード数は、10台程度と考えられる。ちなみに、VME接続マルチプロセッサで計測したスループットは、2.2 Mbyte/sであった。

3.2.3　SCI上のノード通過時間

SCIを使用したマルチプロセッサにおいて、ノード数を増加するに従い、検討が必要な問題点として、ノードをバイパスするときの遅延時間がある。すなわち、リング上の離れたノードにデータを転送する場合、SCIのpoint-to-point接続では、図7に示すような遅延が積み重なり、数十、数百といったノードが接続されると問題になる可能性がある。そのため、このノード通過時間を計測し、表3の結果を得た。なお、時間の単位は、表2と同じである。

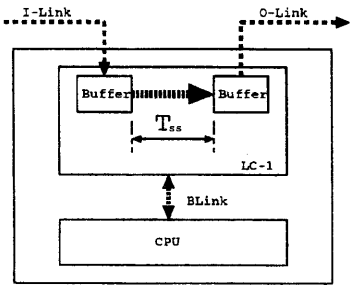

図7　ノードバイパス説明

SCIパケット長	相対遅延時間
16B	0.2
32B	0.4
80B	0.4

表3　ノードバイパス時間

4. 考　　察

(1) システム全体のスループットは、VMEバスに比べ、約30倍 (64:2.2) と非常に高く、point-to-point結合方式の長所が十分に発揮された。しかも、今回の4台結合においての比較であり、理想的には、10台程度接続しても状況は変わらないと考えられる。この場合には、72倍程度の差が出ると考えられる。

(2) SCIのデータ通過時間に関しては、性能を十分発揮できる範囲での接続台数（10台）程度では、ノードをバイパスすることによる遅延は十分に小さいものであると判断できる。しかしながら、数十台、数百台と台数を増加した場合、メモリアクセス時間に比べ、その値は無視できないものとなる。この場合には、リングの構成の変更など別途対策が必要になると考えられる。

(3) SCI経由のグローバルメモリのアクセス時間に関しては、読み出しでは、VMEバスを経由したものよりも遅くなり、かつ、書き込みに比べ、約3.5倍も遅くなるという結果であった。この原因を解析すると以下のものが考えられる。

- CPUとSCIのバストランザクションの不整合
  アルファCPUは、キャッシュしない場合、1回のアクセスで8バイト、また、キャッシュした場合でも、1回のアクセスで32バイトを読み込む。これに対し、SCIは64バイト単位のアクセスであるため、ここで無駄が生じる。
- 読み出しと書き込みの差は、レスポンスの有無
  グローバルメモリのアクセスに関し、書き込みの場合は、dmove64命令を使用したため、書き込み先のノードはレスポンスを返す必要がない。そのため、リクエストを発行した時点でCPUのライトサイクルは終了する。ところが、読み出しの場合は、CPUはレスポンスが返って来るまでウエイトする。このため、他の一切の処理が停止してしまう。
- SCIアクセス時のパケット生成が占める時間が大きい
  パケット生成においては、ブリッジ回路ですべての定型のパケットを生成し、一旦、DPMに格納してから送信する形態とした。これが転送時間の増加を招いた。

以上の結果から、（ここで手法を示すことは割愛するが）改良による高速化を図ると約2－4倍の高速化が図れることがわかった。

5. おわりに

並列プロセッサを構成する高速バスとして、近年米国を中心にして注目を浴びているバスにSCIについて、この実証とSCI自身の性能を評価することを目的として設計した試作機の概要とその評価結果について述べた。

SCIを経由したグローバルメモリのアクセスに関しては、読み出しと書き込みで差があったが、問題となる読み出しに関しても改良により、ローカルメモリと比較ができる程度の高速化が図れることがわかった。

また、今回は、キャッシュのコヒーレンシに関しては、実装しなかったが、性能向上からは必須事項であるため、今後の課題としたい。

今後は、まず、これまでに検討してきた動的負荷分散などの評価結果を試作機上で調べ、その後、改良とコヒーレンシ関係を行っていく予定である。

参考文献

[1] *"IEEE Standard for Scalable Coherent Interface (SCI)"* ,IEEE Computer Society, IEEE Std 1596-1992

[2] *"Proceedings of the First International Workshop on SCI-based High-Performance Low-Cost Computing"*, Aug.17-18, 1994, at Snta Clara Univ., SCIzzL

[3] *"Proceedings of the Second International Workshop on SCI-based High-Performance Low-Cost Computing"*, March 21, 1995, at Snta Clara Univ., SCIzzL

[4] *"Proceedings of the Fourth International Workshop on SCI-based High-Performance Low-Cost Computing"*, Oct. 3, 1995, at Snta Clara Univ., SCIzzL

[5] *"Proceedings of the Fifth International Workshop on SCI-based High-Performance Low-Cost Computing"*, March.26, 1996, at Snta Clara Univ., SCIzzL

[6] 高橋、青山、宮田、菅「SCIを構成要素として用いた各種基本的相互結合網の評価」情処学会 ARC113-10, pp.73-80, Aug. 1995.

[7] 高橋、青山、高野、宮田「リアルタイム信号処理用マルチプロセッサにおける動的負荷分散方式の評価」情処学会 ARC117-3, pp.13-18, March 1996.

[8] H.Miyata, M.Takahashi, H.Takano, K.Aoyama, *"An Evaluation of Dynamic Scheduling Algorithms of Real-Time Signal Processing on SCI-based Multiprocessor Systems,"* SCIzzL-6, pp.49-58, Sept. 24, 1996.