

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

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

本稿では超並列プロトタイプ計算機 JUMP-1 の入出力サブシステムのアーキテクチャについて概説する。JUMP-1 は、種々のコヒーレンス・プロトコルをサポートするとともに、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とる。クラスタと画像／ディスク入出力ユニットの間は、仮想 FIFO と呼ばれる高速シリアルリンクにより接続され、入出力サブシステム全体としては、入出力バッファが共有メモリ空間にマッピングされた共有メモリアーキテクチャに適合したの形態をとる。本稿では、JUMP-1 の入出力サブシステムの構成と特徴について説明する。さらにディスク入出力ユニットの構成や、画像入出力システムのハードウェアについて述べる。

I/O Subsystem for the Massively Parallel Computer JUMP-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 Systems 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

This paper summarizes the input/output subsystem architecture of a massively parallel computer, JUMP-1. JUMP-1 consists of multiple clusters providing inter-processor communication and synchronization mechanism via broad bandwidth inter-connection network, and supports an efficient distributed shared-memory system with multiple coherence protocols in order to reduce access latency. We introduce a scalable I/O subsystem configuration which consists of image and disk I/O systems connected via fast serial links called Virtual-FIFO. In this paper, we describe the features of the I/O subsystem. Moreover, the hardware configurations of a disk I/O unit and an image I/O system are shown.

1 はじめに

現在、さまざまな大学および研究機関において、将来の超並列計算機のアーキテクチャ、オペレーティングシステム、入出力システムやアプリケーションについて研究が進められている。

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]の開発が進められている。分散共有メモリアーキテクチャでは、少数の PE からなるクラスタどうしを相互に接続した形態が有効であると考えられている。クラスタ内においては、従来のスヌープバスをベースにした構成技術が確立されているため、台数にほぼ比例した性能向上が見込まれ、局所処理と非局所処理を分離分割した場合に前者の処理を効率良く行なえとされている。しかし、複数のクラスタから構成されるシステムにおいて、データが格納されるディスクシステムやサイエンティフィック・ビジュアライゼーションのための画像入出力システムを接続する場合に、十分な入出力バンド幅を確保しなければ、システム全体の性能を発揮することはできない。したがって、今後の高性能な超並列計算機のアーキテクチャについて研究する上で、入出力サブシステムに対して十分考慮する必要があると考えられる。

入出力サブシステムの構成として、インテル社の Paragon[2]などの分散メモリ型の並列計算機においては、ディスク装置をクラスタや要素プロセッサに分散させた形態が実現されている。この場合、入出力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が、ディスク装置やそのインタフェースと要素プロセッサ間のケーブルを短くする、もしくは要素プロセッサボード上にディスクユニットを直接実装するなどの物理的な制限が大きい。

これに対して、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、CRAY などのスーパーコンピュータでは一般的である。代表的なものに HIPPI[3]がある。しかしながら、多数の要素プロセッサやクラスタから構成される超並列システムに対して専用バスを接続する場合、接続されるノードやその近傍においてボトルネックが生じ、システム全体にわたる円滑なデータの入出力を行なうことは困難となる。

また、入出力デバイスを要素プロセッサに分散させ、独自の入出力相互接続網を要素プロセッサ間の

相互接続網とは独立に構成した形態も考えられている [4][5]。

しかしながら、今後の超並列計算機において入出力装置のスケラビリティを考慮した場合、種々の入出力装置の設置場所や接続形態に関しては柔軟に対応しなければならない。そのためには、分散された多数のクラスタから入出力サブシステムへの通信路の両端間に存在する物理的距離に対する制限を、なんらかの形で緩和する必要がある。そして、その通信路の出入口には、フロー制御などに必要な複雑な通信手順をできるだけ簡略化したようなインタフェースを提供する必要がある。

また、いくつかのクラスタでパーティションを形成し、それぞれのパーティション内において独立したタスクを実行するとき、パーティション間で実行性能の影響が及ばないように、ハードウェア、ソフトウェアの両面から支援する必要がある。そのためには、入出力アクセスに対して、できる限りパーティションからのアクセスを分離した機構を提供する必要がある。

そこで、複数のクラスタから画像入出力/ディスクシステムへの複数の高速シリアルリンクを設け、入出力ユニット間で、コントロールネットワークを持つ、図 1 に示すような入出力サブシステムの形態を考える。

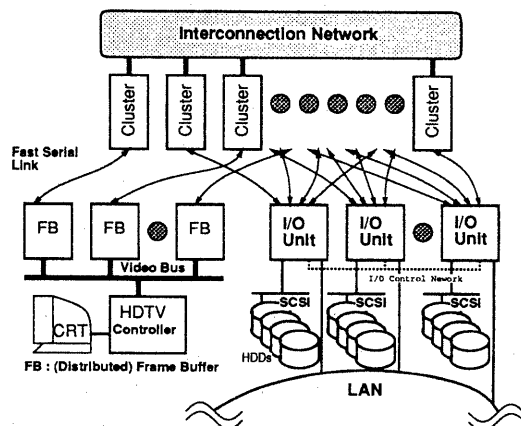


図 1: 超並列計算機の入出力サブシステム

その通信路の両端から見ると高速シリアルリンクが双方向の FIFO に見えるような仮想 FIFO と呼

ばれる機構を提供する。次節では、仮想 FIFO の構成について簡単に触れ、続いて JUMP-1 の入出力サブシステムの構成と特徴について説明する。さらに具体的なディスク入出力ユニットの構成や、画像入出力システムのハードウェアについて述べる。最後に現状と今後について説明を行なう。

2 高速シリアル通信による仮想 FIFO 機構

2.1 入出力インタフェースと狭域・広域ネットワーク

並列計算機システムにおいて、要素プロセッサ間の通信方式やネットワークポロジはシステム全体の性能を左右する重要な要素となる。要素プロセッサ同士はバックプレーン上の高速なバスで接続したり、物理的に離れたルータ間をフラットケーブルなどで結ぶのが一般的である。

しかしながら、多数の信号線を有するケーブルで高速にデータ通信を行なう場合、ケーブルの持つ誘導特性などから生じるノイズの影響によりケーブル長には限界が存在する。また、転送ビット幅を広く取れば、ルータのポート数の増加にしたがってコネクタの基板上の占有面積などの空間的な制約から逃れることはできない。また、入出力装置と計算機本体を接続する場合において、機器を分散して配置したり、カメラや各種センサーからの情報を入力したり、遠隔で操作する場合など、その入出力装置と計算機本体との間の物理的距離は深刻な問題となる。

そこで我々は、シリアル通信の可能性に着目し、物理的、空間的な制約と転送スピードとのトレードオフを考慮した上で、並列計算機システムにおける通信方式の一形態として、仮想 FIFO 機構を提案している [6]。

技術的な背景として、B-ISDN や ATM 技術などの広域ネットワークの分野で開発された高速シリアル通信用 LSI が安価に提供されるようになったことがある。これらの LSI によるシリアル通信の信頼性が高まり、さらに基板への実装技術の進歩が加わり、我々はシリアル通信が今後の並列計算機内における一つの通信形態として位置づけられるのではないかと考えた。

仮想 FIFO による入出力システムの伝送速度と伝

送距離の位置付けを示すために、現状の入出力インタフェース、LAN および WAN の伝送速度と伝送距離を図 2 に示す [7]。仮想 FIFO による入出力システムは、伝送速度においては数十 Mbps から、数 Gbps に及ぶ範囲をカバーし、通信距離においては数メートルから、数百メートルまたはそれ以上の距離に及ぶ入出力データの通信に利用できると考えられる。

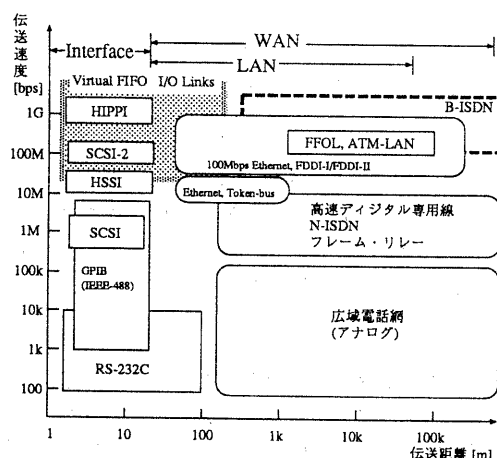


図 2: 入出力インタフェース / LAN / WAN の伝送速度と伝送距離

2.2 仮想 FIFO の構成

入出力システムを高速シリアル・リンクにより構成する場合に要求される事項を以下に示す。

1. 通信ノード間の距離の隠蔽

これは、シリアル通信の最大の利点である。超並列計算機を構成する要素プロセッサが増加するに従い、システムの筐体も大きくなり、分散したクラスタと入出力システムの物理的な距離を考慮すれば、シリアル・リンクによる接続が最も自然な実現方式となる。

また、今後の計算機システムにおいて、計算機本体と入出力機器とが同じ場所に存在する必要はなくなり、ディスク機器は隣の部屋にあり、画像入出力システムは別の建物にあるといった状況も十分考えられ、そのためには空間的距離の隠蔽は必要不可欠な要素となると考えられる。

2. 通信スループットの向上

従来のシリアル通信インタフェースにおいて、パラレルデータをシリアルに変換するために要する時間による通信遅延が生じ、通信性能が上げられなかった。シリアル通信は、

- (a) データの書き込み
- (b) パラレル-シリアル変換
- (c) シリアル通信
- (d) シリアル-パラレル変換
- (e) データの読み出し

の5つのフェーズに分けることができる。(b)～(d)のフェーズは、シリアル通信用LSIにより高速に処理し、送信側と受信側にバッファを設けて、5つのフェーズをオーバーラップさせることにより、通信スループットを向上させることができる。

図3に、仮想FIFO機構の構成を示す。送受信用高速シリアル通信用LSI(TAXIチップ[8])と送信用/受信用の2つのFIFO、さらにFIFOが溢れないようにハンドシェイク(Xフロー制御)を行ないながら非同期通信制御を行なう通信コントローラから通信ブロック(Communication Block)は構成される。

2つの通信ブロック間を、ツイストペアまたは同軸ケーブルで接続することによって、その両端には仮想的に双方向のFIFOが形成され、双方のノードから見た場合にノード間の物理的な通信距離は隠蔽され、透過な通信路を構成することができる。このように構成されたFIFOを仮想FIFOと呼ぶ。

通信コントローラはXon/Xoff制御を行ないながら非同期通信制御を行なう。すなわち、転送先の受信FIFOの残り容量が半分を割ると、受信側はXoffメッセージを転送し、送信の中断を要求する。その後、受信FIFOが受信可能となった時点でXonメッセージを転送し、送信再開を要求する。これらの制御は、通信コントローラ間において自動的に行なうため、仮想FIFOの両端では、シリアル通信に関して考慮することは何もない。送信側では、送信FIFOが満杯になるまでデータを書き込むことができ、受信側では、受信FIFOが空になるまでデータを読み出すことができる。

この仮想FIFOにより複数のリンクを構成することによって、広いレンジの入出力通信バンド幅を持ち、超並列計算機の要素クラスタと入出力機器との伝送距離の制限を緩和した入出力システムの実現が可能となる。

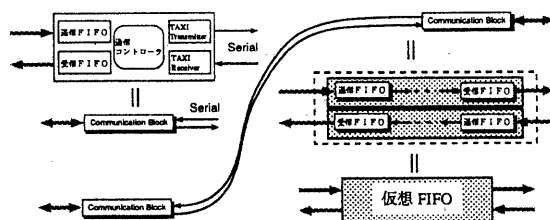


図3: 仮想FIFOの概念

3 JUMP-1の入出力アーキテクチャ

JUMP-1は分散共有メモリによる通信機構や同期機構を専用ハードウェアであるMemory-Based Processor(MBP)[9]により強力にサポートされ、強力な相互ネットワーク装備した超並列計算機である。したがって、入出力システムについてもMBPの機能や、相互ネットワークのバンド幅を有効に利用し、共有メモリアーキテクチャに適合した形態を考えるべきである。

そこで、MBPから仮想FIFOにより接続されるディスク入出力ユニットや画像入出力ユニットなどの入出力機器を共有メモリ空間の一部にマッピングを行なうことによって、クラスタ上の要素プロセッサからは、これらの機器に対してメモリアクセスによりデータの授受を行なえるような機構を考える。

また、要素プロセッサ間のネットワークと独立に、入出力専用ネットワークを設けたシステム[5][10][11]があるが、そのコストに見合うだけの性能を発揮しなければ広バンド幅の2つの独立したネットワークを持つ意味はない。しかしながら、要素プロセッサ/クラスタ間ネットワーク上にディスクや画像データなどの粒度の大きいデータをそのままのサイズで転送した場合、同期やコンシステンスを維持するための制御メッセージなど緊急を要するメッセージの転送の妨げとなる恐れが生じる。

そこで、我々はJUMP-1におけるクラスタ間を

接続する Recursive Diagonal Torus (RDT) の持つ高い転送バンド幅を考慮した上で、入出力専用ネットワークを RDT と併設する形態ではなく、入出力データについてもメモリアクセスと同じ手続きを取り、RDT を有効に活用する方針を取る。

すなわち、画像データやディスクからのデータに対しても RDT 上のパケットの形で転送することにより、クラスタと分散共有メモリ空間、およびクラスタと入出力機器との間でバランスの取れたデータの流れを供給する機構を提案する。

以上の点を考慮して、JUMP-1 における入出力サブシステムの特徴を以下に列挙する。

1. シリアルリンクによる設置場所の柔軟性

入出力機器は、本体に密接した位置に設置するのではなく、比較的離れた場所に設置し、非同期のシリアルリンクである仮想 FIFO により結合することによって分散した共有入出力装置群を形成する。装置の低レベルの管理運営は入出力ユニット内のコントローラ (CPU) が処理を行うことによって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

入出力ユニットのコントローラと 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4に示すように、入出力バッファをクラスタから見た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。

入出力バッファ領域はハードウェアによるコンシステンシ制御は行わないプロトコル (non-coherent) を使ったメモリアクセスによって使用され、入出力ユニット内のバッファへの書き込みやバッファ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。ハードウェアによる細粒度のコンシステンシ維持を行わない点以外は、仮想 FIFO は遠隔メモリアクセスに対して透明であり、JUMP-1 本体の相互結合網 RDT の延長のように振舞う。ディスクキャッシュ等の OS が管理するバッファは JUMP-1 本体上のメインメモリに持ち、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。

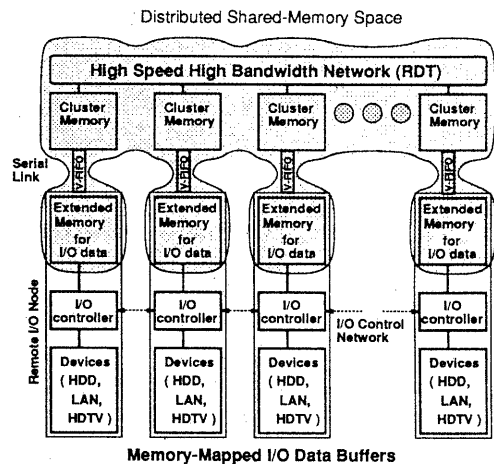


図 4: JUMP-1 の共有入出力バッファ

3. 入出力アクセスの種類

入出力ユニットには、コンソールや LAN などのスカラアクセスが有効なデバイスが接続されるため、バッファメモリへのアクセスはページアクセスやブロックアクセスに限定せず、ワードアクセスやバイトアクセスも使用可能とする。

4. 入出力制御用ネットワーク

入出力用バッファメモリ上において、読み込みデータの準備が整った時や、書き込み時に領域が確保された時には、直接割り込みの原因となる入出力アクセスを行なったクラスタに対して割り込みメッセージを送信しなければならない。

しかしながら、複数のパーティション間において実行性能に影響が及ばないようにパーティションの独立性を維持するためには、入出力ユニットから本体への割り込みメッセージは、入出力を要求したクラスタに直接に返送される必要がある。そのためには、ユニット間の入出力制御ネットワーク (I/O Control Network) を介して直接対象となるクラスタに割り込みメッセージを返送する。また、このネットワークのバンド幅を拡大することによって、ファイルシステムの長期的な再構成、すなわちファイルアクセス

の履歴を元に最適なファイルの配置を行なった
り、ファイルシステムレベルでのガベージコレ
クションなどにも対応することが可能である。

5. JUMP-1 と入出力ユニットの接続リンク数

JUMP-1 本体と入出力ユニットを接続する経路
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、ユニットと 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。

6. 入出力ユニット上の制御用 OS

入出力ユニットのデバイス制御にはリアルタイム
に対応したものが望ましい。また、入出力ユ
ニットの接続されるディスクの総容量は 1 ユニ
ット当たり 2GB ~ 8GB にもなるため、大容量の
ファイルシステムに対応したものが必要となる。

3.1 ディスク入出力ユニットの構成

ディスク入出力ユニットの構成を図 5 に示す。ユ
ニットの構成は以下に示す要素から構成される。

- 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、SCSI インタ
フェースを通じてディスクに対するアクセス制
御を行なう。
- 入出力拡張メモリ (I/O Extended Memory)
トラックバッファとして働き、ディスク入出力
コントローラにより管理される。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)
仮想 FIFO を通じて送られてくるデータパケ
ットを入出力拡張メモリに連続的に格納したり、
ディスクに対して要求を行なったクラスタへの
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。
- 入出力制御ネットワーク (I/O Control Network)
割り込みメッセージなどの制御パケットを要求
元のクラスタに直接転送するための制御用ネッ

トワークである。クラスタ間を接続する RDT
ほどのバンド幅は必要ない。

- 制御パケットルータ (Control Packet Router)
入出力制御ネットワーク上を流れる制御パケ
ットのルーティングを行なう。

入出力ユニット上の制御用 OS には、大容量の
ディスクの接続と実時間制御への対応も考慮して、
Real-Time Mach を候補として考えている。

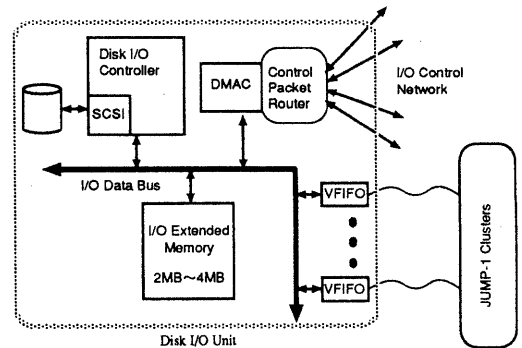


図 5: JUMP-1 における入出力ユニット

3.2 画像入出力ユニット

次に、JUMP-1 における仮想 FIFO を用いた高
密度画像入出力システムについて述べる。画像入出
力システムの構成を図 6 に示す。

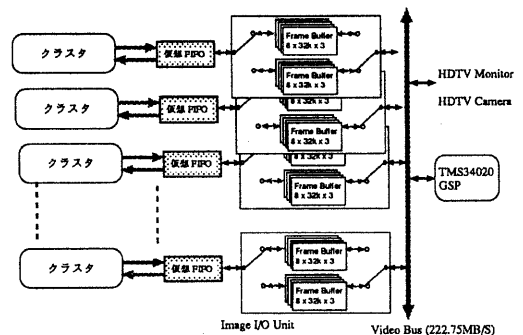


図 6: JUMP-1 の画像入出力システム

ハイビジョン規格の画像データを処理する場合に、
クラスタと画像入出力インタフェース間とのリンク

に要求される転送速度は、リンクを持つクラスタの数を n とすると、

$$180 / n \text{ MB/S} = 1440 / n \text{ Mbps}$$

となる。例えば、 $n = 16$ とした場合に要求される転送速度は 90 Mbps となる。現在開発を進めている仮想 FIFO の転送能力は最高 220 Mbps であるが、画面の分割方法や描画アルゴリズム、パケットの生成・解凍の負荷を考慮して、 n の値を選定する必要がある。

クラスタから出力される仮想 FIFO の一端は、2 バンクのフレームバッファに接続される。仮想 FIFO からの画像データが一方のフレームバッファ・バンクに転送されている間に、他方のバンクからハイビジョンモニタへ出力画像データを表示する。また、カメラなどの画像入力装置からのデータがフレームバッファへ入力画像を書き込んでいる間にもう一方のバンクから画像データを読み出し、仮想 FIFO に流し込むことによって、クラスタに処理すべき画像データを転送する。このように画像データの入出力と仮想 FIFO によるデータ転送をオーバーラップさせることにより、リアルタイムの画像処理を可能とする。また、表示画像の移動や回転などは、TMS34020 Graphics System Processor (GSP) が行なう。

フレームバッファへのアクセスを考えると、表示の面ではフレームバッファはビデオ回路の近くに置くのがよく、計算の面ではプロセッサの近くに置くのがよい。そこで、JUMP-1 のクラスタメモリの一部にフレームバッファのコピーを持たせることにし、以下のような方式でクラスタメモリの中の表示イメージと実際の表示が一致するようにする。

1. 仮想 FIFO によりフレームバッファに接続されるクラスタメモリは、分割された画面のイメージを持っている。フレームバッファメモリは共有空間にマッピングされ、クラスタ上の要素プロセッサはこの領域に対して直接ブロックデータ転送を行なうことによって画像の入力・表示を行なう。分割は静的で、ブロックやラインなどの単位で行なう。
2. フレームバッファは垂直同期信号に同期して、画像表示を担うクラスタに対して割り込みメッセージを転送する。
3. クラスタは 1/30 秒（ビデオ 1 フレーム）以内に自分の持っている分割部分のイメージデータ

をフレームバッファに送る。この次の垂直同期信号のタイミングでバッファを切替える。

以上のシーケンスを定期的（毎秒 n コマ）に行なうと一定時間内に JUMP-1 クラスタ上のメモリ内の画面イメージとフレームバッファのデータは一致する。これにより、計算結果（過程）がリアルタイムで表示できる。

画面の分割については、縦方向と横方向のそれぞれで行ない、きざみ幅（ブロックの 1 辺）と繰り返し周期が変えられるようにする。たとえばフレームバッファのユニット数が 16 の場合、縦方向のみについて、きざみを 1、周期を 16 にすると縦方向のライン分割となる。また、縦横のきざみを 1、周期を 4 ずつにすると 2 次元ドット分割となる。この動作を図 7 に示す。

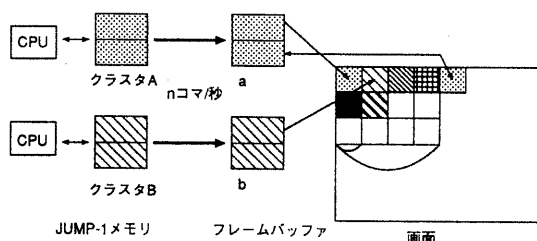


図 7: 表示動作

分割方法や転送周期（1 秒間のコマ数）などは JUMP-1 から仮想 FIFO を通じて変更できるようにする。

実装については、VME ダブルハイトの基板を用い、標準 VME ラックに収めることを目標として設計を行なっている。

1 ユニットのハードウェア（暫定）仕様を以下に示す。

- メモリ：24 個（3 色×4 ウェイインタリーブ×ダブルバッファ）
- バススイッチ：48 個（メモリ数×2 ポート）
- レジスタ：30 個（データ、アドレス）
- コントローラ：PLD, FPGA（アドレス生成、読み書き、バススイッチ）
- 画像バス：96 ビット幅、18.56MHz

4 今後について

現在, JUMP-1 からのファイルアクセスのための制御プロトコルの検討を進めており, クラスタと入出力ユニット間のパケットフォーマットを設計している. 図 5 に示した入出力ユニット上の DMA コントローラについては, LCA による実装を進めている. それと並行して図 8 に示すディスク入出力システムの実装を進めている. このシステムは, ワークステーションのファイルシステムをそのまま流用し, 拡張バッファメモリへのアクセスをソフトウェアエミュレーションにより行なうことによって初期ファイルシステムのデバッグ環境を提供することを目的とする.

図 8 において, MBP からは UNIX のファイルアクセスに関するシステムコールに準拠したメッセージを転送し, そのメッセージに従って, ワークステーション上のカーネルが実際にファイルアクセスを行ない, 要求されたデータを仮想 FIFO に転送する.

今後は, 図 5, 8 の実装を進めるとともに, ユニット上の入出力制御のため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.

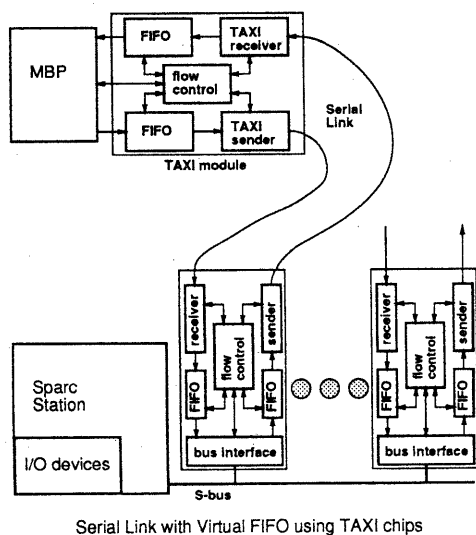


図 8: JUMP-1 の初期入出力ユニット

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] 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, 森真一郎, “超並列プロトタイプ計算機 JUMP-1 の構想”, 情報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.
- [2] Intel Corporation, “Paragon XP/S, Product Overview”, 1991.
- [3] ANSI document, “High Performance Parallel Interface, document #X3T9.3”
- [4] Parsytec, “Technical Summary Parsytec GC”, 1991
- [5] 廣野英雄, 松岡浩司, 岡本一見, 横田隆史, 堀敦史, 児玉祐悦, 佐藤三久, 坂井修一, “超並列計算機 RWC-1 における入出力機構”, 情報処理学会研究会報告 ARC101-5, pp.33-40, Aug 1993.
- [6] 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.
- [7] 上原政二編集, “標準 LAN 教科書”, アスキー出版局, 1993.
- [8] 日本 AMD 株式会社, “Am79168/Am79169-275 TAXI-275 Transmitter / Receiver Preliminary Data Sheet”, 1993.
- [9] 松本尚, 平木敬, “Memory-Based Processor による分散共有メモリ” 並列処理シンポジウム JSPP'93 論文集, pp.245-252, May 1993.
- [10] Luis-Felipe Cabrera and Darrell D. E. Long, “Swift: Using Distributed Disk Striping to Provide High I/O Data Rates”, Technical Report of UC Santa Cruz CRL-91-46, 1991
- [11] Kang G. Shin and Greg Dykema, “A Distributed I/O Architecture for HARTS”, In *Proceedings of the 1990 Annual International Symposium on Computer Architecture*, pp.332-342, 1990.