

並列計算機 EM-X のプロセッサ・ネットワークインターフェースの
最適化の検討

坂根広史 児玉祐悦 佐藤三久 山名早人 坂井修一† 山口喜教

電子技術総合研究所 †新情報処理開発機構

我々は現在並列計算機 EM-X に用いるシングルチッププロセッサ EMC-Y の設計を進めている。EMC-Y は RISC アーキテクチャを採用した演算部およびネットワークインタフェースを内蔵しており、両者を密に融合している。すでに開発した EM-4 の評価に基づき、ネットワークの性能を十分に引き出すと同時に演算部の稼働率を高めるための工夫として、メモリアクセス方法の改良やパケットバッファの追加等をおこなう。新しい機能について、レジスタ転送レベルシミュレータを用いて EMC-Y の内部動作を詳しく調べることによってその効果を確かめ、より効率のよい設計パラメータを得た。

Optimization of network interface in a processing element for
a parallel computer EM-X

Hirofumi SAKANE Yuetsu KODAMA Mitsuhiisa SATO Hayato YAMANA
Shuichi SAKAI† Yoshinori YAMAGUCHI

Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki 305 Japan
†Real World Computing Partnership

This paper discusses some of the design parameters for the network interface of a single chip processor EMC-Y to achieve high throughput and high performance. We are currently designing the EMC-Y, a processing element of a parallel computer EM-X. The design parameters include the arbitration method in the network switch, memory access priority and the size of internal FIFOs. To optimize parallel execution performance, we have examined the parameters of the network interface by using a register transfer level simulator of the EM-X.

1 はじめに

並列計算機では要素プロセッサ (PE: Processing Element) 単体の演算能力ばかりではなく、相互結合網の性能およびそのインタフェースが全体の処理能力に大きく影響を与える。つまり演算部を最高の能率で動作させるためには十分なアクティビティやデータを供給する必要がある、演算処理と通信の整合性が問題となる。データ駆動モデルを改良した強連結枝モデルはその問題を解決する試みであり、本モデルに基づいた並列計算機 EM-4 の開発および評価により、その有効性が確認されてきた。

EM-4 は分散メモリ型並列計算機であり、80 台の要素プロセッサ EMC-R がサーキュラオメガネットワークによって接続されている。EMC-R では強連結枝モデルを実現するためデータ駆動モデルの循環パイプラインとプログラムカウンタを用いた逐次演算パイプラインを融合している。それぞれのパイプラインは前者がバケットベースの高速な通信・同期を提供する直接マッチング機構やバケットバッファにより、また後者はレジスタベースの RISC アーキテクチャを用いることにより高い性能を実現している。両者はパイプラインレベルで融合されており高い整合性をもつため、ネットワークの通信性能と演算処理部の演算能力がバランス良く引き出せる。これによって効率の良い細粒度並列処理を実現することができ、高い並列処理性能が得られた。しかし同時に、ネットワーク負荷が大きくなるとシステムの挙動が大きく変化する現象がみられた。この原因はネットワークインターフェースの高負荷時の入出力性能のアンバランスによるものと考えられ、このような場合には演算パイプラインもネットワークの影響を受けて稼働率が低下していた。

これまでの評価・検討から、アーキテクチャの改良によりそれらを効果的に改善できることがわかった。またリモートメモリアクセスのレイテンシを低減する機構や優先度を考慮したバケットバッファ等のアイデア、命令セットの見直しなどが並列処理性能向上へ大きく貢献することが示された [3][5]。

我々はこれらの研究結果をもとに、さらに優れた並列処理をおこなうための要素プロセッサ EMC-Y の設計を進めている。並列計算機 EM-X は EM-4 と同じくサーキュラオメガネットワークを用い、80 台の EMC-Y が接続される並列計算機である。EMC-Y の設計と動作確認には VLSI 設計用 CAD を用いている。並列動作時のオーバーヘッドの削減やバッファサイズ等のパラメータチューニングをおこなうためには EM-X 全体の動作確認が別の方法が必要である。また EM-4/X は、その要素プロセッサを構成する 32bit RISC 演算部、メモリインタフェース、ネットワーク通信インタフェースおよびそれらを融合する機構を 1 チップに収めていることにより PE 台数の拡大が容

易なシステムになっている。本研究を進める中で、並列処理の高効率化のためのアイデアや仕様が数多く検討されたが、全てを 1 チップに収めることは不可能である。1 チップ化のメリットは EM-4/X の研究指針の基幹をなすものであることから、性能向上アイデアの取捨選択の必要がある。すなわち、性能向上のためのアイデアを設計にとり入れる際にはシミュレーションによってその効果を確認しながら作業を進める必要がある。また同時に 1 チップ化という制約条件のもとで洗練されたアーキテクチャを追求しなければならない。我々はこれらの評価・検討を、クロック単位で動作するレジスタ転送レベルシミュレータを用いておこない、EMC-Y の設計にフィードバックしている。

本稿では、主にネットワーク通信インタフェースに注目してそのアーキテクチャについて述べ、80 台の PE を相互結合網で接続し設計条件を変化させて動作シミュレーションをおこなった結果を示し、その評価に基づき演算と通信が効率良く整合するような設計パラメータについて考察する。

2 EM-X

現在開発中の EM-X は細粒度並列処理に適した並列計算機である。EM-4 のアーキテクチャをベースにその評価と改良点の検討に基づき、より柔軟で効率の良い並列処理を目指している。現在 EM-X の要素プロセッサとして、新たに 1 チップ VLSI の EMC-Y の設計を進めている。EM-X では相互結合網としてサーキュラオメガネットワークを採用し、これによって 80 台の EMC-Y を接続する。

EM-4 は通信の高速化 (低レイテンシと高スループット) と命令体系の単純化 (RISC アーキテクチャ) を兼ね備え、それらを融合したアーキテクチャの有効性を実証した。EM-4 から EM-X への進化の基本コンセプトは、大きく分けて二つある。一つはアーキテクチャの最適化で、EM-4 の評価により明らかとなったネットワークインターフェースのオーバーヘッドや通信のレイテンシの削減、命令セットの見直しによる演算パイプラインの動作効率の向上である。もう一つは VLSI テクノロジーの向上によるもので、これはアーキテクチャの改良にも寄与しているが、そのほかに EMC-R で実現できなかった浮動小数点演算回路の内蔵やレジスタファイルの増設 (16word→32word)、トラップ機構の実装、クロック周期の短縮等が EMC-Y で実現される。

前者のアーキテクチャ上の最適化について詳しく見ると、メモリバンド幅の拡大、リモートメモリアクセスの低レイテンシ化、優先バケットの考慮、内部バケットバッファ再配置等がポイントである。まず、メモリバンド幅がクロック当たり 2 倍になったことがアーキテクチャ設計上の柔軟性を増し、数多くの改良や追加機能に結び付いている。

EMC-R では入力パケットバッファのメモリアクセス優先度が演算部より低かったため、内部 FIFO があふれた場合にしばしばネットワーク負荷の増大を招いた。メモリバンド幅倍増により演算部の命令フェッチとパケットバッファのメモリアクセスが独立しておこなえるようになった結果、ネットワーク負荷の軽減が期待される。同様に命令フェッチとマッチング処理の重畳化もおこなわれ、性能向上に大きく寄与する。リモートメモリアクセスのレイテンシを抑える機構はメモリバンド幅拡大の恩恵を直接受けている。すなわちメモリアクセス頻度の多い命令フェッチにオーバーラップして、他 PE の要求に従い入力パケットバッファ部が直接メモリ操作することによって、リモートメモリアクセスがブロックされることが避けられる。優先パケット処理は柔軟なパケットスケジューリングを提供し、リモートメモリアクセスの低レイテンシ化への貢献のほか、動的負荷分散に対しても有効である。出力パケットバッファの新設は演算部稼働率の向上を目的に新設される。EMC-R ではネットワークがビジーの場合はパケット出力ができず演算パイプラインが停止してしまい、その稼働率が下がることがあった。ネットワークへの出力インタフェースにバッファを設けて混雑の影響を吸収することにより演算パイプラインの実行性能低下を防ぐことができる。

ワード構成は、メモリインタフェースが 38bit/word、ネットワークインタフェースは 39bit/word であり EM-4/EMC-R を継承する。

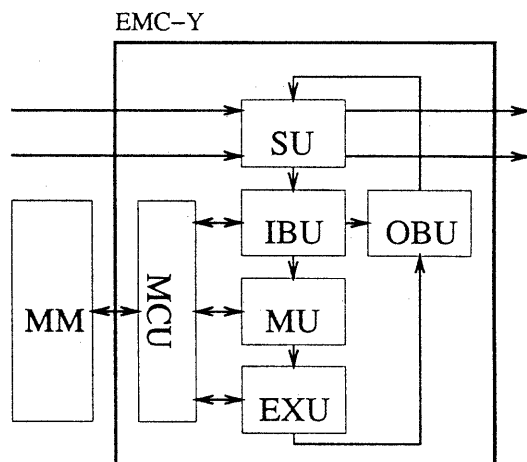
2.1 ネットワーク

EM-X ではプロセッサ間相互結合網として、EM-4 でその高い性能が実証されたサーキュラオメガネットワークを続けて採用する。これは間接網であるオメガネットワークの両端を接続してリング状にし、スイッチノードに PE を接続して直接網としたものである。EMC-R/Y ではネットワークスイッチを内蔵してハードウェア量削減に成功している。

本ネットワークは PE 台数 N に対してハードウェア量が $O(N)$ 、直径が $O(\log N)$ に抑えられるという優れた特徴を持つ。ルーティングはセルフルーティング方式でパケットのアドレス部と自 PE アドレスを比較して転送先が一意に決定される。EMC-Y のルーティングは EMC-R では実現されなかった最小距離アルゴリズムを用いている [3]。

2.2 パケット構成

EM-X/EMC-Y のパケットは EM-4/EMC-R と同じく 2word の固定長で、1word のアドレス部と 1word のデータ部からなり、パケットの出力は 1 命令でおこなうことができる。このパケットの単純化は通信と演算のパイプライン融合と、細粒度並列処理の鍵となっている。



SU: Switching Unit
 IBU: Input Buffer Unit
 MU: Matching Unit
 EXU: EXecution Unit
 OBU: Output Buffer Unit
 MCU: Memory Control Unit
 MM: Memory Module

図 1: EMC-Y の機能ブロック

2.3 EMC-Y のアーキテクチャ

EMC-Y はゲート数約 85,000、入出力ピン数 255 本の CMOS ゲートアレイチップである。動作クロック周波数は 20MHz に設定した。

図 1 の機能ブロックに示すように EMC-Y は大きく分けて 6 個のユニットから構成される。このほかにチップ内部のフリップフロップを個別に読み書きするためのメンテナンス回路がある。以下に各ユニットについて述べる。

SU(Switching Unit) (図 2)

EMC-Y はサーキュラオメガネットワークのノードとなるためネットワーク・スイッチとして SU をチップに内蔵している。入出力はそれぞれネットワークが 2 ポート、自 PE 1 ポートで、 3×3 のクロスバスイッチである。転送先が競合する場合の調停方法によりネットワーク負荷が影響を受ける。このほかサーキュラオメガネットワークのストアアンドフォワード・デッドロックを回避するための 3 バンクバッファを備えている。

IBU(Input packet Buffer Unit) (図 3)

演算実行部 (EXU) や待ち合わせ部 (MU) がビジーの間もネットワークの混雑を避けるため、取り込んだパケットをバッファリングする。内部バッファ (FIFO) が溢れる場合は外部メモリ (MIB) にパケットを退避

する。EMC-R では内部にバケット 16 個分の FIFO を持っていたが、EMC-Y ではその効果とゲート数の制約のトレードオフによりサイズを検討する。また、新しい機能として待ち合わせ動作を MU に代わっておこなう機構や、他 PE のリモートメモリアクセス要求を EXU へ送らず直接にメモリアクセスして処理する機構などを持っている。優先バケット処理に対応するため、バッファリング機構は MIB の管理、内部 FIFO(Ph,Pl) ともに 2 系統備えている。IBU 内部ではリモートメモリアクセス、MIB アクセス、マッチングメモリアクセス等多くのメモリアクセス要求が競合する。これらの要求は MCU に送られる前に IBU 内で調整がおこなわれる。その優先順位はバケット取り込み能力に直接影響するので慎重に決定する必要がある。

OBU(Output packet Buffer Unit) (図 4)

OBU は、ネットワーク混雑時にも EXU がパイプラインを止めることなく能率的に動作するために混雑状況を緩衝するためのユニットである。EXU からは 1 バケットを 1 クロックで受けとり、演算の最大スループットを維持することができる。FIFO サイズ、および循環パイプラインを使用する自 PE 宛バケットのバイパスを設けるか否かが設計要素である。図は自 PE 宛バケットバッファ (SELF) を設けた場合のブロック図である。NET は他 PE 宛バケット用 FIFO である。自 PE 向けバイパス経路によって、ネットワークの混雑による自 PE のアクティビティ低下を防ぐことが期待できる。SELF を設けない場合は SU へのパスが一つになりデマルチプレクサ (demux.) が省略される。

これまでの研究で OBU のサイズをむやみに増やしても全体の時間短縮に結び付くとは限らないことがわかっている。これは演算のスループットが上がり、バケット出力頻度が増え、ネットワークが混雑するためであった。さらにその原因を追求すると IBU のバケット取り込み能力とのバランスがとれていないためであることがわかった。バケットの出力能力と入力能力がともに向上する必要があり、設計時にはそのバランスを考慮することが重要である。

MU(Matching Unit)

改良されたデータ駆動モデルに基づく待ち合わせ処理と強連結ブロック先頭の命令の発火 (フェッチ) をおこなう。

MCU(Memory Control Unit)

EMC-Y のメモリアインタフェースは物理的には 1word、1 ポートしかないが、クロックの前半と後半で 1 回ずつメモリアクセスが可能である。すなわち 1 クロ

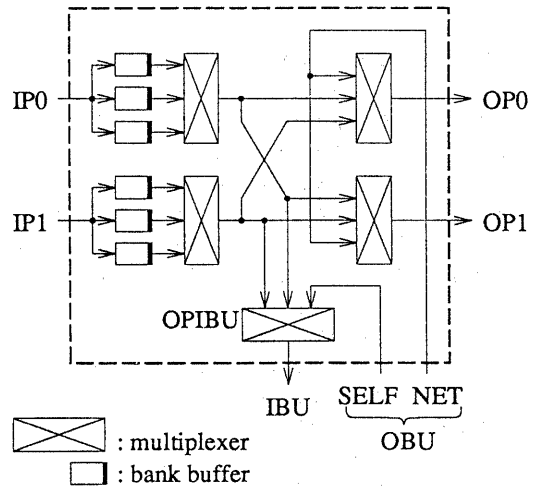


図 2: SU

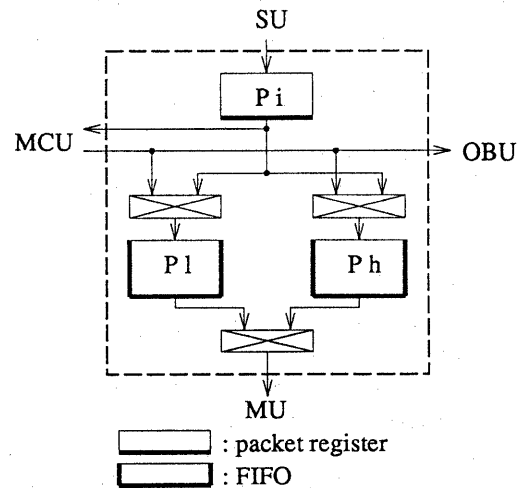


図 3: IBU

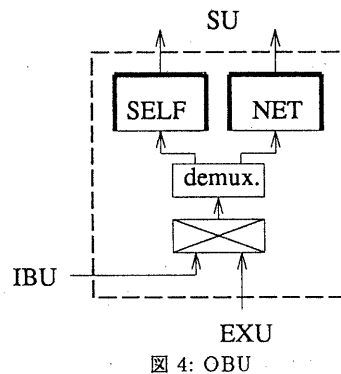


図 4: OBU

ク時間の間にメモリアクセスが2回でき、これは EMC-R に対して1クロック当たり2倍のメモリバンド幅である。本ユニットはクロック前後半それぞれにおいて独立したメモリアクセス調停をおこなう。前半フェーズでは EXU の命令フェッチ要求と MU のメモリ読み出し要求の調停、後半フェーズでは MU のメモリ書き戻し要求と EXU のメモリ操作命令の要求および IBU の各メモリアクセス要求の調停がおこなわれる。

EXU(Execution Unit)

独自の命令セットアーキテクチャを持ち、バケット出力命令等、循環パイプラインに直結する機構を備えた 32bit RISC 演算部である。また EMC-R で実現されなかった浮動小数点演算回路を内蔵しており、科学技術計算の用途にも対応できる。本ユニットの詳細は別の機会に述べる。

MM(Memory Module)

EMC-Y では各種セグメントエリアやバケットバッファを外部に持ち、それを単一のアドレス空間で管理する。EM-X は1PE 当たり、その最大限度である 38 ビット × 1Mword を高速 SRAM で実装する。

本稿では IBU と OBU をネットワークインターフェースと称する。SU はネットワークの一部であるが、IBU、OBU とのインターフェース部分でネットワーク対プロセッサ(OBU)の調停方法を検討する必要がある、これを含めて性能チューニングの対象とする。

3 シミュレーション

並列計算機の要素プロセッサを設計する場合、高いシステム性能を確実に得るために次のような点で設計に即した動作シミュレーションが重要である。

機能検証：設計意図の通りに回路が動作するか

実現可能性：使用する VLSI テクノロジーの制約条件を満たすか

性能チューニング：並列計算機の構成要素として最高の性能を発揮するための設計パラメータを追求する

3.1 シミュレータ

EM-X の開発においては以下に挙げた計算機ハードウェアシミュレータを設計レベルに応じて使い分けている。いずれも効率的な並列計算機システム開発のためには必要でありそれぞれ固有の特徴をもっている。

1. ゲートレベルシミュレータ

論理回路の設計データに基づき、ハードウェアに密

着した論理検証、遅延時間評価等をおこなう。EMC-Y の設計には VLSI 設計用 CAD が持つシミュレータを用いている。プロセッサ単体が対象であるうえ、実行の時間がかかる。

2. レジスタ転送レベル (RTL) シミュレータ

設計データをもとにして、システムクロック毎に状態遷移やレジスタ (FlipFlop) 間のロジックの信号伝搬をシミュレートする。相互結合網を含めたシステム全体のシミュレーションをおこなう。ゲートレベルより高速。後述のシミュレーションは本シミュレータによるものである。

3. 命令レベルシミュレータ

システムのシミュレーションを高速におこなうために、プロセッサの設計データよりもむしろ命令仕様に基づいてシミュレーションをおこなう。コンパイラの出力コードなど、大きなプログラムのテストに適している。

3.2 レジスタ転送レベル (RTL) シミュレータ

並列計算機としての動作検証とチューニングを行なうため、EMC-Y の論理設計データを反映させながら、内部動作に忠実な RTL シミュレータ作成した。本シミュレータは、並列システム全体が動作する場合の論理回路の総合チェックのほかに、設計アイデアや最適なアーキテクチャを探る目的も持っている。すなわち、処理の高速化に寄与することが期待される機能の追加や、設計要素が可変量である場合の最適パラメータをチェックすることができる。注意が必要なのは RTL シミュレーションによる設計やテストで最適と考えられた仕様設定でも、それが1チップに収まるものでなければならないことである。また同様にチップ内のクリティカルパスが、仕様として決定したクロック周期内に収まるかどうかチェックしなければならない。ゲートレベルシミュレータを含む CAD システムを用いてこの作業が進められるが、設計の最終段階では両レベルのシミュレーションと綿密な検討を繰り返すことが重要である。すなわち1チップ化および高速化という制約条件のもとで洗練されたアーキテクチャを選択することが必要となる。

本 RTL シミュレータには先に言及したものを含め、以下のような役割がある。

- 論理機能検証
- ゲートレベルシミュレータ用テストベクタ作成
- 最適パラメータの追求
- 高効率化アイデアのテスト
- 基本プログラムのプロトタイプ検証
- ゲートレベルシミュレータ、命令レベルシミュレータとの関係による設計支援
- 並列プログラムのデバッグ

3.3 設計パラメータ

シミュレーションによる評価と最適パラメータの考察は以下の設計要素についておこなう。

- IBU: メモリアクセスの優先度
- OBU,IBU: 各バケットバッファのサイズ
- OBU: 自 PE 宛バケット用バッファ
- SU: マルチプレクサの調停

4 例題プログラムによる評価

以下では、フィボナッチ数を再帰的に求める EM-X 用並列プログラム (fib) の実行時間に基づいて、ネットワークインタフェースの効率の評価をおこなう。そのアルゴリズムは EM-4 の評価した際にも用いたもので、強連結ブロック (SCB: Strongly Connected Block) が小さくバケット出力頻度が比較的多いという特徴をもつ。強連結ブロックは強連結枝モデルに基づく、排他的に逐次実行される命令ブロックであり、ここでは並列処理の粒度として扱う。このプログラムでは再帰的に呼び出される手続き (強連結ブロック) は静的に PE に割り当てられる。つまり全出力バケット数や転送経路は最初に与える引数のみで決まり、システムパラメータに関して不変である。

評価結果に一般性をもたせるため、実験パラメータに強連結ブロック平均長をもちいた。ここではフィボナッチプログラムに nop 命令を埋め込んでバケット出力頻度を変化できるようにした。nop を埋め込むことにより強連結ブロックの平均長が長くなり、1 命令当たりのバケット出力頻度は減少する。これによって粒度あるいはネットワークの混雑度を視点とした議論をすることができる。ここで混雑度とは実レイテンシあるいはバケット衝突頻度を表す尺度であり、fib では強連結ブロック長が短くなるとほぼ単調に増加する。

比較の基準として EMC-Y 設計初期の設計条件による実行時間を用いる。すなわち、

- IBU のメモリアクセス優先度: restore > MM,store
- OBU FIFO サイズ: 8 packet
- IBU FIFO サイズ: 1 packet
- OBU の SELF バッファ: なし
- SU のマルチプレクサの調停:
OP: IP < OBU
OPIBU: IP > OBU

とする。以下のシミュレーション結果は全て fib(22) を求める場合についておこなったものであり、上記基準条件による実行時間を 1 に正規化し、平均強連結ブロック長 (あるいは粒度、ネットワーク混雑度) に対する実行時間の変化をプロットした。

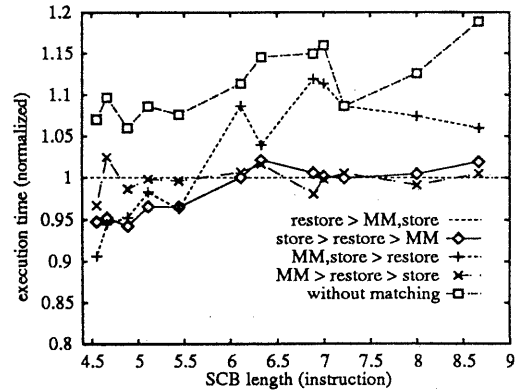


図 5: IBU メモリアクセス優先度の影響

4.1 IBU のメモリアクセス優先度

メモリアクセス優先度の変更組合せは以下の 4 通り。store/restore はバケットの MIB 退避/復帰に伴うメモリアクセスを、MM はマッチング処理によるメモリアクセスを表す。待ち合わせバケットであればバッファリングせず即マッチングをおこなうので MM と store の要求は競合しない。

- restore > MM,store
- store > restore > MM
- MM,store > restore
- MM > restore > store

図 5 はメモリアクセス優先度の各組合せについてシミュレーションした結果である。

restore を優先すると store が抑えられるのでネットワーク混雑を引き起こしやすと考えられる。store を優先するとバケット取り込み能力が向上するが、restore が抑えられるので EXU へのアクティビティ供給は減少する傾向があると考えられる。シミュレーション結果では、store を優先する効果は強連結ブロックが平均約 6 命令以下の領域で現れることがわかった。

マッチング処理は、それが MU や EXU の動作にオーバーラップする場合の効率改善を狙ったものであるが、MM、store とともに restore よりも優先させてしまうと、粒度の荒い領域で効率低下を招いている。これは store の効果が小さくなり、restore 抑制の影響が表面化したためと考えられる。MM だけを restore より優先させた場合は restore 優先と大きな違いはない。

参考のため IBU でマッチング処理をおこなわない場合も図に示した (図中 without matching)。この場合、マッチングミス時のオーバーヘッドが表面化し、IBU でのマッチング処理の有効性を確認した。

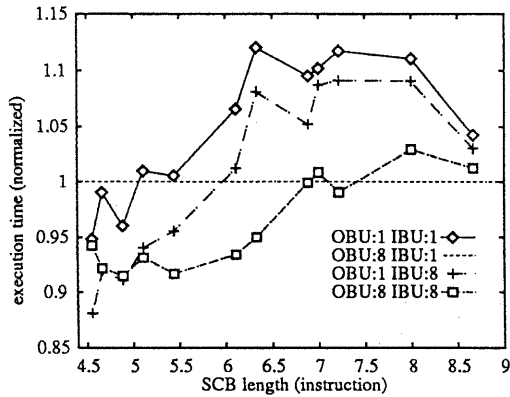


図 6: 内部 FIFO の組合せの効果

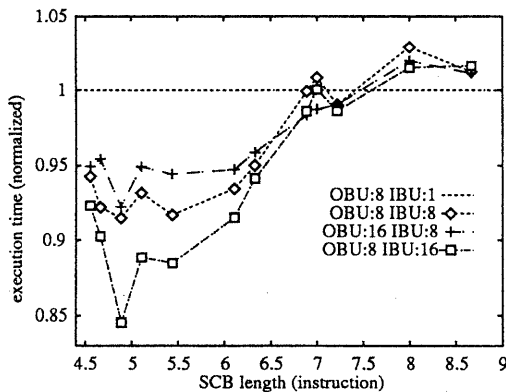


図 7: 内部 FIFO 増設の効果

4.2 OBU、IBU の FIFO サイズ

OBU と IBU の内部 FIFO サイズはネットワークインターフェースの入出力バランスを担う重要なパラメータである。FIFO に用いるゲート数には制限があり、その配分について考察する。

図 6 はそれぞれの FIFO の効果が、サイズが共に 1 のときに比べてどう現れるかを表したものである。なお実行時間はほかの図と比較のため OBU:8, IBU:1 の場合を基準として正規化されている。OBU:8, IBU:1 のグラフは、OBU はネットワーク負荷の小さい領域(粒度の荒い領域)で効果があることを示している。粒度が小さい領域では OBU の能力に対して過剰にパケットが生成されるため効率向上効果は得られない。また OBU:1, IBU:8 ではネットワーク負荷の大きい領域で実行時間が短縮されており、パケット取り込み能力拡大の効果が確認できる。OBU:8, IBU:8 の場合は両者の長所を得ることができる。粒度が小さい領

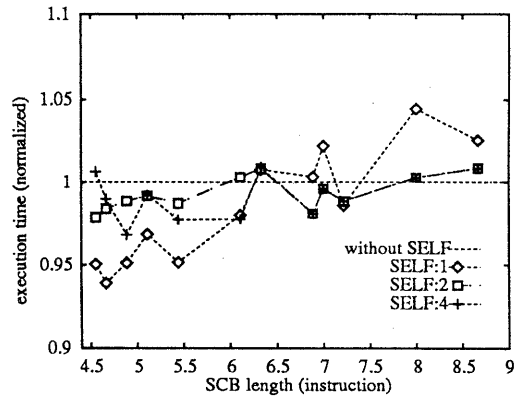


図 8: SELF バッファの効果

域で OBU サイズの増加によるパフォーマンスの低下が起きているのはパケットの過剰生成によるネットワーク負荷増大によるものと考えられる。

図 7 は OBU と IBU の FIFO をさらに増設した場合の効果を示している。これによれば OBU は単独で 8 以上増設しても効果はないばかりか、効率の低下をもたらす。ゲート数制限の下では IBU を増設する方がより効果的である。複雑になるためグラフには示していないが OBU:16, IBU:16 では OBU:8, IBU:16 より実行時間が長くなる。これは OBU を増設するには IBU をさらに増設することが必要であるため、OBU:16, IBU:32 で OBU 増設の効果が現れることがわかっている。これまでの考察と実現可能な FIFO 合計サイズを考慮して、設計に反映させる予定である。

4.3 OBU の自 PE 宛 パケットバッファ (SELF)

図 8 は SELF バッファの効果を示すシミュレーション結果である。粒度の小さい領域で、SELF バッファをつけることによる実行時間短縮が確認できる。4.2項では SELF のない OBU(NET バッファ)について検討したが SELF のサイズの単純な増加は NET と同様にパフォーマンスの低下をもたらす。今後、自 PE 宛パケットが連続して出力されるプログラムのように SELF の効果が期待できる場合での実験をおこなう予定である。

4.4 SU におけるネットワークとプロセッサの調停

図 9 は SU でのネットワークとプロセッサの調停方法の影響を示したものである。SU 内部では、IP 同士の要求の調停および IP と OBU の要求の調停がおこなわれる。OP, OPIBU の 2 種類の 3 入力マルチプレクサがあ

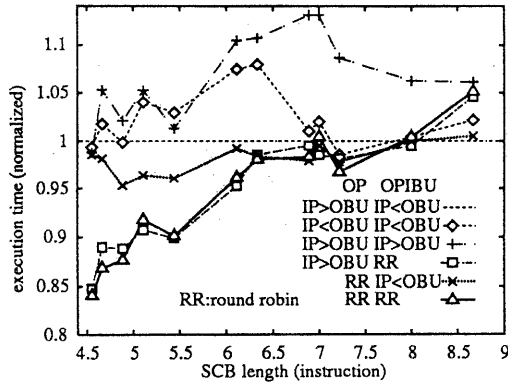


図 9: SU の調停方法の影響

り(図2)、まず IP 同士の調停がラウンドロビンによりおこなわれる。次に、調停された IP の要求と OBU の要求の調停がおこなわれる。このうち IP 対 OBU の調停方法の変更が評価の対象で、OP と OPIBU の調停方法をそれぞれ独立して設定してシミュレーションをおこなう。IP<OBU、IP>OBU、IP=OBU 間でラウンドロビン、の3種類がある。設計当初は OP : IP> OBU、OPIBU : IP<OBU であったが、図9に示すように OPIBU、あるいは OPIBU と OP の両方をラウンドロビンにした方がよいことがわかった。OPIBU で OBU 優先にすると IP から取り込まれるはずのバケットがネットワークで衝突しやすくなり、ネットワークの混雑を引き起こす。IP 優先の場合は OBU の要求がブロックされやすくなり、EXU の稼働率低下を引き起こす。OP で OBU 優先にするとネットワークへのバケット出力が増えネットワークの負荷増大を引き起こす。IP 優先では OBU の要求がブロックされやすくなる。いずれも優先度固定の場合はバケット出力スループットの低下を引き起こすが、ラウンドロビンでは適度な優先度分配がおこなわれ、スループット向上に寄与している。

プロセッサ動作の詳細な解析によれば、バケットの実レイテンシのうち、OBU 内でのバケット存在時間の割合が大きい(おおむね9割以上)。OPIBU でラウンドロビンをおこなうと実レイテンシが最大29%低減していた。そのため特に OPIBU での実行時間短縮効果が大きいことがわかった。

5 おわりに

シングルチッププロセッサ EMC-Y の設計において、ネットワークインタフェース部の最適化をシミュレーションの結果に基づいて考察した。その結果、新しいアーキテクチャの有効性が確認でき、ネットワークの性能を十分引

き出すヒントとなる設計パラメータが得られた。

改良点はおおむねネットワークの混雑に対して有効で、これによりネットワーク負荷の影響に左右されにくい効率の良い並列処理を実現できると考えられる。さらに、パラメータを制御可能にしてネットワークの負荷状況に対応して適切なパラメータを選択できれば高い性能を維持できると考えられ、そのインプリメントと選択方法を検討中である。

本稿ではフィボナッチ数を求めるという限定されたプログラムによって評価してきたが、それだけでは最適なプロセッサを設計するのに十分ではないため、現在全対全のランダム通信や優先バケット処理の評価を進めている。これらは EMC-Y アセンブリ言語を用いて記述されているが、今後 EM-X の基本並列核言語となる EM-C をもちいて各種アルゴリズムを記述した場合の評価もおこなっていく予定である。

現在、基準パラメータによる EMC-Y の設計はほぼ終了し、シミュレーションで評価した結果と進行中の研究成果を反映させた最適化をおこなっている。EMC-Y を今年度中に完成させ、80PE からなる並列計算機 EM-X を構築する予定である。

謝辞

本研究を遂行するにあたり御指導、御討論いただいた太田情報アーキテクチャ部長ならびに計算機方式研究室の同僚諸氏に感謝いたします。

参考文献

- [1] Yamaguchi, Y., Sakai, S., Hiraki, K., Kodama, Y. and Yuba, T. An Architectural Design of a Highly Parallel Dataflow Machine, Proc. of IFIP 89, pp.1155-1160, (1989).
- [2] Sakai, S., Yamaguchi, Y., Hiraki, K., Kodama, Y. and Yuba, T. Design of the dataflow single-chip processor EMC-R, J. Inf. Process., 13, 2, pp.165-173, (1990).
- [3] 児玉, 甲村, 佐藤, 坂井, 山口. 高並列処理向け要素プロセッサ EMC-Y の設計, JSPP '92, pp.329-336, (1992).
- [4] Kodama, K., Koumura, Y., Sato, M., Sakane, H., Sakai, S. and Yamaguchi, Y. EMC-Y : Parallel Processing Element Optimizing Communication and Computation, Proc. of ICS'93, pp.167-174, (1993).
- [5] 児玉, 佐藤, 坂根, 坂井, 山口. 高並列計算機 EM-X のアーキテクチャ, 情報処理学会 計算機アーキテクチャ研究会 101-7, pp.49-56, (1993).