

## リコンフィギュラブルシステム REX への並列計算機 EM-X の実装

児玉 祐悦<sup>†</sup> 片下 敏宏<sup>†</sup> 佐谷野 健二<sup>†</sup>

REX は大規模 FPGA を複数搭載し、並列計算機システム全体をプロセッサ内部からネットワーク構成まで自由に再構成することのできる、並列計算機アーキテクチャの評価や開発を支援する実験装置である。REX 上に 32 プロセッサからなる並列計算機システム EM-X を実装し、その評価を行ったので報告する。

### Implementation of parallel computer EM-X on reconfigurable system REX

YUETSU KODAMA,<sup>†</sup> TOSHIHIRO KATASHITA<sup>†</sup>  
and KENJI SAYANO<sup>†</sup>

REX is a reconfigurable experimental system for parallel computer systems. It consists of large-scale FPGAs, and enables the systems, from their processors to the network topology, reconfigure in order to support their evaluation and development. We report the implementation of the parallel computer EM-X, which includes 32 processors, on REX, and its evaluation.

#### 1. はじめに

計算機システムの開発期間を短縮するために、ハードウェアとソフトウェアを同時に開発する必要が生じることが多い。単体の計算機システムの場合には、ハードウェア部分をシミュレータやエミュレータを用いてコードデザインを行う環境が構築されてきている。しかし、比較的大規模な並列計算機システムを開発する場合には、既存のシミュレータやエミュレータでは対処できず、そのような開発環境を構築することは難しくかった。

近年、再構成可能デバイスである FPGA のゲート容量が急速に増加して、1 個の FPGA に簡単な RISC プロセッサなら複数搭載できるようになってきた。そこで我々は大規模 FPGA を複数搭載し、並列計算機システム全体をプロセッサ内部からネットワーク構成まで自由に再構成することができる、並列計算機アーキテクチャの評価や開発を支援する実験装置 REX を開発した。本稿では、REX 上に 32 プロセッサからなる並列計算機システム EM-X を実装し、その評価を行ったので報告する。

2 章で我々の考える並列処理実験装置に対する要件についてまとめる。次に、3 章で REX について、4 章で今回 REX に実装した並列計算機システム EM-X について述べる。4 章で REX への EM-X の実装並び

に、それを用いた評価について述べ、5 章でまとめを述べる。

#### 2. 並列処理実験装置に対する要件

並列計算機アーキテクチャの評価や開発を支援するための実験装置として、並列計算機システム全体を自由に再構成することを可能にし、高速にエミュレーションができる装置が有効である。我々が、このようなシステムに必要なと考えた項目は以下の通りである。

- (1) 柔軟性。特定の並列計算機のためのブレッドボードではなく、プロセッサ内部やネットワークインタフェース、ネットワークルータなどあらゆる部分を再構成可能であること。
- (2) ゲート規模。少なくとも 10 プロセッサ以上の並列計算機システムを実装できるだけのゲート規模を備える、あるいは拡張可能なものであること。
- (3) メモリ階層。プロセッサの高速化には、キャッシュシステムは不可欠である。レジスタ、一次キャッシュ、二次キャッシュ、主メモリといったメモリ階層を実現できるだけのメモリバンド幅を備えること。
- (4) 通信性能。並列計算機にとってプロセッサ間の通信性能は重要である。したがって、実験システムでも十分な性能の通信性能を引き出せるよう、メモリバンド幅と同等のネットワークバンド幅を持つこと。
- (5) 高速性。十分な規模のアプリケーションを走ら

<sup>†</sup> 産業技術総合研究所  
National Institute of Advanced Industrial Science and  
Technology (AIST)

せるだけの高速なエミュレーションが可能であること。

- (6) コストパフォーマンス。並列計算機アーキテクチャの研究目的に利用できる程度に安価に実現できること。

上記の要件を満たすシステムとして、既存のエミュレーション装置等が利用できないか検討した。

特定のターゲットシステムのためのブレッドボードは、上記(1)の要件を満たさない。プロトタイピングのためのFPGAボードは多数販売されているが、上記(2),(3),(4)の要件を満たさない。市販されているFPGAボードは比較的小規模のものが多く、たとえ複数のボードを接続できるものでも、ボード間の通信性能は非常に限られたものであったり、多種類のメモリを接続できなかつたりと全ての要件を満たすものはなかった。汎用エミュレーション装置は、(1),(2),(3)の要件を満たすが、(4),(5)の要件を満たさない。また、汎用エミュレータ装置は、デバッグ容易性を汎用FPGAとの差別化としているが、ゲート規模やエミュレーション速度の点では汎用FPGAの方が有利である。さらに、エミュレーション対象としている並列計算機システムでは各ノードは同一の構成を持っており、その特性を生かすことによりエミュレーション環境を構築することが容易となるが、汎用エミュレータ装置ではこの特性を生かすことができない。

上記の検討の結果、最新の大規模FPGAを用いて新たに並列処理実験装置を開発することが有効であると判断し、以下のREXの開発を行うこととした。ただし、ゲート数や動作速度などは最新の大規模FPGAで可能な範囲とすることにより、すぐに利用できる現実的な解を選択した。そのため、各プロセッサのゲート規模としてはキャッシュメモリなどを除き20万-50万ゲート程度の比較的シンプルなアーキテクチャとし、16プロセッサから32プロセッサ程度の並列計算機を数十MHz程度の動作速度でエミュレーションすることを想定した。これはエミュレータとしてはかなり高速であるとともに、実プロセッサの数十分の1程度の速度であり十分な規模のベンチマークを実行できると考えられる。これだけの規模の並列計算機をプロセッサからネットワークまで全て再構成可能な状態で、かつ高速にエミュレーションできるシステムはこれまでなかった。

### 3. リコンフィギュラブル並列処理実験装置 REX

図1にREXの構成を示す。REXはフルタワー型PCと同程度のサイズであり、同等のエミュレーション容量を持つ汎用エミュレータ装置と比べて非常にコンパクトである。REXシステムはホストPCと8枚のREXボードから成る。ホストPCとREXボード

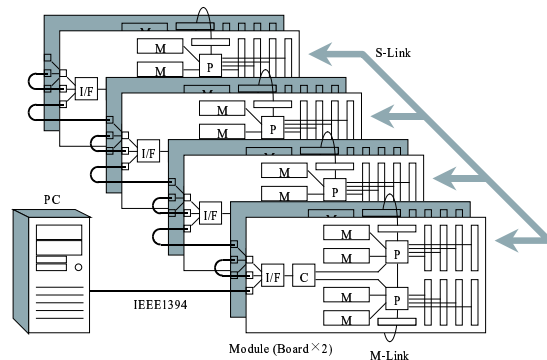


図1 REXシステムの構成

はIEEE1394により400Mbpsで接続される。ホストPCはFPGAのコンフィギュレーションデータのロードやプログラムデータの転送などREXシステムの制御を行う。

REX向けの特別なEDAツールは必要なく、汎用のFPGA向けツールを利用する。すなわち、各ノードの回路を単一のFPGAとして開発して、それをすべてのFPGAにロードする。汎用FPGAを用いて開発を行う際の大きな問題の1つはデバッグである。しかし、FPGA上にロジックアナライザの機能をマッピングする手法が開発されており、汎用FPGA用のツールとして利用可能である。このような新しいツールが簡単に利用できることが汎用FPGAおよび汎用開発環境を用いることの利点である。

2つのREXボードはボード上のコネクタにより接続される。これをモジュールと呼ぶ。モジュール内の4つのFPGAはM-linkによってリング状に接続される。M-linkは74bitのシンプルなリンクで、各ビットの入出力等は自由に設定できる。4つのモジュールはケーブルにより接続される。モジュール間の接続には21組の差動信号を1ポートとするS-linkが用いられ、各ボードに入出力各4ポートのS-linkがある。モジュール間ではリンクのビット数を大きくできないが、作動信号を用いることにより、転送速度をM-linkより高く設定することにより、モジュール内とモジュール間の通信スループットを同程度確保可能にしている。

表1はREXシステムの仕様である。32プロセッサの並列計算機システムをREXに実装することを想定している。プロセッサは簡単なRISCプロセッサであり、そのゲート数はネットワークスイッチを含んでおよそ10万ゲートである。これに、各命令/データ1次キャッシュが各16Kバイト、2次キャッシュが1Mバイト、主メモリが128Mバイトのメモリ階層を持つ。初期評価でMIPS R3000互換の簡単なRISCコアを単純に詰め込んだときには、1FPGAに10プロセッサまで搭載でき、それぞれは33MHz程度で動作することが確認されている<sup>1)</sup>。

#### 4. マルチスレッド並列計算機 EM-X

EM-X<sup>2)</sup> は電子技術総合研究所で開発されたマルチスレッド並列計算機である。ここでマルチスレッドとは、プログラムをいくつかのスレッドに分割し、あるスレッド実行がリモートアクセスなどでアイドル状態になると、他の実行可能なスレッドの実行に切り替えて、プロセッサをアクティブなままに保つ並列実行モデルである。共有メモリ計算機のオペレーティングシステム(OS)でいうマルチスレッド実行とは概念的には近いが、そのスレッド切り替えのタイミングが大きく異なる。OS レベルのマルチスレッドでは、スレッド切り替えのために数 100 マイクロ秒程度必要であるが、EM-X のマルチスレッドでは 1 マイクロ秒以下しか必要としない。このため、より細粒度な通信レイテンシも隠蔽することが可能である。

80 台のプロセッサから成る EM-X プロトタイプは 1995 年に完成した。ノードプロセッサとしては ASIC によって設計/開発された独自アーキテクチャのプロセッサ EMC-Y を用いている。このプロセッサには 1.0 ミクロンルールが用いられており、およそ 8 万ゲート、20MHz で動作を検証した。ネットワークルータ、プロセッサコアおよびメモリインタフェースをシングルチップに含んでおり、チップとメモリを接続するだけで容易に並列計算機を構成することができる。EM-X のプログラミング環境としてはマルチスレッド向けの拡張を施した C コンパイラを開発した。多くのマルチスレッド向けライブラリ関数を数命令で実現するとともに、それらの関数をインライン展開することにより並列オーバーヘッドが低く抑えられている。メモリ保

護機構と例外ハンドラにより、資源管理や入出力処理などを含む並列プログラム実行環境を備えている。

EM-X ではプロセッサ間の通信に、細粒度通信と呼ぶ固定長の小さなパケットのみを用いる。パケットはアドレス部とデータ部各 1 ワードからなり、1 つのパケットでは 32 ビットデータのみが転送される。したがって、より大きな、あるいは複雑なデータを転送するには、複数のパケットをソフトウェアにより組み合わせることが必要となる。これは RISC の思想と同様である。すなわち、ハードウェアは基本的に頻繁に使用される機能だけをサポートして、シンプルなハードウェアで性能を向上させるというものである。アドレス部はパケットタイプとグローバルメモリアドレスを含む。グローバルメモリアドレスは、プロセッサ番号とローカルメモリアドレスから成り、システム内のメモリアドレスを一意に指し示すことができる。EM-X にはパケットを作成/送出する send 命令がある。send 命令は、他の RISC 命令と同じく、2 つのソースレジスタと命令内の埋め込み型即値を用いて、1 クロックサイクルで実行を完了する。これにより、通信レイテンシは非常に小さく抑えられる。言い換えると、通信レイテンシを抑えるために、細粒度パケットのみを用いるアーキテクチャとなっている。EM-X のパケットは単にデータを転送するだけではなく、スレッドの起動、リモートメモリアクセス、局所同期などの機能も持っている。

#### 5. 評価

まず、REX にオリジナルの EM-X を実装して、次に性能比較を容易にするための変更を加えていく。

##### 5.1 REX への EM-X の実装

EM-X の当初の設計は、ASIC ベンダー独自のハードウェア記述言語を用いて行われたので、Verilog を用いて書き直した。最初に、FPGA 上でのゲート数を見積もるために、単一プロセッサを合成してみたところ、およそ 35 % のセル利用率であった。この合成には Xilinx ISE5.1.03i および XST を用いた。動作周波数は 10MHz であったが、これは主に浮動小数点演算ユニット (FPU) の遅延のためであった。FPU に関しては、以下で改良を検討する。

EM-X では 1 クロックサイクル中に 2 回のメモリアクセスを行っているため、CPU クロックの他に、2 倍の周波数のメモリクロックを必要とする。さらに以下で述べるように 4 倍の周波数の S-link クロックを必要とするため、3 種類のクロックを用いている。FPGA 内の DLL (Delay-Locked Loop) を組み合わせることにより定倍のクロックを生成できるので、いづれか一つのクロックを分配すれば良い。REX では 1 つのボードから他のボードにクロックを供給することにより同一クロックに同期したシステムを容易に構築できる。

表 1 REX システム仕様

項目	1 FPGA	システム
登録ゲート		
容量	2.5M ゲート (43K 論理セル+614K ビットメモリ)	40M ゲート
周波数	40MHz(想定)	-
メモリ		
SRAM 容量	2.25M バイト (576K バイト*4 ポート)	36M バイト
バンド幅	1.8G バイト/秒 (36 ビット*4 ポート*100MHz)	28.8G バイト/秒
DRAM 容量	512M バイト (128M バイト*2 ポート)	4G バイト
バンド幅	1.6G バイト/秒 (64 ビット*2 ポート*100MHz)	25.6G バイト/秒
FPGA 間の通信		
M-link	1.8G バイト/秒 (72 ビット*2 ポート*100MHz)	14.4G バイト/秒
S-link	1.6G バイト/秒 (16 ビット*4 ポート*200MHz)	12.8G バイト/秒

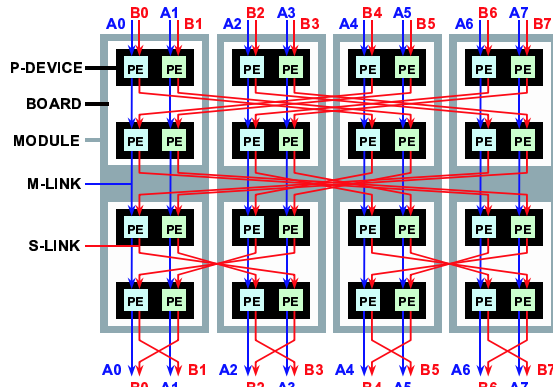


図 2 32PE 版 EM-X の REX 上の構成

しかし、CPU クロックは DLL の最低周波数の制限に違反するため、メモリクロックを分配するクロックとしている。ただし、メモリクロックから CPU クロックを生成するには注意が必要である。CPU クロックは全プロセッサで位相が同じである必要がある。しかし、単に DLL を用いて分周しただけでは位相が異ってしまう可能性がある。そのため、マスタ/スレーブ方式を用いて分周したクロックの位相を合わせる必要があるであった。

EM-X ではメモリインタフェースを簡素化するために主記憶として SRAM を用いている。このため、REX における実装でも主記憶として SRAM を用いる事とした。EM-X のメモリポートは 40 ビットであり、32 ビットのデータ、6 ビットのデータタグおよび 2 ビットのパリティからなる。一方、REX の SRAM ポートは 36 ビットしかない。このため、EM-X のメモリポートとして 2 つの SRAM ポートを並列に用いることとした。このため、EM-X プロトタイプでは各プロセッサあたり 1M ワードのメモリが搭載されているが、REX 上では各プロセッサあたり 128K ワードしかメモリが搭載できなかつた。

単一プロセッサのゲート利用率から、1 FPGA に 2 プロセッサを搭載することとし、図 2 のように各 FPGA に実装することとした。EM-X のネットワークはサーキュラーオメガ網と呼ばれるトポロジーを採用している。このネットワークにはリング構造が含まれており、同じそのリング構造を持つ M-link をその部分に割り当てている。

EM-X のネットワークポートは 44 ビット幅である。これは 39 ビットのデータ、2 ビットの要求制御信号、3 ビットのレディ制御信号からなる。これが FPGA あたり 2 プロセッサ分、すなわち入出力ともに 4 ポートずつ必要である。しかし、REX のネットワークポートのビット幅は、M-link、S-link とともにそれよりも少ない。そのため、M-link の転送レートを CPU クロックの 2 倍として実装することとした。具体的には、ク

ロックの前半で要求制御信号とデータの上位 7 ビットを、後半でデータの低位 32 ビットを送ることとし、レディ制御信号は分割せずに送るものとした。これにより 1 ポートに必要なビット数は 35 ビットなり、M-link で 2 ポート分送ることが可能になる。同様に、S-link は CPU クロックの 4 倍を転送レートとした。具体的には、要求制御信号と転送データを 16 ビットづつに 3 分割して送る。S-link 自体は、差動ペアを利用して 200MHz までのクロックが利用できるように設計されているのでまだ余裕がある。しかし、EM-X ではバッファはバッファ入力部のみであり、出力部にはバッファはないため、データ出力部が用意できるまでの遅延が大きく、4 倍のクロックには間に合わない。そのため、データ出力側にもバッファを持たせることによりこの遅延を小さくすることとした。このバッファを加えることによるゲート規模の増加は 2 プロセッサ分合わせて 5 % 未満であった。

上に述べた以外は REX に実装したプロセッサは、プロトタイプと同一である。ハードウェアの実装とともに、プログラミング環境およびランタイム環境の実装を行った。REX は C-device と IEEE1394 を通してホスト PC と接続されている。ホスト PC 上のプログラムにより、FPGA のコンフィギュレーションデータのロードやリセットが行える。これらの機能はライブラリ化されており、ホスト PC 上のプログラムは容易である。また、EM-X のネットワークは P-device 上のバッファ変換回路によりホスト PC と接続されており、ホスト PC から EM-X にデータを送ったり、EM-X からホスト PC にデータを受け取ることができる。これを用いて、EM-X にプログラムをロード/実行したり、実行結果を取り出すことができる。さらに、EM-X 上の例外ハンドラによりホスト PC との間でファイル I/O システムコールが実現されており、ホスト PC 上のファイルの読み書きが可能である。

## 5.2 EM-X/REX を用いた評価

まず REX に実装した EM-X(EM-X/REX) 上で、mp3d を用いた評価を行う。mp3d はスタンフォード大学で作成された共有メモリ計算機向け並列ベンチマーク集 SPLASH に含まれる 3 次元粒子シミュレーションを行うプログラムである。プログラムは、空間をセルに分割し、各粒子の速度と位置を計算し、同じ空間セルにある粒子との確率的なモデルを用いて衝突をシミュレーションする。これを時間ステップ毎に繰り返す。粒子を各プロセッサに割り当て、プロセッサに分散させた空間セルへはリモートアクセスを行うことにより並列化している。このため、小さなリモートデータが頻繁にアクセスされるネットワーク負荷の高いプログラムとなっている。mp3d のプログラムはすでに EM-X プロトタイプ上で作成されているため、ほとんど修正なしで動かすことが可能であった。

ただ 1 つ変更が必要だったのは実行時間計測の方式

表 2 MP3D の実行結果

スレッド数	実行時間 (秒)	EXE (%)	PWAIT (%)	LS (%)	IDLE (%)
1	2.12	45.6	0.1	0.4	53.9
2	1.43	69.7	0.3	0.6	29.4
4	1.19	83.7	1.2	0.7	14.5
6	1.17	85.8	5.2	0.7	8.3
8	1.15	86.6	6.8	0.7	5.9
10	1.17	86.0	7.7	0.7	5.6

である。EM-X には時間取得の機能がなく、プロトタイプではホストインタフェース上のプロセッサを用いて、時間計測をソフトウェアにより実現していた。時間計測自体は単純なカウンタで実現できるので、REX の再構成機能により簡単に追加できる。EM-X にはデバッグや性能測定用に、プロセッサ内部の全てのフリップフロップにアクセスするメンテナンス機構が備わっている。このカウンタをメンテナンス機構からアクセスして、プログラムから自由に時間を取得できるようにライブラリを整備した。

時間計測はクロック数を数えるだけであるが、プロセッサの実行状態の統計情報を取得するように簡単に拡張することができる。EM-X プロセッサには、IDLE、EXEC、PWAIT、LS の 4 つの実行状態がある。PWAIT はネットワークビジー等によりパイプラインが止まっていることを、LS は実行に 2 サイクル必要な命令の 2 サイクル目を実行していることを、EXEC はそれ以外の命令実行を、IDLE は実行する命令がないことをそれぞれ表わす。各プロセッサの状態を計測するカウンタを追加した。

EM-X/REX 上の 32 プロセッサを使用して 25600 個の粒子のシミュレーションを行った。上記の状態カウンタによる実行状況を表 2 に示す。表では、各プロセッサで動作させるスレッド数を変化させた場合の各実行状況の違いを示している。4 スレッド程度までは、マルチスレッドによるレイテンシ隠蔽の効果により、実行時間が減少しているが、6 スレッドを越えたところから、あまり実行時間の減少はみられず、パケット出力待ちの割合が増加していることが分かる。これは、マルチスレッドによりネットワークへの負荷が増大し、ネットワークが飽和に近くなっていることを示していると思われる。これを検証するため、さらにネットワークの混雑度を測定するように変更を加えることも容易である。

EM-X プロトタイプでも、メンテナンス機構を用いて、実行状態を 2 ミリ秒に 1 回程度のサンプリングにより取得することはできるが、やや精度にかけていた。また、一定周期にクロックを停止して実行状態を取得することも可能であるが、多大な時間 (毎サイクル停止では 4 万分の 1 以下の速度) を必要とした。さらに、元々プロセッサに組込まれていない情報は取得することが困難である。ユーザが観測したいと思う情

表 3 FPU パイプライン化の影響

	1 段 FPU	2 段 FPU
セル利用率	75 %	75 %
最大周波数	10 MHz	18 MHz
mp3d 実行時間	1.19 秒	0.70 秒 (性能比 1.7)

報は、時と場合によって大きく異なるため、システムのデザイン時にそれらをあらかじめ選択するのは非常に難しい。しかし、REX の再構成機能を用いれば、望みの情報を得るように簡単にハードウェアを追加できる。

次に、プロセッサの高速化について検討する。クロックスピードが 10MHz に抑えられているのは、浮動小数点演算ユニット (FPU) がパイプライン化されていないためである。そこで FPU の改良について検討を行う。FPU は既に数個のサブユニットに分割して記述されているので、パイプラインレジスタの追加と若干の制御の変更で、FPU 自体は容易にパイプライン化できる。しかし、EM-X では整数レジスタと浮動小数点レジスタは分離していないため、レジスタ書き込みの競合が生じる。また、深いパイプラインを有効に利用するためにはコンパイラ側の改良も不可欠である。ここでは、単純に FPU を 2 段のパイプラインとして、FPU パイプラインの 2 段目を実行中は演算パイプラインを停めることにより、他の命令実行へ影響を与えない方式を考える。この方式では、クロック周波数が元のクロックの 2 倍になれば、FPU 性能は同じで他の命令実行性能は 2 倍になる。クロック周波数が 2 倍よりも低ければ、FPU 性能はもとよりも低下してしまい、総実行性能は FPU 命令の比率により変化する。この FPU を 2 段にしたプロセッサの実装結果を表 3 に示す。これによると、セルの利用率はほとんど変化がなく、パイプライン周波数は 1.8 倍になっている。mp3d の実行 (表 2 スレッド数 4 の場合) では性能は 1.7 倍となった。FPGA による速度向上はあくまでも目安であり、ASIC で実装したときにこの通りの性能を出せるわけではない。しかし、パイプラインが  $n$  倍になった時の実アプリケーションで性能が類推できるわけで、設計/開発には役立つと考えられる。

### 5.3 汎用エミュレータとの比較

REX は 10MHz で EM-X をエミュレーションできる。この速度を汎用エミュレータ装置である Quickturn 社の CoBALT<sup>4)</sup> と比較を行う。

CoBALT はサイクルベースシミュレーションコードを直接実行する専用プロセッサによるエミュレーションシステムである。エミュレーションボードには、64 個の専用チップが搭載され、各チップには 64 台のプロセッサが内蔵されており、合計 4096 個のプロセッサからなる。各プロセッサは 100MHz で動作し、各実行サイクルでは、3 入力 1 出力のルックアップテーブル (LUT) をエミュレートする。各プロセッサは 128 サイクル毎に 128 個の別々の LUT をエミュレートす

表 4 CoBALT によるエミュレーション速度

実装プロセッサ数	CoBALT(KHz)	VCS (KHz)
1	47.1	1.00
2	46.5	0.43
4	37.3	0.23
6	31.4	0.17

ることができ、1 ボードでは最大 50 万 LUT(100 万ゲート相当) を 768KHz の速度でエミュレーションできる。回路が 128 サイクルに収まらない場合には、複数のステップに分割してエミュレーションすることにより、より大きな回路もエミュレーションできるが、それだけエミュレーション速度は低下する。その他、プロセッサ内では 64K ビット相当のメモリ機能が、ボード上のメモリモジュールでは最大 72M バイトのメモリ機能がエミュレーションできる。

CoBALT にも EM-X アーキテクチャを実装して評価を行った。EM-X のプロセッサ 1 台を実装したところ、ゲート利用率は 9 % であったが、合成ツール (QuestII v3.1) の制限のため最大でも 6 プロセッサシステムまでしか実装できなかった。

表 4 は、実装プロセッサ台数を変化させた場合のエミュレーション速度を、ターゲットの動作周波数により示している。6 プロセッサシステムのエミュレーション速度は、31.4KHz と REX の 1/300 と低いものであった。このエミュレーションは STB モードと呼ばれる、テストベンチをエミュレータ上に合成して実行するモードを利用しており、エミュレータの速度を最大限に引き出している。最大動作周波数 768KHz に対してエミュレーション速度が 31.4KHz と低いのは、主に 2 つの原因による。1 つは、EM-X では主メモリとして非同期メモリを用いているのに対して、CoBALT では大規模メモリとしてレイテンシ付きの同期メモリしかサポートしておらず、その対応をとるために EM-X のクロックを 4 分周したものを CoBALT のクロックとして用いているためである。もう 1 つは、EM-X の回路のクリティカルパス長が 128 サイクルを越えており、1 プロセッサのエミュレーションにも 4 ステップ (128\*4=512 サイクル) 必要であったためである。また、プロセッサ数が多くなるにつれてエミュレーションスピードが低下していることも分かった。それに対し、REX では 32 台のプロセッサまで同じ 10MHz でエミュレーションできる。

シミュレーションによるスピードは汎用エミュレーションよりもさらに桁違いに遅い。表 4 には VCS(v3.1.1) というコンパイルベース方式を用いたシミュレータの結果も示している。これは Ultra Sparc II(296MHz) で実行した結果である。また、REX へのコンフィグレーションデータを開発する際に利用した ModelSim PE(5.7a) でもシミュレーション速度の評価を行った。Pentium4 2.8GHz で 1 FPGA 上の 2

プロセッサをシミュレーションした場合、RTL シミュレーション速度は 508Hz とほぼ VCS と同等であった。また、タイミングシミュレーションでは 11Hz とさらに低下した。さらに表 4 でも分かる通り、シミュレーション時間はプロセッサ数に比例して増大するため、ソフトウェアシミュレーションでは十分な規模のプログラムを実行することは不可能である。

## 6. おわりに

EM-X 並列計算機アーキテクチャを実装して、リコンフィギュラブル並列実験システム REX の評価を示した。REX は大規模 FPGA を用いた論理エミュレータであり、特に並列計算機システムを評価する目的で設計された。REX 上に 32 プロセッサからなる並列計算機システム EM-X を実装して、全体で 300 万ゲートを越える並列システム全体を全て再構成可能な状態で、他のエミュレータやシミュレータより遥かに高速な 10MHz という動作速度でプログラム開発/システム評価を行えることを示した。また、REX の再構成可能という特性を生かして、各種統計量取得のためのカウンタを加えたり、内部アーキテクチャを変更したりといった評価を行い、その有効性を確認した。今後は、REX をチップマルチプロセッサ向けプロセッサアーキテクチャ評価およびロボット視覚システムアクセラレータ等にも利用していく予定である。

### 謝辞

本研究にあたり、有益なコメントをいただいた EM-X の開発メンバおよび REX の利用者に感謝します。また、本研究の遂行をサポートいただいた関口智嗣グリッド技術研究センター長ならびに工藤宏クラスタ技術チーム長に感謝します。

### 参考文献

- 1) 佐谷野, 片下, 小池, 児玉, 坂根, 甲村, “大容量 FPGA の応用によるマルチプロセッサエミュレーションシステムの評価,” 情報処理学会計算機アーキテクチャ研究会, 2001-ARC-144, pp.25-30, July 2001.
- 2) Y. Kodama, H. Sakane, M. Sato, H. Yamana, S. Sakai and Y. Yamaguchi, “The EM-X Parallel Computer: Architecture and Basic Performance,” Proc. the 22nd Annual Int. Symp. on Computer Architecture, pp.14-23, June 1995.
- 3) Xilinx Inc., “Virtex-E 1.8V extended memory field programmable gate arrays,” DS025-1(v1.5), July 2002.
- 4) R.Goeing, “Quickturn offers compiled-code emulation systems,” eetimes, 1997.