計算機アーキテクチャ 139－26
（２０００．８．５）

MIPSベースマルチスレッドプロセッサのFPGAによる実装と評価

佐谷野健二[†] 児玉祐悦[†] 坂根広史[†,††] 山口喜教[†,†††]

[†] 電子技術総合研究所 情報アーキテクチャ部
[††] 電機通信大学大学院 情報システム学研究科
[†††] 筑波大学 電子・情報工学系

本研究では，FPGA上にMIPSアーキテクチャをベースとしたプロセッサコアの実装を行い，このプロセッサコアにEM-Xのマルチスレッド処理メカニズムを試験的に導入した．一般に広く使用されているMIPSアーキテクチャにEM-Xのマルチスレッド処理メカニズムを導入することで，これらのメカニズムの有用性を検証するとともに，パイプライン段数の増加による高周波数動作に対応したプロセッサの開発を目指す．またEM-Xのネットワークルータに対しても高周波数動作に対応する為の変更を行い，プロセッサ全体の性能のバランスを考慮して改良を行った．

Implementation and Evaluation of a Multi-Threading Processor Based on MIPS Architecture by Using FPGA

KENJI SAYANO,[†] YUETSU KODAMA,[†] HIROFUMI SAKANE,[†,††] and YOSHINORI YAMAGUCHI[†,†††]

[†] Computer Science Division, Electrotechnical Laboratory
[††] Graduate School of Information Systems, The University of Electro-Communications
[†††] Institute of Information Sciences and Electronics, University of Tsukuba

In this research we developed a MIPS based processor by using FPGA in which the multi-threading mechanisms of EM-X ware introduced experimentally. The goal of this research is to prove the efficiency of the mechanisms and to enhance the multi-threading processor architecture with long pipeline approach, by introducing the multi-threading mechanisms of EM-X into ordinary MIPS architecture. Also, the network router was enhanced to operate on higher frequency to keep the performance balance of network and processor.

1. はじめに

分散メモリ型マルチプロセッサによる並列処理技術がハイパフォーマンスコンピューティングの分野で広く用いられている現在においても，PE数の増大に伴うプロセッサの動作効率の低下は依然として解決すべき重要な課題となっている．また，多数のプロセッサを効率良く動作させる為のメカニズムとしては，マルチスレッド処理が有力な選択肢の一つであると考えられている．

この様な状況の中，EM-X[1)-4)]ではオーバーヘッドの小さなスレッド切り替え機構や細粒度なパケット通信といった特徴的なアプローチによって，効率の高いマルチスレッド処理を実現している．発展の目覚しい昨今の半導体技術を用いれば，EM-Xの持つ優れた特徴を継承しながら，より高性能な並列計算機の実現が期待できる．

この一方で，近年FPGAの高集積化と高速化が急速に進行している．AlteraやXilinxといった米国企業の間で熾烈な開発競争が行われていることが，集積度と速度の向上に関して強力な促進要因となっている．この為，現時点においても数百万のシステムゲート規模を持つデバイスが利用可能な状況にあり，今後もさらなる向上が期待される．これにより，従来はプログラマブルデバイスでの実装が困難であった高機能なマイクロプロセッサを，1つのFPGAの中に実装することも十分に可能となっている．

また，FPGAを用いれば基板実装後もプロセッサの構造を自由に変更可能であることから，従来はシミュレータを使って行われていた方式検討も，実機に近い環境で行うことが可能である．そこで我々は，この様な大容量のFPGAを利用して，EM-Xで培われた高度なマルチスレッド処理技

術を応用したマイクロプロセッサの開発を進めている．

本研究では，MIPS [5] アーキテクチャをベースとしたプロセッサコアを開発し，このプロセッサコアにEM-Xの要素プロセッサであるEMC-Y [6),7)] のメカニズムを試験的に導入したマルチスレッドプロセッサ(以下MTR)の開発を進めている．一般に広く使用されている MIPS プロセッサにEM-X のメカニズムを導入することで，これらのマルチスレッド処理メカニズムの有用性を検証するとともに，パイプライン段数の増加によるより高い周波数の動作に対応したプロセッサの開発を目指す．また，EMC-Y のネットワークルータに対しても，より高い周波数での動作に対応した変更を行い，プロセッサ全体の性能のバランスを考慮して改良を行った．

以降，2.では本研究で開発を進めているマルチスレッドプロセッサに導入されたメカニズムについて述べ，3.では，このマルチスレッドプロセッサの実装について述べる．さらに4.では，FPGA をターゲットとして EDA ツールで論理合成，配置配線を行った結果を，EMC-Y の場合と比較，考察する．

2. マルチスレッド処理機構

本研究では，EMC-Y が持つマルチスレッド処理機構の中から，主要なメカニズムであるスレッドの起動，リモートメモリアクセス，パケットの送信に絞って MTR に実装し実験を行った．以下ではこれらのメカニズムについて説明する．

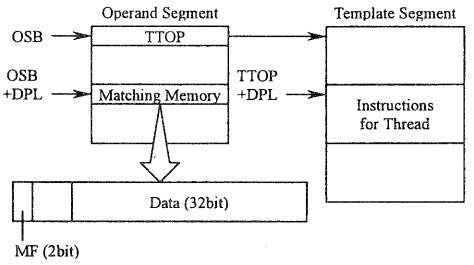

図 1　スレッド起動時のメモリアクセス
(OSB: operand segment base, DPL: displacement, TTOP: template top, MF: matching memory flag.)

2.1 スレッドの起動

スレッドの起動は，ネットワークを経由して受信されたパケットによって行われる．この際，スレッドの起動方法には，バイナリマッチングを用いた同期処理を伴うものと，パケット間の同期をとらずに起動するものの2種類に分けられる．

図 1は，スレッド起動処理の際の，メモリアクセスの手順を示している．operand segment と template segment は，各々がプロセッサのメインメモリ上に配置されている．図中の OSB と DPL はパケット内のフィールドで指定される値である．

同期処理を伴わないスレッド起動では，スレッド起動パケットの到着後に TTOP が読み出されて，TTOP+DPL が示すアドレスから直にスレッドが起動される．同期処理を伴うスレッド起動で

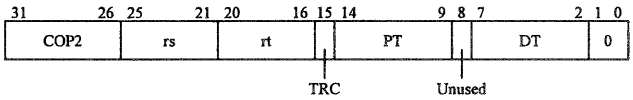

図 2　MTR のパケット送信命令のフォーマット
(TRC: trace bit, PT: packet type, DT: data tag.)

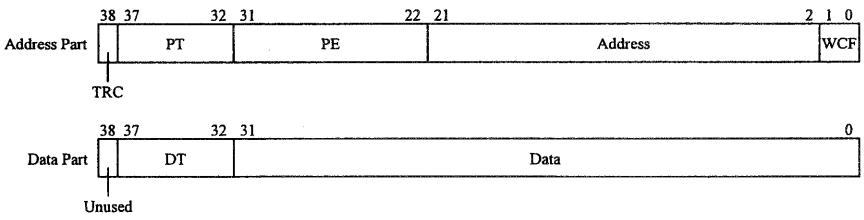

図 3　EMC-Y と MTR のパケットフォーマット
(TRC: trace bit, PT: packet type, PE: target PE number, WCF: wait condition flag, DT: data tag.)

は，「左」と「右」という名前で区別される2種類のパケットが揃った場合にスレッドの起動が行われる．

同期処理を伴うスレッド起動では，スレッド起動パケットの到着後にmatching memoryが読み出されてMFの値がチェックされる．MFは2つのパケットが揃ったかどうかをチェックする為のフラグであり，初期状態ではクリアされている．この状態で左か右のパケットが到着すると，左右どちらのパケットが到着したかという情報がMFに書き込まれる．次にもう一方のパケットが到着すると，先に書きこまれていたMFの値が再びチェックされ，左右が揃えばMFはクリアされる．この時点ではじめてTTOPの値が読み出され，TTOP+DPLが示すアドレスからスレッドの実行が開始される．

2.2 リモートメモリアクセス

PE間のリモートメモリアクセスを行うパケットにはSYSWR，SYSRDの2種類が用意されている．SYSWRはワードデータのリモート書き込みを行うパケット，SYSRDはワードデータのリモート読み出しを行うパケットである．あるPEから他のPEにSYSWRパケットが送信されると，このパケットを受け取ったPEでは，パケット中のフィールドで指定されたメモリアドレスに，パケット中のデータフィールドの値が書き込まれる．また，SYSRDパケットが送信されると，このパケットを受け取ったPEでは，パケット中のフィールドで指定されたアドレスのメモリが読み出され，この値が再びパケットとしてSYSRDパケットを送信したPEに送り返される．

2.3 パケットの送信

MTRでは，MIPSコアにパケット送信命令が追加されている．このパケット送信命令は，コプロセッサ命令として拡張されている．MTRのパケット送信命令のフォーマットを図2に，EMC-YとMTRのパケットのフォーマットを図3に示す．

通信パケットは，アドレス部とデータ部で構成され，1つのパケットが2サイクルで処理される．パケット送信命令ではrsフィールドで指定されたレジスタの内容がパケットアドレス部のPE，Address，WCFの部分を指定し，rtフィールドで指定されたレジスタの内容がパケットデータ部のDataフィールドを指定する．パケット送信命令中のTRC，PT，Unused，DTフィールドは，そのままパケット内の同一名のフィールドに対応する．この命令により，パケットの送信は1命令で行われる．

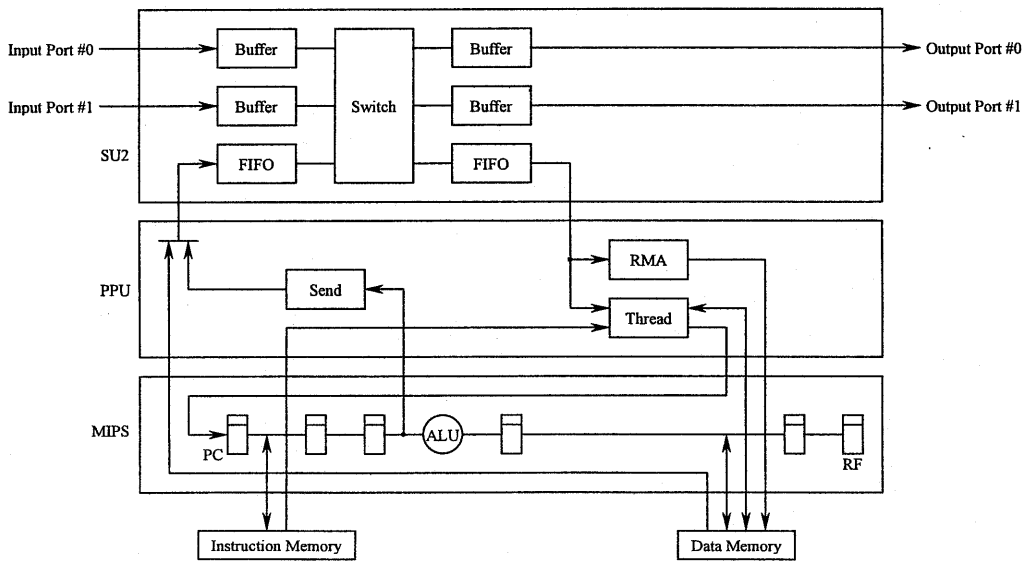

図 4　MTRの内部ブロック図
(SU2: 2-cycle latency switching unit, PPU: packet processing unit, MIPS: MIPS architecture processor core, RMA: remote memory access, PC: program counter, RF: register file.)

3. プロセッサの実装

図 4に MTR 内部のブロック図を示す．このプロセッサは，ルータ部分(図中 SU2)とプロセッサ部分(図中 MIPS)，そしてこれらをつなぐパケット処理ユニット(図中 PPU)から構成されている．ネットワークを通して SU2 に届いたパケットは SU2 内の FIFO に溜められ，PPU によって順に処理される．

PPU 内では，外部から到着したパケットに対して，スレッドの起動とリモートメモリアクセスに分けて処理が行われる．図中の thread と書かれたブロックでは，**2.1**で述べたスレッド起動時の処理が行われる．スレッド起動の際にアクセスされるメモリは，プロセッサのデータメモリと共有されている．スレッドの起動は，プロセッサの PC を直接変更することで行われる．また，スレッドの終了判定は，命令メモリ中のスレッド継続ビットを読むことで行われる．

リモートメモリアクセスは，図中の RMA と書かれたブロックで処理される．リモートメモリアクセスの場合も，スレッド起動の場合と同様に，プロセッサのデータメモリがアクセスされる．データメモリのアクセスに関してはプロセッサの優先度が最も高く，スレッド起動時のメモリアクセスやリモートメモリアクセスは，プロセッサがメモリアクセスを行っている場合には待たされる．

SYSWR パケットの場合はパケットで指定された内容がメモリに書き込まれ，SYSRD パケットの場合はメモリから読み出された内容が PPU から SU2 の FIFO に送られる．また，パケット送信命令によって送られるパケットも，図中の send と書かれたブロックによって SU2 の FIFO に送られる．

3.1 MIPS ベースプロセッサコア

プロセッサコアは MIPS の命令セットに基づいて独自に開発を行った．プロセッサコアのブロック図を図 5に示す．このプロセッサはシングル issue のシンプルな 5 段パイプライン構成を持つ．汎用レジスタとデータパスは，ともに 32 ビット幅である．EMC-Y のプロセッサ部分は 2 段パイプライン構造であった為，パイプライン段数の増加による動作周波数の向上が見込まれる．尚，現時点ではキャッシュコントローラが未実装である為，命令とデータは各々独立したメモリ格納される．

プロセッサ部分はごく基本的な構成を持つが，スレッドの起動をプロセッサ外部から制御する必要があることから，PC の値を操作する為のデータパス(図中 instruction address)が追加されている．また，パケット送信命令を実行する為に，プロセッサ内の ALU ステージの命令とオペランドを出力するパス(図中 instruction / operand)も追加した．この変更は，MIPS アーキテクチャにおける一般的なコプロセッサの拡張に準じている．

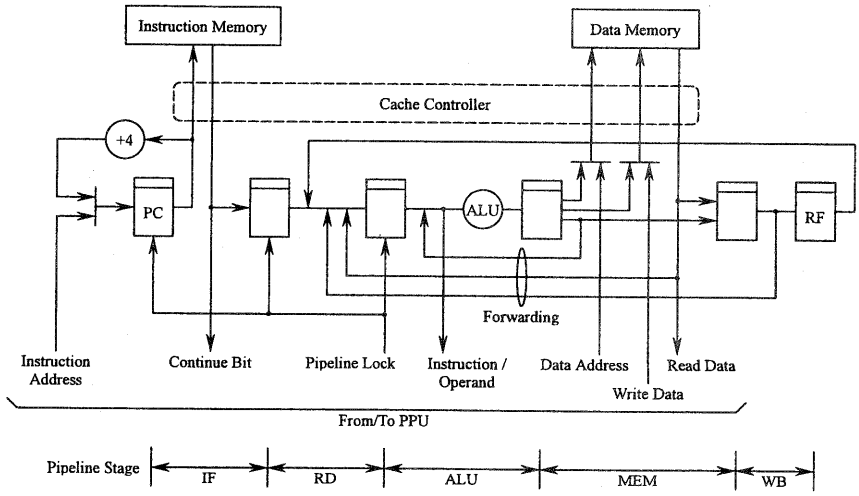

図 5 MIPS ベースプロセッサコアのブロック図
(PC: program counter, ALU: arithmetic & logic operation unit, RF: register file, IF: instruction fetch, RD: register read, ALU: ALU operation, MEM: memory access, WB: register write back.)

3.2 ルータ部分

図 6にMTRのルータ部分のブロック図を示す. MTRの通信ポートは, EMC-Yと同様に入出力各2ポート(図中#0と#1)となっており, 通信方向は片方向のみである. EMC-Yのルータが1ステージ構成であるのに対して, MTRのルータは2ステージ構成となっている. 各ステージは, デッドロックを防ぐ目的からパケットバッファが3重化されている.

MTRのパケットバッファが2ステージ構成となっているのは, プロセッサコアのパイプライン段数の増加に伴って動作クロックの向上が予想される為, ルータ部分の動作クロックとのバランスを保つことを目的としている.

通信ポートから入力されたパケットは, 一旦1ステージめのパケットバッファに溜められ, ルータ内のセレクタでルーティングを行った後, 2ステージめのパケットバッファに再び溜められる. 目的の PE に到着したパケットはルータ内のFIFOに溜められ, PEから出力されるパケットはFIFOから読み出される.

4. 実験と評価

実験では, AlteraのFPGA[8]をターゲットとして論理合成, 配置配線を行った. 表 1に実験で使用した FPGA の仕様を示す. システムゲート数(表中 maximum system gates)は1,052,000ゲートであるが, 回路を実装した際の実効的なゲート規模(表中 typical gates)は400,000ゲートとなっている. このデバイスは, 16,640 個のロジックセル(表中 logic elements)を持ち, 各々のロジックセルは4入力1出力のLUTと1ビットのFFがペアとなった構造を持つ. また, メモリ専用セルのビット数(表中 memory bits)はトータルで212,992ビットである.

表 1 実験で使用したAltera EP20K400Eの仕様

	EP20K400E
Maximum System Gates	1,052,000
Typical Gates	400,000
Logic Elements	16,640
Memory Bits	212,992

論理合成にはSynopsysのFPGA Compiler IIを, 配置配線にはAlteraのQuartusを使用した. 実機上での動作検証には, FPGAとメモリを搭載した実験基板[9]をEM-Xに接続して使用している. また, この実験の為に, 上記の実験装置に搭載可能なFPGAをAlteraのFLEXシリーズからAPEXシリーズに拡張する為のアダプタを開発して使用した.

論理合成と配置配線の結果を表 2に示す. 論理合成時には, ソースコード中のモジュールの階層構造を残して最適化を行った場合(表中 hierachical)と, 階層構造を残さずに最適化を行った場合(表中 flat)の2種類について測定を行った. どちらのプロセッサでも, 階層構造を残さずに最適化を行った方が, 回路規模が小さく動作クロック周波数の高い回路が合成されている.

ロジックセルの使用量(表中 logic elements)は

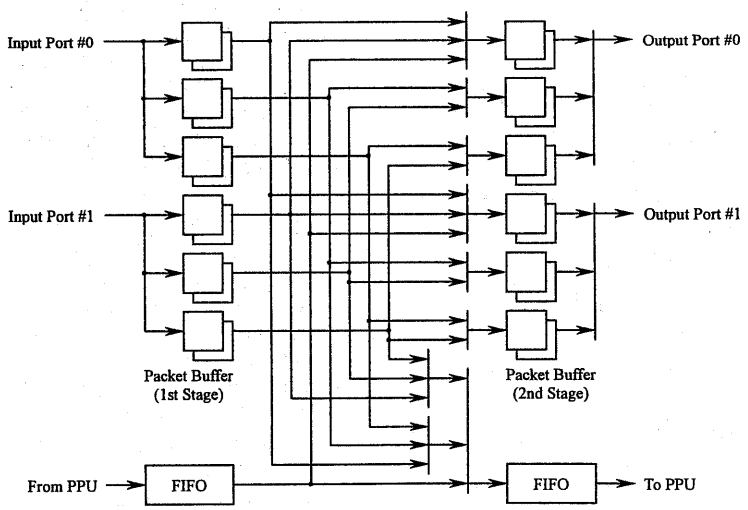

図 6 ルータ部分のブロック図

表 2　MTR と EMC-Y の論理合成，配置配線結果

	MTR		EMC-Y	
Synthesis Style	Hierarchical	Flat	Hierachical	Flat
Logic Elements	7,863	6,473	9,561	7,848
Memory Bits	23,040	23,040	29,312	29,312
FMAX(MHz)	28.10	31.10	15.14	21.88

hierachical，flat とも MTR の方が少なくなっている．**3.2**で述べた様に，MTR ではパケットバッファの 2 ステージ化が行われており，EMC-Y に比べてルータ部分の回路が複雑になっている．その一方で，プロセッサコアにシンプルな MIPS アーキテクチャを用いている為，プロセッサコアのサイズが EMC-Y に比べて小さくなっており，結果として全体のロジックセルの使用量が 20%程度縮小されている．

MTR のメモリセルの使用量(表中 memory bits)は EMC-Y に対して 20%程度減少している．これは，EMC-Y のパケット受信用 FIFO がパケットの種類に対応して 2 系統に分かれているのに対して，MTR では 1 系統のみの実装となっていることが大きな要因である．このため，MTR のパケット受信用 FIFO の構造を EMC-Y と同様にすれば，メモリセルの使用量も同程度になるものと思われる．

MTR の動作周波数の最大値(表中 FMAX)を，EMC-Y を基準として比較すると，階層構造を残した場合で 1.86 倍，階層構造を残さない場合で 1.42 倍の向上が見られた．プロセッサのパイプライン段数の増加と，ネットワークルータの 2 ステージ化の効果が現れている．

5. まとめ

MIPS アーキテクチャをベースとしたプロセッサに EMC-Y のマルチスレッド処理機構を導入した結果，FPGA 上の実装では EMC-Y に対して動作周波数の大幅な向上が得られた．また，プロセッサコアにシンプルな構造の MIPS アーキテクチャを採用することで回路規模が縮小された．通信バッファの 2 ステージ化を行うことで動作周波数が向上しているが，これに伴う回路規模の増大をシンプルなプロセッサコアを用いることで押さえることができた．

現時点では，実験装置が単一の FPGA による実装にしか対応していない為，マルチプロセッサシステムの実装や動作検証は行っていないが，今後はマルチプロセッサに対応した実験装置の開発を行い並列動作時の検証を行う予定である．

謝　辞

本研究を進めるにあたり，貴重なご意見とご指導を頂いた電子技術総合研究所情報アーキテクチャ部の大蒔和仁部長と，同部並列アーキテクチャラボの皆様に感謝致します．また，本研究で FPGA を使用する為の実験装置は画像技研，同装置に Altera の APEX デバイスを搭載する為のアダプタは昭英電機の各社に開発して頂きました．ここに感謝の意を表します．

参　考　文　献

1) 児玉祐悦，佐藤三久，坂根広史，坂井修一，山口喜教，“高並列計算機 EM-X のアーキテクチャ，” 情報処理学会研究報告，ARC-101-7, pp. 49-56, 1993.
2) Y. Kodama, H. Sakane, M. Saito, H. Yamana, S. Sakai, and Y. Yamaguchi, “The EM-X Parallel Computer: Architecture and Basic Performance,” ISCA’95, pp. 14-23, 1995.
3) 児玉祐悦，坂根広史，佐藤三久，坂井修一，山口喜教，“高並列計算機 EM-X のリモートメモリ参照機構の評価，” 情報処理学会論文誌，Vol. 36, No. 7, pp. 1691-1699, 1995.
4) Y. Kodama, H. Sakane, M. Sato, H. Yamana, S. Sakai, and Y. Yamaguchi, “Message-Based Efficient Remote Memory Access on a Highly Parallel Computer EM-X,” IEICE Tans. Inf. & Syst., Vol. E79-D, No. 8, 1996.
5) G. Kane and J. Heinrich, “MIPS RISC Architecture,” Prentice Hall, 1992.
6) 児玉祐悦，甲村康人，佐藤三久，坂井修一，山口喜教，“高並列処理向け要素プロセッサ EMC-Y の設計，” 並列処理シンポジウム JSPP’92, pp. 329-336, 1992.
7) Y. Kodama, Y. Koumura, M. Sato, H. Sakane, S. Sakai, and Y. Yamaguchi, “EMC-Y: Parallel Processing Element Optimizing Communication and Computation,” ICS’93, pp. 167-174, 1993.
8) “APEX20K Programmable Logic Device Family,” Altera Corporation, 2000.
9) 佐谷野健二，児玉祐悦，坂根広史，山口喜教，“並列計算機ノードプロセッサの FPGA を用いた実装と評価，” 情報処理学会研究報告，ARC-134-9, pp. 49-53, 1999.