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

山本淳二[†1,☆] 渡邊幸之介[†2] 土屋潤一郎[†2]
原田浩[†1,☆☆] 今城英樹[†3] 寺川博昭[†3]
西宏章[†1,☆] 田邊昇[†4] 上嶋利明[†3]
工藤知宏[†1,☆☆☆] 天野英晴[†2]

ユーザレベル・ゼロコピー通信を用いることで低レイテンシ・広バンド幅な通信を提供するネットワーク RHiNET のネットワークインタフェースコントローラ Martini について述べる．Martini では基本の RDMA 機能をすべてハードワイヤード処理することで，低レイテンシ・広バンド幅の RHiNET ネットワークを最大限利用する．その一方で内部にプロセッサを持つことで複雑な処理についても柔軟に対処できる構成となっている．

Martini: A Network Interface Controller Chip for High Performance Computing

Junji Yamamoto,[†1,☆] Kounosuke Watanabe,[†2]
Jun'ichiro Tsuchiya,[†2] Hiroshi Harada,[†1,☆☆] Hideki Imashiro,[†3]
Hiroaki Terakawa,[†3] Hiroaki Nishi,[†1,☆] Noboru Tanabe,[†4]
Toshiaki Uejima,[†3] Tomohiro Kudoh[†1,☆☆☆]
and Hideharu Amano[†2]

In this paper, a network interface controller for RHiNET, which is called Martini is described. Martini supports user level zero-processor-copy communication and makes good use of low latency, high bandwidth RHiNET network. While Martini processes fundamental RDMA and PIO-based transfer functions by hard-wired logic, it is provided with an on-core processor to support complex functions flexibly.

1. はじめに

近年のパーソナルコンピュータ（PC）の性能向上はめざましい．また PC をノードとしたクラスタシステムを用いた並列処理も広く行われるようになっている．このようなクラスタではノード間を Myrinet [1)] のようなシステムエリアネットワーク（SAN）で結合することが多い．

SAN はパケットを途中で破棄せず，送信順に到着するネットワークであり，低遅延・広バンド幅の通信を提供する．その一方で最大リンク長やネットワークトポロジに対する制限が厳しいため，クラスタなど 1 カ所にまとまって設置されたシステムに用いられてきた．

一方，フロア内やビル内などある程度広範囲に分散している計算機間はイーサネットに代表されるローカルエリアネットワーク（LAN）で接続されている．一般に LAN は敷設可能な距離が 100 m から 1 km 程度と長く，またとりうるネットワークのトポロジの制限も緩い．しかし，パケットの破棄・重複を認めている

†1 新情報処理開発機構
Real World Computing Partnership
†2 慶應義塾大学
Keio University
†3 日立インフォメーションテクノロジー
Hitachi Information Technology
†4 株式会社東芝
Toshiba Co., Ltd.
☆ 現在，株式会社日立製作所中央研究所
Presently with Central Research Laboratory, Hitachi, Ltd.
☆☆ 現在，コンパックコンピュータ
Presently with Compaq
☆☆☆ 現在，産業技術総合研究所
Presently with National Institute of Advanced Industrial Science and Technology (AIST)

ため，TCP など上位層によるパケットの順序制御・再送制御が必要である．

現在は高性能な並列分散処理は SAN で接続されたクラスタで行われているが，SAN に匹敵する性能を持ち，LAN と同様に広範囲に分散した計算機を接続できるネットワークがあれば，通常業務に使用している計算機の余剰性能を利用したり，複数のクラスタ群を接続したりすることでさらなる高性能な計算機資源を作り出すことが可能である．

そこで我々はこのような LAN と SAN の双方の利点を持つ，新しいネットワーククラスであるローカルエリアシステムネットワーク（LASN）を提案している[2]．

LASN は LAN と同様にネットワークのトポロジを比較的自由に設定でき，また接続距離も 100 m から 1 km まで延ばせると同時に SAN と同じくパケットの破棄は行わず，送信順に届くという長所を持つネットワーククラスである．

LASN の 1 つの実装として RHiNET[3),4)]がある．本稿ではこの RHiNET の概要と，RHiNET で用いるネットワークインタフェースの核となるネットワークプロセッサ Martini[5)~8)]の設計と実装について述べる．

2. RHiNET ネットワークと Martini

RHiNET は自由なトポロジを認め，パケットを破棄することなくデッドロックフリーを保証するネットワークであり，従来の SAN 同様の低レイテンシでパケットを破棄しないという性質を持ちながら，ビル内やフロア内に分散して配置された計算機群をつなぐことができる．リンク長は最大 1 km で，多数の仮想チャネルを設けることにより，自由なトポロジを用いることができる．

Martini は多くの計算機を RHiNET ネットワークで接続して高性能並列処理環境を実現することを目的として設計されたネットワークインタフェースコントローラチップであり，PCI バスに装着される RHiNET ネットワークインタフェースおよび DIMM メモリスロットに装着される DIMMnet-1 ネットワークインタフェースで用いられる．

Martini は，RHiNET と DIMMnet-1 の両方のネットワークインタフェースで用いるために，

(1) 64 bit/66 MHz PCI バス準拠
(2) PC-133 DIMM メモリ相当動作

の 2 種類のホストインタフェースを持つ必要がある．

また，RHiNET および DIMMnet のネットワークインタフェースは，RHiNET プロジェクトで開発された 3 種類のスイッチ RHiNET-2/SW，RHiNET-3/SW，OIP-SW に接続される．

RHiNET-2 システムで用いている光インタコネクションは，エラーレートが非常に低い（BER: Bit Error Rate $= 10^{-20}$）ため，ECC の付加により実質的にはエラーフリーの転送を実現できる．この光インタコネクションモジュールは DC レベルの伝送もできるためレイテンシを小さくするのにも有利である．また，並列伝送されるチャネル間のスキューが問題にならない程度に小さいため，使用する際にスキューを考慮する必要がない．しかし，このような光インタコネクションはどうしても構造上高価なものとなる．

RHiNET-3 システムは，より一般的な，エラーレートが高く（BER $= 10^{-12}$ 程度）DC レベルの伝送ができないモジュールを用いるための実験機で，論理回路によるスキュー調整，8B10B エンコーディング，エラー発生時の再送などを行う．Martini はこれらのうち再送機構のみを備える．スキュー調整，エンコーディング機能は Deskew チップと呼ぶ専用に開発した LSI で行う[9)]．

OIP-SW は，新情報処理開発機構の光インタコネクション NEC 研究室が開発した光モジュールと論理 LSI を同一パッケージに実装する OIP[10)]技術を用いた実験機であり，Martini はこのスイッチの機能を検証するために OIP-SW のインタフェースを備える．

なお，RHiNET-3，OIP-SW は新しい機能を多く盛り込んだ開発中の実験システムであるため，本稿で示す Martini の評価には RHiNET-2 を用いている．

これらのスイッチは以下のように仕様が異なるので，Martini のスイッチインタフェース部はこれに対応しなくてはならない．

(1) RHiNET-2/SW
- 800 Mbps × 10 data channel，1 framing channel，1 clock channel を用いる
- Go-and-stop flow control によるパケット単位のフロー制御
- ECC によるエラー検出，回復

(2) RHiNET-3/SW
- 1.25 Gbps × 8 data channel，1 framing channel，1 clock channel を用いる
- Credit 方式によるライン（80 bit）単位フロー制御
- CRC によるエラー検出とハードウェアによる hop-by-hop 再送

(3) OIP-SW

- 250 Mbps × 9 data channel，1 framing channel，1 clock channel を用いる
- Go-and-stop flow control によるライン（32 bit）単位のフロー制御
- エラー検出機能なし

一方，Martini には，高性能並列処理環境を実現するために，

- 広い通信バンド幅を提供すること，
- 通信のレイテンシが小さいこと，

が要求される．

2.1 RHiNET の通信モデルと Martini への要求

以下，PCI バスに装着される RHiNET ネットワークインタフェースを中心に，通信モデルとそれにともなう Martini への要求仕様について述べる．

RHiNET は，特に新情報処理開発機構で開発された通信ライブラリ PMv2[11]および，その上で動作するソフトウェア分散共有メモリシステム SCASH[12]を効率良く実装できることを目標に設計を行った．SCASH は，PMv2 を用いているだけでなく，バリア同期などでメッセージ通信を用いる一方，ノード間のページコピーに読み込みリモート DMA を採用している．特に，読み込みリモート DMA で高い性能を得ることが重要である．このためリモート DMA をハードウェアによりサポートする．

RHiNET ネットワークは物理層で信頼性のある通信を実現する．そのため，TCP/IP などの信頼性確保のために行われる処理は不要である．この性質を生かして低レイテンシで広バンド幅の通信を実現するために，ユーザレベル・ゼロコピー通信を用いる．

ユーザレベル通信ではユーザプロセスが直接ネットワークインタフェース（NI）にアクセスして通信を起動する．また，ゼロコピー通信ではユーザプロセス上にあるデータを NI が直接 DMA アクセスしてネットワークへ送信したり，ネットワークから受信したデータを直接ユーザプロセス上の受信領域へ書き込んだりする．これらの手法を用いることにより，通常の通信で必要なシステムコールやユーザ空間とカーネル空間の間で行われるコピーなどが不要になるため，低レイテンシで広バンド幅の通信が可能になる．

この際，ユーザプロセスは送信データや受信バッファのアドレスを直接 NI に書き込み，それに従って NI はデータやバッファに対して DMA でアクセスする．ユーザプロセスが使用するアドレスはすべて仮想アドレスであるが，NI からホストメモリへの DMA 時に必要なアドレスは実アドレスである．そのため，NI は仮想アドレスから実アドレスへ変換する機構を備える必要がある．

また，ソフトウェア分散共有メモリをサポートするため，遠隔ノード上のプロセスのメモリ空間に対してアドレスを指定して読み出し/書き込み DMA を行う必要がある．ここで，遠隔ノード上の仮想アドレスは，共有に供されるメモリ領域の先頭の仮想アドレスからのオフセットによって指定する．これにより実際に領域が割り付けられる仮想アドレスを遠隔ノードがあらかじめ知る必要がなくなる．また，多数のプロセスが共有に参加する場合，相手ごとにアドレスを指定する必要がなくなる．これにより柔軟性を増し，異機種，異 OS での使用も可能にする．このため遠隔側の NI ではメモリ領域の先頭アドレスをまず得て，オフセットを加えて仮想アドレスを計算し，これを元に実アドレスに変換する必要がある．

また，SMP やマルチプロセス環境での利用を想定し，同時に複数のプロセスがネットワークインタフェースに対してユーザレベルでアクセスすることを想定する．このため，プロセス間の不当な干渉を防ぐために通信の保護機構が必要である．ユーザレベル通信では通信に関わる情報がユーザプロセスから直接 NI に渡されるため，NI 自身がその情報の正当性を判断する必要がある．

3. Martini の機能

3.1 要求事項

様々な状況と目的に応じて最適な LASN を構成するために，NI に要求される機能をまとめると以下のようになる．

- 保護機構とアドレス変換を含んだゼロコピー転送プリミティブを高速に実行する．特に読み出しリモート DMA でも十分な性能を提供する必要がある．
- DMA ベースの高バンド幅通信とホストから直接データを書き込むことによる低レイテンシ通信の両方を実現する．
- アドレス変換テーブル設定，同期，共有メモリのサポートなど，複雑な処理を扱う必要がある．
- ホスト側 2 種類，ネットワーク側 3 種類のインタフェースを必要とする．

Martini では，以下の手法により，これらの要求に対応した．

- 基本的な転送プリミティブは，保護機構とアドレス変換を含めてすべてハードウェアで実現する．
- DMA 転送用ハードウェアを基本とするが，低レイテンシ通信用データパスも共存させる．

- コアプロセッサを設け，そのソフトウェアで，ハードウェアのモジュール単位で部分的なエミュレートを可能にする．このことにより，性能を大きく犠牲にせずに複雑な処理を実行する．
- 複数のインタフェースに対応するため，それぞれのインタフェースをモジュール化する．各モジュールはそれぞれ適切な周波数で動作させ，モジュール境界に FIFO を置いて異なった周波数間のデータ転送を実現する．

ここでは，このうち，最も大きな特徴である保護機構とアドレス変換を含むゼロコピー転送プリミティブおよび，DMA 通信と低レイテンシ通信に関してその機能面を紹介する．

3.2 ゼロコピー転送プリミティブ

3.2.1 Window

Martini では保護機構を実現するため，ユーザプロセスからの要求を受付ける window と呼ぶ領域を持つ．ユーザプロセスは，window に通信に必要な情報を書き，window 上の特定のアドレス（キックアドレス）に書き込む（キックする）ことにより Martini に通信を要求する．

Martini は，複数のプロセスが並行して通信を行えるように複数の window を提供する．個々の window はホストプロセッサのページ単位に配置されている．ユーザプロセスはあらかじめシステムコールにより window を自プロセスのアドレス空間にマップしておく．

ホストプロセッサの仮想記憶機構により，プロセスは自らのアドレス空間にマップされていない window にはアクセスできない．これにより window へのアクセス権の管理を行うことができる．

3.2.2 PGID

システム内で同時に複数の並列処理プログラムが実行される場合，これらのプログラムを構成するプロセス群相互間での保護を提供する必要がある．すなわち，異なる群に属するプロセスに対して不当な通信を行うことができないような機構が必要である．

このため，RHiNET による通信を行うプロセスには Process Group ID（PGID）と Process ID（PID）の 2 つの ID が割り振られる．

PGID は相互に通信するプロセス群につけられるシステムグローバルな ID で，PID は同一 PGID を持つプロセス群内でそのプロセスを特定するユニークな ID である．

ホストプロセッサのシステムソフトウェアが window をユーザ空間にマップする際に，Martini 内部のテーブル（PGIDTBL）に window の番号（WINID）と，ユーザプロセスの Process Group ID（PGID），Process ID（PID）の対応を登録する．PGID はユーザからは操作できない ID であり，保護機構のキーとして用いることができる．PGID が異なるプロセスのアドレスは指定できないようにすることで，関連のないプロセスが不当に干渉することを防ぐ．

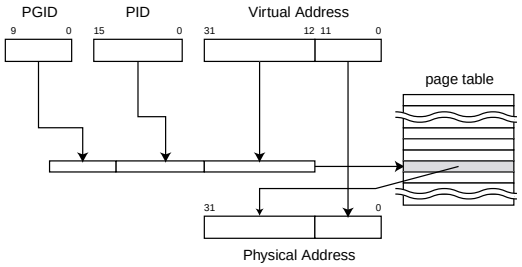

図 1 アドレス変換機構（送信側）
Fig. 1 Address conversion (sender side).

3.2.3 アドレス変換機構

自ホストおよび通信相手方ホストの双方に関して，直接主記憶に DMA 転送を実行するためにはユーザプロセスから与えられた仮想アドレスを実アドレスに変換する機構（**図 1**）が必要である．このため，Martini はアドレス変換をサポートする TLB（PATLB）を備える．

TLB ミスをした場合，コアプロセッサに割込みがかかりコアプロセッサによるハンドリングが行われる．

DMA を行う際，対象メモリ領域は実メモリ上に存在しなくてはならない．これは，あらかじめ pin down しておく方法が最も簡単である．TLB ミスによるハンドリングはコアプロセッサにより行われるため，コアプロセッサがさらにホストプロセッサに割込みをかけて動的に対象領域を実メモリ上に置く実装も可能である．

現在の評価プログラムの実装では，静的に pin down されている領域に対する転送を対象としている．NIC 上のメモリ（Martini チップ外の DRAM）にホストプロセッサが pin down された領域のページテーブルを pin down 時に置き，TLB ミスの際にはコアプロセッサがこの DRAM を参照する．DRAM 上のページテーブルにエントリがない場合にはエラーとなる．

また，リモート DMA の際に，遠隔ノード側のアドレスは，共有に供されるメモリ領域の先頭の仮想アドレスからのオフセットによって指定する．このため，セグメント ID（SID）と呼ぶ ID を提供する．遠隔ノード側のアドレスはこの SID とオフセット（ROFS）の組合せで指定できる．SID とそのセグメントの先頭アド

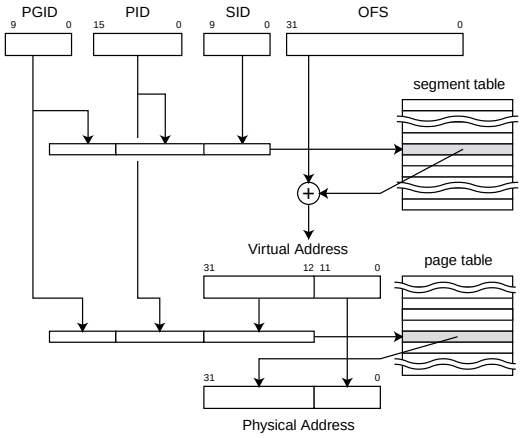

図 2 アドレス変換機構（受信側）
Fig. 2 Address conversion (receiver side).

レスの対応は，そのノードのユーザプロセスがあらかじめ NI 上に登録しておく（**図 2**）．このため，Martini は PATLB に加えて，SID から仮想アドレス（RVA）への変換をサポートする TLB（RVATLB）を備えている．

なお，通信の起動側では NI は先頭アドレスとオフセットによる仮想アドレス指定はサポートしていない．これは，起動側では当然ユーザプロセスが自らの共有領域の先頭アドレスを知っており，プログラム中でこれにオフセットを加えて仮想アドレスにすることは容易であるためである．

3.3 DMA と低レイテンシ通信

3.3.1 RDMA 通信

Martini で最も基本となるプリミティブは PUSH と PULL で，それぞれ DMA を用いたリモートメモリ書き込み，リモートメモリ読み出しに対応する．この 2 つのプリミティブは通常ハードウェアだけで処理される．

PUSH/PULL の実行に必要な情報を**表 1** に示す．ユーザプロセスはこれらの情報を window に書き込みプリミティブを起動する．

3.3.2 低レイテンシ送信機構

大量のデータ転送では DMA を利用した PUSH/PULL が有効であるが，少量のデータ転送ではアドレス変換や DMA のセットアップタイムが無視できない．このため，Martini は PIO によるデータの送信機構も提供している．PIO 送信機構にはある程度の大きさのデータを送ることができる Block On-The-Fly（BOTF）と，1 ワードアクセスで送信が可能な Atomic On-The-Fly（AOTF）の 2 種類を用意する．

表 1 PUSH/PULL で必要な情報
Table 1 Information for PUSH/PULL.

変数名	内 容
RPID	相手プロセス番号
IVA	自プロセス側アドレス
SID	相手プロセスのセグメント番号
ROFS	セグメントに対するオフセット
SIZE	転送データサイズ
STATUS	結果フラグの格納アドレス
RRID	相手ノードまでのルーティング情報
IRID	自ノードまでのルーティング情報

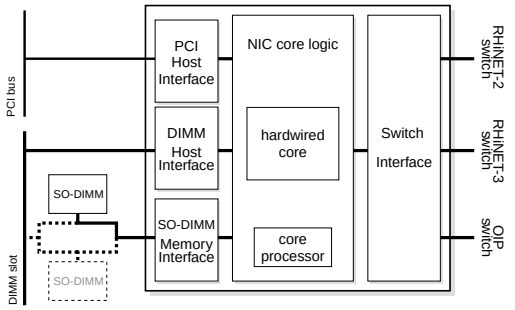

図 3 Martini ブロック図
Fig. 3 Block diagram of Martini.

3.3.2.1 BOTF

BOTF はユーザプロセスがパケットを作成し，送信する機構である．ユーザプロセスは window 上にヘッダを含めたパケットのすべてを構築する．最後にキックアドレスをキックすることで Martini に送信開始を指示する．

3.3.2.2 AOTF

AOTF はホストからのライトアクセスによってパケットを送信する機構である．ユーザプロセスは AOTF 用の領域に書き込む．Martini はこの書き込みを検出し，書き込まれたデータをあらかじめ Martini 内に用意されているパケットヘッダと合わせてパケットを作成し，送信する．

パケットヘッダはあらかじめシステムソフトウェアにより Martini 内に設定される．

4. Martini の構造

Martini は大きく 5 つのブロックから構成される（**図 3**）．

PCI ホストインタフェース部と DIMM ホストインタフェース部はそれぞれ PCI バス，メモリバスに接続され，ホストからのアクセスを Martini 内部に伝える．メモリインタフェース部は Martini に接続される SO-DIMM を制御する．コア部は Martini の中核となるブロックで，ハードワイヤー処理部とコアプロセッ

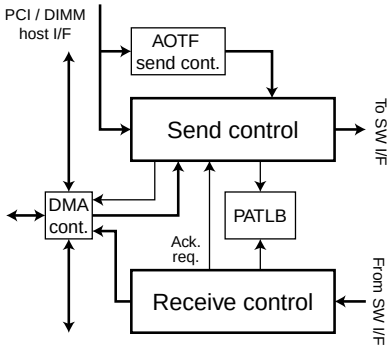

図 4 ハードワイヤー処理部ブロック図
Fig. 4 Block diagram of hardwired core.

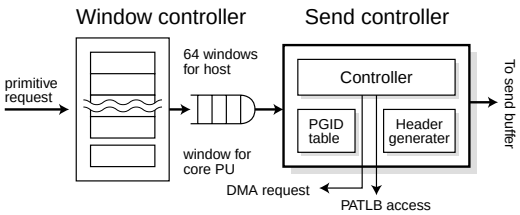

図 5 送信部ブロック図
Fig. 5 Block diagram of send controller.

サからなる．スイッチインタフェース部は RHiNET-2/SW，RHiNET-3/SW および OIP スイッチのプロトコルをサポートする．

4.1 ハードワイヤー処理部

図 4 にハードワイヤー処理部のブロック図を示す．

ハードワイヤー処理部は送信部と受信部に分かれており，送信処理と受信処理を並列に行うことができる．

4.1.1 送 信 部

図 5 に送信部の構造を示す．

PUSH プリミティブの処理を例に送信部を説明する．

(1) ユーザプロセスは window にプリミティブ起動に必要な情報を書き込む．
(2) 送信部は window 上のキックアドレスへの書き込みを契機に，その window の WINID と，書き込まれた情報をキューにコピーする．
(3) PGIDTBL を参照して，キューから取り出した WINID から対応する PGID と PID を得る．
(4) PATLB を参照して IVA を実アドレス（PVA）に変換する（図 1）．
(5) キューから得た情報と PGID，PID を元にパケットヘッダを作成し，送信バッファに送り出す．
(6) DMA コントローラにホスト上のデータの実アドレス（PVA）とデータ長を渡してデータを読み出す．DMA で読み出したデータは先のヘッダに連結され送信バッファを通じてスイッチインタフェースに送り出される．

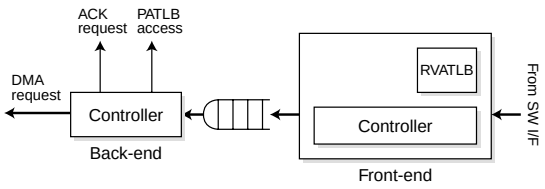

図 6 受信部ブロック図
Fig. 6 Block diagram of receive controller.

データが複数のページにまたがっている場合は，ページ単位に (4) から処理を繰り返し，複数のパケットとして送信する．この場合，2 ページ目以降のアドレス変換は前ページの送信中に行われる．このためページ境界をまたぐことによる転送バンド幅に対するオーバヘッドはほとんどない．

PGIDTBL のエントリが無効の場合や PATLB にミスヒットした場合は，そこで処理を中断し，コアプロセッサに割り込む．コアプロセッサ上のソフトウェアは割込みを受け取るとエラー処理や TLB のリフィル処理を行い，処理終了後，送信部の動作を再開する．

4.1.2 受 信 部

図 6 に受信部の構造を示す．

PUSH プリミティブから生成されたパケット（PUSH パケット）の処理を例に受信部の動作を説明する．

(1) スイッチインタフェースを通して受信したパケットはフロントエンド部でヘッダが解釈される．RVATLB を参照してヘッダに含まれる PGID，RPID と SID からセグメントの先頭アドレス（RVAbase）を得る．これに ROFS を加え，受信側プロセスの仮想アドレス RVA を求める（図 2）．
(2) PUSH パケットのように，ヘッダに続きデータを DMA 転送するパケットではフロントエンドに引き続きバックエンドの処理が開始される．バックエンドは PATLB を用いて RVA から実アドレス（RPA）を求める．次に DMA コントローラを起動し，受信データの DMA を開始する．
(3) さらにバックエンドは受信側と送信側のデータのアライメントが異なる場合にも正しくコピーを行うために，ヘッダにある送信側アドレスの下位 3 ビットと，受信側アドレスの下位 3 ビットを元に受信データをシフトしつつ DMA コントローラにデータを渡す．
送信側と同様に複数ページにわたる DMA が必要な場合は (2) から処理を繰り返す．アドレス

変換は転送中に行われる．

(4) すべての DMA が終了した時点で，送信側に ACK パケットを送るよう送信部に要求する．要求を受けた送信部はプリミティブを発行したノードに ACK パケットを返送する．

ACK パケットを受け取ったノードは，ACK パケットのヘッダの情報に従い，終了フラグを書き込む．この終了フラグを受け取る変数のアドレスは window の STATUS フィールドに書き込まれたもので，ユーザプロセスはその変数をポーリングすることで PUSH プリミティブの終了を知る．

受信部も送信部と同様に RVATLB や PATLB でのミスヒットや，ハードワイヤー処理できないパケットを受信した場合には処理を中断し，コアプロセッサに割り込む．

4.1.3 代行処理機構

Martini は PUSH と PULL の 2 つのプリミティブに関してはハードワイヤー処理部のみで処理が行えるよう設計されている．しかし，それ以外の LOCK，BARRIER，SEND，RECEIVE などの複雑なプリミティブや各種 TLB でのミスヒットなどの例外事項が発生した場合には，ハードワイヤー処理部はコアプロセッサに割り込み，自身は処理を中断する．ハードワイヤー処理部の各モジュールは，コアプロセッサがそのモジュールになり代わって処理を行うことができるように，状態の遷移を停止できるように設計されている．コアプロセッサのソフトウェアは，各モジュールを状態遷移レベルで詳細に信号を制御することが可能であり，ハードワイヤー処理の一部または全部を代行してから，ハードウェアに制御を戻す．この機構を代行処理機構と呼ぶ．コアプロセッサがあるモジュールの処理を代行している間も，他のモジュールのハードウェアは並列に動作することが可能である．Martini では，代行機構により，コアプロセッサのソフトウェアがハードワイヤー処理部とプリミティブの処理を細粒度で分担することが可能となり，基本性能を落とさずに柔軟できめ細かな例外処理や複雑なプリミティブ処理を行うことが可能である．

4.1.4 DMA 制御部

DMA 制御部は Martini 内部での DMA 転送（コアプロセッサやホストプロセッサのコピーによらない転送）の制御を行う（**図 7**）．DMA 転送の対象は，ハードワイヤー処理部（送信バッファ，受信バッファ），PCI ホスト I/F 部，メモリ I/F 部，内部メモリ（SRAM）である．このうちの任意の 2 つのモジュール間での DMA が可能である．さらに，重ならない 2 つの DMA 転送は同時に実行できるので，たとえば，PCI ホスト I/F 部（ホストメモリ）とハードワイヤー処理部内の送信バッファ間で DMA 転送を行いつつ，SRAM とメモリ I/F 部（SO-DIMM）間での DMA 転送を実行することが可能である．

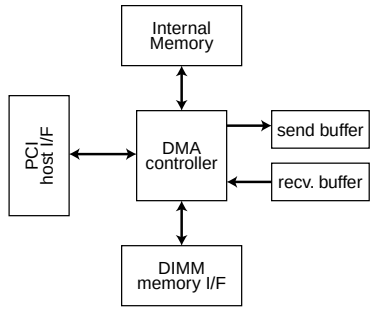

図 7 DMA 制御部
Fig. 7 DMA controller.

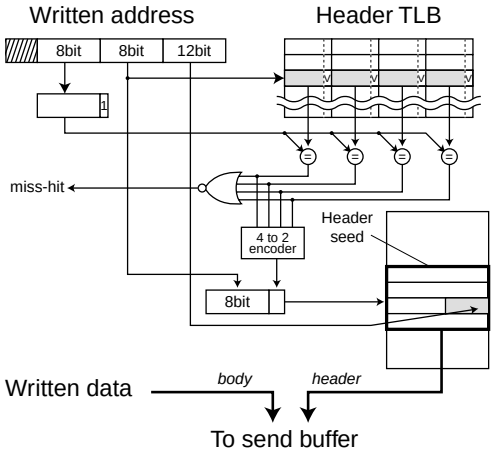

図 8 AOTF 送信部構造
Fig. 8 Structure of AOTF send controller.

4.1.5 AOTF 送信部

ホストが AOTF 用領域に書き込むと，AOTF 送信部が処理を開始する．AOTF 送信部は書き込まれたアドレスのページ番号からパケットヘッダの種（ヘッダシード）を求める（**図 8**）．AOTF 送信部はヘッダシードに書き込みアドレスのオフセット部，アクセスサイズを追加し，パケットヘッダを完成させる．そして，パケットヘッダと書き込まれたデータから構成されるパケットを，送信バッファを通してスイッチインタフェース部に引き渡す．

4.2 コアプロセッサ部

コアプロセッサ部は，R3000 をベースとした 32 bit プロセッサと，256 Kbyte のメモリ，割込みコントローラから構成される．主にハードワイヤー処理部で処理しきれない複雑な処理を受け持つ．

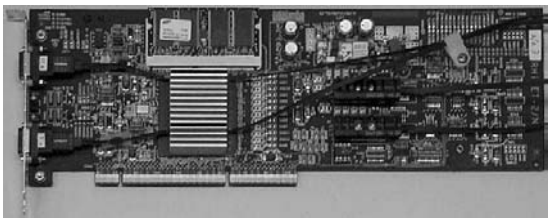

図 9 Martini が実装された RHiNET-2 ネットワークインタフェース
Fig. 9 RHiNET-2 Network Interface.

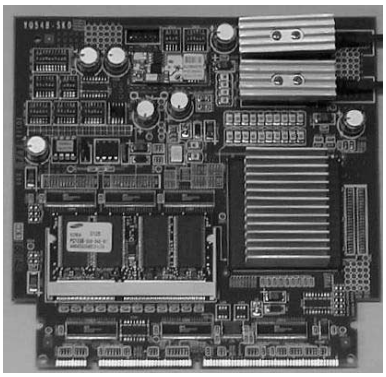

図 10 Martini が実装された DIMMnet-1 ネットワークインタフェース
Fig. 10 DIMMnet-1 Network Interface.

4.3 スイッチインタフェース部

Martini は RHiNET-2/SW，RHiNET-3/SW，OIP-SW の 3 種のスイッチと接続できるインタフェースを持つ．それぞれのスイッチは異なるパケットフォーマットを用いる．

スイッチインタフェースはその 3 種のパケットフォーマットと，コントロール部で使用するパケットフォーマットの相互変換を受け持つ．また，各スイッチの仕様に合わせたフローコントロールを行う．

5. Martini の諸元

Martini は現在第二次試作チップが完成して各ネットワークインタフェースに実装されており，各種評価を行っている．

図 9，**図 10** に RHiNET-2 および DIMMnet-1 ネットワークインタフェースを示す．大きな放熱板が実装されているチップが Martini である．

表 2 に Martini の諸元を示す．**表 3** に各部のゲート数一覧を示す

スイッチ I/F 部のうち，各スイッチ専用の論理のゲート数は**表 4** のとおりである．各スイッチの伝送速度に合わせるためのマルチプレクサ部は同一の回路を使用している．

表 2 Martini の諸元
Table 2 Specifications of Martini.

項 目	
デザインルール	0.14 μm
ダイサイズ	272.91 mm^2
メモリ総量	538 Kbyte
I/O 伝送周波数	
RHiNET-2/SW	800 MHz
RHiNET-3/SW	800 MHz
OIP SW.	250 MHz
内部動作周波数	
コア部	66 MHz
DIMM ホスト I/F	133 MHz
スイッチ I/F	125 MHz
パッケージ	784BGA

表 3 Martini の各部のゲート数
Table 3 Estimation of gate size of Martini.

ブロック名	ゲート数
ハードワイヤード処理部	908 K
スイッチ I/F 部	396 K
DMA 制御部	80 K
SRAM I/F 部	15 K
コアプロセッサ	107 K
その他	107 K
PCI ホスト I/F	272 K
DIMM I/F	366 K
合計	2,163 K

表 4 スイッチ I/F 部の各スイッチ専用論理のゲート数
Table 4 Gate size of special logics for each type of switches.

スイッチ名	ゲート数	説明
sw3	46 K	主に再送系の論理
sw2	9 K	主に ECC 部の論理
oip	8 K	主にプロトコル変換とバンド幅調整

6. 実機評価

Martini チップが実装された RHiNET-2/NI を RHiNET-2/SW により接続して実機評価を行った．評価環境は以下のとおりである．

- CPU：Intel Pentium III 933 MHz
- Chipset：Serverworks Serverset LE
- Memory：512 Mbytes
- OS：Linux (RedHat Linux 6.2, kernel 2.2.17-score)
- PCI：64 bit/66 MHz
- ネットワークスイッチ：RHiNET-2/SW
- ノード–スイッチ間リンク長：5 m

この環境上に SCore システムソフトウェア[13),14)]

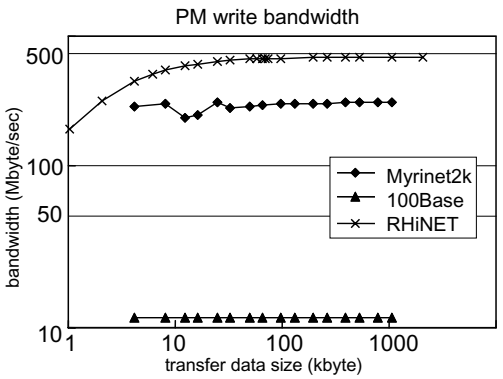

図 11 PM write によるバースト転送性能
Fig. 11 Burst transfer bandwidth of PM write operation.

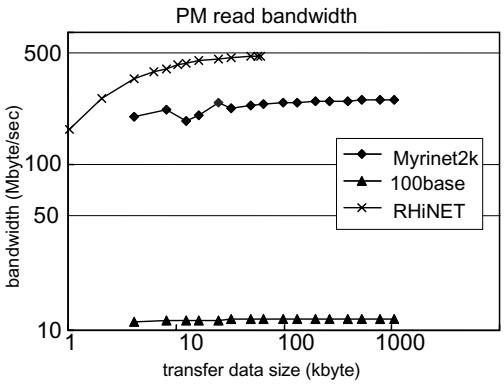

図 12 PM read によるバースト転送性能
Fig. 12 Burst transfer bandwidth of PM read operation.

を実装し，RHiNET の PUSH，PULL 機能を用いた SCore の通信ライブラリ PM のリモートライトとリモートリードのバースト転送性能を測定した☆．

測定結果を**図 11** と**図 12** にそれぞれ示す．比較のため，Myrinet2000 および 100 Mbps の Ethernet を用いた場合の性能も示した．RHiNET でのリモートライトの最大バンド幅は 470 MByte/秒，リモートリードの最大バンド幅は 455 MByte/秒である．

PCI の理論バンド幅は

$$66\,\mathrm{MHz} \times 64\,\mathrm{bit}/8 = 528\,\mathrm{MByte/sec}$$

であるから，RHiNET は PCI のバンド幅をほぼ使いきる性能を出しているといえる．リモートリードについては，RHiNET の PULL 機能は 64 Kbyte 未満の大きさの転送までしかサポートしていないため，この大きさ以上の転送は評価していない．しかし 64 Kbyte 未満でも PCI のバンド幅に対して十分な性能が得られている．

☆ Martini 二次試作チップには PUSH 機能と PULL 機能を同時にハードウェアで実現できないという問題点が見つかり，現在これを改良した三次試作チップを開発中である．

7. シミュレーションによるマイクロベンチマーク

基本機能の処理時間の内訳の計測は実機上ではほぼ不可能なため，Martini の HDL 記述を使用したシミュレーションにより計測した．

7.1 評価環境

シミュレーションによる評価には Cadence 社の Verilog シミュレータ NC Verilog Simulator v3.30 を用い，PCI バスのシミュレーションモデルには Synopsys 社の Smart Model PCI を用いた．

また，シミュレーション上でのホストの動作は，C++ で記述した．これには，Verilog シミュレータ側からホスト上のプログラムを起動し，シミュレータとプログラムの間に通信路を形成するための独自開発のライブラリ[15]を用いている．

7.2 評価条件

シミュレーションの評価条件を以下に示す．

- システム：RHiNET-2
- ネットワークスイッチ：RHiNET-2/SW2
- ノード–スイッチ間の伝送遅延：100 ns
- PCI バス：64 bit/66 MHz
- Martini 動作周波数：66 MHz

上記環境で 2 台のホスト間でのデータ転送を行い，Martini の通信性能を評価した．

7.3 PUSH/PULL プリミティブの評価

7.3.1 PUSH プリミティブの処理時間の内訳

あるホストで PUSH プリミティブを起動し，別ホストに 1,024 Byte のデータを転送した際の，PUSH パケットの発行側における処理時間の内訳を**図 13** に，受信側における処理時間の内訳を**図 14** に示す．

PUSH プリミティブを起動してからリモートにデータが書かれ始めるまでのレイテンシは，ネットワークによる遅延を除くと約 1.5 μs となる．

7.3.2 PULL プリミティブの処理時間の内訳

1,024 byte のデータを転送した際の，PULL パケットの発行側における処理時間の内訳を**図 15** と**図 17** に，受信側における処理時間の内訳を**図 16** に，それぞれ示す．

プリミティブの起動からリモートのデータがローカルメモリに書かれ始めるまでに要するレイテンシは，ネットワークの遅延を含めない場合約 2.0 μs となり，PUSH と同様にネットワークの遅延を往復分含めると約 3.0 μs となる．

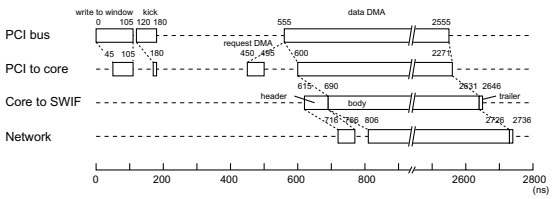

図 13 PUSH パケット発行側の処理時間内訳
Fig. 13 Break down of a PUSH packet sending time.

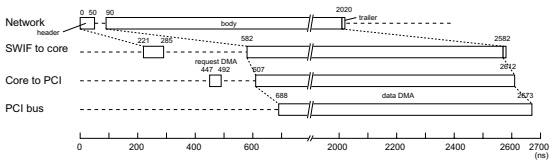

図 14 PUSH パケット受信側の処理時間内訳
Fig. 14 Break down of a PUSH packet receiving time.

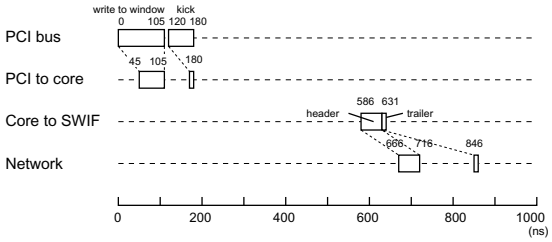

図 15 PULL パケット発行側の送信処理時間内訳
Fig. 15 Break down of a PULL packet sending time.

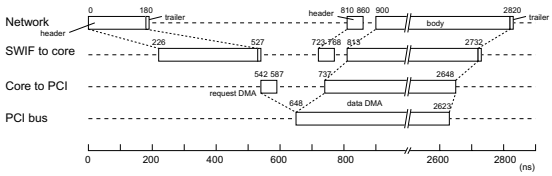

図 16 PULL パケット受信側の処理時間内訳
Fig. 16 Break down of a PULL packet receiving time.

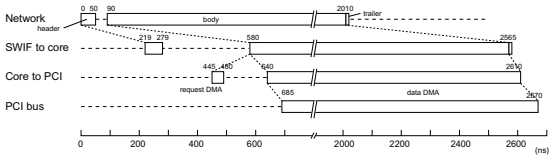

図 17 PULL パケット発行側の受信処理時間内訳
Fig. 17 Break down of a PULL acknowlege packet receiving time.

8. 関連研究

ユーザレベル通信を行う通信方式としては VIA や，これを発展させた InfiniBand [16] がある．InfiniBand は通信リンクの物理層までを規定しているが，VIA では規定していない．これらの通信方式では，信頼性のある高速転送は 1 対 1 通信が基本であり，原則として通信を行うホスト間にあらかじめコネクションを確保してから通信を行う．コネクションにはピンダウンされたメモリ領域が必要で，一度に利用できるコネクション数はハードウェア資源により制限される．また，リモート DMA では，リモートノードの仮想アドレスを指定する．

これに対して Martini では，1 つの Window を介して複数のプロセスと通信を行うことができ，ソフトウェア分散共有メモリの実装に適している．また，リモートノード側のアドレスは，SID とオフセット（ROFS）により決まる．そのため，動的に決まる可能性がある相手側のプロセスでの実際の仮想アドレスを知る必要がない．このためより柔軟で，異機種や異 OS 環境にも適する．

VIA をサポートするハードウェアとしては，Giganet 社の cLAN [17] シリーズと QLogic 社の QLA2300-VI [18] がある．cLAN は Gigabit Ethernet で用いられる物理層を用いておりリンクバンド幅は 1.25 Gbps，QLA2300-VI は 2 Gbps のファイバチャネルの物理層を用いている．このためいずれもリンクバンド幅の制限から Martini よりも転送バンド幅は小さい．

InfiniBand は，2.5 Gbps のリンクを基本として，これを 1 本用いた x1，4 本，12 本用いた x4，x12 の規格を持っている．現在のところ Essential 社や Intel 社などが x1 の Host Channel Adaptor と呼ぶ PCI バスに装着するネットワークインタフェースを発表している [19),20)] が，実装の詳細は明らかにされていない．x1 では PCI バスの転送性能よりもリンクバンド幅が小さいため，これが転送性能の隘路となる．

Quadrics 社の QsNet [21] は ELAN と呼ぶ Martini に大変よく似た構造のネットワークインタフェースコントローラを用いている．リモート DMA では，リモートノードの仮想アドレスを直接指定する．64 bit/66 MHz の PCI バスインタフェースを持つが，ネットワークのリンクバンド幅が 340 MByte（プロトコルオーバヘッドを除く実効データレート）であるため，これが実効転送性能の隘路となっている．

Myrinet 社の最新の Myrinet2000 は，2 Gbps のリンクを持ち，NI 上に 200 MHz で動作する RISC プロ

セッサを持つ．ユーザレベルゼロコピー通信を行う場合，アドレス変換などはこの RISC プロセッサのソフトウェアが行うため，Martini に比べてオーバヘッドが大きいと考えられる．

Martini は，64 bit/66 MHz の PCI バスのバンド幅をほぼ最大限に利用することができるリンクインタフェースとリモート DMA 転送ハードウェアを備えており，これらのネットワークインタフェースに対し最大実効バンド幅が広い．

9. おわりに

高性能並列処理環境を実現するためのネットワークインタフェースコントローラ Martini の設計と内部構成について述べた．

Martini はユーザレベル・ゼロコピー通信により低レイテンシ・広バンド幅な通信を提供する．最も基本となるプリミティブ PUSH，PULL をすべてハードワイヤードで実装しているため，高速に処理できる一方，内部にプロセッサを持ち，複雑な機能や例外処理などに対しても柔軟な対応が可能な構成となっている．

Martini は，現在稼働中の実機を用いてアプリケーションレベルの評価を行うとともに，実機において発見された問題点を改善したチップを開発する予定である．

参考文献

1) Myricom, Inc. http://www.myri.com/
2) 西　宏章，多昌廣治，西村信治，山本淳二，工藤知宏，天野英晴：LASN 用 8 Gbps/port 8*times*8 One-chip スイッチ：RHiNET-2/SW, *JSPP2000*, pp.173–180 (2000).
3) 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).
4) Kudoh, T., Tanabe, N., Yamamoto, J. and Nishi, H.: RHiNET: A network for high performance parallel computing using locally distributed computers, *2000 RWC Symposium*, Real World Computing Partnership, pp.5–10 (2000).
5) 山本淳二，田辺　昇，西　宏章，土屋潤一郎，渡辺幸之介，今城英樹，上島利明，金野英俊，寺川博昭，慶光院利映，工藤知宏，天野英晴：高速性と柔軟性を併せ持つネットワークインタフェース用ネットワークインタフェースチップ：Martini, 情報処理学会研究報告，Vol. コンピュータアーキテクチャ（ARC），No.2000-ARC-140, pp.19–24 (2000).
6) 山本淳二，土屋潤一郎，寺川博昭，田邊　昇，渡邊幸之介，今城英樹，西　宏章，工藤知宏：RHiNET の概要と Martini の設計/実装，*SWoPP 2001*, ARC-144-7, pp.37–42 (2001).
7) 渡邊幸之介，山本淳二，土屋潤一郎，田邊　昇，西　宏章，今城英樹，寺川博昭，上嶋利明：RHiNET/MEMOnet ネットワークインタフェース用コントローラチップ Martini の予備評価，*SWoPP 2001*, ARC-144-9, pp.49–54 (2001).
8) 宮脇達朗，山本淳二，工藤知宏：RHiNET のソフトウェアレイア，*SWoPP 2001*, ARC-144-8, pp.43–48 (2001).
9) Nishi, H., Yamamoto, J., Ohsugi, K., Harasawa, K. and Nishimura, S.: Deskew-LSI for 10-Gbit/s parallel optical links in RHiNET-3 system, *COOL Chips V* (2002).
10) Yoshikawa, T., Hatakeyama, I., Miyosi, K. and Kurata, K.: Optical Interconnection as an Intellectual Property of a CMOS Library, *Hot Interconnects 9* (2001).
11) Tezuka, H., Hori, A., Ishikawa, Y. and Sato, M.: PM: A Operating System Coordinated High Performance Communication Library, *High-Performance Computing and Networking '97* (1997).
12) 原田　浩，手塚宏史，堀　敦史，住元真司，高橋俊行，石川　裕：Myrinet を用いた分散共有メモリにおけるメモリバリアの実装と評価，並列処理シンポジウム JSPP'99，pp.237–244，情報処理学会 (1999).
13) Hori, A., Tezuka, H. and Ishikawa, Y.: Highly Efficient Gang Scheduling Implementation, *SC'98* (1998).
14) 堀　敦史，手塚宏史，石川　裕：ギャングスケジューリングの PC クラスタ上での実装，*HPC*, Vol.67-14, pp.79–84 (1997).
15) 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，デザインガイア 2001 (2001).
16) InfiniBand Trade Association. http://www.infinibandta.org/
17) Emulex. http://www.emulex.com/
18) QLogic. http://www.qlogic.com/
19) Essentail Communications. *IB-Now InfiniBand Adaptor Product Brief.* http://www.esscom.com/
20) Intel. http://www.intel.com/technology/infiniband
21) Petrini, F., chun Feng, W., Hoisie, A., Coll, S. and Frachtenberg, E.: The Quadrics Network (QsNet): High-Performance Clustering Tech-

nology, *Hot Interconnects 9* (2001).

(平成 14 年 1 月 29 日受付)
(平成 14 年 5 月 24 日採録)

山本 淳二（正会員）
1991 年慶應義塾大学理工学部卒業．1997 年同大学大学院理工学研究科博士課程単位取得退学．同年新情報処理開発機構入社．現在，日立製作所中央研究所に勤務．並列処理，ネットワークに関する研究に従事．博士（工学）．

渡邊幸之介
慶應義塾大学大学院理工学研究科に在籍．

土屋潤一郎
2002 年慶應義塾大学大学院理工学研究科修了，現在 NEC コンピュータ事業部に勤務．

原田 浩
2001 年度新情報処理開発機構に所属，現在，コンパックコンピュータスーパーコンピューティング技術部に勤務．

今城 英樹
現在，日立インフォメーションテクノロジーに勤務．

寺川 博昭
現在，日立インフォメーションテクノロジーに勤務．

西 宏章
1991 年慶應義塾大学理工学部卒業．1997 年同大学大学院理工学研究科博士課程単位取得退学．同年新情報処理開発機構入社．2002 年より（株）日立製作所中央研究所に勤務．IP ネットワークの研究に従事．博士（工学）．

田邊 昇（正会員）
1985 年横浜国立大学工学部卒業．1987 年同大学大学院工学研究科修了．同年（株）東芝に入社．1998 年より 2001 年まで新情報処理開発機構つくば研究センターに出向．並列処理，並列アーキテクチャに関する研究に従事．現在,（株）東芝・研究開発センター勤務．工学博士．電子情報通信学会会員．

上嶋 利明
現在，日立インフォメーションテクノロジー勤務．

工藤 知宏（正会員）
1991 年慶應義塾大学大学院理工学研究科博士課程単位取得退学．東京工科大学講師，助教授を経て，1997 年より新情報処理開発機構並列分散システムアーキテクチャつくば研究室長，2002 年より産業技術総合研究所．工学博士．並列処理，通信アーキテクチャに関する研究に従事．

天野 英晴（正会員）
工学博士．慶應義塾大学理工学部情報工学科教授，計算機アーキテクチャの研究に従事．