

メモリーマルチプロセッサ一体型 ASSP「PPRAM」用 標準通信インタフェース『PPRAM-Link Standard』 Draft 0.0 の概要

村上 和 彰^{††} 岩下 茂 信[†] 宮 嶋 浩 志[†]

本稿は、メモリーマルチプロセッサ一体型 ASSP「PPRAM」向けの標準通信規格『PPRAM-Link』Draft 0.0 について、その概要を紹介している。PPRAM-Link は PPRAM ノードならびに (PPRAM ノードを 1 個以上含む) PPRAM チップを 1 対 1 単方向パラレル/シリアル・リンクで接続し、1G バイト/秒 (パラレル・リンク当り) あるいは 1G ビット/秒 (シリアル・リンク当り) 以上の高速データ転送を可能とする。その論理階層の通信プロトコルは、基本的には SCI (IEEE Std 1596-1992) に基づいて PPRAM 向けに修正を施したもので、トランザクション、フロー制御、エラー検出、初期化の各プロトコルを含む。ただし、SCI と異なりハードウェア・レベルでキャッシュ・コヒーレンスを保証しないので、SCI のキャッシュ・コヒーレンス・プロトコル相当のプロトコルは含まない。PPRAM-Link の標準化作業は、PPRAM コンソーシアム (仮称) の正式発足 (1996 年 10 月の予定) を待って開始し、まずは 1 年間の予定で『PPRAM-Link Standard (仮称)』Draft 1.0 を策定する計画である。

Overview of the Draft Standard for PPRAM Interface (PPRAM-Link)

KAZUAKI MURAKAMI,^{†,††} SHIGENOBU IWASHITA[†]
and HIROSHI MIYAJIMA[†]

This paper outlines an unapproved draft standard for PPRAM-Link. The PPRAM-Link provides a high-bandwidth interface needed for communicating among two or more PPRAM chips, or merged-memory/logic LSIs, by using a collection of fast point-to-point unidirectional links. Based on SCI (IEEE Std 1596-1992) signaling technology, the PPRAM-Link would be defined at 1 Gbyte/s (16-bit parallel) and 1 Gbit/s (serial). The PPRAM-Link supports the SCI-like transaction and flow-control protocols, although it does not provide any cache coherence protocols since PPRAM-based system is NCC-NUMA. The standardization will begin soon, and a proposed standard for PPRAM-Link will complete in a year.

1. はじめに

本稿は、メモリーマルチプロセッサ一体型 ASSP「PPRAM」向けの標準通信規格『PPRAM-Link』Draft 0.0⁴⁾の内容を抜粋したものである。

PPRAM (Parallel Processing Random Access Memory) とは、DRAM-プロセッサ混載 LSI ならびに並列処理時代における新しいコンピュータ・システム構成法のための構成要素であり、

- 大容量メモリ: DRAM を始めとして, SRAM,

Flash EEPROM, 等を単独あるいは組合せて構成

- マルチプロセッサ: 汎用プロセッサ, 特殊用途向けプロセッサ, FPGA, 等の 0 個以上のプロセッサ/ロジックから構成されるホモジニアスまたはヘテロジニアスなマルチプロセッサ
- 通信: PPRAM チップ内のプロセッサ間通信, および, PPRAM チップ間通信を「ある標準の通信インタフェースおよびプロトコル」に準拠して制御

を 1 チップに集積した ASSP (Application-Specific Standard Product) である。

PPRAM チップの開発者は、「標準」である通信には触らずに、対象とするアプリケーションに応じて、「内部アーキテクチャ」、すなわち、

- メモリの種類/容量/バンド巾
- プロセッサの数/機能/性能

[†] 九州大学 大学院システム情報科学研究科 情報工学専攻
Department of Computer Science, Kyushu University
ppram@c.csce.kyushu-u.ac.jp

http://kasuga.csce.kyushu-u.ac.jp/~ppram

^{††} PPRAM コンソーシアム設立準備会

PPRAM Consortium: Launch Working Group

を決めればよい。たとえば、我々は、内部アーキテクチャとして以下の特徴を有する PPRAM[®] (Reference PPRAM) を提案している[☆]。

- 大容量 DRAM + シンプル・マルチプロセッサ：まずその時代で入手可能な最大容量の DRAM を搭載、次にそれが提供するメモリ・バンド巾に見合っただけの性能 (単体性能×数) のプロセッサを搭載する。
- 分散メモリ型オンチップ・マルチプロセッサ：本質的に非常に高いオンチップ・メモリ・バンド巾を活用するために DRAM を各プロセッサにローカル・メモリとして分散配置し、さらにローカル・メモリに行バッファを複数個設けてこれをキャッシュとして活用する。
- 共有グローバル・レジスタ型オンチップ・マルチプロセッサ：さらに、本質的に低いチップ内通信レイテンシを活用するために、共有グローバル・レジスタ・ファイルを設けてチップ内プロセッサ間通信/同期を行う。

また、PPRAM の上記の定義から明らかなように、たとえば、「プロセッサを備えない DRAM だけの PPRAM」も存在し得る (これは、Rambus DRAM や SyncLink DRAM に相当する)。

一方、システム開発者は、対象とするアプリケーションに応じて必要な PPRAM チップの種類/数を決定し、それらを「標準インタフェース」に基づいて相互結合すればよい。

上述の LSI およびシステム開発法を可能とし、その普及を図るには、複数ベンダーの PPRAM チップ同士の相互接続性 (interconnectivity)、相互運用性 (interoperability)、および、同システム上のソフトウェアの可搬性 (portability) を保証する必要がある。すなわち、上記の「ある標準の通信インタフェースおよびプロトコル」が「標準」でなければならない所以はここにある。

そこで、PPRAM の通信インタフェースおよびプロトコル (これらをまとめて PPRAM-Link と総称する) の標準化を目的に、PPRAM コンソーシアム (仮称) を 1996 年 10 月を目処に発足する予定である。当コンソーシアムは、まずは 1 年間の予定で『PPRAM-Link Standard (仮称)』Draft 1.0 を策定する。標準化の対象 (scope) は、以下の通り (図 1 参照)。

- 物理階層：
 - － チップ内：ベンダー依存とし、標準には含まない。
 - － ボード内チップ間、および、ボード間：パラレル・リンク当り 1G バイト/秒以上、あるい

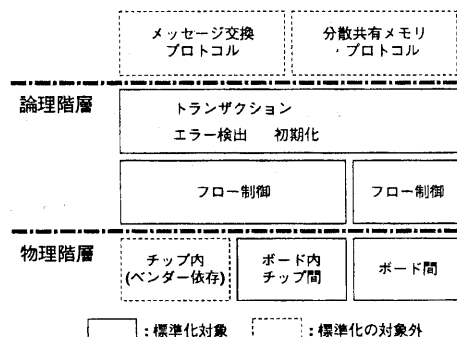


図1 標準化の対象

は、シリアル・リンク当り 1G ビット/秒以上の高スループット転送を可能とする 1 対 1 単方向リンクに関する電氣的/機械的仕様、等を定める。これには、パラレル/シリアルの違いに加えて、複数の仕様が存在し得る。

- 論理階層：複数の仕様が存在し得る物理階層とは独立に、論理的な 1 式の通信プロトコルを定める。これは、次のものから成る。

- － トランザクション・プロトコル
- － フロー制御プロトコル
- － エラー検出プロトコル
- － 初期化プロトコル

上記の各プロトコルは、SCI (Scalable Coherent Interface: IEEE Std 1596-1992)⁶⁾の対応するプロトコルを基に、PPRAM 向けに変更を施したものである。SCIにはあとキャッシュ・コヒーレンス・プロトコルが含まれているが、PPRAMはCC-NUMA (Cache Coherent, NonUniform Memory Access) のSCIとは異なりNCC-NUMA (Non Cache Coherent NUMA) であることから、当該プロトコルは不要である。なお、NCC-NUMA上でもソフトウェアの支援により、コヒーレントな共有メモリをユーザ・プログラムに提供可能である³⁾が、そのプロトコルは標準化の対象外である。

以下、PPRAM コンソーシアムでの今後の標準化作業のたたき台となる Draft 0.0 について、その論理プロトコルを中心に紹介する。

2. PPRAM ベース・システム構成

PPRAM の PE (Processing Element) は、

- 0 バイト以上のローカル・メモリ
- 0 個以上のプロセッサ/ロジック
- 1 個以上のネットワーク・インタフェース

から構成される。PEはSCIのノード (node) に相当し、以下の記述ではPEとノードを同じ意味で用いる。PPRAM ノードは、図2に示すようにモデル化さ

[☆] 以前の文献^{1), 2)}では、この PPRAM[®]のことを PPRAM と呼んでいた。現在の PPRAM の定義は本稿で述べた通り、より広義になっている。

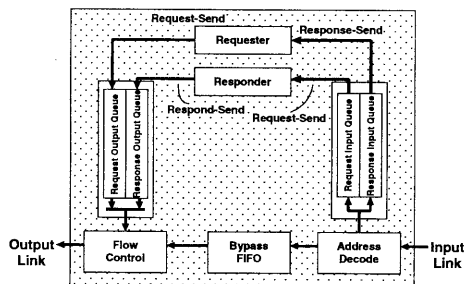


図2 PPRAM ノード・モデル

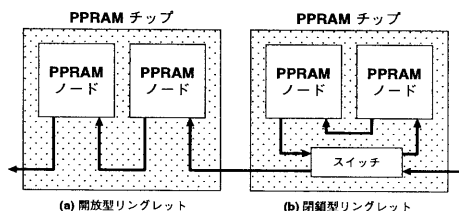
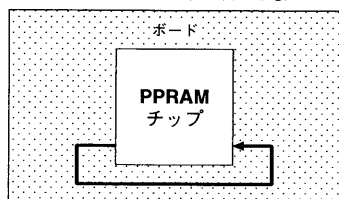
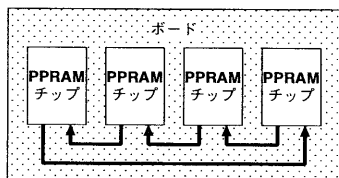


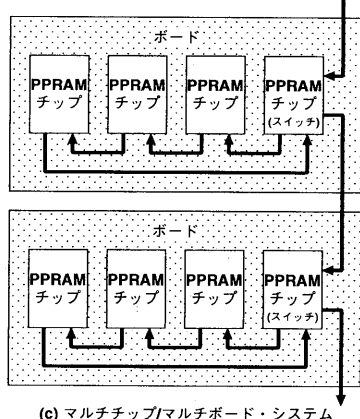
図3 チップ内相互結合形態



(a) シングルチップ・システム



(b) マルチチップ/シングルボード・システム



(c) マルチチップ/マルチボード・システム

図4 PPRAM ベース・システムの構成例

れる。

1 個の PPRAM チップは、1 個以上の PPRAM ノードを含む。2 個以上の PPRAM ノードを含む場合、それらは PPRAM-Link で相互結合されるが、そのチップ内結合形態として次の 2 通りが可能である。

- 開放型リングレット：リングレット (ringlet) ☆がチップ内で閉じていない結合形態 (図 3(a) 参照)。
- 閉鎖型リングレット：リングレットがチップ内で閉じている結合形態 (図 3(b) 参照)。チップ内リングレットとチップ間リングレットとはスイッチにより接続する。

PPRAM ベース・システムは、1 個以上の PPRAM チップから構成される。2 個以上の PPRAM チップから成る場合、それらは PPRAM-Link で相互結合される。図 4 に PPRAM ベース・システムの構成例を示す。なお、PPRAM ベース・システムが複数のプリント基板 (PCB) から成る場合、PCB 間はスイッチを介した PPRAM-Link による結合、あるいは、ブリッジを介した SCI 等の他の相互結合網による結合の双方が可能である。

PPRAM ベース・システムは SCI 同様、CSR (Control and Status Register) Architecture (IEEE Std 1212-1991)⁵⁾ が定めた「64 ビット・アドレス空間モデル」を用いる。ただし、ノード識別子 (nodeId) が 64 ビット・アドレスの上位 32 ビット (SCI では 16 ビット)、ノード内オフセットが下位 32 ビット (SCI では 48 ビット) と変更している。これにより、PPRAM ベース・システムは最大で 4G 個 (SCI では 64K 個) のノードから構成される。また、各ノードの最大メモリ容量は 4G バイト (SCI では 256T バイト) となる。システム全体として、64 ビット・アドレス長の単一かつグローバルな物理アドレス空間を提供する。

3. トランザクション・プロトコル

3.1 トランザクション、サブアクション、パケット

2 個のノード間のデータ交換をトランザクション (transaction) と呼ぶ。1 個のトランザクションは SCI 同様、一般に次の 2 個のサブアクション (subaction) から成る。

- (1) 要求サブアクション (request subaction)：要求側ノード (requester) から応答側ノード (responder) へ要求を送る。
- (2) 応答サブアクション (response subaction)：応答側ノードから要求側ノードへ応答を返す。

さらに、1 個のサブアクションは、一般に次の 2 個のパケット (packet) から構成される (図 5 参照)。

- (1) 送出パケット (send packet)：サブアクションのタイプに応じて、次の 2 種類がある。

☆ SCI の定義で、あるノードの出力リンクから同一ノードの入力リンクに至る閉路。

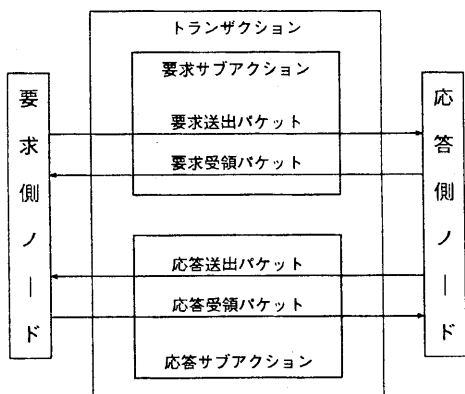


図5 トランザクション、サブアクション、バケット

- 要求送出バケット (request-send packet): 要求側ノードから応答側ノードへ、要求を送出するのに用いる。
 - 応答送出バケット (response-send packet): 応答側ノードから要求側ノードへ、応答を送出するのに用いる。
- (2) 受領バケット (echo packet): サブアクションのタイプに応じて、次の2種類がある。
- 要求受領バケット (request-echo packet): 応答側ノードから要求側ノードへ、要求を受領したことを通知するのに用いる。
 - 応答受領バケット (response-echo packet): 要求側ノードから応答側ノードへ、応答を受領したことを通知するのに用いる。

3.2 バケット・フォーマット

バケットを構成する不可分な単位は、16ビット長のシンボル (symbol) である。バケット長は最小8シンボル (16バイト) で、常に8シンボル (16バイト) の倍数となる。バケット・フォーマットには、バケットの種類に応じて次の3種類がある。

- (1) 要求送出 (request-send) バケット・フォーマット (図6参照)
 - (2) 応答送出 (response-send) バケット・フォーマット (図7参照)
 - (3) 受領 (echo) バケット・フォーマット (図8参照)
- 各フィールドの意味は基本的に SCI のそれに準ずる。なお、ノード識別子が SCI の16ビット (1シンボル) から32ビット (2シンボル) へと増加したことに伴って、以下の措置を採っている。

- ターゲット・ノード識別子 (targetId) は、32ビットすべてがヘッダ内に常に置かれる。
- ターゲット・ノード識別子の上位16ビットとソース・ノード識別子 (sourceId) の上位16ビットとが一致しているか否かをコマンド・フィールド内で示す。
- 一致している場合は、ソース・ノード識別子の上

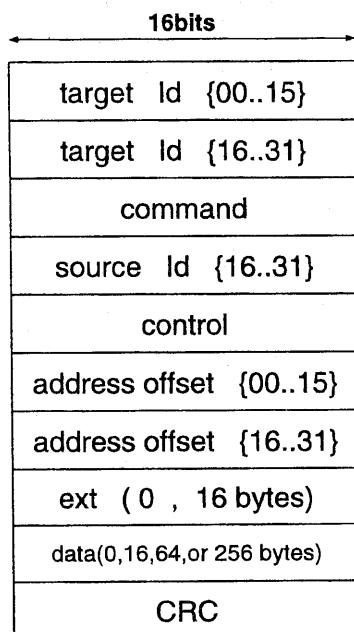


図6 要求送出バケット・フォーマット

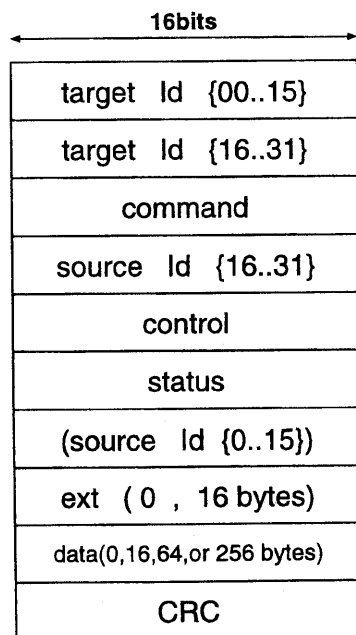


図7 応答送出バケット・フォーマット

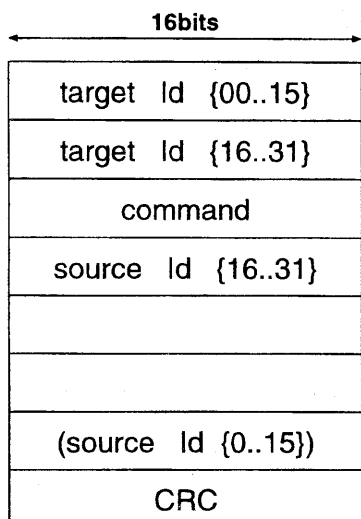


図8 受領パケット・フォーマット

位 16 ビットはパケット内に置かない。

- 一致していない場合は、以下の所定フィールドにソース・ノード識別子の上位 16 ビットを置く。
 - － 要求送出パケットの場合：拡張ヘッダ内
 - － 応答送出パケットおよび受領パケットの場合：ヘッダ内

3.3 トランザクション・タイプ

対ソフトウェア・インタフェースとして以下の通信プリミティブを提供するプロセッサおよびコントローラを対象に、PPRAM-Link のトランザクション・セットを定める。

- ローカル・メモリへのアクセスと同様の、リモート・メモリに対するメモリ・アクセス (ただし、キャッシュ・コヒーレンスを保証しないノンコヒーレント・アクセス)
 - － LOAD/STORE 命令によるリモート・メモリへの通常のアクセス
 - － Fetch&Add 命令などによるリモート・メモリへの不可分アクセス
 - ローカル・メモリとリモート・メモリとの間での非同期 DMA 転送[☆]
 - リモート・メモリのある一定サイズの領域に対しての、選択的ワード書込み (ソフトウェアでメモリ・コヒーレンスを保証する際、変更済みワードのライト・バックのために使用³⁾)
 - Active Messages によるメッセージ交換
- トランザクションには現在、次の 6 種類がある (図

9 参照)。

- (1) 読出し (reads, read64, read256) : 指定されたアドレスから指定されたサイズ S ($1 \leq S \leq 16, 64, 256$ バイト) だけ読み出す。SCI の “noncoherent memory read (nread)” に相当する。
- (2) 書込み (writes, write64, write256) : 指定されたアドレスに指定されたサイズ S ($1 \leq S \leq 16, 64, 256$ バイト) だけ書き込む。SCI の “noncoherent memory write (nwrite)” に相当する。
- (3) 選択ワード書込み (writesw64, writesw256) : 指定されたアドレスから始まる指定サイズ (64, 256 バイト) の範囲において、選択されたワード (4 バイト) に対してのみ (最小 0 ワードから最大 16/64 ワードまでの任意数) 書込みを行なう。SCI には存在しない PPRAM-Link 独自のトランザクション。選択ワードの指定フィールド (16/64 ビット) を必要とするので、要求送出パケットは必ず拡張ヘッダを伴う。
- (4) 移動 (moves, move64, move256) : 指定されたアドレスに指定されたサイズ S ($1 \leq S \leq 16, 64, 256$ バイト) だけ書き込む。「書込み」とは異なり、応答サブアクションは伴わない。SCI の “move” と同等。
- (5) ロック (locks) : 指定されたアドレスから指定されたサイズ S ($1 \leq S \leq 16$) だけ読み出すと同時に、それを用いて指定された不可分操作を施して同一ロケーションに書き込む。SCI の “locks” と同等。
- (6) イベント (event00, event16, event64, event256) : 受領パケットも応答サブアクションも伴わない特殊なトランザクション。SCI 同様、event00 (clockStrobe) しか現在では定義していない。

4. フロー制御プロトコル

SCI⁹⁾のフロー制御プロトコルは以下の 2 つから成っており、“deadlock free” および “starvation free” を保証している。

- バンド巾割当て (bandwidth allocation) プロトコル
- キュー割当て (queue allocation) プロトコル

PPRAM-Link でも同様のフロー制御プロトコルを定義する。ただし、“fair/unfair” に関しては “fair” のみに限定し、プロトコルを簡略化している⁴⁾。

5. おわりに

以上、PPRAM 標準通信インタフェース PPRAM-Link の Draft 0.0⁴⁾を概観した。

PPRAM-Link は PPRAM ノードならびに (PPRAM ノードを 1 個以上含む) PPRAM チップを 1 対 1 単方向パラレル/シリアル・リンクで接続し、1G バイト/

[☆] PPRAM ベース・システムでは、従来の I/O コントローラも PPRAM ノードとしてシステムに組み込まれる。よって、I/O のための DMA 転送もメモリ間 DMA 転送と同じ扱いになる。

要求	応答
readxx	ヘッダ
writexx	ヘッダ
writeswxx	ヘッダ
movexx	ヘッダ
locksx	ヘッダ
eventxx	ヘッダ

図 9 トランザクション・タイプ

秒 (パラレル・リンク当り)あるいは1Gビット/秒 (シリアル・リンク当り)以上の高速データ転送を可能とする。その論理階層の通信プロトコルは、基本的にはSCIに基づいてPPRAM向けに修正を施したもので、トランザクション、フロー制御、エラー検出、初期化の各プロトコルを含む。ただし、SCIと異なりハードウェア・レベルでキャッシュ・コヒーレンスを保証しないので、SCIのキャッシュ・コヒーレンス・プロトコル相当のプロトコルは含まない。

今後のPPRAM-Linkの標準化作業の予定は、以下の通り。

- (1) 1996年9月：第2回PPRAMコンソーシアム設立準備会。
- (2) 1996年10月：PPRAMコンソーシアム正式発足。その後、月に1回の頻度で標準化作業部会を開催。
- (3) 1997年10月：『PPRAM-Link Standard (仮称)』Draft 1.0完了。

PPRAMコンソーシアム、および、同設立準備会に関する問い合わせは下記まで。

ppram@c.sce.kyushu-u.ac.jp

謝辞 日頃から御討論頂く、九州大学 大学院システム情報科学研究科 安浦寛人教授、岩井原瑞穂 助教授、PPRAMプロジェクト・メンバ、安浦・村上・岩井原研究室の諸氏、ならびに、PPRAMコンソーシアム設立準備会の会員諸氏に感謝致します。

参 考 文 献

- 1) 岩下茂信, 宮嶋浩志, 村上和彰, “次々世代汎用マイクロプロセッサ・アーキテクチャPPRAMの概要,” 情処研報, ARC-113-1, 1995年8月。
- 2) 村上和彰, 岩下茂信, 宮嶋浩志, 白川 暁, 吉井 卓, “メモリ・マルチプロセッサ型 ASSP (Application-Specific Standard Product) アーキテクチャ: PPRAM,” 信学技報, ICD96-13, CPSY96-13, FTS96-13, 1996年4月。
- 3) 村上和彰, 吉井 卓, 岩下茂信, 宮嶋浩志, “PPRAMベース・システム向け分散共有メモリ・

システムの提案,” 情処研報, OS-73-2, 1996年8月。

- 4) PPRAM-Link Standard Draft 0.0, (to appear in) 第2回PPRAMコンソーシアム設立準備会, 1996年9月。
- 5) IEEE Std 1212-1991, IEEE Standard Control and Status Register Architecture for Microcomputer Buses, IEEE, 1991。
- 6) IEEE Std 1596-1992, IEEE Standard for Scalable Coherent Interface (SCI), IEEE Computer Society, 1992。