2004－ARC－159　(26)
2004／7／31

DIMMnet-2 ネットワークインタフェースボードの試作

北村　聡[†]　伊豆直之[†]　田邊　昇[††]
濱田芳博[†††]　中條拓伯[†††]　渡邊幸之介[†]
大塚智宏[†]　天野英晴[†]

近年, PC クラスタが企業・大学をはじめとする様々な研究機関で用いられている. これは PC の価格対性能比の著しい向上から, 比較的安価に高性能なシステムを構築できるためである. このようなシステムには Myrinet, QsNET, InfiniBand のような高速な専用インターコネクト, もしくは Gigabit Ethernet などが用いられるが, インターコネクトの性能の向上につれて, これらのネットワークインタフェースの接続に用いられてきた標準的なバスである PCI バスの性能がシステムの性能のボトルネックになりつつある.

そこで, 我々はこの問題を解決するべく, DIMMnet-2 システムの開発を行っている. 今回, FPGA を用いた DIMMnet-2 ネットワークインタフェースの試作ボードが完成したため, このボード, 及びネットワークコントローラに搭載する予定であるハードワイヤード TLB リフィル機構の実装, 評価について報告する.

A Prototype of DIMMnet-2 Network Interface Board

Akira Kitamura,[†] Naoyuki Izu,[†] Noboru Tanabe,[††] Yoshihiro Hamada,[†††]
Hironori Nakajo,[†††] Konosuke Watanabe,[†] Tomohiro Otsuka[†]
and Hideharu Amano [†]

PC clusters have been popularly used because of their high degree of performance per cost based on remarkable performance improvement of common PCs. In such systems, performance of the PCI bus, which is generally used for connecting network interface, dose not balance with recent high speed interconnections: Myrinet, QsNET, InfiniBand and Gigabit Ethernet.

DIMMnet-2, that uses memory slot of common PCs for connecting a network interface, has been developed to address this problem. In this report, a prototype board of DIMMnet-2 network interface is introduced and its fundamental performance is evaluated.

1. はじめに

Personal Computer (PC) の価格対性能比の著しい向上により, これらをノードとして用いた PC クラスタが実用的な計算資源として企業・大学などで広く用いられるようになった. PC クラスタの多くは Myrinet[1] や QsNET[2], InfiniBand[3] などの高速なインターコネクトによりノード PC を相互接続することで高い性能を実現する. 従来, これらインターコネクトのネットワークインタフェースは PC における標準的な入出力バスである PCI バスに接続されてきた. しかし, 近年のインターコネクトの性能の向上に従い, PCI バスの性能がシステム性能のボトルネックになりつつある.

最近になって, PCI-X などの高速入出力バスが規格化され, それらに対応したインターコネクトのネットワークインタフェースが開発, 販売され始めてはいるが, 主としてサーバ用途であるため, 高価である. また, これら高速入出力バスでは, バンド幅は大幅に向上しているものの, バスのレイテンシについてはほとんど改善されていないという問題点がある. そこで, 我々はこれらの問題を解決するべくメモリスロットに着目し, メモリスロット装着型のネットワークインタフェースを用いた PC クラスタ向けインターコネクト DIMMnet の研究, 開発を行っている.

本稿では, 現在研究, 開発を行っているメモリスロット装着型ネットワークインタフェースである DIMMnet-2[5] ネットワークインタフェース (DIMMnet-2/NI) の試作ボードの構成について述べ, ネットワークコントローラの機能とその性能について報告する.

以下, 第 2 章で DIMMnet-2 の概要, 第 3 章で DIMMnet-2/NI の試作ボードについて触れた後, 第 4 章でネットワークコントローラ部について解説し, 第 5 章でその予備評価を示す. 最後に第 6 章でまとめる.

2. DIMMnet-2

DIMMnet はネットワークインタフェースを PC の DIMM

† 慶應義塾大学
Keio University
†† (株) 東芝, 研究開発センター
Corporate Research and Development Center,Toshiba
††† 東京農工大学
Tokyo University of Agriculture and Technology

スロットに装着する PC クラスタ向けインターコネクトである. DIMMnet の最大の特徴はメモリバスの利用にある. これまでネットワークインタフェースの多くは, Local Area Network (LAN), System Area Network (SAN) を問わず, PCI バスに装着されてきた. しかしながら, PCI バスのクロック周波数はホスト CPU や FSB のクロックに比べて大幅に低いため, ホスト CPU からのアクセスレイテンシが大きくなってしまう. 例えば, ホスト CPU からネットワークインタフェースをポーリングするには μs オーダーのレイテンシを要することになる.

PCI バスより高い性能を持つ次世代 I/O 規格として PCI-Express[4] が存在するが, x16, x32 のような高速な規格は高価なサーバやワークステーションのみへの搭載が予定されており, PC クラスタの全ノードに装備するにはコストが大きい. また, x1, x2 など, PCI-Express の中でも低速な規格では PCI, PCI-X 世代からの著しい性能向上は見込めないものと思われる.

これに対して, メモリバスは PCI バスよりもアクセスレイテンシが小さく, また, バスのバンド幅は近年のメモリクロックの著しい向上から, PCI-Express の x16, x32 に匹敵する性能に達している. さらに, メモリスロットはどのような PC にも搭載されていることから, ここにネットワークインタフェースを装着した場合, 安価な PC をノードとして用いることができる. DIMMnet では, このようなメモリスロットを介してネットワークインタフェースを接続することにより, 高性能 I/O バスを用いたインターコネクトに比べ, 低コストで高性能なシステムを構築することを目標としている.

DIMMnet-2 は, 東京農工大学, 及び新情報処理開発機構によって開発された DIMMnet-1[7] の実装, 及び評価の結果に基づいて開発が行われている. 独自設計のネットワークスイッチを利用した DIMMnet-1 とは異なり, DIMMnet-2 ではネットワークスイッチに商用のスイッチを用いることで, 高性能なシステムを安価に構築することを目指す.

DIMMnet-2/NI と DIMMnet-1/NI の相違点を以下に示す.

(1) DIMMnet-1/NI はホストとの接続に SDR-SDRAM 用のメモリスロットを用いているのに対して, DIMMnet-2/NI では, PC で用いられるメモリの発展に合わせて, DDR-SDRAM に対応する.

(2) DIMMnet-1/NI では, ホストからネットワークインタフェース上の SDR-SDRAM を直接アクセスしたが, DDR-SDRAM は転送が高速であるためにこれが困難となる. そこで, Window と呼ばれるバッファを経由してメッセージ転送やネットワークインタフェース上のメモリの読み書きを行う. このことにより, ローカルとリモート両方のネットワークインタフェース上のメモリへのアクセスに統一された手法を提供することが可能となる.

(3) さらに, Window とネットワークインタフェース上のメモリ間の転送機能を拡張し, プリフェッチ機構[6] を付加することによって CPU のキャッシュやメモリバスの利用効率を向上させることで, PC 単体で用いた際にも, ネットワークインタフェース上のメモリを有効利用し, アプリケーションの実行性能を向上させる機能を持つ.

(4) インターコネクト, 及びネットワークスイッチには商用の InfiniBand スイッチを用いる.

本稿ではこの DIMMnet-2/NI の試作ボード, 及びその CoreLogic の設計, 実装に関して述べる.

3. DIMMnet-2/NI 試作ボード

DIMMnet-2/NI の試作ボードでは, ボード上の FPGA にネットワークコントローラを搭載する. メモリスロット装着型ネットワークインタフェースを用いて安価に PC クラスタを構築するためにはネットワークインタフェースの ASIC 化が必要であり, このボードはそのための機能検証, 論理検証を目的とする. FPGA 上には高速インターコネクトの IP が実装されており, InfiniBand への接続が容易に実現可能である.

また, メモリスロットとのインタフェース, 及び高速ネットワークインタフェースを持つボードにより, メッセージ転送に関連する処理を FPGA 上で実現すると共に, 演算処理の一部も FPGA 上で実行できる. このようなボードはホスト PC の処理の一部を高速化する Reconfigurable System として魅力的である. 過去に SDR-SDRAM に対しては実現例がある[12] が, DDR-SDRAM に対して装着可能で, かつネットワークインタフェースを持つボードは存在せず, バイオインフォマティクス[13] をはじめとして数多くの応用分野が期待される.

図 1 に試作ボードの構成図を示す.

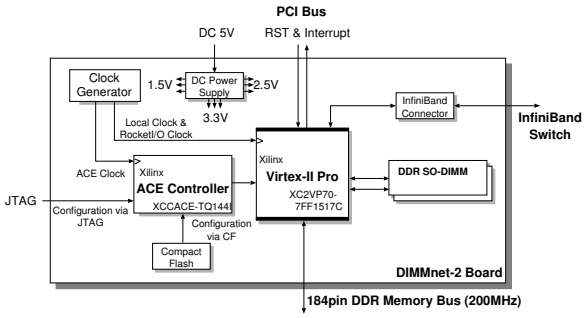

図 1 DIMMnet-2/NI 試作基板の構成図

試作ボードは以下の要素から構成される.

- Xilinx VirtexII Pro XC2VP70-7FF1517C
- Xilinx System ACE Controller
- DDR SO-DIMM × 2
- InfiniBand Connector
- DDR Host Interface

ネットワークインタフェースのコアとなるコントローラ部は Xilinx 社の VirtexII Pro XC2VP70-7FF1517C 上に

実装される. このチップは 8,272 個の Configuration Logic Block (CLB) と最大 5,904kbit の内部 RAM, 996 本のユーザ I/O ピンを持っており, さらに PowerPC を 2 個, RocketIO トランシーバを 16 個備えた, 大規模かつハイエンドな FPGA である. この RocketIO を用いて InfiniBand スイッチとの接続を行う.

試作ボードは, PC のメモリバスに装着されるため, リセット時に BIOS によりアクセスされるが, その際に正常に PC を起動させるには, FPGA が実際のメモリのふりをして BIOS のアクセスに応答しなければならない. そのため, リセット直後の段階で FPGA がコンフィギュレーションされている必要が生じ, それにはコンフィギュレーションが不揮発であることが望ましい. しかし, VirtexII Pro などの大規模な FPGA には大容量のコンフィギュレーションメモリが必要となるため, PROM をコンフィギュレーションメモリに用いると部品点数が増加してしまう. そこで, Xilinx 社より提供されている, 大規模 FPGA のコンフィギュレーションソリューションである System ACE Compact Flash を採用した. この方式では Compact Flash の装着により, 電源投入と同時のコンフィギュレーションが可能となる.

また, ネットワークインタフェース上にはノート PC 用の汎用メモリである 200pin DDR SO-DIMM を 2 枚搭載する. これは通信用のバッファに使用するほか, ホスト PC のデータ記憶領域として使用する. 現在, 256MB の SO-DIMM を 2 枚, 計 512MB 分のメモリを搭載している. 将来的にはより大容量のメモリを搭載し, DIMMnet-2 クラスタシステム全体で大規模な共有メモリシステムを構築することを目指す.

図 2 に完成した試作ボードの概観を示す. このボードをホスト PC の DDR メモリスロットに装着する.

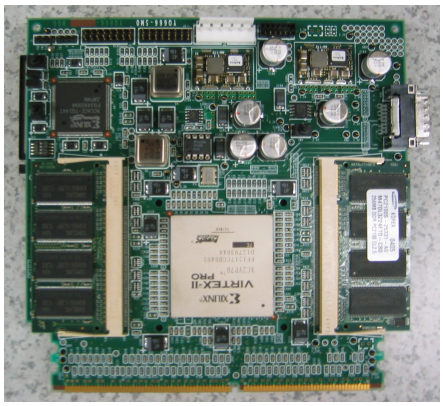

図 2 DIMMnet-2NI 試作ボード

なお, 今回は FPGA を使用した試作ボードであり, 高い動作周波数での稼働が困難であるため, DDR200 (PC-1600) での動作を前提としている.

本稿執筆時点では, このボードに対してホスト CPU からのアクセスが可能であることが確認されている.

4. DIMMnet-2/NI ネットワークコントローラ

以下では, FPGA 上に実装されるネットワークコントローラ部について述べる.

4.1 コントローラ部の構造

図 3 に DIMMnet-2/NI コントローラ部のブロック図を示す.

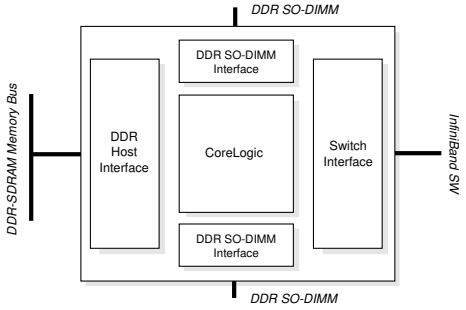

図 3 コントローラ部のブロック図

コントローラ部は大きく以下の 4 つのブロックから構成される.

- DDR Host Interface 部：ホスト CPU とのトランザクションを処理するブロックである. ホスト CPU との転送は, 高速 (100MHz) なクロックの両エッジに対して行う必要がある. 特にホスト CPU がデータ受信を行う際は, 一定以内の遅延時間でデータを送出する必要がある. このため, データ交換は Window と呼ばれる小規模なバッファ, 及びレジスタを介して行う. 両エッジで転送される 64bit データは Interface 内で 128bit の片エッジ転送に変換される.
- DDR SO-DIMM Interface 部：基板上の DDR SO-DIMM を制御するブロックである. DDR Host Interface 部同様, 64bit 幅, 100MHz 両エッジ転送でアクセスを行い, インタフェース内で 128bit 幅, 100MHz 片エッジ転送に変換する.
- Switch Interface 部：InfiniBand スイッチとのインターフェースとなるブロックであり, 内蔵の RocketIO を用いて, ネットワークとのパケットの送受を行う. End-to-End の再送機構を持つ[9].
- CoreLogic 部：ネットワークインタフェースの心臓部に相当し, 送信パケットの生成や受信パケットの解析, プリフェッチ処理など, データの送受信の要求を処理する.

本稿では CoreLogic 部について重点的に紹介する.

4.1.1 CoreLogic 部の構成

図 4 に CoreLigic 部の構成を示す.

DIMMnet-2/NI の CoreLogic は以下のモジュールで構成される.

- Write Window：パケット送信用書き込みバッファ
- Prefetch Window：ブロックデータのプリフェッチバッ

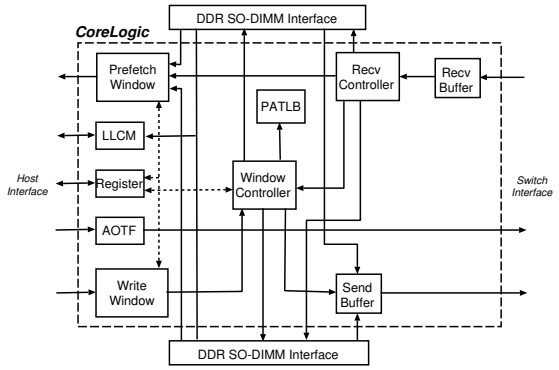

図 4 DIMMnet-2/NI CoreLogic 部の構成

ファ

- LLCM(Low Latency Common Memory)：ホストから物理メモリとして直接読み書き可能な共有メモリ
- Register：コマンドレジスタ, 各種ステータスレジスタ
- AOTF：AOTF[8] パケットの送信部
- Window Controller：送信パケットのヘッダ生成等の送信処理, 及びプリフェッチ処理用コントローラ
- PATLB：仮想アドレスと SO-DIMM の物理アドレス間の変換 TLB
- Recv Controller：受信したパケットの処理用コントローラ

DIMMnet-1/NI では FET スイッチを用いて, ボード上の SO-DIMM を接続するバスを電気的に切り替えることで, ホストとネットワークコントローラの双方から, ボード上の SO-DIMM にアクセスすることが可能であった. しかし, DDR は高速両エッジ転送を行うため, 電気的な特性からこのような切り替えは不可能である. そこで, DIMMnet-2/NI ではホストからボード上の SO-DIMM に直接アクセスせず, Write Window, Prefetch Window, Register を介して, ネットワークインタフェースに対するコマンドの発行やデータの読み書きを行う.

この手法を用いることにより, ホストに接続しているネットワークインタフェースの SO-DIMM への読み書きと, リモートホストのネットワークインタフェース上の SO-DIMM への読み書きを統一的な方法で実現し, この操作で基本転送コマンドを構築することができる. さらに, ストライド転送等インテリジェンスの高い転送をネットワークコントローラが行うことで, キャッシュのヒット率を改善することができる[6].

ホスト PC と DIMMnet-2/NI とのデータ転送は, 基本的には高速小容量のメモリである Window を介して行う. Window は, ユーザプロセスのメモリ空間にマップされ, Write Window はホストからの書き込み用, Prefetch Window は読み出し用として使用される. 転送は以下のように行われる.

- SO-DIMM への書き込み：ホスト PC は, Write Window にデータを書き込み, Register に転送サイズ, 書き込みアドレス等を与えて Window Controller を起動する. Window Controller はアドレス変換を行い, 書き込みアドレスがローカルのボード上の SO-DIMM ならば, SO-DIMM Interface を介して書き込みを行う. リモートホストのネットワークインタフェース上の SO-DIMM ならば, Send Buffer 上にパケットを構築して Switch Interface に渡す.
- SO-DIMM の読み出し：ホスト PC は Register を用いてアドレス, サイズ等を指定してプリフェッチ要求を出す. アドレスに応じて Window Controller がローカルの SO-DIMM を読み出すか, リモートホストのネットワークインタフェース上の SO-DIMM へのアクセス要求を出す. ローカルから読み出したデータは Prefetch Window に格納され, リモートホストからネットワーク経由で転送されてきたデータは Recv Buffer に格納された後, Recv Controller によって Prefetch Window に格納される. ホスト PC は所定の遅延時間後, あるいはフラグセンスによってデータ到着を確認した後に Prefetch Window から読み出しを行う.
- SO-DIMM 間転送：ホスト PC は Register を用いてアドレス, サイズ等を指定して転送要求を出す. アドレスと転送方向に応じて, Window Controller がパケットを生成し, ローカルのネットワークインタフェース上の SO-DIMM とリモートホストのネットワークインタフェース上の SO-DIMM 間でデータ転送を行う.
- その他：コマンドに応じて Window Controller は, ホストによるキャッシュアクセスを高速化するために, ストライドアクセス等の高機能なアクセスを行う[6].

これらの方法では, データ転送は SO-DIMM と Window 間, あるいは SO-DIMM 間で行われる. 一方, ホスト間で直接データをやり取りするための小容量なメモリとして DIMMnet-1/NI 同様, LLCM を用いる. LLCM は小容量であるが, 書き込んだデータを直接, リモートホストの LLCM に転送することが可能である. LLCM は, パケットヘッダをあらかじめネットワークインタフェースが保持することでパケットデータの書き込み時間を低減する AOTF[8] と組み合わせて用いることもできる.

さらに, DIMMnet-2/NI は TLB のミスヒット時にソフトウェアではなく, ハードウェアで内容をリフィルする機構を持つ. この機構により, ミスヒット時のオーバヘッドを削減することが可能である.

5. 評　　価

本章では DIMMnet-2/NI の CoreLogic の基本性能として, ボード上の SO-DIMM からリモートホストへのデータ転送を行う際に要するクロック数, 及び新たに実装されたハードワイヤード TLB リフィル機構のシミュレーションによる評価を示し, DIMMnet-1/NI のネットワークコントローラである Martini[10] と比較する.

5.1 転 送 処 理

DIMMnet-2/NI の CoreLogic での転送処理の流れとそ

れに要するクロック数を以下に示す. なお, 転送サイズは 32Byte (8Byte × バースト長 (=4)) とする.

(1) ホストから Register, Write Window にデータ転送コマンドを発行 (2 クロック)
(2) Write Window が Window Controller を起動 (2 クロック)
(3) Window Controller が PATLB を参照して転送データの SO-DIMM 上の物理アドレスを取得 (7 クロック)
(4) Window Controller から SO-DIMM Interface へデータの読み出し要求を発行すると同時にパケットヘッダを生成し Send Buffer に転送 (8 クロック)
(5) SO-DIMM から読み出されたデータを Send Buffer の所定の位置に書き込み, パケットを生成 (5 クロック)
(6) Switch Interface にパケット転送を開始 (4 クロック)

ホストからコマンドが発行されてから CoreLogic でパケットを生成するまでに計 28 クロック要する.

一方, DIMMnet-1/NI で用いた Martini も同様な処理を行うが, SO-DIMM からのデータ転送には DMA を用いているため, DMA コントローラの起動に時間を要する. このため, (4) に 11 クロック, (5) に 25 クロック要し, 計 51 クロックとなる. このことから, DIMMnet-2/NI ではデータ転送に要するクロック数が DIMMnet-1/NI の約 55%になっており, データ転送時のレイテンシを抑えられると推測される.

5.2 ハードワイヤード TLB リフィル

第 3 で述べた通り, DIMMnet-2/NI の試作ボードは動作確認の段階であるため, 本稿ではハードワイヤード TLB リフィル機構を DIMMnet-2/NI 向けに改良を行った Martini を用いたシミュレーションによって評価した.

ハードワイヤード TLB リフィルとは, TLB にアクセスした際にミスヒットが発生した場合, 例外を発してソフトウェアに処理をゆだねずに, ハードウェアがマスタデータを参照して, 値を TLB にセットする機構である.

今回評価を行ったハードワイヤード TLB リフィルは PATLB を対象としたものである. PATLB の構成は, 72bit 幅 ×1024 エントリのオンチップ SRAM を 2 つ利用した 2-way 構成の TLB となっている.

本稿で実装したハードワイヤード TLB リフィル機構 (= ハードウェアリフィル) を, Maritni に搭載されているオンチッププロセッサを用いて実装されたソフトウェアによるリフィルと比較を行う.

5.3 シミュレーション環境

性能評価には Martini の開発用に作成されたシミュレーション環境[11] を用いた. このシミュレーション環境は Verilog-PLI によって Verilog シミュレータを拡張したものであり, C++で記述されたホストプログラムと通信することが可能である. そのため, 実際にホスト上で動作するプログラムを組み合わせたシミュレーションが可能である.

5.4 ソフトウェアリフィル

オンチッププロセッサを用いた実装では, LLCM を用いてマスタデータの一部をキャッシュしている. PATLB を参照してミスヒットが発生した場合は, まず LLCM にエントリが存在するかを確認する. 有効なデータがなければ, SO-DIMM に DMA でアクセスし, 当該データを含んだ複数エントリを一度に読み出し, LLCM にキャッシュしておく. そのため, リフィルの動作としては, LLCM 上のキャッシュデータからの読み出しで済む場合と, SO-DIMM へのアクセスを伴う場合との 2 通りがある. 基本的な手順は同様であるが, LLCM からリフィルする場合は DMA アクセスに関する処理が行われない.

表 1 にソフトウェアリフィルに要するクロック数を示す. この表の値はマスタデータから DMA で読み出すデータのサイズが 128Byte (32 エントリ) の場合のリフィルに要するクロック数を示している.

表 1 ソフトウェアによるリフィル処理の内訳

処理内容	SO-DIMM からリフィル [clk]	LLCM からリフィル [clk]
割込み信号発生	0	0
割込みハンドラ開始	+ 3	+ 3
内部レジスタ読み出し	+ 64	+ 64
LLCM 読み出し	+ 268	+ 268
DMA 要求準備開始	+ 131	–
DMA 要求受理	+ 239	–
DMA 完了検出	+ 165	–
PATLB 登録開始	+ 261	+ 154
状態復帰	+ 285	+ 285
合計	1416	774

5.5 ハードウェアリフィル

表 2 にハードウェアリフィルの手順と, 各操作に要するクロック数を示す. ハードウェアリフィルでは, ソフトウェアリフィルに比べ, DMA 処理, リフィルデータ読み書きなど各処理に要するクロック数が大きく削減されている.

表 2 ハードウェアによるリフィル処理の内訳

処理内容	処理時間 [clk]
リフィル要求発生	0
リフィル処理開始	+ 1
物理アドレス獲得	+ 1
DMA FIFO 要求	+ 1
DMA 要求発行	+ 1
リフィルデータ読み出し	+ 22
PATLB 要求	+ 1
PATLB 獲得	+ 1
リフィルデータ書き込み	+ 1
状態復帰	+ 1
合計	30

表 1 と表 2 から, ハードウェアリフィルはソフトウェアリフィルで SO-DIMM からリフィルを行った場合の 47.2 倍, LLCM からリフィルを行った場合の 25.8 倍の性能向

上を達成していることが分かる.

5.5.1 リフィル機構に要するハードウェアコスト

リフィル機構の実装に必要としたハードウェアコストを Xilinx 社の提供する合成ツール (ISE6.1i) を用いて測定すると, 表 3 に示されるように, 400 スライス程度の増加であり全体の 5%程度であることから, 低いハードウェアコストで高い性能が得られたと言える.

なお, この値は送受信制御のモジュールである Window Controller, Recv Controller, PATLB の 3 モジュールに要するスライス数である.

表 3 リフィル機構の有無によるハードウェアコストの違い

	実装前	実装後
スライス数	7703	8113

合成ツール: Xilinx 社 ISE 6.1i
デバイス: XC2VP70-7FF1517C

5.6 送受信制御部のハードウェアコスト

従来の DIMMnet-1/NI で用いられた Martini 内の送受信制御部 (従来版) と DIMMnet-2/NI 向けに改良を行った送受信制御部 (改良版, ハードワイヤード TLB リフィル機構を含む) の論理合成の結果を表 4 に示す.

表 4 送受信制御部のハードウェア量の比較

	従来	改良後
スライス数	13211 (39%)	8113 (24%)
内部 RAM	64 (19%)	56 (17%)
(*) 動作周波数	68.709 MHz	121.589 MHz

合成ツール: Xilinx 社 ISE 6.1i
デバイス: XC2VP70-7FF1517C

表 4 の括弧内の数値は FPGA のハードウェアリソースに対する, 合成を行った回路の占める割合を示している.

送受信制御部のみでコントローラを構成するわけではないため, コントローラ全体では動作周波数は表に示した見積り値に比べ, 幾分落ちると予測されるが, 現段階では高い動作周波数を実現できることが示されている.

また, ネットワークコントローラの将来の ASIC 化を想定し, ASIC に実装した際にどの程度の動作周波数が期待できるかを見積もる. 日立の 0.18μm プロセスを用いてコントローラ内の送受信制御部を合成した結果を表 5 に示す. 合成には. Synopsys 社の Design Analyzer を用いた.

表 5 送受信制御部の合成結果

	面積	動作周波数
改良版	155641	159.2 MHz

プロセスルールにおける面積の単位は明らかにされていないため, ゲート数などの具体的なコストが明確でないものの, ASIC の実装時には, より先進的なプロセスルールが使用されると考えられるため, 送受信制御部のみであれば, 150MHz 以上での動作が期待される.

6. ま と め

本報告では PC クラスタ向けインターコネクトである DIMMnet-2 のネットワークインタフェースの試作ボードについて述べ, ネットワークインタフェースのコントローラに搭載する予定であるハードワイヤード TLB リフィル機構の実装, シミュレーションによる評価を行った. 評価の結果, ハードウェアリフィルはソフトウェアリフィルで SO-DIMM からリフィルした場合の 47.2 倍, LLCM からリフィルした場合の 25.8 倍の性能を示した. また, このリフィル機構を低いハードウェアコストで実装可能であることが確認された.

謝辞 本研究は総務省戦略的情報通信研究開発推進制度の一環として行われたものである. DIMMnet-2 の開発に関する議論にご参加頂いている (株) 日立 IT の今城氏, 岩田氏, 上嶋氏, 慶應義塾大学の西助手に感謝致します.

参 考 文 献

1) Myricom,Inc., http://www.myri.com/
2) Quadrics.Ltd., http://www.quadrics.com/
3) InfiniBand Trade Association, http://www.infinibandta.org/
4) PCI-SGI, http://www.pcisig.com/
5) 田邊 昇, 濱田 芳博, 三橋 彰浩, 中條 拓伯, 天野 英晴, メモリスロット装着型ネットワークインタフェース DIMMnet-2 の構想, 情報処理学会アーキテクチャ研究会, 2003-ARC-152, Mar.2003.
6) 田邊 昇, 土肥 康孝, 中條 拓伯, 天野 英晴, プリフェッチ機能を有するメモリモジュール, 情報処理学会アーキテクチャ研究会, 2003-ARC-154, Aug.2003.
7) 田邊 昇, 山本 淳二, 工藤 知宏, メモリスロット搭載型ネットワークインタフェース DIMMnet-1 における細粒度通信機構, 情報処理学会アーキテクチャ研究会, 2000-ARC-137, Mar.2000.
8) 田邊 昇, 濱田 芳博, 山本 淳二, 今城 英樹, 中條 拓伯, 工藤 知宏, 天野 英晴, DIMM スロット搭載型ネットワークインタフェース DIMMnet-1 とその低遅延通信機構 AOTF, 情報処理学会論文誌ハイパフォーマンスコンピューティングシステム, Vol.44 No.SIG01, Jan.2003.
9) 濱田 芳博, 西 宏章, 田邊 昇, 天野 英晴, 中條 拓伯, bDais:DIMMnet-1/InfiniBand 間ルータの開発, 先進的計算基盤システムシンポジウム SACSIS2004, May.2004
10) 山本 淳二, 渡邊 幸之介, 土屋 潤一郎, 原田 浩, 今城 英樹, 寺川 博昭, 西 宏章, 田邊 昇, 上嶋 利明, 工藤 知宏, 天野 英晴, 高性能計算をサポートするネットワークインタフェース用コントローラチップ Martini, 情報処理学会論文誌ハイパフォーマンスコンピューティングシステム, Vol.43 No.SIG06, Nov.2002.
11) 山本 淳二, 渡邊 幸之介, 宮脇 達朗, 西 宏章, 工藤 知宏, 天野 英晴, PLI を用いたネットワークインタフェースコントローラとホストプログラムの協調シミュレーション, 情報処理学会アーキテクチャ研究会, 2001-ARC-145, Nov.2001.
12) Plessl, C., Platzner, M., TKDM - a reconfigurable co-processor in a PC's memory slot, Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on , Dec.2003,
13) Yasunori Osana, Tomonori Fukushima, Hideharu Amano, Implementation of ReCSiP: A ReConfigurable Cell SImulation Platform, 13th International Conference on Field Programmable Logic and Applications(FPL), 2003. Proceedings., Sep,2003.