計算機アーキテクチャ 137−11
ハイパフォーマンス 80−11
コンピューティング
（2000．3．2）

高性能並列計算用ネットワーク RHiNET-1 の実装と評価

山本淳二[†1] 建部修見[†2] 横山知典[†3]
土屋潤一郎[†3] 宮脇達朗[†4] 清水敏行[†5]
天野英晴[†3] 工藤知宏[†1]

LASN(Loacal Area System Network) は、フロア内のデスク上で実際に業務に用いている PC 間を光インタコネクトと専用スイッチにより接続することにより，高性能並列計算を実現するためのネットワーククラスである．本報告では、LASN の最初のプロトタイプ RHiNET-1 の実装と RHiNET-1 の基本的な性能の評価を行った．RHiNET-1 は PCI バスに接続する RHiNET-1/NI と 1 チップスイッチ RHiNET-1/SW からなる．評価の結果、現在の RHiNET-1 の性能はネットワークインタフェース上の回路の動作速度により制限されていることがわかった．本報告では RHiNET-1 の構成と評価および今後の改善点について述べる．

Implementation and evaluation of RHiNET-1, a network for LASN

JUNJI YAMAMOTO,[†1] OSAMU TATEBE,[†2] TOMONORI YOKOYAMA,[†3]
JUN'ICHIROH TSUCHIYA,[†3] TATSUAKI MIYAWAKI,[†4]
TOSHIYUKI SHIMIZU,[†5] HIDEHARU AMANO[†3] and TOMOHIRO KUDOH[†1]

The LASN (Local Area System Network) is a new class of network which enables high performance parallel processing by connecting PCs distributed on one or more floors of a building. It provides low latency reliable communication as well as relatively free topology design and longer length of links comparable to a LAN (Local Area Network). Here, the implementation and initial performance evaluation of RHiNET-1, the first prototype of LASN are presented. It consists of network interfaces called RHiNET-1/NI and one-chip switches called RHiNET-1/SW. Although the current performance of RHiNET-1 is limited by the RHiNET-1/NI the way for performance improvement is clearized.

1. はじめに

最近では高性能な PC がビル内やフロア内に分散して配置されて利用されているが，これらは常時稼働しているわけではない．これらの計算機を高速低遅延なネットワークで接続し，余剰計算能力を活用することができれば専用の PC クラスタシステムと同等の能力を得ることができる．また，異なる設置スペースにある，サーバやクラスタシステムの計算能力を組み合わせて高性能計算を行なうことも可能になる．

このように分散して配置された計算機群を接続するには従来 LAN が用いられてきた．Gigabit Ethernet や 1.06Gbps の Fibre Channel などが実用・普及段階に入り，LAN 環境でも数ギガビットクラスの高速な通信が可能となっている．しかし，従来の LAN では物理層でのパケットの廃棄や順序の入れ変えを許しているため，信頼性のある通信を行う場合には，上位レイヤにおいて再送や順序の入れ換えを行う TCP/IP などのプロトコルを用いる必要がある．しかしこの様なプロトコルはホストプロセッサが行う処理のオーバヘッドが大きく，数ギガビットのバンド幅を持つネットワークを有効利用するだけの実効バンド幅を得ることが困難である[3]．このように，LAN は高性能並列計算のためのネットワークとしては問題がある．

そこで，我々は新しいネットワーククラス，LASN (Local Area System Network) を提唱している．LASN は図 1 に示すようにフロアやビルに分散して配置されている PC を接続し，PC クラスタと同様の並列処理を行うシステムである．LASN は専用 PC クラスタで用いられている Myrinet[2] を代表とする SAN(System Area Network) と同程度の広いバンド

†1 新情報処理開発機構
Real World Computing Partnership
†2 電子総合技術研究所
Electrotechnical Laboratory
†3 慶應義塾大学
Keio University
†4 NEC 情報システムズ
NEC Informatec Systems
†5 シナジェテック
Synergetech

幅を持ち，レイテンシが小さく，パケットの到着順序を保証し，さらにLANと同様な環境で用いるために，自由なトポロジおよび，十分なリンク長を提供する必要がある[4),5)]．

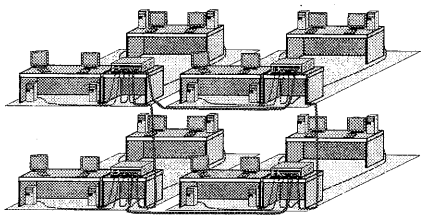

図1 LASN

LASNの一つとして我々は，RHiNET(RWCP High Performance Network)と呼ぶシステムを開発している．RHiNETはネットワークとネットワークインタフェース(NI)から構成され，上記LASNの性質に加えてNIが並列処理をサポートする様々な機能を持つ．RHiNETの最初のプロトタイプとしてRHiNET-1を開発した[6)]．RHiNET-1は，RHiNET-1/NIおよび大容量パケットバッファを備えた1チップスイッチRHiNET-1/SWを高速光リンクで接続する構成を持つ．リンクは1.3Gbpsの転送容量を持ち，100mの距離で配置された100台規模のPCを，ほぼ自由なトポロジで接続することができる．

RHiNET-1/NIは，33MHz/32bitsのPCIバスに装着されるNIである．NIによる並列処理のサポートについて試行実験を行なうためにPCIバスを介してプログラム可能なCPLD(Complex Programmable Logic Device)をコントローラに採用している．このCPLD上の回路構成を変えることにより，様々な機能をNIに持たせて動作検証，性能評価を行なうことができる．現在入手可能なreconfigurableなCPLDでは，やや規模の大きい回路を実装した場合，33MHzで動作させることは困難であるが，試行実験を容易にするため敢えてこの構成を採った．

33MHz/32bitsのPCIバスは単純計算で1.05Gbpsのデータ転送バンド幅を持つ．RHiNET-1のリンクは1.3Gbpsであるから，NI上の回路が十分な速度で動作すればPCIバスがシステム全体のデータ転送バンド幅の隘路となるはずである．しかし，実際にはCPLDの動作速度の問題や現在採用しているPCIバスインタフェース回路の構成などにより，このバンド幅は得られていない．また，通信レイテンシについてもいくつかの改良の余地がある．

本報告では，RHiNET-1/NIを用いてデータ転送を行なう際の各部の動作速度についての評価結果を報告し，今後の改良の可能性について考察する．

表1 プリミティブ一覧

プリミティブ	(基本)	twin/diff	bitmap	stride
push	○	diff	○	○
pull	○	twin	○	○
mcast	○	diff	○	○
isend	○			
irecv	○			
lock	○			
barrier	○			

2. プリミティブ

RHiNETを用いるプログラムはプリミティブと呼ぶAPIを通じてNIを利用することで他のプロセスとの通信を行う．

表1にRHiNETで提供する主なプリミティブの一覧を示す．RHiNETではこれらのプリミティブはすべてNIが処理する．そのため，ホストプロセッサはNIに処理を発行した後は，別の処理を継続して実行することが可能である．

RHiNETが提供するプリミティブは大きく3つに分けられる．

- リモートメモリアクセス
  リモートメモリライトであるpush，リモートメモリリードであるpullを中心に構成されている．バリエーションとして不規則に並んだデータの転送を行うbitmapや，規則的な不連続なデータを転送するstrideなどもサポートする．データのリード時のコピー(twin)の作成と，twinとメインメモリのdiffを用いた転送により multiple writer protocolをサポートする．
  複数の宛先に対してメモリライトを行うマルチキャスト(mcast)も提供する．マルチキャストはスイッチの機能を用いるため，宛先の数に関わらず処理時間はほぼ一定となる．
- メッセージ通信
  MPIに代表されるメッセージ通信をサポートするプリミティブとしてisendとirecvを提供する．
- 同期機構
  排他処理や同期操作をサポートするプリミティブとしてlock(unlock)とbarrierを提供する．

最も基本的なプリミティブであるpushのライブラリインタフェースを図2に示す．ライブラリは与えられた引数をNIの所定のレジスタに書き込むことでプリミティブの実行を開始する．

送信データのアドレスなどは仮想アドレスで与える．NIは必要に応じてTLBによる変換を行う．受信側のデータ領域のアドレスはプログラムの状態により変化するため，仮想アドレスではなく通信エリアIDとオフセットにより与える．通信エリアIDと実際の仮想アドレスの対応は別途受信側のプロセスによって設定

```
push( src, remid, comid, offset, flag, status_ret, size );
  void *src;              /* 送信データのアドレス */
  PPID remid;             /* 宛先プロセスの ID */
  GID comid;              /* 受信側通信エリア ID */
  unsigned offset;        /* オフセット */
  FLAG flag;              /* 各種動作設定用フラグ */
  STATUS *status_ret;     /* 終了フラグアドレス */
  unsigned size;          /* 転送データサイズ */
```

図 2 プリミティブの書式 (push)

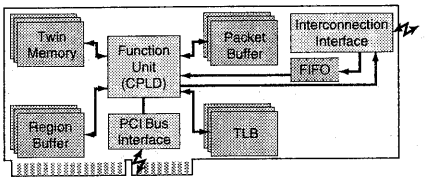

図 3 RHiNET-1/NI Block Diagram

される．NI はメインメモリ上の変数 (**status_ret**) を用いてプリミティブの終了やエラーを通知する．

3. RHiNET-1 システム

RHiNET-1 は RHiNET-1/NI，および RHiNET-1/SW，および相互結合網，プリミティブにより構成される．

3.1 ネットワークインタフェース:RHiNET-1/NI

3.1.1 RHiNET-1/NI 上の構成:

RHiNET-1/NI は，並列処理を効率よく行うための PCI インタフェースを用いたプロトコル処理ハードウエアである．アプリケーション開発は message passing (MPI) ベース，共有メモリ (Open/MP) ベースの両方のスタイルで記述でき，専用の LINUX のライブラリおよびデバイスドライバを用いる．このプリミティブ処理は NI で行われ，これらを組み合わせることにより，マルチタスク環境での zero-copy 通信を実現する．

図 3 に RHiNET-1/NI のブロック図を示す．RHiNET-1/NI は，パケット送受信部 (Interconnection Interface)，PCI バス制御部 (PCI Bus Interface)，プリミティブ処理部 (Function Unit) およびプリミティブ処理用に用いるメモリから構成される．高速動作が要求されるパケット送受信部および PCI バス制御部は，高速な QuickLogic 社のアンチヒューズ型 FPGA を用い，プリミティブ処理部は，in-system programming が可能な FPGA(Altera 社 CPLD) を用いている．

基板の写真を図 3.1.1 に示す．

3.1.2 RHiNET-1/NI 上でのプリミティブ処理:

ローカルノード (Initiator) からリモートノード (Remote) にパケットを転送する push プリミティブの実行 (図 5) を例に RHiNET-1/NI の動作を解説する．

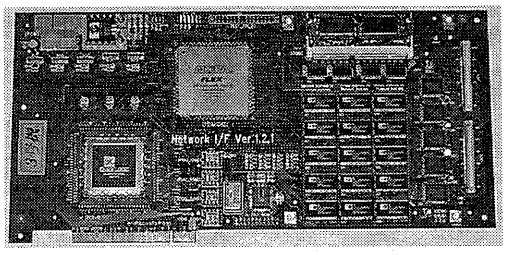

図 4 RHiNET-1/NI 基板

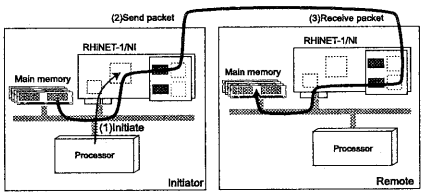

図 5 Push の動作

(1) 起 動:

プリミティブ処理部は，プリミティブ起動に必要な情報を格納するプリミティブ起動用情報レジスタを持つ．ユーザプロセスが，ライブラリ経由でプリミティブ実行に必要な情報 (送信データアドレス，宛先プロセス ID，サイズ等) をプリミティブ処理部のレジスタに書き込むことで，プリミティブが起動される．

プリミティブ処理部は，ボード上の TLB を参照して仮想アドレスを物理アドレスに変換した後，PCI バス制御部に転送すべきアドレスおよびサイズを与えることで，DMA 転送の要求を出す．PCI バス制御部とプリミティブ処理部は，データ転送用のローカルバスと，DMA 転送起動用の DMA バスで二重に接続されており，データ転送中も次回の DMA の起動をかけることができる．

(2) パケット送信:

PCI バス制御部は，要求に従い主記憶からプリミティブ処理部にデータ転送を開始する．この間，プリミティブ処理部は，パケットヘッダを準備し，パケット送受信部の送信用 FIFO にヘッダの送信を開始する．DMA 転送された 32bit データは，即座に送信用 FIFO に入れられる．パケット送信部は，送信用 FIFO にデータが入り次第，32bit データを光インターコネクトで転送する 8bit 形式に変換し，転送を開始する．データ転送量が多い場合，レイテンシ・タイマが働くため，一時 DMA 転送が中断されることがある．中断された結果，送信用 FIFO が空になった場合は，パケット送信部は自動的に idle flit をパケット中に挿入

する．DMA による転送が終了した場合，プリミティブ処理部は，主記憶の所定の番地に対して，終了通知フラグを書き込むために，PCI バス制御部に対して再び DMA 要求を出す．この処理は単に転送終了を示すフラグを転送するだけのために DMA を起動するため，一見無駄が多いように見える．しかし，ホストプロセッサが NI 上のフラグをポーリングすると，その度に PCI バスを利用するため，データ転送の効率が落ちる．一方，メインメモリ上のフラグのポーリングではキャッシュにヒットするため，このような損失を避けることができる．

(3) パケット受信:

リモートノードでは，まずパケット受信部が光インターコネクトから受け取ったデータを 32bit の形に変換し，受信 FIFO に入れる．受信 FIFO に少しでもデータが入れば，リモートノードのプリミティブ処理部が起動される．プリミティブ処理部は，ヘッダ中のリモートアドレスを，送信側同様に TLB を参照することにより，物理アドレスに変換して，PCI バス制御部に対して DMA 要求を発生する．PCI バス制御部は，DMA 要求を受け付けると，受信 FIFO 内のデータを主記憶に転送する．

一連の操作は，PCI バス制御部中で PCI バスの転送を効率化のためにデータをバッファする以外は，ほとんどの部分でデータをバッファリングすることなしに，パケット転送を行うことができる．また，プリミティブ処理部の構造は，in-system programming 可能であり，様々な機能のプリミティブを状況に応じて実行することができる．さらに，RHiNET-1/NI は，SDRAM から成る TWIN メモリと呼ばれるメモリを装備している．このメモリを利用することにより，複数のプロセッサが同じリージョンに書き込むことが可能な multiple writer protocol をサポートすることができる．現在，各種プリミティブは実装中の段階で，現在は push プリミティブと pull プリミティブのみが稼働している．

3.2 リンクおよびスイッチ

スイッチ (RHiNET-1/SW) およびノード間を接続するリンクには，Gigabit Ethernet で用いられているインタコネクション（GBIC およびトランシーバとして VITESSE VSC7211），もしくは日立製の光インタコネクション（MDS2211A,MDR2211A）を用いる．これらのリンクは，最大で 133Mbps × 9bit（1Gbps）の伝送容量を持つ．チップの入出力部に 1:2 の multiplexer/demultiplexer を持ち，内部では 18bit のデータ幅のパケットを扱う．

ホップ数を低減して多くのノードを接続するため，スイッチは 8 入力，8 出力とした．

RHiNET-1/SW は 0.35μm CMOS エンベッデッドアレイを用いて構成した 1 チップスイッチで，8 × 8 のクロスバを内蔵している．RHiNET-1/SW は，自由なトポロジと，長距離のパケット転送に対応するために，縮約構造化チャネル法, 拡張 slack buffer, 仮想チャネルキャッシュなどの新しい技術を用いている [6]．さらに，ハンドシェイク用パケットを利用した断線の検出と，活線挿抜も可能である．

表 2 測定条件

バス	周波数 (MHz)
PCI バス	33
プリミティブ制御部	16
ネットワーク	80

さらに，RHiNET-1/SW はトポロジフリーを実現するためにチップ内部のルーティングテーブルの内容に従ってルーティングを行う機能を持つ．このため，電源投入時や活線挿抜によるノードの追加削除に応じてルーティングテーブルを自動的に更新し，必要な場合には管理ノードにトポロジの変更情報を伝える必要がある．これらの動作は複雑ではあるが，特に高速な処理は要求されない．そこで，スイッチチップ外部に日立製 H8 プロセッサを用いたメンテナンスプロセッサボードを接続し，ソフトウエアによりルーティングテーブルの設定を行う．このテーブル設定は，送り手の物理 ID を格納した設定用パケットをやり取りすることによって自動的に行う．

RHiNET-1/SW は富士通製の 0.35μCMOS マクロ埋め込み型セルアレイ CE61 上に実装した．アルミ 3 層配線で，利用可能なエリア数は 120 万エリアであり，最大 506 の I/O ピンを持つことができる．スイッチボードの写真を図 3.2 に示す．

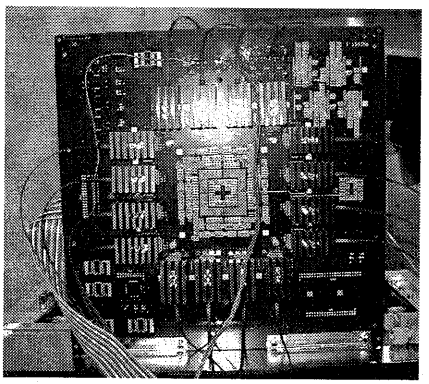

図 6 RHiNET-1/SW のボード

4. 評　　価

RHiNET-1 の性能を評価するため，基本的なプリミティブの処理時間やデータ転送でのバンド幅の測定を行う．測定に用いた条件を表 2 に示す．

プリミティブ制御部は，Altera 社 FLEX

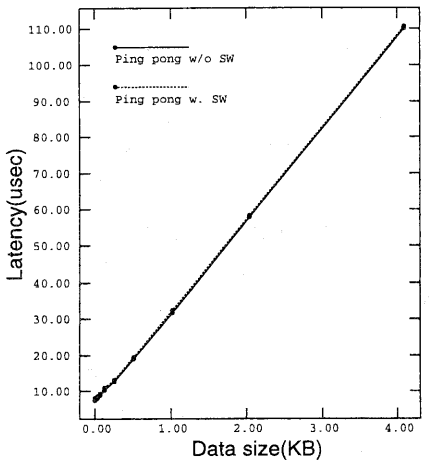

図 7 片方向レイテンシ

10K250ABC600-2(25 万ゲート相当) を用いている．現在，push プリミティブ，pull プリミティブのみを実装した段階でゲート利用率は，約 17%で，内部最大動作周波数は 27MHz である．PCI バス制御部，パケット送受信部との転送遅延の問題で最大動作周波数は現在の段階では 20MHz にとどまっており，今回の測定では，PCI バスの半分の 16MHz の周波数で動作させている．

図 7 に 2 つのノード間で互いに push 操作を行うプログラム (ping-pong) を実行し，片方向のレイテンシを実測した結果を示す．

Push 実行時の各部の処理時間を測定した．計測はロジック・アナライザを用い，プリミティブの起動開始のアクセスが PCI バスに現われた時点を基準点として測定した．なお，ホスト上のプログラムによる測定ではさらに 1.0μs ほど長く観測される．

RHiNET-1/NI の PCI バス，PCI 制御部とプリミティブ制御部間のローカルバス上でタイミングの概要を図 8 に示す．図 8 中に示した番号に対応する処理時間を表 3 に示す．この図，表は 64byte のデータを転送した時の処理内容を示している．

4.1 レイテンシに関する考察

さて，この結果を SAN で接続された専用クラスタと比較して検討する．RWCP の Myrinet を用いた PC クラスタ上での PM で 8 バイトの ping-pong 転送を同様の条件で行った際のレイテンシィは，約 7.5μs である [1]．RHiNET-1 では表 3 に見られるように約 8.1μs を要する．これにソフトウェアの処理にかかる約 1.0μs を加えると，ユーザレベルで見えるレイテンシは約 9.1μs となる．このレイテンシは，RWCP の Myrinet を用いているクラスタに比べて，1.6μs ほど大きな値となっている．RHiNET では SAN と同程度のレイテンシの目指しているが，今回の RHiNET-1 では SAN である Myrinet に比べて大きなレイテンシを持つ．この原因と性能の改善法を検討する．

現在の PCI バス制御部は PCI バス利用率の向上を優先した設計となっている．例えば，PCI バスとローカルバス間のデータ転送では PCI バス制御部内の FIFO が full になるか，データ転送の終了まで送信先側のバスのトランザクションが開始されないように設計されている．そのため，大きなデータでは FIFO が full になるまで反対側のバスのトランザクションを開始しないため，レイテンシが大きくなる．

現在，送信側で PCI バスのデータを取り込み始めてからローカルバスのトランザクションを開始するまで 0.90μs かかっており，また受信側でローカルバスにデータが出始めてから PCI バスの要求を開始するまでに 1.17μs かかっている．送受信側ともに PCI バスとローカルバスのトランザクションが片方のバスが起動されると即座にもう片方のバスが起動されるように設計を変更することにより，現在より 2.07μs のレイテンシ削減が可能である．

4.2 バンド幅に関する考察

次にバンド幅を比較すると，現状の RHiNET-1 では 4K バイトの転送で約 37MB/s の転送バンド幅を得ているが，Myrinet 上での PM では約 119MB/s の転送バンド幅を得られている [1]．RHiNET はネットワークのバンド幅が PCI バスのそれより大きいため，最も効率良くデータ転送を行うと PCI バスのバンド幅が律速となるはずであるが，現在の RHiNET-1 ではそこまでの性能が得られていない．この原因は以下のように考えられる．

測定の結果，データの送信時に PCI バスのトランザクションが 32byte(8word) に 1 回中断されていることが分かった (図 8 中網掛け部)．原因はターゲット側が #STOP をアサートしていることである．この中断は，メインメモリからのリード時に，メモリ・リードと呼ばれる PCI コマンドを用いていることが原因と考えられる．このコマンドはキャッシュライン

表 3 RHiNET-1/NI 処理時間

番号	処理内容		時刻 (μs)
(1)	PCI	プリミティブ書込み開始	0.00
(2)	LB	プリミティブ書き込み開始	0.29
(3)	LB	データ DMA 開始要求	2.04
(4)	NW	ヘッダ出力	2.21
(5)	PCI	データ DMA 開始	2.64
(6)	LB	データ書込み開始	3.54
(7)	NW	データ出力開始	3.96
(8)	LB	フラグ書込み DMA 要求	4.98
(9)	PCI	フラグ書込み DMA	5.94
(10)	LB	DMA 開始要求	6.23
(11)	LB	データ書出し開始	6.97
(12)	PCI	データ DMA 開始	8.14

PCI: PCI バストランザクション
LB: ローカルバストランザクション
NW: ネットワークトランザクション

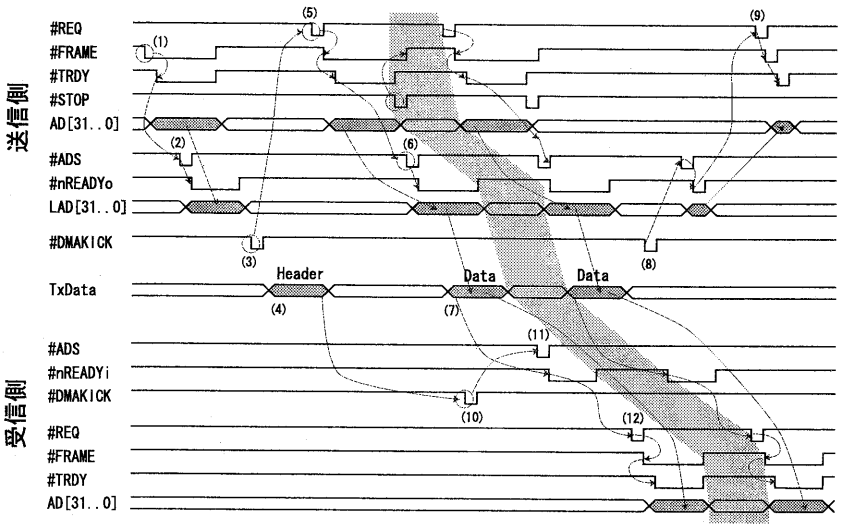

図 8 RHiNET-1/NI 処理概要

サイズ以下のサイズのデータを読む時に利用するコマンドであるため，ターゲット (すなわちチップセット) は違うラインのデータのプリフェッチを行うことをしない．そのため，次のキャッシュラインのデータの準備が間に合わないため，ターゲットによる中断が行われていると考えられる．

このメモリ・リードをキャッシュラインサイズ以上の転送を行うメモリ・リード・マルチプルを用いるよう変更することで，データ転送時の中断を防ぎ，バンド幅を改善することができる．

また，現在，RHiNET-1/NI のプロトコル処理部は，最適化をさほど行っていないため， PCI バスの約半分の 16MHz で動作している．データ転送が連続して行えるようになるとプロトコル処理部の処理が間に合わなくなると考えられる．この動作周波数は最適化により 25MHz 程度には改善できると思われる．

5. おわりに

LASN をサポートする RHiNET の一次試作である RHiNET-1 の実装を行い，初期評価としてデータ転送能力の測定を行った．その結果，現在の実装では Myrinet に代表される SAN と比較してレイテンシ，転送バンド幅ともに劣ることが分かった．

レイテンシに関しては PCI バスコントローラの動作を変更する事で，現在より 2.07μs ほど短くする事は可能であると考えられる．さらに，PCI のコマンドを変更することで連続した DMA 転送が可能となり，転送バンド幅も PCI バスのバンド幅いっぱいまで使用できるようになると考えられる．しかし，その場合にはプロトコル制御部の動作速度を上げる必要があるため，今後，プロトコル制御部の記述を変更し，さらなるパフォーマンスの向上を目指す．これらの改善により SAN 並みのレイテンシ，バンド幅を実現できる可能性は十分あると考えられる．

現在，並列処理に必要な各種プリミティブの実装と，並列アプリケーションの実装を行っている．

参考文献

1) 手塚宏史, 堀敦史, O'Carroll, F., 石川裕: RWC PC Cluster II の構築と性能評価, *HOKKE'98*, 情報処理学会 (1998).
2) Myricom, Inc.: *http://www.myri.com/*.
3) 石川裕: コモディティハードウェアを用いた並列処理技術, 情報処理, Vol. 39, No. 8, pp. 784–791 (1998).
4) 工藤知宏, 山本淳二, 建部修見, 佐藤三久, 西宏章, 天野英晴, 石川裕: PC 間ネットワークによる共有アドレス空間を持つ並列処理システム, 情報処理学会研究報告, Hokke '99, No. 99-ARC-132-21,99-OS-80-21,99-HPC-75-21, pp. 121–126 (1999).
5) Kudoh, T., Nishimura, S., Yamamoto, J., Nishi, H., Tatebe, O. and Amano, H.: RHiNET: A network for high performance parallel processing using locally distributed computers, *IWIA 99* (1999).
6) 西宏章, 多昌廣治, 工藤知宏, 天野英晴: 仮想チャネルキャッシュを持つネットワークルータの構成と性能, 並列処理シンポジウム JSPP'99, Vol. 99, No. 6, pp. 71–78 (1999).