

超低消費電力化 イベント駆動型ヘテロジニアスチップマルチプロセッサ

黒田亮太郎[†], 内田裕之^{††}, Jihane B. Abderrazak^{††},

富安洋史^{††}, 西川博昭^{††}

[†]筑波大学第三学群情報学類

^{††}筑波大学大学院システム情報工学研究科

著者らはこれまでネットワーク向きデータ駆動プロセッサである CUE シリーズの研究を行ってきた。現在、CUE シリーズで得た知見を基に超低消費電力化イベント駆動型ヘテロジニアスチップマルチプロセッサの設計開発を進めている。本プロセッサは効率のよい小型で簡素なプロセッサを多数集積し、必要に応じて作動プロセッサ数を切り替えることによって、低消費電力と十分な性能を両立することを目指す。本報告ではこのプロセッサの設計方針とその目標を示し、これに基づく設計概要について述べる。

Ultra-Low-Power Event-Driven Heterogeneous Chip Multi-Processor

Ryotaro Kuroda[†], Hiroyuki Uchida^{††}, Jihane B. Abderrazak^{††},

Hiroshi Tomiyasu^{††} and Hiroaki Nishikawa^{††}

[†]College of Information Sciences, Third Cluster of Colleges, University of Tsukuba

^{††}Graduate School of Systems and Information Engineering, University of Tsukuba

The authors have been studying CUE-series networking-oriented data-driven processors. An ultra-low-power event-driven heterogeneous chip multi-processor is currently being designed by evaluating the latest version event-driven chip multi-processor; CUE-v3. In order to meet both ultra-low power consumption and sufficient performance, the processor discussed will be realized as a many-core architecture, even though each processor itself is not so powerful. This paper discusses design philosophy and logical structure of the ultra-low-power event-driven heterogeneous chip multi-processor.

1. 序論

近年ネットワーク上を流れるデータ量は日を追うごとに増大している¹⁾。将来的にインターネットの維持に必要な電力の消費量は莫大なものとなるはずであり²⁾、早急なネットワークキング処理の低消費電力化が求められている。

ネットワーク全体の低消費電力化を図るためには、ネットワークの構造はもとより、各ノード上で動作するOSやアプリケーションまでを含めた考察が必要であるが、著者らは特にプロセッサの低消費電力化に注力しネットワークキング処理向き低電力化プロセッサの設計開発を行っている。本プロセッサでは著者らがこ

れまで設計してきたネットワーク向きデータ駆動プロセッサであるCUEシリーズ³⁾⁻¹¹⁾で得た知見を基に効率を高め低消費電力化を目指す。

2. 本プロセッサの目的

本プロセッサは低消費電力化とネットワーク処理に対する最適化を目的として設計開発を行う。CUE シリーズは性能の向上を目標として開発され、その目標を実現してきた。特に CUE-v3 は CUE-v2 を Processing Element (以下 PE)として4つ搭載したクアドコアチップマルチプロセッサであり、高い性能を実現できた。

しかし、CUE シリーズは純粋に性能を追求してきたため、消費電力についての取り組みは施されていない。

ネットワーク処理向きプロセッサは低消費電力化の要求が強いため、新たに設計開発を行うプロセッサは電力あたりの処理能力の向上と、待機時の低消費電力化を主な目標とする。

具体的な目標として CUE-v3 の電力消費の10分の1程度の電力消費を目指す。電力あたりの性能は命令セットの見直しとパイプラインの効率向上によって改善を図り、待機時の消費電力はスリープモードを細かく適用することによって削減を目指す。

3. 本プロセッサの概要

本プロセッサはデータ駆動と制御駆動とのヘテロジニアスチップマルチプロセッサである。本章では本プロセッサの構成とスリープ機能、制御駆動プロセッサの概要について述べる。

3.1. 本プロセッサの構成

データ駆動プロセッサは応答性が高く、制御駆動プロセッサは逐次処理のスループットが高い。CUE-v2でもデータ駆動と制御駆動を同一パイプラインで実行することにより両方の長所を得ることに成功している。

それに対して本プロセッサではデータ駆動と制御駆動を別種の PE としてヘテロジニアス構成をとり、それぞれを複数搭載する。専用の PE とする理由は、PE あたりの回路規模を抑えるためである。一般に PE 一つあたりの機

能が多くなるほど回路規模は大きくなる。しかし、一つの処理がプロセッサの回路全体を利用することはなく、その処理に関係する部分だけを利用するため、回路規模が大きいほど電力的なロスが大きくなる。

本プロセッサではデータ駆動プロセッサと制御駆動プロセッサを別個なものとして各々の PE の回路規模を小型にし、必要な部分のみを動作させることによって電力的なロスを抑える。また、使用する PE 数を変えることによって幅広い負荷変動に対応する。特にモバイル機器では使用状態によって負荷が大きく変動するため、これに対応する電力制御は重要である。

3.2. スリープ機能

本プロセッサは待機時の消費電力を抑えるためにスリープ機能の搭載を行う。

動作させる PE の数やスリープモードの選択はソフトウェアスケジューラによって行われる。スケジューラは、各 PE の負荷と処理待ちプロセスの量から各 PE の動作モードを決定する。これを補助するためハードウェアによって各 PE の負荷を測定する機構を設ける。この機構は、各 PE の FIFO に蓄えられているパケット数やパイプラインの充填率などを積算してレジスタに保持することによって実現する。

スリープの状態は数段階用意する。現段階では5つの状態を考えている。スリープの状態が深い順に、

- ・ 完全なスリープ状態
- ・ 瞬時に起動することができるが内部メモリの読み出しを保障しない状態
- ・ 瞬時に起動し内部メモリの読み出しを保障する状態
- ・ 電源電圧とクロックレートを下げた状態
- ・ 通常の動作を行う状態

である。

3.3. 制御駆動プロセッサの概要

本プロセッサの制御駆動プロセッサについて述べる。CUE-v2は2命令同時発行のスーパースカラであるため、命令レベル並列性を利用して性能を高めることができたが、ハードウェアが大きくなり消費電力あたりの効率は芳しくなかった。本プロセッサにおける制御駆動プ

ロセッサはシングルパイプラインに戻した上で高効率化を図ることにより、消費電力あたりの効率の向上を実現する。

本制御駆動プロセッサの命令は可変長命令である。レジスタのみを使い即値を使わない場合は、短い16ビット長の基本命令で処理を行うことができる。即値を使う場合は(16+16)ビットの拡張命令を用いる。このように命令長を短くすることで命令メモリから読み出すデータを小さくすることができ、読み出しに掛かる電力消費を下げるができる。

パイプラインはステージ数を多くし、各ステージの行う処理を軽くし、クロックレートを高めにする。リーク電流はトランジスタの占める面積に比例するため、同一の処理能力であれば並列度を上げるよりクロックレートを上げた方が消費電力が小さくなる傾向がある。

パイプラインが深くなると分岐時のペナルティも大きくなるため、分岐予測機構を搭載する。分岐予測機構はわずかな面積の増加で大きな性能向上が見込めるため電力あたりの効率向上も期待できる。

4. データ駆動プロセッサの構成と低消費電力化機能

本章ではデータ駆動プロセッサの構成と低消費電力化機能について述べる。CUE-v2では発火処理に非効率的な部分が存在した。これらを改善するためのパイプラインと命令の構成について述べる。また、新たに搭載する逐次処理機構とその目的について述べる。

4.1. 命令セット

本節では本プロセッサの命令について述べる。本プロセッサでは命令メモリのアクセスに関する改良と新しい命令の追加を行い、低消費電力化とネットワーク処理能力の向上を実現する。

4.1.1. 命令形式

メモリからの命令の読み出しには電力を消費するため、メモリ上のプログラムの大きさを小さくすることは低消費電力化に有効である。

本プロセッサにおける命令の基本形式を図1に示す。本プロセッサの命令の長さは基本部16ビットと拡張部16ビットの計32ビットで

ある。2オペランドの整数演算、浮動小数点演算、シフト、論理演算命令は基本部16ビットのみで処理を行う。この場合拡張部16ビットの読み出しは行わないので、その分のメモリアクセスに必要な電力消費を削減することができる。即値を用いる場合や分岐を行う場合などは拡張部16ビットが必要となる。

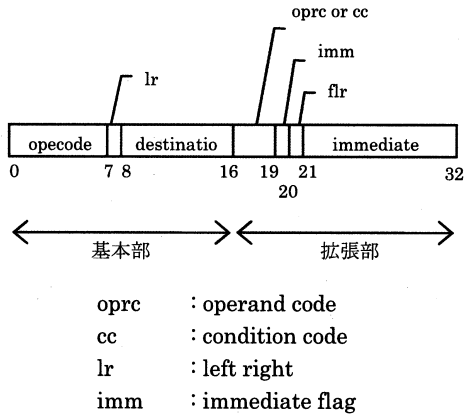


図1 命令の基本形式

4.1.2. 新規追加命令

命令セットは新規のものにネットワーク処理でよく使われる処理を行う命令や2つの命令の機能を一つの命令へまとめたものを追加することにより処理効率を向上させる。処理能力が向上すればスリープ状態の時間を伸ばすことにもつながり、低消費電力化に有効である。

ここでは一例としてマスク処理を行う命令を挙げる。この命令は2つの即値を用いる。5ビットの即値で始まりを指定し、もう一方の即値でビット数を指定することにより範囲を指定しマスクを取る。本プロセッサのオペランドは32ビットであるので、5ビットあれば0から31の数字を指定でき32ビット中どの範囲でもマスクを取ることができる。ヘッダ処理等で1ワード中に複数データが含まれている場合にマスクを取るケースは頻繁に現れる。1命令において自由にマスクを取る範囲を指定できれば、マスク値を用意する必要がなく効率の向上が見込める。

4.2. パイプライン構成

本節では本プロセッサのパイプライン構成について述べる。CUE-v2 のパイプラインと今回設計を行ったデータ駆動プロセッサのパイプラインをそれぞれ図 2, 図 3 に示す。本プロセッサのパイプラインとの比較を容易にするため CUE-v2 のパイプライン図からは制御駆動のみにかかわるものを除外している。

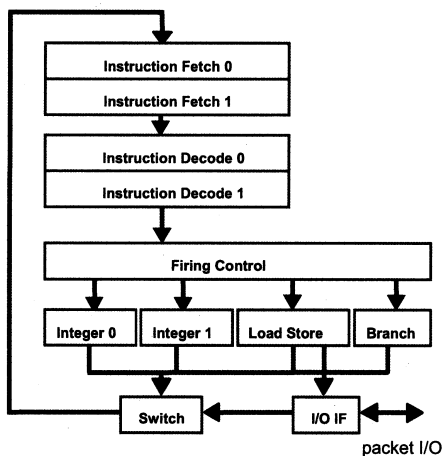


図 2 CUE-v2 パイプライン

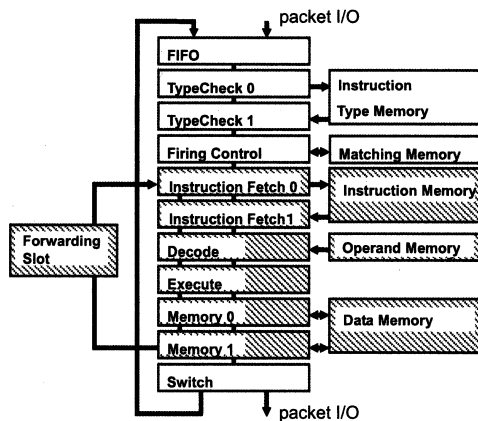


図 3 パイプライン構成

データ駆動プロセッサとしてみると CUE-v2, CUE-v3 とも循環パイプライン型であり、Matching Memory (以下 MM)を用いてオペランドの待ち合わせをすることによって命令の実行を制御する。本プロセッサでは CUE-v3 に比してオペランドマッチングをパイプライ

ンの始めの方で行うこととし、オペランドを格納する場所を Operand Memory (以下 OM)として MM から分離した。これによりオペランドが通るステージを減らし消費電力を抑えることができる。

これらにともない2オペランド命令であるか否か、また32ビット命令であるか否かを示すフラグをチェックするステージである Type Check (以下 TC)を設けた。

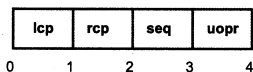
Forwarding Slot (以下 FS)は後述する逐次処理で用いられるオペランド保持機構である。

4.2.1. オペランドマッチングと命令読み出しの改善

データ駆動では 2 オペランド命令の場合、2 つの packets が揃わなければ発火しない。これは、2 オペランド命令の packets が到着しても二回に一回は発火しないということである。CUE-v2 では Instruction Fetch (以下 IF)と Decode (以下 Dec)の後に Firing Control (以下 FC)があるため、発火しなかった場合 IF で読み出し Dec で処理した命令は破棄されることになる。そのため、メモリアクセスとデコードに掛かった電力が無駄になってしまう。本プロセッサではこれを防ぐために FC を IF と Dec の前に移した。このように命令読み出しはオペランドマッチングの後であることが望ましいが、それでは FC に packets が来た段階で 1 オペランド命令と 2 オペランド命令の区別が付かない。そこで、命令の情報の一部(図 4 参照)を持つ小型のメモリ Instruction Type Memory (以下 ITM)とそれにアクセスするパイプラインステージ TC をパイプラインの先頭近くに搭載する形をとることにした。この ITM は命令メモリに保存される命令と一対一の関係で対応し、1 オペランド命令判別フラグと逐次実行開始フラグ、データ駆動でオペランドのコピーを示すフラグを持つ。

1 オペランド命令判別フラグ(uni-oprand)は 1 オペランドであることを示すフラグである。このフラグを用いて 1 オペランドであることを発火処理よりも前に判別し、必要に応じてマッチング処理を行うためのものである。

逐次処理開始フラグ(sequential execution)は本プロセッサが新たに搭載する逐次処理機構の動作を開始させるためのフラグである。こ



lcp : left copy
 rcp : right copy
 seq : sequential execution
 uopr : uni-operand

図4 ITM 保持データ

の機構の詳細については次節で述べる。逐次処理開始のフラグを命令メモリから読み出す場合、パイプライン構成の関係上逐次処理の最初の命令を読み出した段階で逐次処理の対象ではないパッケージが逐次処理パイプラインに流入していることになる。そのため、1ステージ分処理の巻き戻しが必要となり、行った処理が無駄になってしまう。このことから、フェッチを行うよりも前に逐次処理を行うことを知っておくことが望ましい。逐次処理機構の動作開始を示すこのフラグをITMから読み出すことによりフェッチを行うよりも前に逐次処理の開始を知ることができる。

ITMにはすべての命令がアクセスすることとなるのでアクセス分の電力消費が増加することとなるが、このメモリのサイズは小さなものであるため消費する電力は小さく、フラグを有効に用いて処理を助けることで、電力の増加以上の電力の低減を得ることができる。2オペランド命令ばかりを処理する場合このメモリへのアクセスのオーバーヘッドの方が大きくなってしまいが、即値を用いた算術演算やロードが1オペランド命令に含まれるのでこのような状況はあまり生じないと考えられる。

4.3. 逐次処理機構

本プロセッサは簡素な逐次処理機構を搭載する。発火制御を行わず逐次的に命令を発行することによって並列度の小さいプログラムのスループットを上げることができる。また、逐次処理中は発火制御を行わないためMMの動作率を下げるができる。MMは内部に多数の比較器を持つ連想記憶であり回路規模が大きいため、この使用頻度を抑えることは重要である。

逐次処理は図4.2中の斜線が引かれているパイプラインとメモリにより実行され、逐次処理

開始のフラグを持つ命令の発火を契機に開始される。逐次処理中に新しいパッケージが送られてきた場合、そのパッケージはFIFOに蓄えられる。逐次処理終了のための命令が処理されれば逐次処理状態は解除され、通常の動作に戻る。逐次処理時は中間結果をFSに蓄え、必要なタイミングにあわせてIFへと吐き出すことによってオペランドを供給する。FSからデータを吐き出すタイミングは、データ駆動命令中で指定された次命令のアドレスから算出する。

この方法により発火制御なしに複数の連続した命令を処理することができる。逐次処理が開始されれば逐次処理機構に含まれるパイプラインステージのみが動作する状態となり、それ以外のステージに存在するデータは保持される。

逐次処理を用いることでのデメリットも存在する。それは、処理中の応答性の悪化である。処理中は排他的に処理することになるため、動作中新たに受け取ったパッケージに対する応答性が悪化する可能性がある。また、逐次処理終了時に1命令処理が必要となることから逐次処理を行う命令数が極少ない場合かえって効率が悪くなる。連続した命令発行をどの程度にすれば良好な応答性を維持したまま逐次処理の効率を向上できるかは今後の課題である。

5. 今後の展開

本報告では、新しく設計開発を行うプロセッサについて述べた。また、データ駆動プロセッサについては低電力化のための方法について検討した。命令とパイプライン構成はメモリから読み出す情報の削減により低消費電力化を行い、局所的逐次処理はMMの使用率を下げるにより低消費電力化を行う。

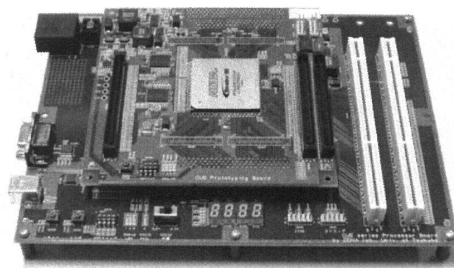


図5 FPGAを搭載したプロトタイプボード

現在, これらの仕様を検討した段階であるので, 実装についてのより詳細な議論が必要である. その上で設計開発を行い, FPGA において動作検証を行う予定である. このため, 新たに大規模 FPGA を実装した図 5 に示すプロトタイピング用ボードを開発した.

謝辞

本研究の一部は, 戦略的創造研究推進事業 (CREST) の支援による共同研究により遂行したものである. また, CUE-v3 は総務省戦略的情報通信研究開発推進制度 (SCOPE), CUE-v2 は半導体理工学研究センター (STARC) の支援による共同研究によって試作したものである. また, これら全ての開発, および性能評価に用いたツールは, 東京大学大規模集積システム設計教育研究センターを通し, シノプシス株式会社の協力により提供されたものである.

参考文献

- 1) 総務省情報通信政策研究所, 「インターネット概観統計集(平成18年改訂)」(インターネット研究会報告)
<http://www.soumu.go.jp/iicp/chousakenkyu/data/research/survey/telecom/2007/2007-1-01-2.pdf> (2007).
- 2) 小笠原敦, "科学技術動向研究レポート 1: 情報通信のエネルギー問題 - 求められる通信インフラの省電力化 -" 科学技術動向・月報, 2006年6月号, 文部科学省 科学技術政策研究所 科学技術動向研究センター,
http://www.nistep.go.jp/achiev/ftx/jpn/stfc/stt063j/0606_03_featurearticles/0606fa01/200606_fa01.html (2006).
- 3) S. Ito, R. Kurebayashi, H. Tomiyasu and H. Nishikawa: A Processor Architecture for Simultaneously Processing Dataflow and Controlflow Threads, Proceedings of the 15th IASTED International Conference on Parallel and Distributed Computing and Systems, pp.339-344 (2003).
- 4) 青木一浩, 工藤慎也, 西川博昭: ボトルネックのないレイヤ2/3 間インタフェースのデータ駆動型実現法とその実験的検討,

- 電子情報通信学会論文誌 D-I, J87-D-I 巻, 5 号, pp.591-598 (2004).
- 5) 伊藤伸也, 野本祥平, 富安洋史, 西川博昭: データ駆動・制御駆動スレッドを同時・多重処理するプロセッサ CUE-v2 の LSI 試作, 電子情報通信学会論文誌 D-I, J88-D-I 巻, 2 号, pp.113-124 (2005).
 - 6) H. Nishikawa: Design Philosophy of a Networking-Oriented Data-Driven Processor: CUE, IEICE Transactions on Electronics, Vol.E89-C No.3, pp.221-229 (2006).
 - 7) 西川博昭, 我孫子泰祐, " タグ操作を許すデータ駆動プログラムの開発・再利用支援手法, " 電子情報通信学会論文誌 D-I, J85-D-I 巻, 3 号, pp.294-302 (2002, 3).
 - 8) 樽林亮介, 西川博昭, " データ駆動プロセッサによる実時間処理のためのプログラム割当手法, " 電子情報通信学会論文誌 D-I, J86-D-I 巻, 10 号, pp.721-732 (2003).
 - 9) 富安洋史, 岡本政信, 西川博昭, " ネットワーキング向きイベント駆動型チップマルチプロセッサ CUE-v3 の性能予測, " 情報処理学会論文誌, 48巻 SIG 13(ACS 19) 号, pp.49-57 (2007).
 - 10) Hiroaki Nishikawa, Hiroshi Tomiyasu, Hiroyuki Uchida: VLSI Design of Networking-Oriented Chip Multi-Processor: CUE-v3, Proceedings of the 2008 International Conference on Parallel and Distributed Processing Techniques and Applications, 655-661 (2008).
 - 11) Hiroaki Nishikawa, Hiroshi Tomiyasu, Masanobu Okamoto, Masayoshi Sugiyama, Hiroyuki Uchida, Osamu Mizuno, Hiroshi Ishii, Makoto Iwata, CUE-v3: Data-Driven Chip Multi-Processor for Ad hoc and Ubiquitous Networking Environment, Proceedings of the 2007 International Conference on Parallel and Distributed Processing Techniques and Applications, 623-629 (2007).