2008－ARC－179 (20)
2008／8／7

組み込み向けマルチコア SoC における周期的プロセッサ間通信手法の提案

上久保雅規[†] 森義和[††] 菱沼智道[†††] 鳥居淳[†]

組み込み向けマルチコア SoC をモバイル通信アプリケーションに用いた場合の PE 間通信について，直ちに受信させる細粒度逐次型とまとめて受信させる粗粒度タイマ駆動型との比較検討行った．モバイル通信 Layer2 処理への適用を仮定してシミュレーションを行なった結果，処理速度およびリアルタイム性ともに粗粒度タイマ駆動型が優れ（処理速度で 3 割），それは，細粒度逐次型での割り込み処理時におけるオーバーヘッドとキャッシュヒット率低下による処理遅延要因の低減が原因とわかった．これによりシステムのタイマ周期中に複数回の PE 間通信が毎周期において発生するシステムでは粗粒度タイマ駆動型が有利であることを示した．

Periodic Inter-processor Communication Scheme on Embedded Multicore SoC

MASAKI UEKUBO[†] YOSHIKAZU MORI[††]
TOMOMICHI HISHINUMA[†††] SUNAO TORII[†]

In this article, the difference between two types of inter-processor communication scheme on embedded multicore SoC for mobile communication is discussed: one is fine-grain type, where each communication data is immediately received, and anther is coarse-grain type, where all communication data are received in every hardware timer event. The system simulation, for layer2 processing in mobile communication standard, revealed that the coarse-grain type with timer event is superior to the fine-grain type in processing speed by 30% and real-time property. It is because the communication with fine-grain type involves overhead of interrupt processing and degradation of cache hit ratio which causes processing delay of all the tasks after interrupt event. The result shows that coarse-grain communication scheme has advantages for the system where more than one communication data occurs in every system timer event.

1. はじめに

マルチコアを採用する動きは，サーバーや PC といった汎用処理系はもとより組み込み機器においても一般的になろうとしており，処理速度の向上と消費電力の抑制を両立する手段として広く認められつつある．

この組み込み機器では，PE（プロセッサエレメント）ごとにタスクがあらかじめ静的に割り付けられている AMP（非対称マルチプロセッサ）構成を採ることが多い．これは，複数アプリケーションを構成する要素であるタスク群をあらかじめ決められた PE 上に静的に割り付けておくことによってリアルタイム性能を維持するための最適化作業が行いやすくするものである．また，SMP（対称マルチプロセッサ）構成とは違って複数の PE をまたがったタスクスケジューリングをする必要がないので PE 間でデータのコヒーレンスを取る回路を削減できる．しかし，タスク群の間であらかじめ決められたデータの授受ができるような PE 間通信機構は実装する必要がある．

そこで，本研究では組み込み機器向けの PE 間通信機構について検討を行った．対象としたのは，リアルタイム処理が周期的に行われ，かつ，PE 間通信が毎周期発生し，そのレイテンシ制約がシステムの処理周期よりも長いシステムであり，ここではモバイル通信規格における Layer2 処理を題材として選んだ．PE 間での通信発生毎に受信を行う細粒度逐次型と，周期的にまとめて受信を行う粗粒度タイマ駆動型という二つの機構をそれぞれシミュレータ上で実行比較して，粗粒度のタイマ駆動型が処理速度向上とリアルタイム性の維持に有効な手段といえるかを確認した．

2. 課題と提案

組み込み機器において，リソースの最大限活用とリアルタイム性能の実現は重要な要件である．リソースを最大限活用するのはコスト削減対応が目的で，搭載されるプロセッサやメモリといったリソースを極力削減しつつ，この限られたリソースの中で最大限の演算処理性能を得る必要がある．リアルタイム性能を実現するためには，タスクフローやそれに影響を与えるデータフローを検討しながらシステム全体の設計をしな

† 日本電気（株） NEC Corp.
†† （株）NEC情報システムズ NEC Informatics Systems Corp.
††† コアーズ（株） Cores Corp.

ければならないが，そのためにはタスクスイッチの予測が容易であることが重要である．組み込み機器をマルチコア化して PE 間通信機構を導入する際にも，リソースの最大限活用とリアルタイム性能の実現といった要件を満たす必要があることは言うまでもない．

PE 間通信機構自体に関する研究はすでにいくつかある．たとえば，池原［池原 1981］は，PE 間通信機構を，送受信間での同期（通信のために処理を中断）の有無や媒体共用（共有メモリなど）の有無を指標として分類し，それぞれの方式の間での相違を調べ，通信要求が小さい場合は各 PE が独自の処理をどの程度連続して実行できるかがシステム性能向上の鍵であることを明らかにした．また，広瀬ら[広瀬 1996]は，多数のPE間で通信をする際に個々のPEにかかるソフトウェア処理負荷を低減するため，PE 間通信処理を行う専用ハードウェアを用いる方法を提唱した．しかし，これら既存の PE 間通信機構は組み込み機器を特に想定しているものではなく，限られたリソースの最大活用（オーバーヘッド時間の極小化）やリアルタイム性の確保について必ずしも考慮されている訳ではない．

組み込み機器でAMP構成を採るときのPE間通信では，通信発生毎に PE 間割り込みを用いて通信イベントの発生を受信側 PE へ通知するという細粒度での通信方法がよく用いられる．しかし，通信頻度が高くなるに従い単位時間あたりの割り込み処理回数が増えるため，通常は，先行するデータの処理中は後続データ受信の割り込みを割り込みマスクで抑制することによって割り込みオーバーヘッドの低減を図る方法が採られる．

このオーバーヘッド回避を目的として IP フレームをまとめて送信するという処理を，タスク機能の一部として実装するというのも現実的な解の一つである．しかしながら，それぞれが異なる PE 間通信を独自に行うようなタスク群を同時に実行・スケジューリングしているようなソフトウェアの作りになっている場合には，どれだけのデータをまとめて一括送信すべきかを実装段階で適切に見極めるのは難しい．

そこで，この「まとめる」という考え方を発展させて，複数のデータをまとめる操作を送信側のタスクは意識せず，受信側がハードウェアタイマを用いて粗粒度でまとめて処理を行うという方法を提案する(図 1)．この際，送信側ではPE間通信データをキューにひたすら入れていくだけでよい．この手法により，割り込みハンドラとそれに付随するタスクによる受信処理による時間オーバーヘッドを削減することができる．これは，タイマ周期が通信データ間隔よりも長いほど時間

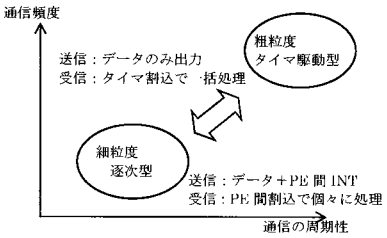


図 1　二つの PE 間通信手法

オーバーヘッド削減効果が高い．加えて，周期的に所定のタイミングで受信処理を行うことが予めわかっているため，設計時にPE間通信による影響予測が容易となり，リアルタイム性向上に寄与する．

3.　シミュレータ実装

前章で提案した粗粒度タイマ駆動型 PE 間通信による効果を，従来からの細粒度逐次型 PE 間通信と定量比較するため，それぞれをシミュレータで実行評価した．サンプルアプリケーションには 3GPP によるモバイル通信規格の Layer2 下り処理を用いた．タスクスイッチログを元に，システムの処理にかかる時間とリアルタイム性指標となるタスクスイッチ規則性をそれぞれ評価した．

3.1　シミュレーション環境

シミュレータには，NEC エレクトロニクス社の ClassMate[黒川 1998]を用いた．このシミュレータの特徴は，C++によってシステムレベルでのハードウェア記述が可能で，ソフトウェアを実行可能なプロセッサやその他の周辺回路のモデルが用意されていることである．

ハードウェアのブロック図を図 1に示す．プロセッサはARM926EJ-S，バスはAHB仕様のシングルレイヤバスである．これらのCPU，INTC，Timerの組をマルチコア環境として２組配置し，おのおののプロセッサはAHBバスを共有している．メモリはAHBバスを介して外部にSDRAMとして接続され，PE間で共有される．

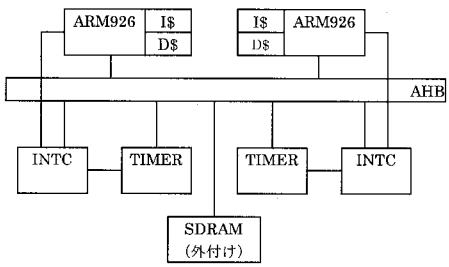


図 2　HW ブロック図

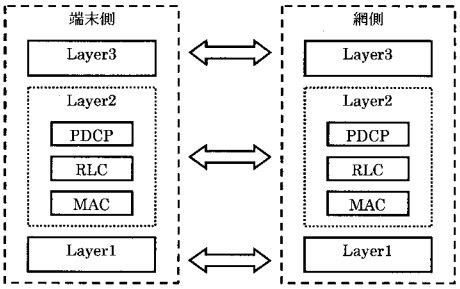


図 3　モバイル通信 Layer2 プロトコルスタック

表 1　ハードウェアモデル設定パラメータ

CPU コア	
動作周波数	200MHz
キャッシュ	4-way セットアソシアティブ ハーバードアーキテクチャ ランダムリプレース
L1 命令$	容量=4kB，ブロック=256B
L1 データ$	容量=4kB，ブロック=256B ライトバック方式(PE 間通信用共有メモリは非キャッシュ)
SDRAM	
動作周波数	100MHz
レイテンシ	最小 4cyc（CPU clk 換算）
ページ長	1kB

3.2　ワークロード

本検討で用いた 3GPP モバイル通信規格の Layer 2 処理の特徴は，通信プロトコルのサブレイヤ毎にヘッダ処理や対向局への送達確認，フレーム組み立てなどの処理を行いながら，通信サブレイヤ間をデータが伝播していくというものである．

3GPP のモバイル通信規格の Layer2 仕様では MAC/RLC/PDCP の各通信サブレイヤが存在する（図 3）．これらのサブレイヤのうち，MAC は Layer1 物理層との間で TTI と呼ばれる単位時間毎にトランスポートブロック（以下，TB）を授受する一方，Layer3 に対しては PDCP が IP フレームを授受する．上り送信時は，複数の IP フレームを所定の条件に従って生成したヘッダ部分と合わせて 1 つの TB の中に詰めて Layer1 に渡す．下りはその逆で，TTI 毎に受け取った TB から，複数の IP フレームとその他制御情報を抽出して Layer3 に渡す．

シミュレータ上の主な動作はヘッダ解析と次のレイヤへ渡すフレームの生成および転送であり，ハードウェア的には，メモリアクセスが主となる．

3.3　ソフトウェア構成

OS には，RTOS である TOPPERS/JSP を用いた．各

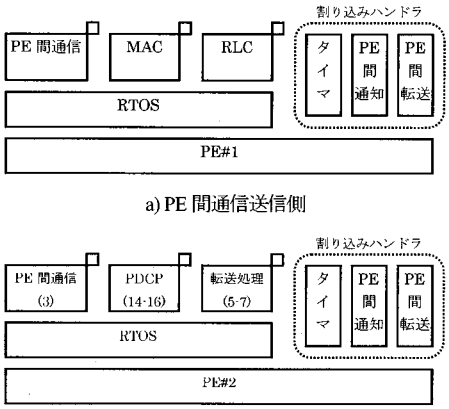


図 4　ソフトウェア構成

プロセッサコアでそれぞれ RTOS を独立して立ち上げており，スタックやヒープはそれぞれ専用の領域を確保してある．メモリ空間は，各 PE のローカル領域の他に共有メモリを SDRAM 中に確保した．この共有メモリは PE 間通信などに用いる．

PE ごとのタスク割り付けでは，MAC/RLC 処理タスクを PE#1 に，PDCP 処理タスクを PE#2 にそれぞれ実装した（図 4）．これにより，RLC と PDCP の間が PE 間をまたぎ，PE 間通信を利用して PDCP フレーム（サイズはほぼ IP フレームと同じ）の授受が行われる．タスク間通信はメールボックスを介して行うが，PE をまたがるタスク間通信については，PE 間通信タスクが 2 つの RTOS のメールボックス間でのデータ授受仲介を行う．

次に，受信側 PE におけるタスク群の処理フローを図 5 のシーケンスチャートと表 2 の受信 PE 上タスク一覧を用いて詳しく説明する．割り込みハンドラによって起床 (i) された受信処理タスク(ID=3)は，PE 間通信バッファから PDCP フレームを取り出してローカルメモ

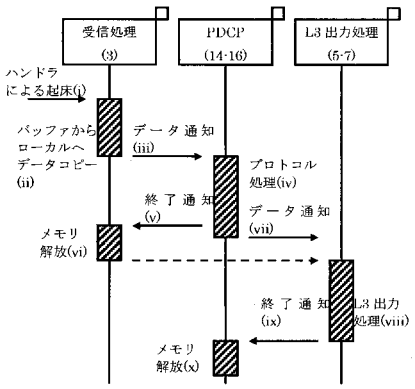


図 5　処理フロー

リ上の動的確保領域へ書き込んだ（ii）ことを，PDCP処理タスクへ通知する（iii）．PDCP処理タスク（ID=14-16）は，受信処理タスクから受け取ったPDCPフレームのヘッダ解析をしてIPフレームをローカルメモリ上の動的確保領域へ生成した（iv）ことをL3出力処理タスクへ通知する（vii）．このとき不要になったPDCPフレームは，PDCP処理タスクからの通知（v）により，受信処理タスク(ID=3)によって領域解放される（vi）．L3出力タスク（ID=5-7）は，PDCP処理タスクから受け取ったIPフレームをチップ外部（今回はUART I/F）へ出力する（viii）．このとき不要になったIPフレームは，L3出力処理タスクからの通知（ix）により，PDCP処理タスクによって領域解放される（x）．

表 2　受信側PE上タスク一覧

ID	タスク名	優先度	機能
1	IDLE	4	アイドルタスク
3	BR_CCPU	1	PE間受信処理
5－7	L2_APP[1-3]_DL	3	データをL3へ
14－16	L2_PDCP[1-3]_DL	2	無線プロトコル
20	timer_handler	－	タイマハンドラ
21-22	pe_notify_handler br_ccpu_handler	－	PE間割り込みハンドラ

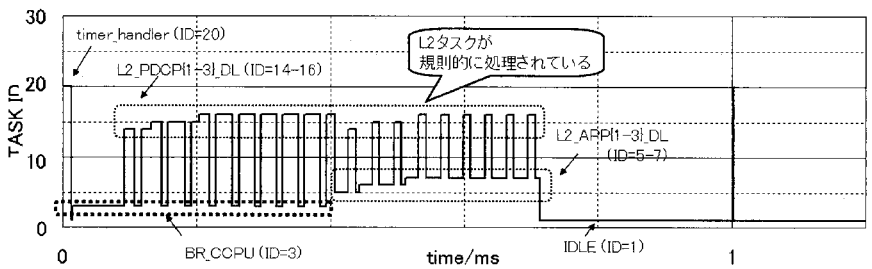
3.4　PE間通信部分の実装

PE間通信方式比較のため，細粒度逐次型と粗粒度タイマ駆動型の二方式での実装をそれぞれ行った．これらの各通信方式間での実装上の相違を，送信側PEと受信側PEに分けて説明する．

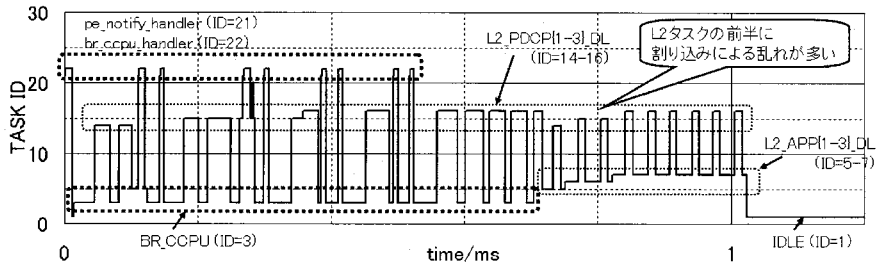
送信側PEでは，PE間で送信すべきデータが発生次第，PE間通信バッファにキューイングする．細粒度逐次型の場合にはこれに加えてデータ送信したことを直接受信側PEに伝えるためのPE間通信を発行する．

受信側PEでは，割り込みハンドラによって起動された受信処理タスクによってPE間通信バッファ内のデータが引き取られる．このとき，キュー内に複数のデータがある場合には，一回のタスク起動中にすべてのキュー要素が引き取られる．ちなみにこの受信契機となるこのハンドラとは，細粒度逐次型ではPE間割り込みハンドラであり，粗粒度タイマ駆動型ではタイマハンドラである．

PE間通信バッファは，SDRAMの共有メモリ上で，キューをリングバッファとして実現した．このキューに入れられるのは，送信先のタスクIDとデータへのポインタのみで，実際の送信データはメモリ空間においてSDRAM上の共有メモリ領域に保持される実装とした．



a) 粗粒度タイマ駆動型受信



b) 細粒度逐次型受信

図 6　PE間受信方式毎のタスクスイッチ時間遷移

4. シミュレーション結果と考察

4.1 通信粒度の違いによる二方式の比較概要

66バイトのPDCPフレーム9個（計594バイト）をPE間通信しているときの受信側PEのタスク遷移を図6に示す．ここに，横軸は時間経過（単位＝ms），縦軸はタスクのID番号である（表2）．なお，時刻0から受信処理を開始し，66バイトのフレーム9個分のPE間通信が終了した後に，アイドルタスクへ移行する．

まず，提案の粗粒度タイマ駆動型におけるタスクスイッチの様子を図6(a)に示す．PE間通信で受信する9つのデータを各タスクで規則的に処理しながら次の段階のタスクに実行が移っている様子がわかる．このとき，外部イベントなどに起因するタスクフローの乱れは無い．このように，処理の間でのばらつきがほとんどないために各処理実行時間の予測性が高くなり，結果として高いリアルタイム性を達成しやすい方式であることがわかる．

次に，細粒度逐次型のタスク遷移を図6(b)に示す．今回の実装では，一つのPE間通信データを処理中に後続データがあれば続けて処理することによってハンドラ起動などのオーバーヘッドを回避する機構は備えてはいるが，送信側でのPE間通信データ発生が散発的なため，結果として一回の受信処理起動によって複数データの受信処理を行うケースは発生しておらず，データフレーム数に等しい9回のPE間割り込みおよび付随タスクが繰り返されている．これらのPE間割り込み（TASK ID=21-22）が受信側タスク処理（タスク遷移）とは非同期に発行されていることにより，その他の受信側PE上タスク動作が都度中断され，各タスク処理間隔が図6(a)の粗粒度タイマ駆動型よりも開いている箇所と同等な箇所が入り交じっている．処理間隔の広い箇所が処理速度の低下原因となるほか，間隔のばらつきによって予測性が粗粒度タイマ駆動型に比べて相対的に落ちてリアルタイム性維持に不利となる．

4.2 性能差分の要因分析

さらにタスク毎の詳細なオーバーヘッド量について解析するため，細粒度逐次型と粗粒度タイマ駆動型について，各タスクの実行積算時間の観測を行った．

上記図6に示したタスク遷移のログから処理時間を抽出して積算したものを図7に示す．横軸にタスクをとり，縦軸には各タスクの実行時間積算値をとってある．また，二方式を棒グラフ（左棒が細粒度逐次型，右棒が粗粒度タイマ駆動型）で並べて表し，また，細粒度逐次型に対する粗粒度タイマ駆動型の改善量を細粒度逐次型全体の処理時間に対する寄与率として百分

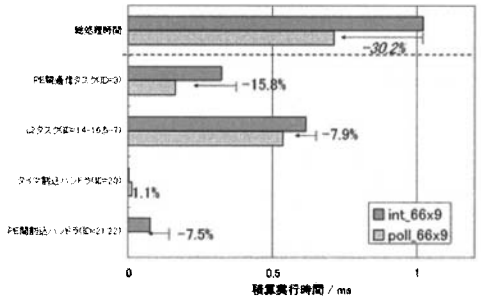


図7　タスク積算実行時間（キャッシュ有り，全体）

率でコメント追加してある．

ここでまず特徴的なのは，粗粒度タイマ駆動型のほうが全体の処理時間が約3割削減できて高速なことである．そこで，内訳を粗粒度タイマ駆動型の立場から見ていくと，まず，PE間の割り込みハンドラが起動不要な分として7.5%（ID=21,22），後続のPE間受信タスクで15.8%（ID=3）分の削減がそれぞれみられる．ただし，タイマハンドラで1.1%の処理増加があり，PE間通信を開始するための時間増である．これらを合計して直接PE間通信に関わる部分で差し引き22.2%分の削減がなされている．ところが加えて，直接PE間通信に関わらないLayer2関連処理タスクでも，計7.9%（ID=5-7,14-16）分の改善効果が得られている．これらが足しあわされて30.1%の改善となる．

上記のPE間通信に関わらないLayer2関連処理で改善した7.9%の内訳を表したのが図8である．これによると，個々のタスク毎の改善量には偏りがあるのがわかる．そこで，この図8と図6を合わせて見ると，図6でPE間割り込みハンドラ（ID=21,22）が起動される時間帯と，図8で性能差分がみられるタスクの実行時間が一致するのがわかる．このことから，本来PE間通信処理を直接行っているわけではないタスクであっても，実行時間中にPE間割り込みが入ることにより，図6(b)の説明で述べた処理開始時刻の遅延だけでなく，これらLayer2関連処理自体の時間も延びていることがわかる．

さらに原因を調べるため，キャッシュをI$/D$とも無効にし，Layer2関連処理の各タスク実行時間積算値をグラフ化したのが図9である．この結果では，PE間通信処理を直接行わないLayer2関連処理について二つのPE間通信方式間で差が見られない．このことから，図8においてPE間通信を直接行わないタスクの積算実行時間がPE間通信の影響を受けたのは，キャッシュの有無の差に原因があり，PE間割り込みに関わる処理が挟まることでキャッシュの一部がリプレースされてヒッ

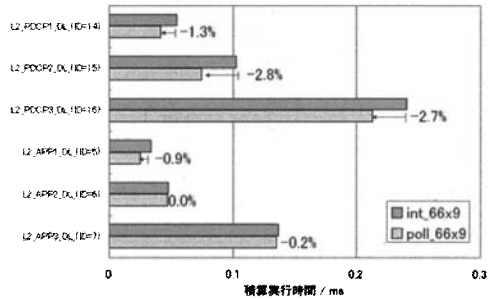


図 8　タスク積算実行時間（キャッシュ有り，無線処理部）

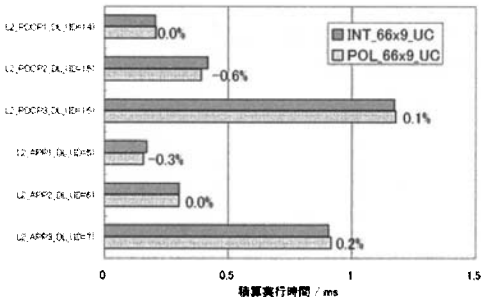


図 9　タスク積算実行時間（キャッシュ無し，無線処理部）

ト率が低下したためだと考えられる．

5. まとめ

組み込み向けマルチコア SoC に用いる PE 間通信について，直ちに受信させる細粒度逐次型とまとめて受信させる粗粒度タイマ駆動型との比較検討行った．モバイル通信 Layer2 処理への適用を仮定してシミュレーションを行うことにより，処理速度およびリアルタイム性ともに粗粒度タイマ駆動型が優れ，それは，細粒度逐次型での割り込み処理時におけるオーバーヘッドとキャッシュヒット率低下による処理遅延要因の低減が原因とわかった．これによりシステムのタイマ周期中に複数回の PE 間通信が毎周期において発生するシステムでは粗粒度タイマ駆動型が有利であることを示した．

粗粒度タイマ駆動型で上記効果を得るには，周期毎に PE 間通信が複数回発生していることが必要である．システムの処理周期と PE 間通信頻度がこの条件を満たせば良いが，そうでなければタイマ周期の設定を長く（システム周期の整数倍など）とったり，必要処理量が一時的に低下して毎周期必ずしも PE 間通信が発生しない期間には細粒度逐次型に一時的に切り替えたりすることで，この粗粒度タイマ駆動型を用いる際の要件を実質的に緩和することが可能である．さらに，これらは実装段階でも容易に試みることができる．

細粒度逐次型で処理性能を改善するには，キャッシュ容量を増やしてヒット率低下を防ぐ方法も場合によって採りうる．しかし，ハードウェア設計段階で今回の Layer2 処理遅延のような間接的な影響を考慮してキャッシュ容量を最適化するのは難しい．別の手として，送信側のタスク内でまとめてから送信を行うのも可能であるが，その実装や実装後の動作確認のために相当量の設計工数を要するのが難点である．

また，粗粒度タイマ駆動型では，細粒度逐次型と比べてより大きな PE 間通信用バッファが必要となる．たとえば，14.4Mbps のモバイル通信の場合，TTI が 2ms として，必要なバッファ量は単純に（ダブルバッファとして）約 7.2kB であり，オフチップメモリ上はもちろん，オンチップ上でも十分確保可能なメモリ量である．このため，搭載可能なバッファサイズ内でできるだけ周期を延ばすことで最も効率的な PE 間通信が行える．

謝辞

シミュレーション環境構築にあたって，NEC エレクトロニクス社の中島氏，長瀬氏，柏木氏に環境提供およびサポートをいただきここにお礼申し上げます．

また，PE 間通信について指導していただいた NEC システム IP コア研究所酒井氏に深く感謝いたします．

参考文献

[3GPP] 3GPP, http://www.3gpp.org/

[Sakai2005] Sakai, J. et al.; "Multi-tasking Parallel Method on MP211 Multi-core Application Processor", COOLChips VIII Proceedings, pp.198-211 (2005)

[TOPPERS] TOPPERS/JSP, http://www.toppers.jp/

[Torii2005a] S. Torii, et al., "A 600MIPS 120mW 70/spl mu/A leakage triple-CPU mobile application processor chip," ISSCC Dig. Tech. Papers, vol. 1, pp.136-137(2005)

[池原1981] 池原悟，マルチプロセッサ方式におけるプロセッサ間通信方式と通信特性について，情報処理，Vol. 22, No. 1, pp. 1-8 (1981)

[黒川1998] 黒川秀文，SOC の事前検証を実現する C++ シミュレータ「ClassMate」，信学会研究報告．VLD, Vol.98, No.287, pp.17-24 (1998)

[田坂2003] 田坂修二，『情報ネットワークの基礎』，数理工学社 (2003)

[広瀬 1996] 広瀬哲也ほか，並列コンピュータ Cenju-3 のプロセッサ間通信方式とその評価，情報処理，Vol. 37, No. 7, pp. 1378-1387 (1996)

[藤倉 2003] 藤倉俊幸，『リアルタイム/マルチタスクシステムの徹底研究』，CQ 出版 (2003)