計算機アーキテクチャ 107−23
（1994. 7. 23）

超並列計算機 JUMP-1 における 入出力サブシステムのアクセス方式

岡田 勉[*1], 中條 拓伯[*1], 松本 尚[*2], 小畑 正貴[*3], 松田 秀雄[*1], 平木 敬[*2], 金田 悠紀夫[*1]

[*1]神戸大学 工学部 情報知能工学科, [*2]東京大学 理学部 情報科学科,
[*3]岡山理科大学 工学部 情報工学科
e-mail: ben@seg.kobe-u.ac.jp

JUMP-1 は，プロセッサ間での高速な通信／同期のための機能を備えた複数のクラスタを，RDT と呼ばれる強力なネットワークで接続した分散共有メモリ型のアーキテクチャを持つ．クラスタと入出力サブシステム間は，STAFF-Link と呼ばれる高速なシリアルリンクにより接続され，入出力バッファが共有メモリ空間にマッピングされた形態をとる．本稿では，JUMP-1 のディスク／画像入出力システムの構成と，共有入出力バッファを用いたディスク入出力／画像表示システムへのアクセス方式とデバイスドライバの役割について述べる．

An I/O Access Method for the Massively Parallel Computer JUMP-1

Tsutomu Okada[*1], Hironori Nakajo[*1], Takashi Matsumoto[*2], Masaki Kohata[*3], Hideo Matsuda[*1], Kei Hiraki[*2] and Yukio Kaneda[*1]

[*1] Department of Computer and System Engineering, Faculty of Engineering, Kobe University,
[*2] Department of Information Science, Faculty of Science, The University of Tokyo,
[*3] Department of Information and Computer Engineering, Faculty of Engineering,
Okayama University of Science

A massively parallel computer JUMP-1 consists of multiple clusters providing inter-processor communication and synchronization mechanism via a broad bandwidth inter-connection network called RDT. We introduce a scalable input/output subsystem configuration which consists of disk/image I/O systems connected via fast serial links each called Serial Transparent Asynchronous First-in First-out Link (STAFF-Link). In this paper, we describe the features and hardware configurations of disk/image I/O systems. Moreover, an I/O access method using shared I/O buffer and also a role of device driver in a cluster are shown.

1 はじめに

近年，半導体集積回路技術や高密度実装技術の進歩によって，複雑なプロセッサや大容量のメモリが低価格で供給されるようになった．このため，多数のプロセッサを相互結合網を用いて接続した並列計算機の研究が，さまざまな大学および研究機関において盛んに行われている．

文部省科学研究費補助金・重点領域研究においても，分散共有メモリ型の超並列計算機のプロトタイプマシン JUMP-1[1][2] の開発が進められている．分散共有メモリ型のアーキテクチャでは，一台もしくは数台の要素プロセッサを一つのクラスタとして，それぞれに共有メモリの一部分を持たせ全体で共有メモリを構築する形態が有効であると考えられている．その理由として，メモリアクセスの局所性を利用することにより，遠隔クラスタへのメモリアクセスを効率良く行えることが挙げられる．しかしながら，この形態ではページングや，長期にわたる演算のチェックポイントを考慮した場合，データが分散しているため，システム全体の性能を発揮するには，入出力サブシステムに対して十分考慮する必要がある．

入出力サブシステムの構成として，ある特定のノードに専用の高速入出力バスを設置し，そのバスに種々の入出力機器を接続する形態が考えられる．nCUBE 社の nCUBE 2[3] では，高速入出力バスとして HiPPI を採用している．しかしながら，多数のプロセッサやクラスタから構成される並列計算機で，専用の高速入出力バスを設置した場合，接続されるクラスタやその近傍においてボトルネックが生じ，システム全体で，円滑なデータの入出力を行うことが困難となる．

これに対して，Intel 社の iPSC/2 hypercube[4] は，ディスク装置を一部の要素プロセッサに分散させた形態を採用している．この場合，ディスクアクセスが分散され，システム全体で広い入出力バンド幅を得ることができる．しかしながら，入出力アクセスが分散されても，SCSI 等のパラレルケーブルではケーブル長に物理的な制限があるため，入出力機器の設置場所が接続されるクラスタの近くに限定されてしまう．

そこで JUMP-1 では，Serial Transparent Asynchronous First-in First-out Link (STAFF-Link)[5] と呼ばれる高速なシリアルリンクによってクラスタと入出力機器を連結した形態の分散入出力システムを構築する．複数の STAFF-Link を用いることでケーブル長の制限を緩和するだけでなく，冗長経路による障害発生時の耐故障性の向上や，ディスクアクセスの分散化も期待できる．

本稿では，JUMP-1 の入出力サブシステムであるディスク入出力／画像表示システムの構成と，複数の STAFF-Link を用いたディスク入出力／画像表示システムを構築する際の，共有入出力バッファを用いたアクセス方式とデバイスドライバの役割について述べる．

2 JUMP-1 の入出力アーキテクチャ

2.1 JUMP-1 の概要

分散共有メモリ型のアーキテクチャでは，通信や同期等のデータアクセスの局所性を利用できない処理が本質的に存在する．従来型の要素プロセッサ (PE) では，このような非局所処理を効率良く実行できない．

そこで，JUMP-1 では Memory Based Processor (MBP)[6] と呼ばれる非局所処理に特化したプロセッサを用いて，通信や同期等の処理を実行させる．PE と MBP を組み合わせることで，局所処理と非局所処理を分離させ，効率的な分散共有メモリを実現する．

JUMP-1 のクラスタ構成を図 1に示す．各クラスタは Recursive Diagonal Torus (RDT)[7] と呼ばれる相互結合網を介して結合される．RDT は 2 次元トーラス構造を基本とする相互結合網で，高い転送バンド幅と，メッシュ構造やハイパーキューブ構造のエミュレーションが可能である等の特徴を持つ．

2.2 入出力サブシステムの構成

まず，JUMP-1 に要求されている入出力機器を挙げる．

- 大容量・高信頼性のディスク
- ハイビジョン・モニタとカメラ
- LAN インターフェースとコンソール

これらの機器のデータ転送路として RDT を用いるが，データが各クラスタに分散しているため入出力装置との接続は，複数のクラスタに分散した形態

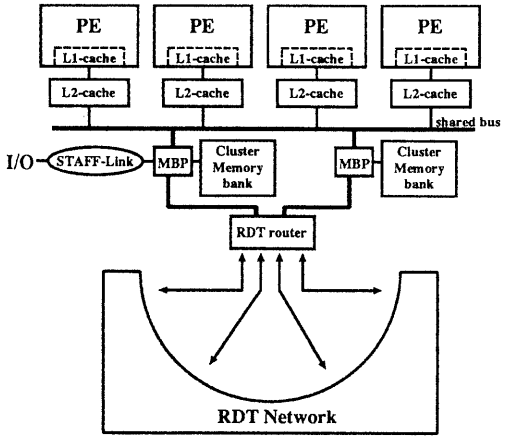

図 1: JUMP-1 のクラスタ構成

が望ましい．また，カメラ等は接続に関して自由度が高い方がよく，人間が接するキーボードやモニタは，安全性，動作環境を考慮すると，計算機本体から分離させた方がよい．以上のような理由により，柔軟かつ実装が容易な STAFF-Link を使用する．入出力バンド幅を確保するため，クラスタとディスク装置間は複数の STAFF-Link を使用する．図 2に JUMP-1 の入出力サブシステムの構成を示す．

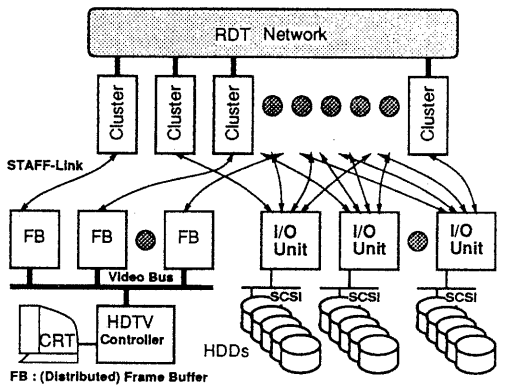

図 2: JUMP-1 の入出力サブシステムの構成

2.3 入出力サブシステムの特徴

JUMP-1 における入出力サブシステムの特徴を以下に挙げる [8].

1. STAFF-Link による設置場所の柔軟性
   入出力機器は本体に近い位置に設置するのではなく，比較的離れた場所に設置し，柔軟で高速な STAFF-Link を介して結合することによって，分散した共有入出力装置を構築する．ディスクのブロックまたはトラック単位のリード／ライト，画像表示のためのフレームメモリへの書き込みなどの入出力装置の低レベルの管理は，入出力サブシステム内のコントローラに処理を行わせることで，データ入出力の負荷分散をはかる．

2. 共有入出力バッファ
   入出力サブシステムのコントローラと JUMP-1 本体とのデータ交換のためのバッファメモリを設け，このメモリを本体の共有メモリマップ上にマッピングする．すなわち，図 3 に示すように，入出力バッファをクラスタから見た入出力拡張メモリとみなすことによって，入出力機器を共有メモリとしてすべてのクラスタ間で共有することが可能となり，種々の入出力機器の特性を吸収することができる．この入出力のための共有メモリを共有入出力バッファと呼ぶ．また，データだけでなく入出力サブシステムに対するコマンドも，本体の共有メモリマップ上にマップされた領域に対するメモリアクセスとして扱う．

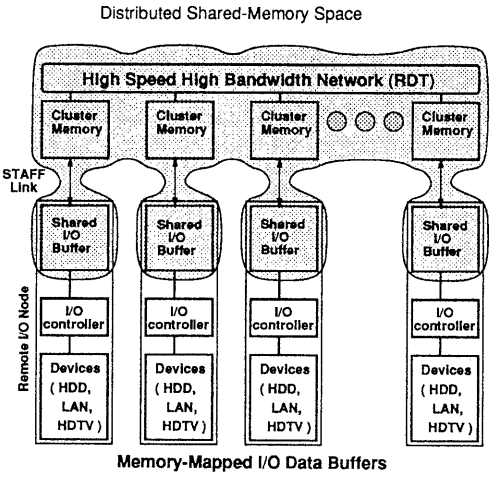

図 3: JUMP-1 の分散共有入出力

入出力サブシステム内の入出力共有メモリへの書き込みや入出力共有メモリからの読み出しはクラスタ上のデバイスドライバから指示される．加えて，入出力サブシステムは遠隔メモリアクセスに対して透過であり，JUMP-1 本体の相互結合網である RDT の延長のように振舞う．ディスクバッファキャッシュ等の OS が管理するバッファは JUMP-1 本体上のメインメモリに作成し，ファイルの共有については，できる限りメモリ共有の形でユーザに提供しアクセスの効率化をはかる．

3. **入出力アクセスの種類**
入出力サブシステムには，コンソールや LAN 等のデバイスが接続されるため，入出力共有メモリへのアクセスはページ単位やブロック単位に限定せず，ワードアクセスやバイトアクセスも使用可能とする．

4. **JUMP-1 と入出力サブシステムの接続リンク数**
JUMP-1 本体と入出力サブシステムを接続する経路が単一のクラスタ経由で固定化されると，ホットスポットの発生やパーティション間の処理の絶縁性の悪化が懸念されるため，1 つのディスク入出力システムと JUMP-1 は 4 系統程度のリンクでシステム内に分散された 4 クラスタ程度の接続ポイントと継れる．

3 ディスク入出力システムおよび画像表示システムの構成

3.1 ディスク入出力システムの構成

JUMP-1 のディスク入出力システムの構成を図 4 に示す．システムは以下の要素より構成される．

- ディスク入出力コントローラ (Disk I/O Controller)
クラスタ上のデバイスドライバから転送されるディスクに対する要求に応じて，ディスク入出力インタフェースを通じてディスクに対するアクセス制御を行う．

- ディスクアレイ (Disk Array)
信頼性を向上させるため，RAID (Redundant Array of Inexpensive Disks) 技術を採り入れた，ディスクアレイをディスク装置として用いる．

- 共有入出力バッファ (Shared I/O Buffer)
トラックバッファとして働き，クラスタ内のデバイスドライバにより管理される．JUMP-1 の共有メモリ空間にマッピングされ，クラスタから直接アクセスすることが可能である．

- DMA コントローラ (DMA Controller)
STAFF-Link を通じて送られてくるデータパケットを共有入出力バッファに連続的に格納したり，ディスクに対して要求を行ったクラスタへの割り込みパケットを生成する機能を持つビルディングブロックである．

- STAFF-Link
高速シリアル通信用 LSI と FIFO および，コントローラから構成される．通信処理の多重化による高速性と，シリアルリンクによる柔軟性を合わせ持つ．本システムでは JUMP-1 のクラスタ内の MBP とディスクおよび画像入出力システムを接続するために使用される．

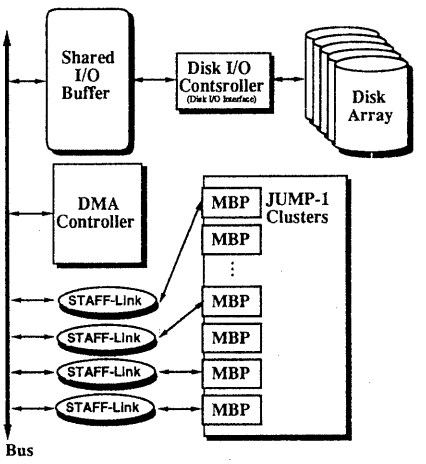

図 4: JUMP-1 におけるディスク入出力システム

3.2 画像表示システムの構成

3.2.1 全体構成

超並列マシンでの画像表示装置には，計算の高速性に見合うだけの表示の高速性が要求される．並列計算機では分割計算した結果を一箇所の表示装置 (フレームメモリ) に集めなければならないが，プロセッサ間接続網を利用すると大量の表示データが計算の

ための通信を圧迫することになる点やフレームバッファの入口で通信が混雑する点で問題がある．このため，並列計算機の画像表示装置ではデータの転送を複数クラスタに分散することが必要となる [9]．

図 5に画像表示システムの全体を示す．ハイビジョン 1 画面分のフレームメモリは，ディスク入出力システムと同様に JUMP-1 のクラスタからは共有入出力バッファとして見え，クラスタからのアクセスは共有空間に対するリード／ライトアクセスとして実現される．その共有フレームメモリは 16 分割され，表示データが 16 クラスタから並列に転送される．クラスタと共有フレームメモリ間は TAXI チップを用いた STAFF-Link により接続される．ハイビジョン規格の画面はドット周波数 74.25MHz で，1920×1035 ドットの構成である．今回は回路簡略化のためフレームメモリの構成を 2048×1024(2M ドット) としている．各ドット 24 ビットなので，1 画面の情報量は 6MB となり，毎秒 30 フレームで 180MB/s の情報量となる．175MHz の TAXI チップで最大 17.5MB/s の転送能力があり，これを 16 本使うと 280MB/s となるので，表示速度に見合うだけのデータ転送幅が確保できる．

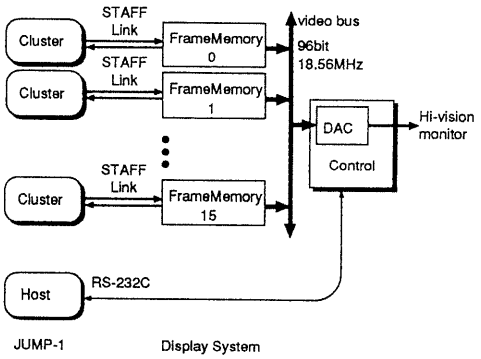

図 5: 画像表示システム

16 枚のフレームメモリボードからディスプレイへのデータ転送はビデオバスによる．ビデオバスには汎用の VME バスを利用する．データ幅を 96 ビットにし 4 ドット分を同時に転送することによって，バス周波数は

$$74.25/4 = 18.56\text{MHz}$$

となる．

実装は，VME ダブルハイトのボードを使い，標準 VME ラックに収める．現在，フレームメモリボードは回路設計を終え，基板製作に取り掛かっている．STAFF-Link の部分はドータボードとして実装する．

3.2.2 共有フレームメモリ

表示とデータ転送を並列に行うため，共有フレームメモリは 2 バンク構成とし，一方が表示中に，他方に次の表示データをクラスタから転送する．フレームメモリボードの構成を図 6に示す．1 枚あたり

$$32\text{K ドット} \times 4 \text{ ページ} = 128\text{K ドット}$$

で，16 枚で 2M ドットとなる．

JUMP-1 からの RDT パケットの制御とビデオバスへのデータ読み出しの制御は，ディスク入出力と同様に DMA コントローラが担い，2 個の FPGA (Field Programmable Gate Array) により実現される．

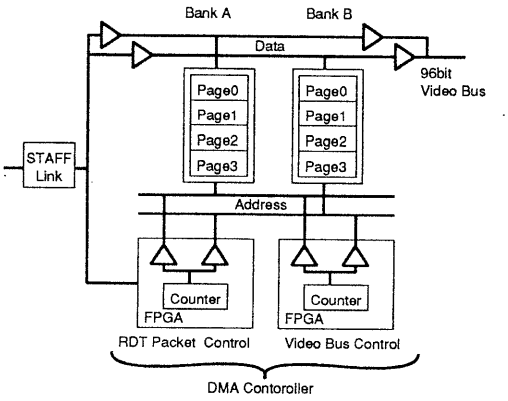

図 6: フレームメモリボード

画面分割は図 7の 6 種類をサポートする．分割方法およびクラスタへの割り当ては静的で，プログラム実行前に分割モードを切替えるものとする．画面分割を変更すると，各フレームメモリからのデータの読み出し順序の変更が必要になるが，これはビデオデータ読み出し側の FPGA の再コンフィギュレーションによって行う．

ドットと縦ライン以外のモードでは連続した 4 ドットは同一のボードの 4 つのページから同時に読み出される．しかし，ドットモードと縦ラインモードでは連続した 4 ドットが 4 枚のボードの同じページから同時に読み出される．このため，この 2 つのモードでは 1 ボードずつページ番号をずらすように設定

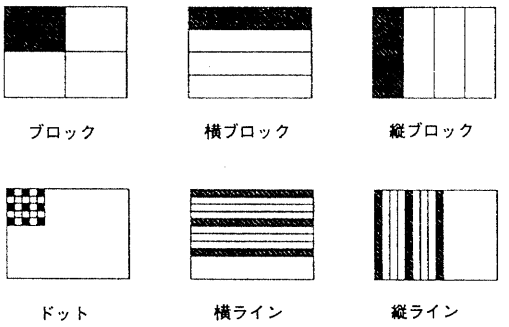

図 7: 分割モード

して 4 ボードの同一ページどうしがバスの異なる位置にデータを出力するようにする.

クラスタからのデータ転送は, 1 画面分のデータを適当なドット数単位で分割して, メモリへのライトアクセスとして行う. この単位をラインと呼び, ラインサイズは固定とする. パケットは, ディスクシステムと同様に RDT パケットに準拠し, ライン番号に対応する共有空間に RGB データを書き込むという形をとる. データ転送後終了コマンドを特定のアドレスに送ることにより, フレームメモリのバンク切替えを行う. 部分変更の場合は必要なデータだけを送って終了コマンドを送ればよい. データ転送時間は各ボードによって違ってくるため, 切替えは全ボードが終了コマンドを受けてからとなる.

誤り検出方法は TAXI チップのエラー検出のみとする. 誤り検出時の処理としては, 表示切替えの周期が短い場合は非再送, 長い場合は再送が適していると考えられるので両方用意する予定である.

3.2.3 コントロールボード

コントロールボードの構成を図 8に示す. ビデオバスからのデータをマルチプレクサを通して DA 変換器 (DAC) に入れ, モニタにつなぐ. 同期信号等のタイミングは FPGA で作成する.

FPGA のコンフィギュレーションや全体の制御を行うためマイクロプロセッサを使用する. このマイクロプロセッサを RS-232C でホスト (コンソール) と接続し, コンフィギュレーションデータのダウンロードや表示モードの切替えなどを行うようにする.

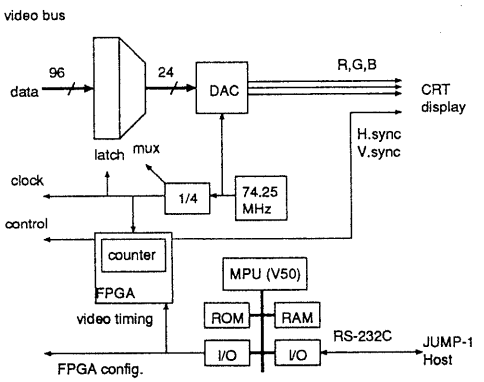

図 8: 画像表示コントロールボード

3.2.4 表示動作

共有フレームメモリへのアクセスを考えると, 表示の面では共有フレームメモリはビデオ回路の近くにあるのがよく, 計算の面ではプロセッサの近くにあるのがよい (物理的にも論理的にも). そこで, 図 9 のように JUMP-1 のクラスタメモリ内にディスプレイの表示イメージを持たせ, これを定期的に共有フレームメモリにブロック転送することで表示を一致させる方式をとる.

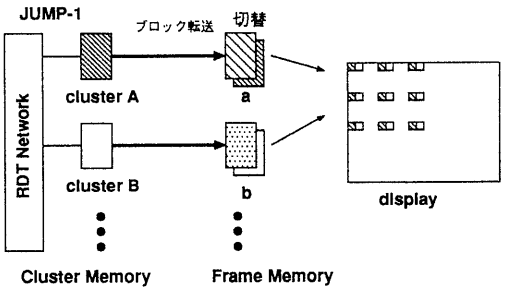

図 9: 表示動作

画面の更新のタイミングとして以下の 3 通りが考えられる. 実際には 1 と 2 の併用を考えている.

1. **ユーザ指定**：プログラム内に共有フレームメモリへのブロック転送関数を記述する.
2. **OS**：JUMP-1 の OS が定期的にブロック転送を実行する.

3. フレームバッファ：フレームバッファ側から定期的に割り込みをかける．

4 ディスク入出力システムにおけるアクセス方式

JUMP-1の入出力サブシステムは，MBPやRDTの特徴を有効に活用し，分散共有メモリ型のアーキテクチャに適合した形態にしなければならない．そのため，入出力サブシステムに対するアクセスも共有メモリアクセスとして扱うのが効率的である．

以下では，ディスク入出力システムに対するリードアクセスとライトアクセスの手順を示す．JUMP-1のクラスタ側のデバイスドライバ(JDD)は，共有入出力バッファ上に既に存在しているディスクブロックに関する情報を管理しているものとする．1つのディスク入出力システムとJUMP-1は4系統のSTAFF-Linkで接続され競合が懸念されるが，クラスタ側で共有入出力バッファの管理情報を共有させることで解決される．また，ブロック番号は全てのディスク入出力システムで一意とし，共有入出力バッファやディスクの空き領域の管理はJUMP-1のクラスタ側で行う．

4.1 リードアクセス

リードアクセスは以下に示す手順で行われる．

1. i-node等より得られる情報を基に，カーネルからリード要求とブロック番号がJDDに渡される．
2. JDDは，そのブロックが既に共有入出力バッファ上に存在しているかどうかを検査する．もし存在していれば，以下の(a)から(e)の手順は省略される．
   (a) JDDは，トラックリード要求，ブロック番号および格納すべき共有入出力バッファのアドレスを入出力コマンド用のメモリに書き込み，休眠する．このトラックリード要求は，MBPによりパケットとして転送される．
   (b) ディスク入出力システムにおいて，DMAコントローラがパケットを解釈し，ディスク入出力コントローラに対してトラックリード要求とブロック番号および格納すべき共有入出力バッファのアドレスを渡す．
   (c) ディスク入出力コントローラは，要求されているブロック番号から1トラック分のディスク領域を，指示された共有入出力バッファのアドレスに転送する．
   (d) ディスク入出力コントローラからDMAコントローラに転送終了が知らされ，DMAコントローラは割り込みパケットを要求元クラスタのJDDに発行する．
   (e) 割り込みパケットを受けとったJDDが休眠状態から復帰する．
3. JDDは，要求されたブロックが格納されている共有入出力バッファのアドレスにリードアクセスを行う．このリード要求は，バイト，ワードまたはページ単位で行われ，実際にはMBPによりパケットとして転送される．
4. ディスク入出力システムのDMAコントローラがパケットを解釈し，要求されたデータを共有入出力バッファから読み出し，パケットにして転送する．
5. JDDは受けとったデータをカーネルに渡す．

4.2 ライトアクセス

ライトアクセスは，以下のような手順で行われる．

1. カーネルから，フリーリスト等より得られる情報をもとに，ブロック番号および書き込むべきデータがJDDに渡される．
2. JDDは，ライト要求，ブロック番号およびデータを，入出力コマンド用およびライト用の共有入出力バッファ領域に書き込み，休眠する．このライト要求は，実際にはMBPによりパケットとして転送される．
3. ディスク入出力システムにパケットが到着すると，DMAコントローラがパケットを解釈し，ライト用の共有入出力バッファ領域に格納し，ディスク入出力コントローラにライト要求とブロック番号を発行する．
4. ディスク入出力コントローラは，ライト用の共有入出力バッファ領域を，指定されたディスクブロック領域に転送する．

5. ディスク入出力コントローラから DMA コントローラに転送終了が知らされ，DMA コントローラは書き込み終了の割り込みパケットを要求元クラスタの JDD に発行する．

6. 割り込みパケットを受けとった JDD が休眠状態から復帰して，カーネルに書き込み終了を通知する:

4.3 パケットフォーマット

入出力サブシステムに関するパケットは，RDT パケットフォーマットに準拠する．具体的には，図 10 に示す **packet command** の部分で，リード，ライト，割り込み等の区別を行う．

ディスク入出力システムにおけるパケットの種類は，

- ディスクに対するトラックリード
- バイト，ワード，ブロック単位のリードとライト
- リード準備完了，ライト完了，障害発生の割り込み

が考えられる．

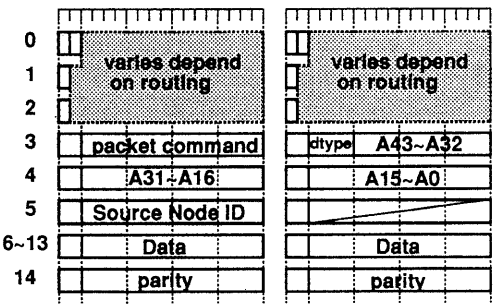

図 10: パケット・フォーマット

5 今後の課題

現在，図 4に示したディスク入出力システム上の DMA コントローラについて，FPGA による実装を進めている．また，障害発生時の回復手順についても検討を進めている．その他，ディスクアクセスを効果的に分散させるための，STAFF-Link の接続形態等についても考慮していく．

今後は，図 4に示したディスク入出力システムの構築を進めるとともに，入出力制御用のデバイスドライバを実装し，JUMP-1 の入出力性能の見積りと評価を行っていく予定である．

謝辞

本研究を進める上で有益なご助言をいただいた神戸大学工学部情報知能工学科 前川禎男教授に感謝いたします．なお，本研究の一部は文部省科学研究費（重点領域研究（1）課題番号 04235130「超並列ハードウェア・アーキテクチャの研究」）による．

参考文献

[1] 松本 尚, 平木 敬“超並列計算機上の共有メモリアーキテクチャ”, 信技報, *CPSY 92-26*, pp.47–55, Aug 1992.

[2] 平木 敬, 天野 英晴, 久我 守弘, 末吉 敏則, 工藤 知宏, 中島 浩, 中條 拓伯, 松田 秀雄, 松本 尚, 森 眞一郎, “超並列プロトタイプ計算機 JUMP-1 の構想”, 情処処理学会計算機アーキテクチャ研究会報告 *ARC102-10*, pp.73–84, Oct 1993.

[3] Juan Miguel del Rosario, “High Performance Parallel I/O on the nCUBE 2”, *Institute of Electronics, Information and Communications Engineers*, vol. J75-D-1, no.8, pp.626–636, Aug 1992.

[4] James C. French, Terrence W. Pratt and Mriganka Das, “Performance Measurement of a Parallel Input/Output System for the Intel iPSC/2 Hypercube”, Technical Report of University of Virginia IPC-TR-91-002, 1991.

[5] 中條 拓伯, 松田 秀雄, 金田 悠紀夫, “超並列計算機における ワークステーションクラスタ・ファイルシステム”, 情報処理学会計算機アーキテクチャ研究会報告 *ARC107-24*, Jul 1994.

[6] 松本 尚, “局所処理と非局所処理を分離並列処理するアーキテクチャ” 第 *43* 回情報処理学会全国大会講演論文集 *(6)*, pp.115–116 Oct 1991.

[7] 文部省重点領域研究「超並列原理に基づく情報処理基本体系」第 3 回シンポジウム予稿集, pp.257–279, Sep 1993.

[8] 中條 拓伯, 松本 尚, 小畑 正貴, 松田 秀雄, 平木 敬, 金田 悠紀夫, “分散共有メモリ型超並列計算機 JUMP-1 の入出力サブシステム”, 情報処理学会計算機アーキテクチャ研究会報告 *ARC104-15*, pp.113–120, Jan 1994.

[9] 中條 拓伯, 小畑 正貴, 金田 悠紀夫, “高速シリアル・リンクを用いた分散画像生成実験システム”, 電子情報通信学会研究会報告 *CPSY93-33*, pp.39–46, Aug 1993.