

## PCI Expressによる省電力・高信頼・高性能通信 リンクのためのコミュニケータチップ: PEACH

埜 敏博<sup>†1,†2</sup> 朴 泰祐<sup>†1,†2</sup> 三浦 信一<sup>†2</sup>  
佐藤 三久<sup>†1,†2</sup> 有本 和民<sup>†3</sup>

我々は、組み込みシステムに適したディペンダブル省電力高性能通信機構として、PCI Express を用いた通信リンク PEARL を提案している。本論文では PEARL を実現するためのコミュニケータチップ、PEACH チップの概要、機能について述べる。PEACH チップは、4レーンの PCI Express Gen2 を 4ポート持ち、4コアの M32R プロセッサ、DMA コントローラを内蔵する。これらは SuperHyway バスによって結合され、高速動作と柔軟な制御を両立する。PEACH チップを搭載したネットワークインタフェースカードとして PEACH ボードを実現することにより、リンク当たり 2GB/s の理論ピーク転送性能を持つ省電力ディペンダブル通信機構を実現する。

### Communicator Chip for Power-aware, Dependable, and High-performance Communication Link Using PCI Express: PEACH

TOSHIHIRO HANAWA,<sup>†1,†2</sup> TAISUKE BOKU,<sup>†1,†2</sup>  
SHIN'ICHI MIURA,<sup>†1</sup> MITSUHISA SATO<sup>†1,†2</sup>  
and KAZUTAMI ARIMOTO<sup>†3</sup>

We have proposed a power-aware, high-performance, and dependable communication link for embedded systems using PCI Express, named PEARL. In this study, we describe the overview, structure, and function of a communicator chip for realizing the PEARL, named PEACH. The PEACH chip has 4 ports of PCI Express Gen2 with 4 lanes, and employs M32R processor with 4 cores and DMAC. These components are connected by the SuperHyway bus, which provides both high speed and flexible control. The PEACH board will be built as the network interface card with the PEACH chip, and it provides the power-aware and dependable communication link theoretical peak performance of which is 2GB/s per link.

### 1. はじめに

デジタル家電やカーナビゲーションシステムのような組み込みシステムでは、機能の複雑化、扱う情報の大規模化などにより、年々高い性能が求められている。その反面、熱対策、環境への配慮などから、組み込みシステムには一層の消費電力削減が求められている。そこで、これまでサーバやデスクトップ向けのプロセッサに導入されてきたマルチコアが、組み込み向けにも広く使われるようになってきている。例えば、ルネサステクノロジ社の SH2A-DUAL や SH4A-MULTI, ARM 社の MPCore, Freescale 社の QorIQ など、組み込み向けのマルチコアプロセッサが登場し、高性能化と消費電力削減とを両立させている。しかしながら、組み込み向けのマルチコアは現状で高々2~4 コア程度であり、より高い性能が必要な場合には、マルチコアプロセッサをネットワークで結合したマルチプロセッサシステムに変わっていくと考えられる。

一方、組み込みシステムには、高い信頼性が必要とされるだけでなく、可用性や耐故障性などのディペンダビリティが求められる。ディペンダビリティを満足するためにはシステムに冗長性を持たせる必要があるが、マルチコア、マルチプロセッサシステムであれば、複数コアや複数ノードにより互いの動作を補完し合うことで冗長性が実現できる。このとき、ディペンダブルなマルチプロセッサシステムを考えると、プロセッサやノードだけでなく、通信機構におけるディペンダビリティも必要不可欠である。通信中のパケットの完全性や到達性を保証し、通信のためにも複数リンクを用意し障害に応じて構成を切り替えたり、通信経路を迂回させることによって通信機構におけるディペンダビリティを実現することができる。さらに、マルチプロセッサシステムにおいてノード間の通信性能は重要であるが、さらにチェックポイントデータや動作ログの転送や、動作ログの収集などを考えると、十分な転送性能が必要であり、経路迂回時にはさらに多くの通信が集中するため、性能の余裕も求められる。

我々は、これまでに述べた、高い通信性能とディペンダビリティとを両立できるような

†1 筑波大学大学院 システム情報工学研究科

Graduate School of Systems and Information Engineering, University of Tsukuba

†2 筑波大学 計算科学研究センター

Center for Computational Sciences, University of Tsukuba

†3 ルネサステクノロジ

Renesas Technology

通信機構として、PCI Express を用いた省電力・高信頼・高性能通信リンク PEARL (PCI Express Adaptive & Reliable Link) を提案している<sup>1),2)</sup>。PCI Express(以降、PCIe と略す) は、PC と周辺機器を接続するための高速なシリアル I/O インタフェースである<sup>3)</sup>。PEARL の通信リンクには PCI Express Gen2 x4 レーンを用いて、高い転送性能を実現するとともに、数 m 程度の近距離通信に限定し、スイッチを使わずノード間をリンクで直結して構成することにより、既存のネットワークに比べて消費電力を低減することができる。また、PCI Express の使用レーン数を減らしたり、リンク速度を半分に落とすことにより、省電力化が可能である。さらに、PCI Express の規格により、エラー検出、フロー制御、再送制御がハードウェアレベルでサポートされており、信頼性の高い通信が可能である。

本研究では、通信リンク PEARL を実現するコミュニケータチップ PEACH (PCI Express Adaptive Communication Hub) について、その概要と設計について述べる。

## 2. 既存の通信リンクと PCI Express によるノード間通信

ここでは、既存の通信リンクについて述べ、性能、信頼性、消費電力について議論する。従来から、クラスタ向けの高性能かつ信頼性の高いネットワークとして、InfiniBand や Myrinet がよく使われている。中でも、Infiniband は低遅延、高バンド幅であり、2 $\mu$ s 程度の遅延で、InfiniBand DDR 4x では、20Gbps の転送性能(実効性能 2GB/s)を持つ<sup>4)</sup>。また、Subnet Manager を用いて、故障が起こっても自動的に故障から回復することも可能である<sup>5)</sup>。しかし、コントローラチップの消費電力は、1 ポート当たり 3~5W 程度である。さらに、3 ノード以上を互いに接続するためにはスイッチが必要であり、スイッチも同様に 1 ポート当たり 3~5W 程度の消費電力を要する。

一方、安価なネットワークである Gigabit Ethernet が組込み向け、またクラスタ向けにも使われることが多い。Gigabit Ethernet のコントローラチップの消費電力は 1 ポート当たり 1~1.5W 程度であり、InfiniBand や Myrinet と比べれば小さい。しかし、3 ノード以上を接続するためには、InfiniBand と同様別途スイッチが必要であり、数 W 程度の消費電力を必要とする。そもそも Ethernet は 100m 程度の伝送距離を対象としたネットワークであり、そのため転送速度に比べて消費電力が比較的大きくなってしまふ。遅延時間も 10 数  $\mu$ s と比較的大きい。

我々は以前より、Gigabit Ethernet をマルチリンクにすることによって、通常時は性能向上を実現し、冗長性により耐故障性も持つ RI2N(Redundant Interconnection with Inexpensive Network) を提案している<sup>6)</sup>。しかしながら、複数リンクを利用するためには、各

ノード当たりのポート数、スイッチ数も増加することになり、消費電力も増加してしまう。また、Ethernet では、リンク自体の信頼性は保証されていないため、本質的にパケットロスを防ぐことはできない。従って、TCP/IP における TCP のように、上位層によってパケットの再送処理、順序制御やフロー制御を行う必要がある。

他にも、組込み向けネットワークとして、車載用ネットワークとして良く用いられている Controller Area Network (CAN)<sup>7)</sup> や FlexRay<sup>8)</sup> なども存在する。これらは、高信頼性、低消費電力、ノード接続の容易さや低コストである点についても考慮されているが、通信速度は高々数 Mbps~数十 Mbps 程度であり、比較的高性能な組込み機器を考えたときに性能面で十分ではない。

そこで我々は、ノード間の通信リンクに PCI Express (PCIe) を用いることを検討した。PCIe は、PCI, PCI-X バスに代わる、PC と周辺機器を接続するための高速なシリアル I/O インタフェースで、PCI-SIG により標準化が行なわれている<sup>3)</sup>。現在では PC に搭載される事実上の標準 I/O インタフェースとなっており、様々なデバイスが PCI Express に対応している。PCIe では本来ホスト側に当たる Root Complex と、周辺デバイス側に当たる Endpoint の間で、メモリ読み出し/書き込みやメッセージ送受信などを行う。しかし、実際の動作としては 2 点間で双方向のパケット転送を行っているに過ぎないため、これをノード間の通信そのものに応用することを考えた。

上で述べた Infiniband や Gigabit Ethernet など、全てのネットワークは、ホストから見ると結局 PCIe 経由で通信することになる。また、Infiniband や RI2N では複数リンクを搭載し、リンク毎に制御することで性能に応じて電力の最適化を行うことは可能であるが、リンク単位で PCIe より先のドメインだけが制御の対象となる。それに対して、PCIe を直接通信に使用し、制御プロセッサにより PCIe のレーン数やレーン速度を柔軟に制御することができれば、より広いドメインで詳細に効果的な電力・性能制御を行うことが可能になる。

PCI Express を拡張し、多数のプロセッサおよび I/O 間を相互接続する ASI (Advanced Switching Interconnect) という規格も存在する<sup>9)</sup>。当初は組込み向けに開発されてきたが、実際には高性能クラスタを念頭においた仕様になっており、ハードウェアも複雑である\*1。また、複数ノードを接続するためには専用のスイッチが必要である。

一方、PCIe スイッチに、Non transparent bridge (NTB) という特殊な機能を持たせることにより、ノード間の通信を可能にする技術も存在している<sup>10)</sup>。PCIe では、論理的に Root

\*1 ASI 規格を制定した ASI-SIG は 2007 年に解散し、ASI 規格は PICMG に譲渡されている。

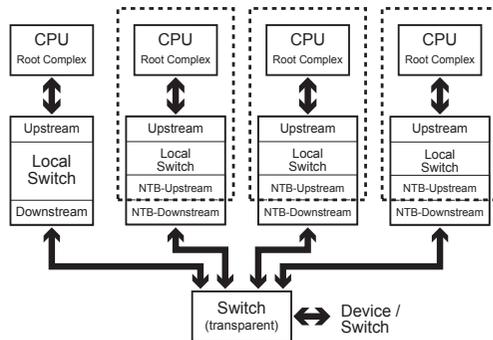


図 1 Non transparent bridge を用いたクラスタ構成例  
Fig. 1 Example of cluster using Non transparent bridge

Complex が根、Endpoint が葉、スイッチが節となる木構造を形成し、アドレス空間を共有するため、Root Complex であるプロセッサを複数存在させることはできない。そこで PCIe における NTB では、PCIe スイッチの下流ポート部分にブリッジ機能を追加し、その下流ポートからのアクセスと、そのポート以外からのアクセスとで異なった別の Endpoint として振る舞い、これらの Endpoint 間でアドレス変換を行うことにより、異なる 2 つの Root Complex との間、つまりプロセッサ間での通信を実現する。

NTB を用いたクラスタの例を図 1 に示す。図中の点線で囲まれた部分は、異なるアドレス空間であることを表す。このように NTB を用いた場合、システム全体では特定のノードを Root とする木構造になり、残りのノードは NTB 機能を持つスイッチを用いて接続することになる。しかし、NTB は PCIe の仕様ではなく、PCIe スイッチチップのベンダがそれぞれに独自の実装を行っているため、互換性がない。さらに、Root に接続されたノードが故障した場合には、別のノードが Root に代わる必要があり、各スイッチの設定も変更しなければならない。そのためノードの再起動が必要になり、ディペンダブルな通信機構として適切ではない。

### 3. PCI Express による通信リンク PEARL

2 節における議論を元に、我々は PCIe を用いた省電力高性能ディペンダブル通信機構 PEARL (PCI Express Adaptive and Reliable Link) を提案してきた<sup>1);2)</sup>。

2 節で述べたように、PCIe は本来デバイスを接続するための規格であるが、本質的には

Root Complex (RC) と Endpoint (EP) との間で双方向の packet 転送を行っているのみであり、初期化や割り込み関連の処理以外はそれぞれの動作や構造に大きな差はない。そこで PEARL においては、各ノードにコミュニケータチップ PEACH を配置し、PEACH 間を PCIe リンクで結ぶことによってノード間通信を実現する。PEACH の各ポートは RC と EP を切り替えられるようにし、各 PCIe リンク毎に、RC と EP が組になるように起動時に設定を行う。

PCI Express Base Spec. Rev. 2.0 (以降 Gen2 と呼ぶ) では、リンク速度は Gen1 での 2.5Gbps に加えて 5Gbps が選択できる。また、複数のレーンを束ねてバンド幅を拡張することが可能であり (レーン数を “x4” のように表記する)、本数に応じて自動的に 1byte 毎にインタリーブで送信する。さらに PEARL では、複数あるレーンのうち、特定のレーンにエラーが生じた場合は、そのレーンを使用しないように、本数を減らしたりレーン番号を反転させて再構成することによって、正常な動作を継続することができる。

一方、PCIe リンクは I/O バスの置き換えを念頭においているため、ボード上では数 10cm 程度のリンク長に制限される。しかし、小型の組み込み用途であれば、1 枚のボード上に、複数チップを隣接して配置することができるため、問題はない。さらに、PCIe 外部ケーブル<sup>11)</sup> を用いれば数 m 程度の距離を伝送できるため、小規模クラスタにおいても十分であり問題にはならない。また、消費電力は PCIe x4 レーンの場合でポート当たり 1W 程度を見込んでおり、他の高速ネットワークに比べるとポート当たりで数分の 1 である。ノードに追加する PEACH チップの他にはスイッチが不要なため、システム全体では消費電力、設置面積、コストの点で、さらに有利になる。

PEARL は、通常動作については PCIe そのものであるため、リンクの先に SATA コントローラなどのデバイスも直接接続できる。PEACH チップにより、PEARL で接続されたノード全てからアクセスすることが可能になる。これにより複数のノードで外部デバイスを切り替えてフェイルオーバーすることも可能になり、ディペンダビリティの向上に役立つ。

### 4. PCI Express コミュニケータ: PEACH

我々は現在 PCIe を用いた通信機構 PEARL を実現するためのコミュニケータチップ PEACH (PCI Express Adaptive Communication Hub) を開発中である。ここでは、PEACH の機能と構成について詳しく述べる。

PEACH チップは、ホストと他の PEACH との中継をする一種のルータチップである。図 2 に PEACH チップの構成を示す。

表 1 レーン数とレーン速度の選択 (物理層における消費電力比)

Table 1 Selection of the number of lanes and lane speed (power consumption ratio on PHY)

レーン速度	レーン数		
	x1	x2	x4
Gen1	2.5Gbps (21)	5Gbps (38)	10Gbps (75)
Gen2	5Gbps (28)	10Gbps (50)	20Gbps (100)

PEACH チップには、PCIe Gen2 コントローラが 4 ポート分配置され、それぞれが 4 レーン分の信号伝送を行なう。そのうち 1 ポートはノード CPU との接続のために使用され、残り 3 ポートを用いて隣接ノードの PEACH と接続される。PCIe Gen2 では、レーン当たり 5Gbps のデータ転送が可能であるため、各接続先毎に最大 20Gbps の転送速度を持ち、8b10b 符号化のため理論ピークバンド幅は 2GB/s となる。

PEACH の PCIe Gen2 コントローラは、動的に転送レートとレーン数を切り替えることができる機能を備えており、動作中にコンフィグレーションレジスタの設定を変更することで、必要な転送速度に応じて最適な転送レートとレーン数に切り替えて、省電力を実現することができる。但し、これらの変更には再設定のための時間に 30 $\mu$  秒程度を要する。表 1 に PEACH における PCIe の設定可能な組み合わせを示す。括弧内の数値は、各動作モードに対して、65nm プロセスで試作した物理層チップにおける消費電力の比を表す。Gen2 4 レーンを使用した最大性能 20Gbps 転送時の消費電力に対して、Gen1 1 レーン 2.5Gbps に設定することで約 80% の電力が削減できることが分かる。この消費電力比は物理層のみであり、リンク層の制御を含めると 80 数% の電力削減効果が期待できる。

一方、PCIe の規格では、一定時間送受信がない状態が続くと、通常動作の“L0”状態から、自動的に省電力モードの“L0s”状態に遷移することができる。L0s 状態では、出力信号のみを停止することにより、物理層レベルで L0 状態の 30~40% の電力まで低減できる。L0s 状態で要求を受け取ると、短時間 (数 10ns~数  $\mu$ s) で L0 に戻る。この機構は物理層ハードウェアで実現され、コンフィグレーションレジスタにより有効/無効を切り替えられる。さらに、データリンク層からの指示により、L0s 状態よりも電力を削減できる“L1”状態にも移ることができる。L1 状態では L0 状態より 90% 程度の消費電力が削減できる。但し、L0 状態への復帰にはリンクの復旧が必要なため数 10 $\mu$ s が必要である。

PEACH チップは、ルネサステクノロジ社 M32R プロセッサを内蔵する。M32R は、コアサイズが比較的小さく、性能に比較して低消費電力のコアとして組込み向けに多く用いられている<sup>12)</sup>。PEACH チップでは、内蔵プロセッサにより、トランザクション層の制御、

PCIe パケット送信・中継・受信などの処理だけでなく、ノード全体の監視、さらに通信リンク全体の管理を行う必要がある。そこで、4 ポートの PCIe を持ち、ルーティング機構を備えること、複数ストリームの制御が必要であること、ホストプロセッサが障害により停止した場合にもそれを検知して対処する必要があること、などを考慮して 4 コアを搭載することにした。また、不要なコアのクロックを停止することによって、消費電力の削減も可能である。PEACH 上の M32R コアが用いるメモリとして、チップ外に DDR3-SDRAM を接続する。実際に M32R コアは、コア毎に存在する命令データ分離 L1 キャッシュ、4 コア共有の単一 L2 キャッシュを介してアクセスする。起動用には別途チップ外に Flash-ROM を備える。また、PEACH チップ内には DMA コントローラ (DMAC) を備えている。これらのモジュールはすべて高速システムバスである SuperHyway を介して接続されている。DMA により高速システムバスである SuperHyway を介して、PCIe ポート間、または PCIe ポートと SRAM との間で、高速にパケット転送を行なうことができる。

図 3 に、PEACH チップのフロアプランを示す。これは各モジュールの必要面積や入出力ピンを考慮した配置の一例である。チップの四辺に PCIe Gen2 ポートが配置され、中央部分に M32R コア、および PCIe パケットを格納するための SRAM を備える。

表 2 に、PEACH チップの諸元を示す。PEACH チップは 45nm プロセスで製造される。チップサイズは試作の都合で 12mm $\times$ 12mm を予定しており、1000 ピンを超えるパッケージになる。電源電圧は、周辺回路用の 3.3V、DDR3 用の 1.5V、コアおよび PCIe 用の 1.2V の 3 種である。

パッケージされた PEACH チップは、PCIe カードエッジを持つボードに搭載する。当初、SiP (System in Package) モジュールとして、PEACH チップと SH-4A プロセッサと同一パッケージに収め、SiP パッケージを複数プリント基板に並べて配線することを想定していた<sup>1)</sup>。しかし、Infiniband, Ethernet などのネットワークインタフェースカード (NIC) と同様、PCIe のフォームファクタに則ったボード形状にすることで、一般的な PC やサーバに標準的に用意されている PCIe 拡張スロットにも挿入して使用することができる。図 4 に PEARL を用いた小規模クラスタの構成例を示す。これにより、PEARL が組込みプロセッサのための通信機構としてだけでなく、汎用のデスクトップあるいはサーバプロセッサを用いた小規模クラスタのための省電力かつ高信頼の通信機構としても使えるようになる。PEACH ボードのブラケット部分には、PCIe 外部接続ケーブル用の計 3 ポートのコネクタが設けられ、そこに PCIe 外部ケーブルを接続し、それぞれノード間を接続する。3 節で述べたように、ノード間のリンクは必ず両端が Root Complex と Endpoint になる必要があ

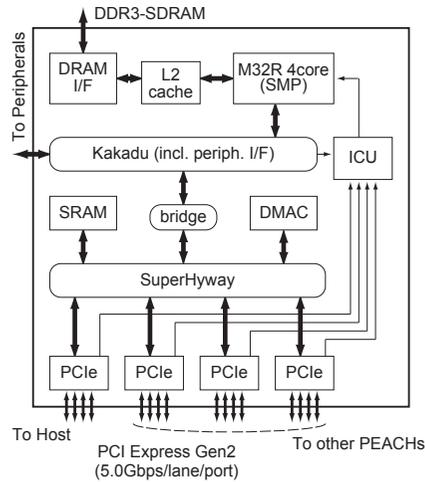


図2 PEACHチップの構成  
Fig.2 Block diagram of PEACH chip

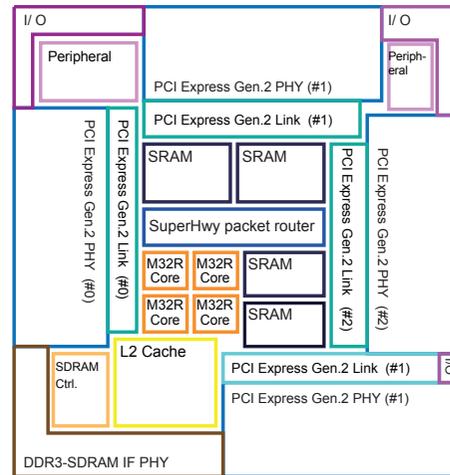


図3 PEACHチップのフロアプラン  
Fig.3 Floorplan of PEACH chip

表2 PEACHチップ諸元  
Table 2 Specification of PEACH chip

内蔵コア	ルネサス M32R 4 コア, SMP
通信リンク	PCI Express Gen2 4 レーン (20Gbps) × 4 ポート
プロセス	45nm Low power, triple-Vth, 8-Layer Metal
パッケージ	FCBGA-1296pin, 37.5mm×37.5mm
チップサイズ	12mm×12mm

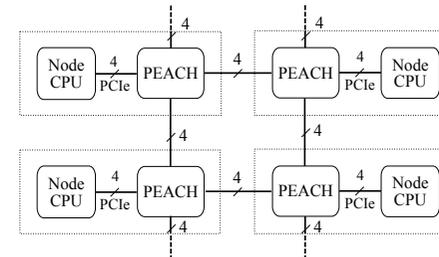


図4 PEARLによるクラスタ構成例  
Fig.4 Example of cluster using PEARL

るため、コネクタ付近に RC/EP 切り替え用のスイッチを設け、接続作業時に設定する。また、ボード上には Compact Flash スロットを用意し、後で述べるように、Compact Flash 上に M32R コア上で動作する Linux のファイルシステムを置く。これによって、ログを収集してホストプロセッサを介さずに Compact Flash 中に直接ファイルとして格納することができ、障害発生時の原因分析などが容易になる。

PEACH に内蔵された M32R コアの制御用 OS としては Linux/M32R を用いる。PCIe パケットのハンドリングなど性能を必要とする機能は Linux ドライバとして実現し、故障検出や回復、モニタリングなどの柔軟な処理が必要な機能は、Linux 上にユーザレベルで実現する。あらかじめ Flash ROM に小規模のブートローダを書き込んでおき、Compact Flash 内のファイルシステムからカーネルイメージを読み込んで、通常の PC と同様に起動することができる。

## 5. PEARL 向け通信 API: XMCAPI

本節では、PEARL 向けのユーザ通信 API について述べる。

マルチコアプロセッサにおける並列プログラムでは、Pthread や OpenMP といったスレッドを利用したライブラリや言語拡張が多く使われてきた。しかし、これらは共有メモリが存在することが前提であり、同期のためにロックが必要である。また組込み向けマルチコアにおいては、共有メモリがない場合や、メモリ空間を共有していても、コヒーレントキャッシュを持たない場合もある。そのため、これらのプログラミング手法では移植性が悪く、また、ロックを使った並列プログラムでは各スレッドの協調動作の状態が明示的には分からないため、プログラミングが困難になる場合がある。

一方、分散メモリ環境においては、ソケットや MPI (Message Passing Interface) が使われてきた。しかし、いずれも長距離、大規模な環境まで想定しているため、比較的通信遅延が大きく、通信の際に必要なメモリ使用量も小さくない。

Multicore Communications API (MCAPI) は、マルチコアチップ内のコア間通信を対象にし、MPI やソケットに比べて通信遅延やメモリフットプリントを小さくすることを目的とした、軽量通信 API である<sup>13)</sup>。そこで我々は、MCAPI の利点を活かしたままチップ間の通信に拡張した、XMCAPI を提案し、これを PEARL のためのユーザ通信ライブラ

りとして提供する<sup>14)</sup>。PEARLにおいては、PCIeでのリモートメモリ書き込み/読み出し、およびメッセージ送受信が操作プリミティブになるため、これらを用いてXMCAP I向けの低レベル通信ドライバを実装する。

PEARLを用いたマルチコアプロセッサによるクラスタでは、XMCAP Iにより、チップ内とチップ間でシームレスなコア間通信を利用することが可能になる。チップ内の通信においては、本来のMCAP Iの性質を活かした軽量の通信を利用し、チップ外の通信においては、PEARLが提供する高信頼で高性能な通信リンクを利用することができる。

XMCAP Iと同時に、TCP/IP向けにsocketを使って書かれた既存のプログラムのために、socketライブラリも開発する予定である。

## 6. おわりに

本研究では、ディペンダブルな省電力高性能通信機構として我々が提案している、PCI Expressを用いた通信リンクPEARLを実現する、PCI Express コミュニケータチップPEACHチップの概要および機能について述べた。

PEACHチップは、4レーンのPCI Express Gen2を4ポート持ち、4コアのM32Rプロセッサ、および各PCIeポート用にDMAコントローラを内蔵する。これらはSuperHywayバスにより結合され、高速動作と柔軟な制御の両立が可能になっている。

現在、PEACHチップおよびPEACHチップ搭載ボードを実装中であり、2010年3月末にPEACHチップ、2010年前半には搭載ボードがそれぞれ完成予定である。

今後は、PEACH搭載ボードのためのファームウェアやPEARL制御・監視用プログラム、ホスト用のドライバ、およびMCAP Iを拡張したXMCAP Iによるユーザ通信ライブラリを開発し、性能評価および、ディペンダブルな並列分散プラットフォームとしての応用を検討していく予定である。

**謝辞** 本研究の一部は、科学技術振興機構 戦略的創造研究推進事業 (CREST) 研究領域「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」、研究課題「省電力高信頼組込み並列プラットフォーム」による。

## 参 考 文 献

1) 埜 敏博, 朴 泰祐, 三浦信一, 岡本高幸, 佐藤三久, 有本和民: ディペンダブルな組込みシステムに適した省電力高性能通信機構, 情報処理学会研究報告 2007-HPC-113, Vol.2007, No.122, pp.31-36 (2007).

- 2) 埜 敏博, 朴 泰祐, 三浦信一, 佐藤三久, 有本和民: 小規模システム向け省電力高性能ディペンダブル通信機構: PEARL, 先進的計算基盤システムシンポジウム 2009, pp.124-125 (2009).
- 3) PCI-SIG: *PCI Express Base Specification, Rev. 2.0* (2006).
- 4) Infiniband Trade Association: *The Infiniband Architecture Specification*. <http://www.infinibandta.org/specs/>.
- 5) OpenFabrics Alliance: *OpenFabrics Enterprise Distribution (OFED)*. <http://www.openfabrics.org/>.
- 6) 岡本高幸, 三浦信一, 埜 敏博, 朴 泰祐, 佐藤三久: ユーザ透過に利用可能な耐故障・高性能マルチリンク Ethernet 結合システム, 情報処理学会論文誌コンピューティングシステム, Vol.1, No.1, pp.12-27 (2008).
- 7) International Organization for Standardization: *ISO 11898: Controller Area Network (CAN)*.
- 8) FlexRay Consortium: *FlexRay specifications*. <http://www.flexray.com/>.
- 9) ASI-SIG (PICMG): *Advanced Switching Core Architecture Specification* (2003).
- 10) Gudmundson, J.: Enabling Multi-Host System Designs with PCI Express Technology, <http://www.plxtech.com/products/expresslane/techinfo> (2004).
- 11) PCI-SIG: *PCI Express External Cabling Specification, Rev. 1.0* (2007).
- 12) Kaneko, S. et al.: A 600MHz Single-Chip Multiprocessor with 4.8GB/s Internal Shared Pipelined Bus and 512kB Internal Memory, *International Solid-State Circuits Conference (ISSCC) 2003*, Vol.1, pp.254-255 (2003).
- 13) Multicore Association: Multicore Communications API V1.063, <http://www.multicore-association.org/workgroup/mcapi.php> (2008).
- 14) 三浦信一, 埜 敏博, 朴 泰祐, 佐藤三久: 組込み機器向け on-chip/off-chip コア間通信機構, 情報処理学会研究報告, Vol.2009-ARC-184, No.2, pp.1-7 (2009).