

CPLD を利用したワークステーションクラスタ用同期コントローラ

早川 潔, 関口 智嗣

電子技術総合研究所

〒305-8568 つくば市 梅園 1-1-4

TEL : 0298-54-3308

E-mail : hayakawa@etl.go.jp

あらまし ワークステーション同士を高速なネットワークに接続することによりワークステーションクラスタを形成し, そのワークステーションクラスタ上で並列処理を行うことにより計算の高速化を試みる研究が行われている. ワークステーション同士を接続させるためのネットワークインターフェースとして, Myrinet 等の高速ネットワークインターフェースが開発されている. それらの多くは, 高機能であるが, 高価である. 我々は, 機能を同期と簡単な Collective 通信機能に絞り, 安価にその機能を高速に処理できるコントローラを開発中である. 本稿では, その同期コントローラについて報告する.

キーワード ネットワークインタフェース, バリア同期機構, Collective 通信, メッセージパッシング

Synchronization Controller using CPLD for Workstation Cluster

Kiyoshi Hayakawa, Satoshi Sekiguchi

Electrotechnical Laboratory

1-1-4 Umezono, Tsukuba, Ibaraki 305-8568

TEL : 0298-54-3308

E-mail : hayakawa@etl.go.jp

Abstract Parallel Processing using workstation cluster connected with high speed network has been studied, which allows to reduce calculation time. To communicate faster, high-speed network interfaces, such as Myrinet, have been developed. Most of these interfaces are high performance, but expensive. We are developing cheaper network interface called synchronization controller, which supports only barrier synchronization and simple collective communication. In this paper, we describe the synchronization controller which allows nodes to synchronize and communicate with shorter latency using the synchronization network.

key words Network Interface, Barrier Synchronization, Collective Communication Message-Passing

1. はじめに

ワークステーションクラスタを用いて並列処理することにより、大規模な科学技術計算を高速に処理する研究が盛んに行われている[1][2][6][7]。ワークステーションクラスタ上での高速な並列処理を実現するためには、主にネットワーク上でのデータ転送時のスループット向上・レイテンシの低減が必要となる。

ネットワークのスループット向上やレイテンシ低減を実現するために、Myrinet[2]などの高機能ネットワークインターフェースが開発されている[1][7]。これらのネットワークインタフェースは、高機能ゆえ比較的高価である。

一方、Gigabit-Ether 等の Ether 系のネットワークインターフェースは、デファクトスタンダードゆえ比較的安価になりつつある。しかし、Ether 系のネットワークインタフェースは、ある程度高スループットであるが、バリア同期や Collective 通信の処理にはある程度時間がかかってしまう。

そこで、高いスループットが必要なデータ通信を伴う処理は Ether 系のネットワークを使用することにし、バリア同期や

Collective 通信は専用ネットワークを使用して処理することを考える。

本稿では、その専用ネットワークをコントロールする同期コントローラについて報告する。

2. ワークステーションクラスタ構成

本同期コントローラを搭載するワークステーションクラスタ構成を図 1 に示す。各ワークステーション間は、Ethernet Switch などの高速ネットワークと同期や Collective 通信専用ネットワークである Synchronization Network で結合する。高スループットを要求する処理は従来ある Ethernet を利用し、低レイテンシを要求し、且つ、同期コントローラがサポートしている処理であれば、Synchronization Network を使用して処理する。

3. 同期コントローラがサポートする機能

本同期コントローラで処理される機能は、ワークステーションクラスタ（分散メモリ）上で行われるメッセージパッシングをサポートする機能に限定する。メッセージパッシングで必要な機能として、バリア系の同期機構

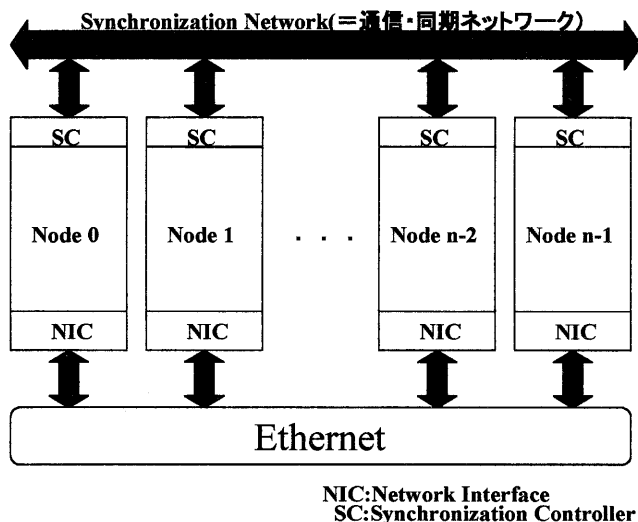


図1. ワークステーションクラスタ構成

や Reduce など Collective 通信の一部をサポートする。リモートメモリアクセスに必要な同期機能（ロック、セマフォ等の排他制御機能）はサポートしない。

バリア系の同期を高速に処理するために、ユーザーレベル通信機能のサブセットをサポートする。

プログラムの実行時間解析を行うために、タイマ機能をサポートする。

3.1. バリア系の同期機構

本同期コントローラで実装する同期機構は、バリア同期機構、RBNQ 同期機構[9]および重複化バリア同期機構[4]の3種類である。

3.1.1. バリア同期機構

バリアは、あるプロセスがバリアの同期ポイントに到達した時点で、同期に参加している全てのプロセスがそれぞれの同期ポイントに到達するまで、次のプログラム実行を延期するという動作を行う。

バリアグループを設定し、そのグループ内でバリア同期を処理するようにする。バリアグループは、プログラム実行中に変更することができるようにする。

3.1.2. RBNQ 同期機構

RBNQ 同期機構では、「面状&強制参加&全順序関係&オーバーラップ不可能」[8]なバリア同期モデルの出口のバリア領域と入口のバリア領域が接しているような動作を行う。つまり、「ある同期ポイントに到達したプロセスは、全てのプロセッサが直前の同期ポイントを通過しなければ、その同期ポイントを通過することはできない」という動作を行う。

3.1.3. 重複化バリア

重複化バリア同期機構では、「面状&強制参加&全順序関係&オーバーラップ可能」なバ

リア同期モデルの動作を行う。重複化バリア同期機構の同期ポイントは入口同期ポイントと出口同期ポイントの2種類ある。入口同期ポイントは出口同期ポイントの前なら（同期が必要な依存関係を保証する範囲で）どこにでも設定して良い。出口同期ポイント間には順序関係があるので、その順序関係を満たすように同期ポイントを配置する。入口同期ポイントはポイント通過を同期コントローラに知らせる動作を行う。出口同期ポイントで、全ての入口同期ポイントが通過されるまで待つ。

3.2. Collective 通信

本同期コントローラでは、Synchronization Network を利用して、Collective 通信も行う。本同期コントローラがサポートする Collective 通信の種類は、Reduce, allreduce, Reduce-scatter, scan の4種類を用意する。

Synchronization Network を使用したデータ通信を行うため、独自に通信プロトコルを定める。

3.3. 本同期コントローラで行われるユーザーレベル通信

従来の通信では、システムコールを使用してデータの送受信が行われることにより、通信レイテンシが長くなる。つまり、送信時には、Write システムコールなどを利用して、カーネルがデータを送信し、受信時には、割り込みが発生しカーネルが受信処理をする。このカーネルの介在が通信レイテンシを長くさせる原因となる。

そこで、一般的には、ユーザーレベル通信を用いて、上記のレイテンシを低減させる[1][6][7]。しかし、ユーザーメモリ通信をサポートする場合、プロセスが同期コントローラに送出するアドレスは、仮想アドレスであり、同期コントローラが PCI を介して主記憶にアクセスする場合は、物理アドレスなので、

仮想アドレスから物理アドレスへの変換が必要となる。

アドレス変換を同期コントローラ上で行う場合、TLB などのページテーブルを持たなければならないので、メモリなどのハードウェアを追加する必要がある。

本同期コントローラでは、同期コントローラが PCI を介して主記憶にアクセスする機能は搭載せず、プロセスが直接同期コントローラにアクセスする機能のみを搭載する。

4. ボードの説明

同期コントローラボードは、PCI バスに挿入する Add-in ボードである。

同期コントローラボードには、アルテラの CPLD (Flex10k30)、通信・同期用ネットワーク、Responsive Processor[5]用コネクタが用意されている (図 2 参照)。

5. 同期コントローラの構成

同期コントローラの機能は、基本的にアルテラの CPLD によって実現する。アルテラの CPLD は、再構成可能なデバイスなので、後で必要になった機能は追加することが可能である。

現段階で CPLD に実装する機能は、PCI

バス制御機能、同期制御機能、データ通信制御機能、ネットワーク制御機能およびタイマ機能である (図 3 参照)。

5.1. PCI バス制御部

ノードと同期コントローラとの間でデータのやり取りを行うために、同期コントローラは PCI バスのマスタ/ターゲット機能を実装する。本同期コントローラでは、アルテラの PCI_a MageCore Function を使用して、PCI バスのマスタ/ターゲット機能を実装する。

PCI_a は、アルテラ Flex10K30 の約 54% (約 950 ロジックエレメント) のゲートアレイを使用する。残りの 46% (約 778 ロジックエレメント) で同期制御部、データ転送部、ネットワーク制御部、タイマ部を構成する。

また、Flex10K30 の EAB (Embedded array blocks : 多機能メモリブロック) は、約 1.5Kbyte ある。EAB を PCI 制御部、同期制御部、タイマ部、データ転送部、同期ネットワーク制御部と 5 分割にする。

同期コントローラでは、ターゲットメモリスペースにコマンドを割り当てる。コマンド実行時に、ノード内のプロセスがそのコマンドに対応したメモリスペースをアクセスする

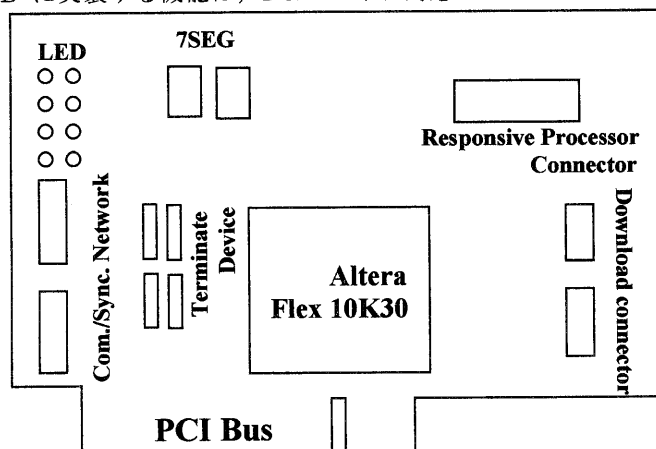


図2.同期コントロール基板

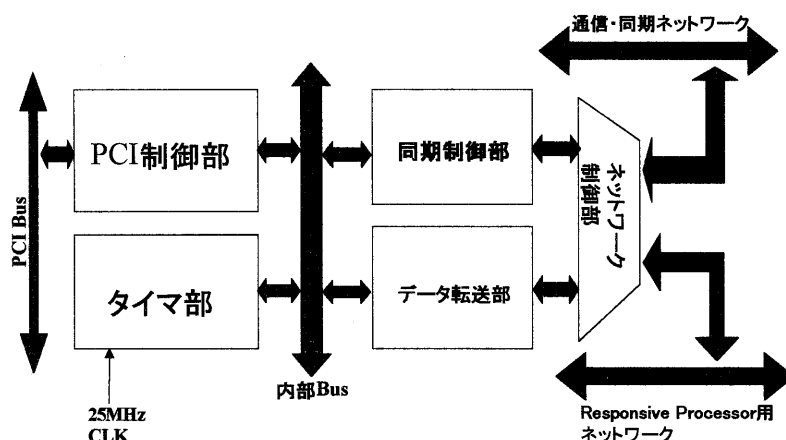


図3. CPLDの内部構成

ことにより、同期コントローラがコマンドを開始する。

同期処理完了を検出する方法として、以下の2種類の方法を用意する。

- 同期処理完了のフラグをターゲットメモリスペースに割り当てる。そのフラグをプロセスがチェックすることにより、同期処理終了を検出する。
- 同期コントローラが、PCIバスのINT A信号を利用して、プロセスに割り込みをかけることにより、プロセスが同期処理終了を検出する。

5.2. ネットワーク制御部

ネットワーク制御部は、通信・同期ネットワークと Responsive Processor 用ネットワークを制御する。

Responsive Processor[5]とは、機能並列分散リアルタイム制御用のプロセッサである。このプロセッサには、Responsive Linkと呼ばれる高速シリアル通信ネットワーク用のPHYやLLCが内蔵されている。将来、それを利用して、リモートメモリアクセスなどのより高機能的な処理も実現できる。

通信・同期ネットワークは、同期のための信号と通信のための信号を同時に流すことが

できない。そこで、ネットワーク制御部は、通信・同期ネットワークをデータ通信で使用するのかまたは同期処理で使用するのかを制御する。

ネットワーク制御部は、通信・同期ネットワークの信号線を通信の場合と同期の場合とで異なった用途に使用する。

5.2.1. 同期処理での通信・同期ネットワーク信号線の用途

同期処理では、図4に示すように、通信・同期ネットワークの信号線を定義する。各PEに1本ずつ同期ポイント到着信号線を用意し、バリア同期ポイント到着時に、ノードは自ノード信号線をアクティブにする。全てのノード信号線がアクティブになったら、同期完了フラグかまたは割り込みを発生させ、ノードに同期処理完了を知らせる。

5.2.2. 通信処理での通信・同期ネットワーク信号線の用途

通信処理では、図5のように、隣接ノード間を16ビット送受信データ線で結合する。隣接ノードに接続されている36ピンを送信データ線16本、受信データ線16本、送信制御線2本、受信制御線2本に分ける。

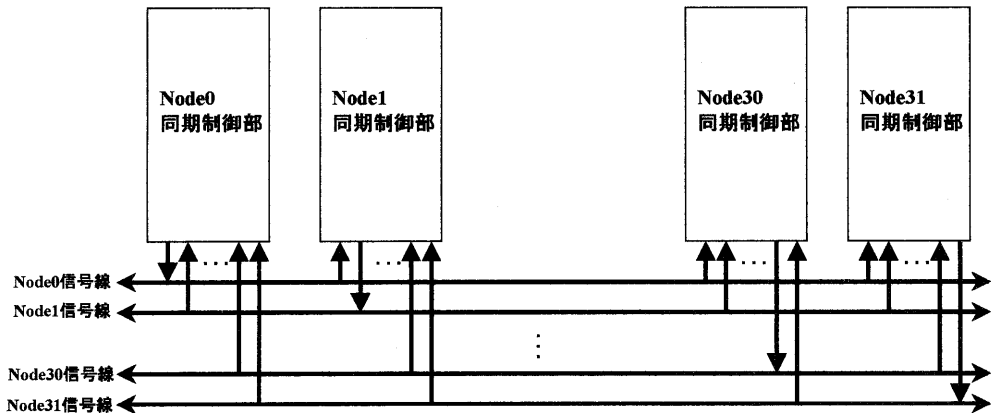


図4. 同期処理時のネットワーク

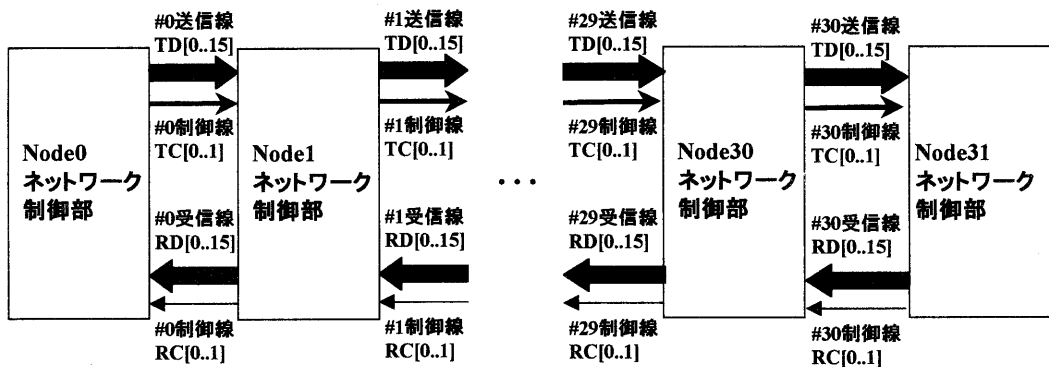


図5. 通信時の通信・同期ネットワーク

図 5. のネットワークを利用してデータ転送するために、パケットフォーマットを規定する。

通信・同期ネットワークを通信で使用するか同期で使用するか設定するために、ネットワーク制御部でネットワークレジスタを用意する。各ノードがそのネットワークレジスタを設定することにより、通信・同期ネットワークの用途を決定する。

5.3. 同期制御部

同期制御部では、バリア系同期機構の制御を行う。本同期コントローラで実装する同期

機構は、バリア同期機構、RBNQ 同期機構および重複化バリア同期機構の 3 種類である。

バリアの種類を設定するために、同期制御部内にレジスタを用意し、そのレジスタを設定することにより、バリア系同期の種類を決定する。

5.3.1. バリア同期機構制御部

バリアグループを設定し、そのグループ内でバリア同期を処理するようにする。

バリアグループを設定するために、各ノードの同期制御部に同期グループレジスタを用意する。同期制御部は、その同期グループレ

ジスタの値に従って、バリア同期を行う。

バリア同期機構の同期ポイントを示すバリア命令を用意する。また、バリア命令が発行されたことを示すフラグおよび同期終了を示すフラグをターゲットアドレススペースに割り当てる。

バリア命令が発行された時、バリア同期機構制御部で、図 5 の **node** 信号を送出し、バリアグループ内の全ての **node** が **node** 信号を送出した時に、同期終了を示すレジスタをセットする。

5.3.2. RBNQ 同期機構制御部

RBNQ 同期機構の同期ポイントを示す RBNQ 命令を用意する。また、RBNQ 命令が発行されたことを示すフラグおよび同期終了を示すフラグをターゲットアドレススペースに割り当てる。また、RBNQ 同期機構には、バリア参加情報と呼ばれるマスクビット情報が必要なので、その情報を保存するスペースを用意する。

RBNQ 同期機構制御部では、**node** 信号とバリア参加情報から RBNQ 同期の成立を検出する。同期が成立した時、同期終了を示すフラグをセットする。

5.3.3. 重複化バリア制御部

重複化バリアの同期ポイントを示す命令（入口同期命令、出口同期命令）を用意する。また、入口同期命令・出口同期命令が発行されたことを示すフラグおよび同期終了を示すフラグをターゲットアドレススペースに割り当てる。

5.4. データ通信部

データ通信部は、Collective 通信のデータ転送時に使用される。

データ通信部では、隣接ノードから送られてくるデータパケットの送受信を行う。データ転送路は、16 ビット双方向パラレルリン

クである。16 ビット単位でハンドシェイクを行いながら、データの送受信を行う。

データパケット容量は、パケットヘッダ 8Byte、ペイロード 32Byte の合計 40Byte にする。

データパケット転送プロトコルは、スプリットトランザクションを採用する。

トランザクションの種類は、Reduce, allreduce, Reduce-scatter, scan の 4 種類を用意する。

データ転送のプロトコルスタックとして、Physical Layer, Link Layer, Transaction Layer を規定する。

データ通信部は、プロトコルスタックの Physical Layer と Link Layer を実装し、Transaction Layer はプロセス内で処理する。つまり、データ通信部は、隣接ノード間のパケット送受信のみを行うこととし、パケットの解析および要求・応答パケットの送出はプロセス内で行われる。

データ通信部では、送受信 FIFO を用意し、プロセスが要求・応答送出パケットを送信 FIFO に投入することにより、パケットが送出される。受信 FIFO に要求・応答受領パケットが到着した場合、PCI バスの INT A の割込みをかけ、パケットを処理するプロセスにパケットが届いたことを通知する。

5.5. タイマ部

タイマ部では、25MHz の外部クロックを 2, 4, 8, 2, 16, 32 分周して、実行時間の計測を行う。外部クロックのカウンタとして、16 ビットの非同期カウンタを用意する。

タイマの値を保存する領域を EAB に設ける。これらの値は、ノードからの命令でまとめて吸い上げる。

タイマ部では、以下の機能を搭載する。

▶ タイマリセット

ノード単位でリセットをかけると同

時に、そのノードのタイマ値を保存している EAB もリセットする。

- ▶ タイマスタート
ノード単位でタイマをスタートさせる。
- ▶ タイマストップ
ノード単位でタイマをストップさせる。
- ▶ タイマラップ
ノード単位でタイマのラップ値を EAB に格納する。
- ▶ バリア&スタート
ノード全体のタイマスタート。同期制御部で全てのノードに対してバリア同期を行い、同期が成立した直後にタイマをスタートさせる。
- ▶ バリア&ラップ
ノード全体のタイマラップ。同期制御部で全てのノードに対してバリア同期を行い、バリアを行う直前のラップタイムと同期が成立した直後のラップタイムを各ノードの EAB に保存する。
- ▶ ラップタイムリード
各ノードが EAB に保存したラップタイムを読み込む。

6. おわりに

プログラマブルロジックデバイス (CPLD) を用いることにより、同期や Collective 通信のオーバーヘッドを低減し、メッセージパッシングを利用した並列処理をサポートする同期コントローラについて述べた。

同期コントロールボードは、メッセージパッシング用の同期や Collective 通信に限定しているため、比較的部品点数を少なく構成することができる。

今後は、数台程度のワークステーションクラスタを使用して、本同期コントローラの性能評価を行う予定である。

参考文献

- [1] 工藤知宏, 横山知典, 周東福強, 清水敏行, 天野秀晴, “Network based Parallel Computing のための Network Interface の評価” 信学技法, CPSY98-60, pp.1-8, 1998.
- [2] Nanette J.Boden, Danny Cohen, Robert E.Felderma, Alan E.Kulawik, Charles L.Seitz, Jakov N.Seizovic, and Wen-King Su, “Myrinet – A Gigabit-per-Second Local-Area Network ” IEEE MICRO, Feb. 1995.
- [3] Jenq-Shyan Yang and Chung-Ta King, “Designing Tree-Based Barrier Synchronization on 2D Mesh Networks” IEEE Trans. Parallel and Distributed System, Vol.9, No.6, 1998.
- [4] 高木浩光, 有田隆也, 曾和将容, “重複可能なバリア型同期のためのスケジューリングアルゴリズムとその性能” 信学技法 CPSY91-15, pp.91-98, 1991.
- [5] 山崎信行, 松井俊浩, “機能並列分散リアルタイム制御用レスポンス・プロセッサの設計” 第 15 回日本ロボット学会学術講演会, No.3, pp.1027-1028, 1997.
- [6] 松本尚, 平木敬, “100BaseTX によるメモリベース通信の性能評価” 信学技法 CPSY97-50, pp.109-116, 1997.
- [7] 手塚宏史, 堀敦史, 石川裕, “ワークステーションクラスタ用通信ライブラリ” 並列処理シンポジウム JSPP'96 論文集, pp.41-48, 1996.
- [8] 山家陽, 村上和彰, “バリア同期モデル –Taxonomy と新モデルの提案” 並列処理シンポジウム JSPP'93 論文集, pp.119-126, 1993.
- [9] 早川潔, 本多弘樹, “ブロッキングを利用した同期方式における冗長同期コード削減手法の検討” 情報処理学会研究報告, 98-ARC-125-4, pp.19-24, 1997.