

卒業論文

題目

可変パイプライン段数プロセッサの
細粒度切り換えコントローラの
ハードウェア設計とその評価

指導教員

佐々木敬泰 助教

平成22年度

三重大学 工学部 情報工学科
計算機アーキテクチャ研究室

目方徹 (407853)

内容梗概

近年、モバイルプロセッサの高性能化に伴い、消費電力が増大するという問題が生じている。消費電力が多いと、発熱、故障のトラブルや、ノートパソコンなどのバッテリー持続時間の減少の原因となってしまうためである。そのため、現在様々な高性能かつ低消費電力を実現する手法が提案されている。

その手法の一つとして、可変パイプライン段数アーキテクチャ (VSP: Variable Stages Pipeline) が提案されている。VSP はプロセッサにかかる負荷が高い場合にはパイプライン段数を増やし高周波数で動作させ、負荷が低い場合にはパイプライン段数を減らし低周波数で動作させるアーキテクチャである。

VSP を実現するためには、実行時に負荷の変動に応じて最適なパイプライン段数を予測し、制御を行うコントローラが必要となる。そのコントローラとして、メインメモリへのアクセス回数に着目して予測を行う細粒度切り換えコントローラが提案されている。しかし、従来の研究では、細粒度切り換えコントローラの評価をプロセッサシミュレータである SimpleScalar を用いて行っており、正確な電力評価やコントローラ追加に必要なハードウェア量の評価が行われていなかった。そこで、本研究では細粒度切り換えコントローラを MIPS R3000 命令互換プロセッサに実装し、評価を行った。その結果、電力遅延積において高速モードと比較して約 63 %、低消費電力モードと比較して約 46 % の改善が得られ、ハードウェア量はプロセッサ全体に対して 1 % 未満であることがわかった。

Abstract

Recently, the increase of the power consumption caused by enhancement of mobile processors becomes a serious problem. It causes the decrease of battery life of laptop PC. So achievement of both high-performance and low energy is demanded. Therefore various high-performance and low energy methods have been proposed.

As one of the high-performance and low energy methods, Variable Stages Pipeline (VSP) is proposed. In order to achieve them, VSP changes the number of pipeline stages and frequency dynamically. When workload for processor is high, VSP increases the number of pipeline stages and runs at high clock frequency. On the other hand, when workload is light, VSP decreases the number of pipeline depth by unifying the pipeline stages and runs at low clock frequency.

In order to use VSP effectively, it needs mode controller that predict the suitable number of pipeline stages according to workload for processor. Therefore fine-grain controller, which predicts using the number of main memory access, is proposed. However, the controller is not implemented, and accurate power consumption and hardware cost are not evaluated. This paper implements fine-grain controller on the MIPS R3000 compatible processor and evaluates the power consumption and hardware costs. According to our evaluation results, the processor with proposed controller can achieve about 63% and 46% better performance than high speed mode and low energy mode, respectively. And the additional hardware cost is less 1% of the processor.

目次

1	背景	1
2	可変パイプライン段数プロセッサ	2
2.1	パイプラインプロセッサに関する概括	2
2.2	可変パイプライン段数プロセッサの概要	4
2.3	パイプライン統合手法	7
2.4	VSP について	7
2.4.1	LDS-cell	9
3	モード切り換えコントローラ	13
3.1	細粒度切り換えについて	14
3.2	モード予測器	15
3.3	切り換えオーバーヘッド	17
3.4	切り換えオーバーヘッド削減手法	18
4	ハードウェア実装	20
4.1	本研究の目的	20
4.2	実装	20
5	消費エネルギーとエネルギー遅延積	23
5.1	消費エネルギーの定義	23
5.2	電力遅延積の定義	24
6	評価	25
6.1	電力遅延積の評価	25
6.2	ハードウェア規模の評価	26
7	まとめ	29
	謝辞	29
	参考文献	30

図目次

2.1	パイプラインプロセッサの動作	3
2.2	モード別パイプライン段数	5
2.3	高速モード	6
2.4	低消費電力モード	6
2.5	D-FF+MUX	8
2.6	VSP プロセッサの構成図	10
2.7	LDS-cell	12
2.8	グリッチ緩和	12
3.9	切り換え周期における違い	14
3.10	切り換えコントローラの概要	16
3.11	削減された加算部分	16
3.12	モード別パイプライン段数	19
4.13	オーバーヘッド削減手法を含めたハードウェア構成	22
5.14	実行時間と消費電力	24
6.15	閾値別電力遅延積	27
6.16	モード別電力遅延積	27
6.17	タイミングチャート	27

表 目 次

4.1	主な信号線の意味	21
6.2	面積結果	28
6.3	面積割合	28

1 背景

近年，プロセッサの高性能化に伴い消費電力が増加しているため高性能と低消費電力の両立が求められている．消費電力の増加は発熱の増加やバッテリー持続時間の減少につながる．

そのため，プロセッサの負荷に応じてプロセッサの動作を切換えることで，高性能かつ低消費電力を実現する手法が提案されている．その手法の一つとして，可変パイプライン段数プロセッサが提案されている．可変パイプライン段数プロセッサとは負荷の変動に応じてパイプライン段数と周波数を動的に切り換えることで低消費電力化を実現する方法である．パイプラインレジスタへのクロック供給の電力は膨大であるため，可変パイプライン段数プロセッサでは，性能があまり必要でない場合にはパイプライン段数を少なくし，クロック供給を行うパイプラインレジスタの個数を減らすことで低消費電力を実現している．この可変パイプライン段数プロセッサの一手法として VSP(Variable Stages Pipeline) [1] [2] [3] が提案されている．

可変パイプライン段数プロセッサである VSP を実用化するためには最適なパイプライン段数を予測し，パイプライン段数を動的に切り換えるコントローラが必要となる．既に Yao らによって切り換えコントローラは

提案されている [4] が, 切替間隔が長いと細かい負荷の変動に対応できない. そのため細粒度に切り換えを行うコントローラが本研究室で提案されている [5]. しかし, このコントローラはプロセッサシミュレータを用いての評価しか行われておらず, ハードウェア実装が行われていない.

そこで, 本論文では細粒度切り換えコントローラのハードウェア実装を行う. また, 細粒度に切り換えを行った場合, 段数切り換えにかかるオーバーヘッドが無視できないので, パイプライン段数変更のオーバーヘッド削減手法が提案されており [5], この手法のハードウェア実装も行う.

細粒度切り換えコントローラを VSP に実装し, 評価を行った結果, 電力遅延積において高速モードと比較して約 63 %, 低消費電力モードと比較して約 46 % の改善が得られた. また, 細粒度切り換えコントローラとオーバーヘッド削減手法のハードウェア量を評価した結果, VSP プロセッサと比較するとハードウェア量は 1 % 未満であるという結果が得られた.

2 可変パイプライン段数プロセッサ

2.1 パイプラインプロセッサに関する概括

プロセッサ内では, 命令の読み込み (Fetch), 解釈 (Decode), レジスタ読み込み (Register Read), 実行 (Execution), メモリへのアクセス (Memory

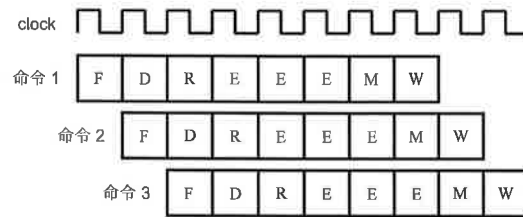


図 2.1: パイプラインプロセッサの動作

Access), 結果の書き込み (Write Back) などの複数の段階的な処理を行うことにより 1 つの命令が処理される。パイプラインの動作を図 2.1 に示す。パイプラインの動作を図 2.1 では Fetch を F, Decode を D, Register Read を R, Execution を E, Memory Access を M, Write Back を W としている。図 2.1 に示すように、それぞれの段階において処理は異なる回路で処理されるため、1 クロックで段階ごとに並列に命令を動作させるプロセッサをパイプラインプロセッサと呼ぶ。パイプラインプロセッサではパイプライン段数を分けることにより、1 クロックで動作させるべき回路は小さくなるため高周波数にすることができ、単位時間当たりに実行できる命令が増えることから高性能を実現している。しかし、パイプライン段数が増えれば、ステージ間で命令の処理内容を記憶しておくためのレジスタが必要となるため、消費電力は増えるという欠点がある。

2.2 可変パイプライン段数プロセッサの概要

可変パイプライン段数プロセッサとは、パイプライン段数と周波数を負荷に応じて動的に切り換えることにより、負荷が高い場合には高性能を発揮させ、負荷が低い場合には性能を低下させる代わりに消費電力を抑えるという手法である。可変パイプライン段数プロセッサには本研究室で提案している VSP とパイプラインステージ統合 (PSU: Pipeline Stage Unification) [6] [7] [8] [9] などの方式が提案されている。これらの可変パイプライン段数プロセッサは以下のような特徴を備えている。

- 多段パイプライン段数時、性能や動作はパイプライン段数が可変でないプロセッサと同等である。
- 少段パイプライン段数時、パイプラインレジスタへのクロック供給の必要がなくなるため、クロックドライバの消費電力低減に繋がる。
- 少段パイプライン段数時、データ依存による待ちサイクルの低減、分岐予測ミスペナルティ低減に繋がる。

VSP などの可変パイプライン段数プロセッサでは、パイプライン段数を 2 種類以上へ切り換え可能であるが、今後、説明を簡潔にするために現在 VSP で用意されている高速モードと低消費電力モードの 2 通りのパ

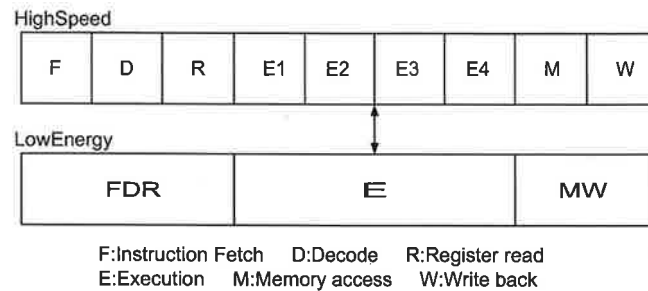


図 2.2: モード別パイプライン段数

パイプライン段数だけを用いて説明を行う。高速モードと低消費電力モードの構成は図 2.2 のようになっている。

モード別のアーキテクチャの様子を図 2.3, 図 2.4 を用いて述べる。ここでは、高速モードにおけるパイプラインの中段である四段分だけを取り出して説明を行う。図 2.3, 図 2.4 を比較すると、高速モードで存在していたパイプラインレジスタ B, D を、低消費電力モードでは使用しないため、その分レジスタへの電力供給を減らすことができ、低消費電力となる。また、高速モードで存在していた論理回路 A と B が、パイプラインレジスタを低消費電力モードでは統合されている。ここで周波数を下げてやると、この統合された論理回路が 1 クロックで動作するようになり、データ依存待ちサイクル時間の低減と分岐予測ミスペナルティの低減が実現される。

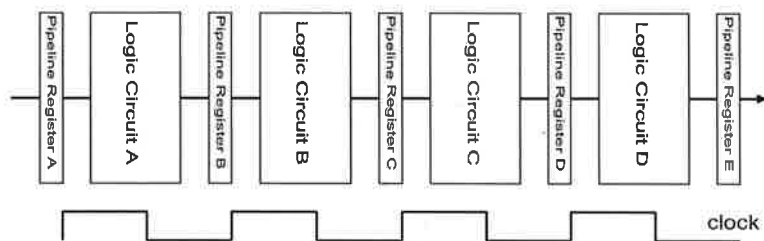


図 2.3: 高速モード

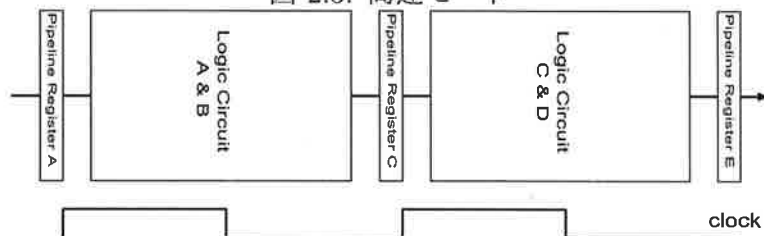


図 2.4: 低消費電力モード

2.3 パイプライン統合手法

可変パイプライン段数プロセッサでは，パイプライン段数を変更できるようにするため，本来は D-FF だけで構成されているパイプラインレジスタにマルチプレクサを追加し，データを次の回路へ流すのか，D-FF で保存するのを選択するようにしている．図 2.5 はパイプラインレジスタを使用するかどうかを選択する回路である．マルチプレクサ MUX に制御信号を与えることで選択を行うことができる．高速モードでは D-FF を使用し，パイプライン段数を増やし，周波数を上げることで高性能を実現する．低消費電力モードでは D-FF は使用せず，周波数を下げることにより低消費電力を実現できる．また，図 2.5 の回路は，図 2.3 に示したパイプラインレジスタ B, D の部分に挿入され，高速モードでは D-FF が選択され，低消費電力モードでは D-FF 使用しないよう選択される．また，他の部分のパイプラインレジスタでは必ず D-FF を選択することとなるため，図 2.5 の回路は挿入されない．

2.4 VSP について

可変パイプライン段数プロセッサの実現方式の一つとして VSP (Variable Stage Pipeline) [1] が存在する．VSP ではさらに低消費電力化を行うため

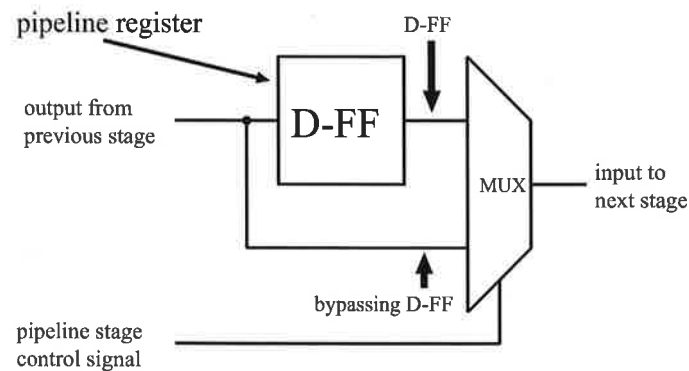


図 2.5: D-FF+MUX

に, LDS-cell(Latch DFF Selector-cell) という回路を加えている. この LDS-cell により, グリッチという無駄な電力の発生を抑えることができる. LDS-cell については第 2.4.1 節で説明を行う.

VSP プロセッサの構成図を図 2.6 に示す. また, 現在 VSP プロセッサは MIPS R3000 命令互換プロセッサに実装されており, 高速 (High Speed) モードと低消費電力 (Low Energy) モードの 2 つの状態を持つ. これらの 2 つのモードの特徴を以下に示す.

- 高速モード

- 9 段パイプラインであり, LDS-cell はパイプラインレジスタとして動作する.
- デコードステージに gshare 分岐予測ユニットを塔載する.

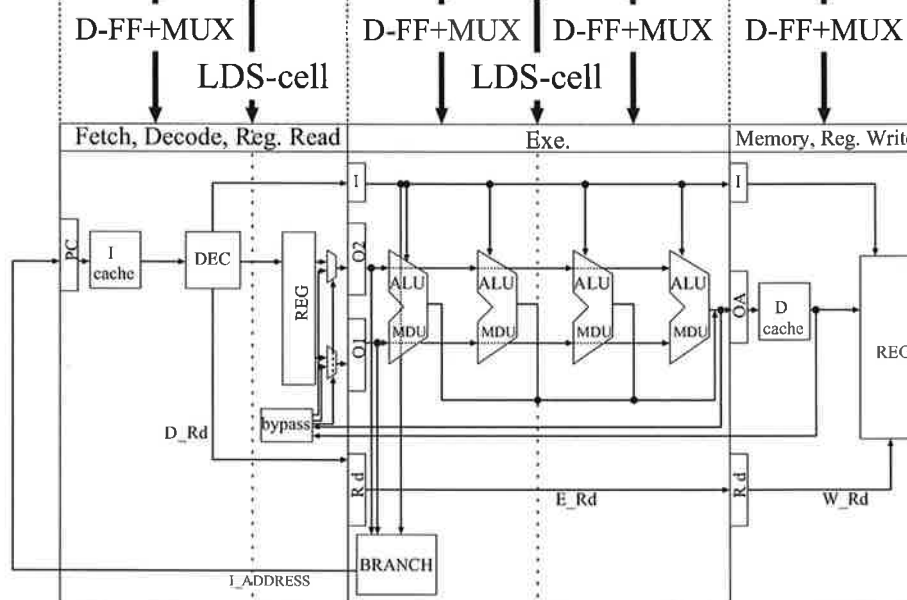
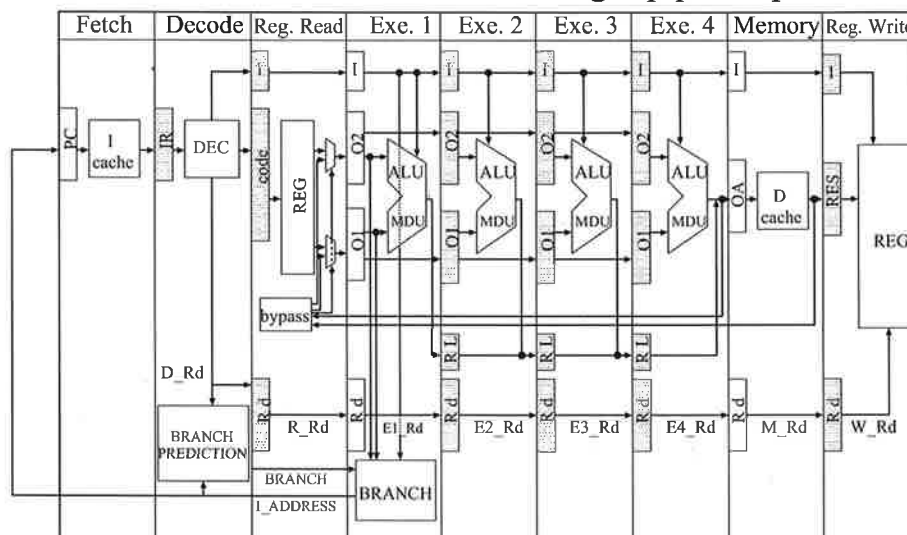
- － レジスタ間接を除く無条件分岐は分岐予測ユニットにおいて 100 % の分岐予測が可能である。
 - － インターロックと演算結果のフォワーディング機構を搭載している。
- 低消費電力モード
 - － 3 段パイプラインであり，LDS-cell はグリッチの緩和を行う D-ラッチとして動作する。
 - － 遅延分岐，遅延ロード，フォワーディングによって分岐ペナルティやデータ依存によるインターロックが発生しない。
 - － 分岐予測ユニットは使用しないので停止する。
 - － 分岐予測ユニットやバイパスされて使用しなくなったパイプラインレジスタのクロックを止めることでパイプラインレジスタで消費される電力を削減することができる。

2.4.1 LDS-cell

VSP の重要な技術として，LDS-cell (図 2.7) と呼ばれるものが存在する。VSP では LDS-cell を使用することにより，無駄な消費電力となるグ

High Speed mode

9 stages pipeline processor



Low Energy mode

3 stages pipeline processor

図 2.6: VSP プロセッサの構成図

リッチを緩和することができ、さらなる低消費電力化を図っている。ここでいうグリッチとは、電子回路にあらわれる無駄な電気信号の変動のことであり、パルス周期の突然の変化や、ゲート遅延・配線遅延のばらつきなどで生じてしまう。グリッチは一度発生すると次の回路へ伝播され、後段の電子回路ではさらにグリッチが発生してしまう。そこで、LDS-cell を挟むことにより、グリッチの緩和を行う。

VSP において、高速モード時にはパイプラインレジスタとして使用しているが、低消費電力モード時にはラッチとして機能させている部分がある。組み合わせ回路の中段付近にラッチを挟むことで、図 2.8 に示すようにグリッチを緩和することができる。これは、ラッチはクロックが下がるまでは後段の組み合わせ回路に電気信号を伝達させない性質があるためである。具体的には、図 2.7 のようなマスター・スレーブ型 D-FF において、高速モード時には 2 つのキーパを直列につなぐことで通常の D-FF の機能を実現する。一方、低消費電力モード時には、スレーブ側のキーパをバイパスさせることで D-latch として機能させることで、グリッチの伝播を緩和する。

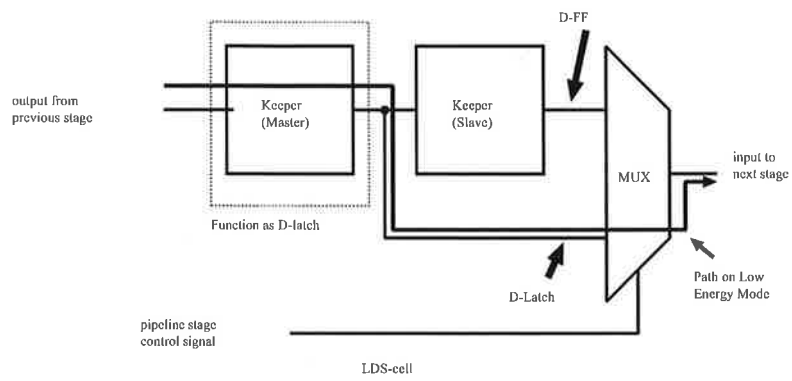


図 2.7: LDS-cell

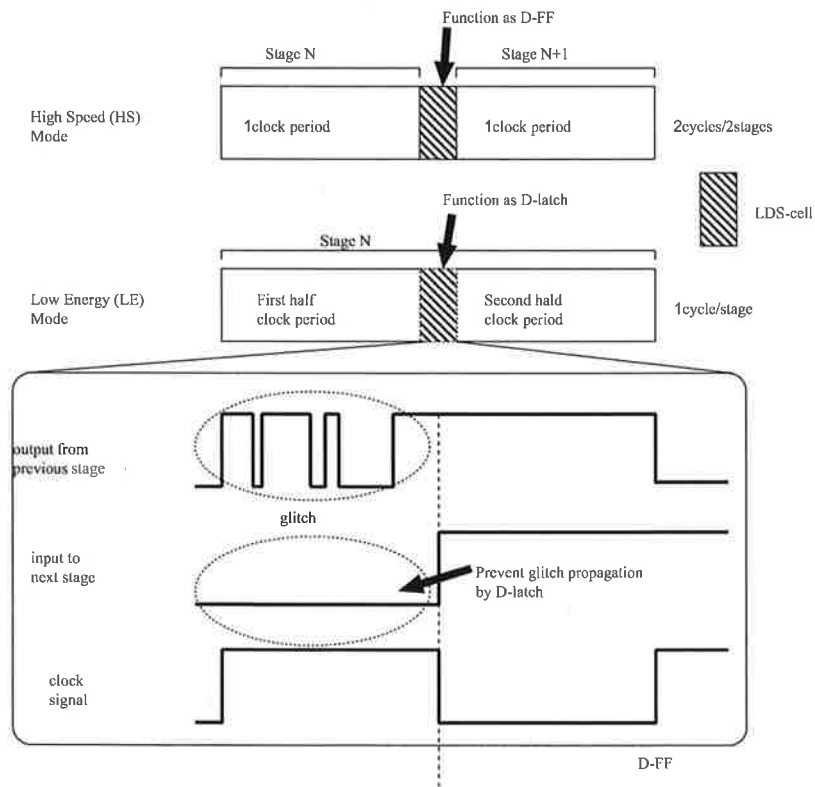


図 2.8: グリッチ緩和

3 モード切り換えコントローラ

これまで VSP ではモードを固定した上で性能評価を行ってきた。しかし、実際には特定のモードで固定して利用する可能性は低いため、動的に負荷を検出し、適宜最適なモードに切り換えるためのコントローラが必要となる。

現在、可変パイプライン段数プロセッサのための切り換えコントローラが、Yao らにより既に提案されている [4](以下、「従来コントローラ」と呼ぶ)。しかし、従来コントローラは最適なパイプライン段数を予測するために 10 万命令程度プログラムを実行する必要がある。その結果、切り換え粒度が粗くなるため細かな負荷の変動には追従できず、可変パイプライン段数プロセッサの性能を十分に発揮できない。さらに、最適なパイプライン段数を予測するために複雑な構造になっており、多数のレジスタを持つため、ハードウェア規模が大きなものとなる問題点がある。そこで、少ないハードウェア規模でより細かな負荷変動に対応するための細粒度切り換えコントローラが提案されている。以下の節では細粒度切り換えコントローラの詳細を示す。

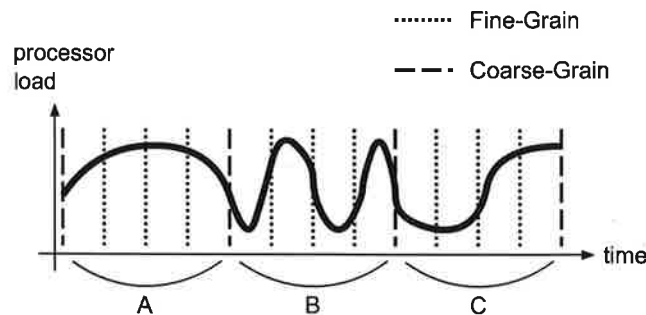


図 3.9: 切り換え周期における違い

3.1 細粒度切り換えについて

本節では細粒度に切り換えを行った場合の利点について述べる。従来コントローラでは10万命令に一度の割合で切り換えを行っている。しかし、10万命令に一度の切り換えでは細かな負荷の変動がある場合に対応できない。図3.9に変動の様子を示す。図3.9では縦軸がプロセッサの負荷、横軸が実行時間である。図3.9のAでは負荷の変動が細かくないため、切り換え周期が粗粒度の場合にも負荷に応じたモード切り換えが可能である。しかし、図3.9のB、Cの場合には切り換え周期よりも負荷の変動が細かいため負荷に応じた切り換えを行うことができない。細粒度切り換えコントローラでは粗粒度では対応できていなかった図3.9のB、Cのような細かい負荷の変動にも対応できる。

3.2 モード予測器

本節では細粒度に切り換えを行うためのモード予測器について述べる。細粒度切り換えコントローラでは負荷の変動をロード命令とストア命令の回数をカウントすることで検出している。これは現在のプロセッサではメインメモリアクセスが命令の流れを遮っておりロード・ストア命令の頻度をカウントすることでプロセッサにかかっている負荷を検出できるためである。

図 3.10 に細粒度切り換えコントローラの概要を示す。Instruction Decoder は一般的なプロセッサのデコーダである。デコーダから現在のサイクルのロード・ストア命令の命令数が 32 entry queue に保存される。32 entry queue は 32 サイクル分保存するために、32 個の 3bit D-FF で構成されるキュー構造になっており、32bit D-FF に 1 サイクル中のロード・ストア命令の命令数が保存できる。デコーダから最新のロード・ストア命令数が送られてくる度に最も古い命令数を保存しているエントリは破棄される。Add は最新 32 サイクル分のロード・ストア命令の合計命令数を計算する回路である。Mode select は Add で得られたロード・ストア命令の合計命令数とあらかじめ設定しておいた閾値とを比較し、モード切り換えを行うかを制御する部分である。また、Add の部分はハードウェア規

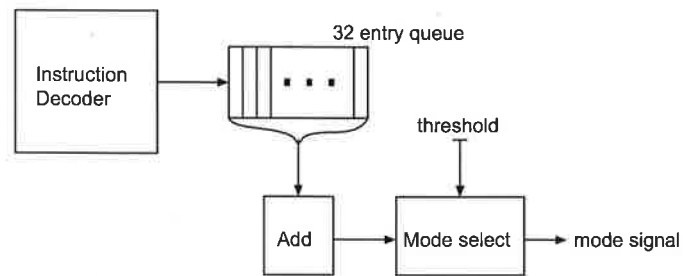


図 3.10: 切り換えコントローラの概要

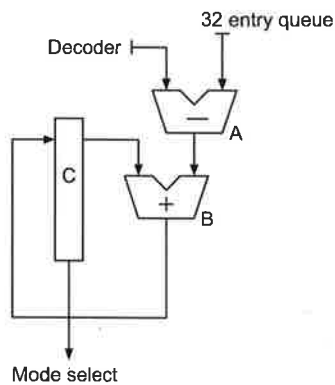


図 3.11: 削減された加算部分

模を削減するために図 3.11 のような構成になっている。C には 32 entry queue に保存されているロード・ストア命令数の総和が保存されている。A の減算器はデコーダから得られた最新のロード・ストア命令数から 32 entry queue に保存されている最も古いロード・ストア命令数の減算を行う。その後、B の加算器で C と A から得られたロード・ストア命令数の加算を行うことで、32 entry queue に保存されているロード・ストア命令数の総和が C に保存される。

3.3 切り換えオーバーヘッド

細粒度にモード切り換えを行うと、モード切り換えの際に発生するオーバーヘッドの影響を多く受けてしまう。VSPにおいてモード切り換えを行うには以下の2つのオーバーヘッドが発生してしまう。

1つ目のオーバーヘッドは周波数変更を行うために発生するもので、このオーバーヘッドは必ず発生してしまうため回避できない。しかし、周波数変更にかかるオーバーヘッドはたかだか2, 3サイクルであるため無視できる。

2つ目のオーバーヘッドはパイプラインレジスタ数が減少する場合に、パイプラインフラッシュをしなければならないため発生するものである。このオーバーヘッドは高速モードから低消費電力モードへ切り換えを行う場合に発生する。VSPには高速モードでは使用されるが低消費電力モードでは使用されないパイプラインレジスタがある。このようなパイプラインレジスタは高速モード動作時ではデータが保存されるが、高速モードから低消費電力モードへモード切り換えを行った場合、低消費電力モード時ではこのパイプラインレジスタを使用できないため、高速モード時に保存したはずのデータを呼び出すことができなくなり処理が正常に行えない。この問題を解決するために、高速モードから低消費電力モード

へ切り換えを行う場合パイプラインレジスタを一度フラッシュさせ、パイプラインレジスタ内に残っていた命令を初めからやり直すという手法が考えられる。しかし、この方法では処理が進んでいたはずの命令を初めからやり直さなければならないので性能が悪化してしまう。

3.4 切り換えオーバーヘッド削減手法

そこで、細粒度コントローラでは、高速モードから低消費電力モードへの切り換えを分岐予測ミス発生時に限定することで、このオーバーヘッドを隠蔽している。切り換えオーバーヘッドの削減方法として、新たにマイグレーションモードを設ける。図3にマイグレーションモードを含む状態の遷移を示す。マイグレーションモードは、実行ステージ以前（以降、前段）は低消費電力モードで動作し、実行ステージ以降（以降、後段）は高速モードで動作する。あるサイクルにおいて、高速モードで命令が実行されていたとする。ここで、分岐予測ミスが発生した場合、前段の命令はフラッシュされてパイプラインレジスタは空になる（図3のA）。この時、高速モードから低消費電力モードへ移行する際、前段はパイプラインステージを統合し低周波数で動作し、後段は高速モードと同様に高周波数で動作させるマイグレーションモードに移行する。マイグレー

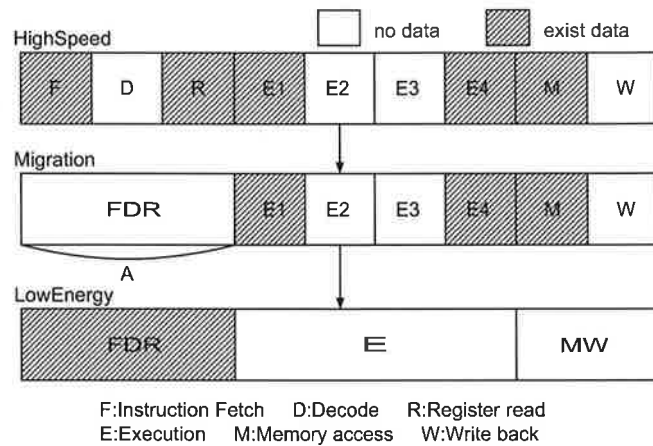


図 3.12: モード別パイプライン段数

シオンモードでは、後段のステージは高周波数で動作するため、低周波数で動作する前段でフェッチされた命令が後段に到着する頃には、後段の命令の実行は完了しており、後段のパイプラインレジスタは空になる。その後、低消費電力モードへと完全に移行することで、モード切り換えの際に発生するオーバーヘッドを隠蔽している。

また、一般的なプログラムの分岐命令の出現頻度は10～20%であり、高度な分岐予測を行ったとしても数百命令に一度は分岐予測ミスが発生する。そのため、高速モードから低消費電力モードへの切り換えを分岐予測ミスの発生時に限定しても細粒度な切り換えを実現することが可能である。

4 ハードウェア実装

4.1 本研究の目的

本研究室で提案されてる細粒度切り換えコントローラはハードウェア実装されておらず，プロセッサシミュレータでの評価しか行われていなかった．

そこで，本研究では，細粒度切り換えコントローラと切り換えオーバーヘッド削減手法を実現するために必要なハードウェアを MIPS R3000 命令互換の VSP プロセッサに Verilog HDL を用いて実装し，評価を行った．

4.2 実装

本節では細粒度切り換えコントローラと切り換えオーバーヘッド削減手法の具体的な実装方法について述べる．図 4.13 に細粒度切り換えコントローラとオーバーヘッド削減手法を導入した概要図を示す．図 4.13 において，[] がついているものは使用するビットを指定しているものであり，[0] は 0bit 目を示しており，[1] は 1bit 目を示す．それぞれの信号線の意味を表 4.1 に示す．

低消費電力モードから高速モードへの切り換えの際に，低消費電力モードでは使用されていなかったパイプラインレジスタ内に余分なデータが

表 4.1: 主な信号線の意味

信号名	意味
current mode	現在のモードの信号
low mode signal	低消費電力モードの信号
hi mode signal	高速モードの信号
mig mode signal	マイグレーションモードの信号
instruction read	命令を読み込みを行ったかどうか
threshold(HItoLOW)	高速モードから低消費電力モードへの閾値
threshold(LOWtoHI)	低消費電力モードから高速モードへの閾値
branch miss	プロセッサ側で分岐予測ミスがおきたかどうか
instruction exist	パイプラインレジスタ内に命令が存在するかどうか
mode change	次サイクルのモード信号
fore stall	実行ステージ以前をストールさせる信号
pipeline clock	モード切り換えを行う際周波数を変更するために必要となる信号

保存されている場合があるので、低消費電力モードで使用されていなかった

パイプラインレジスタをクリアするための制御線を追加した。

また、マイグレーションモードで動作する場合、実行ステージ以降の命令がすべてフラッシュする前にあらたな命令が実行ステージに到達してしまう場合がある。そのため、マイグレーションモードで動作する場合に、実行ステージ以前は低消費電力モードの周波数で動作させるだけでなく、実行ステージ以前のパイプラインをすべてストールするように変更した。

5 消費エネルギーとエネルギー遅延積

本章では回路の消費エネルギーについての評価指標の定義を行う。高性能なプロセッサを評価する場合には実行時間を見比べればよく、低消費電力プロセッサにおける評価では主に消費エネルギーを見比べればよい。しかし、今回は高性能かつ低消費エネルギーのプロセッサを作成することを目標としているため、実行時間と消費エネルギー両方について見比べる必要がある。そのため、次節で消費エネルギーについての定義を述べ、第5.2節で高性能かつ低消費エネルギーのプロセッサの評価を行う場合によく用いられる電力遅延積について述べる。

5.1 消費エネルギーの定義

CMOSで構成されたプロセッサの消費エネルギー E は消費電力 P 、実行時間 T を用いて以下の式によって求められる。

$$E = P * T = St * C * V^2 * G * cycle$$

St はゲートのスイッチング確率、 C は容量（ゲート容量、配線容量を含む）、 V は電源電圧、 G はアクティブなゲート数、 $cycle$ はアプリケーションを実行するサイクル数である。図5.14は式を用いて動作周波数を変更した時の消費エネルギーを計算したものである。図5.14からも分か

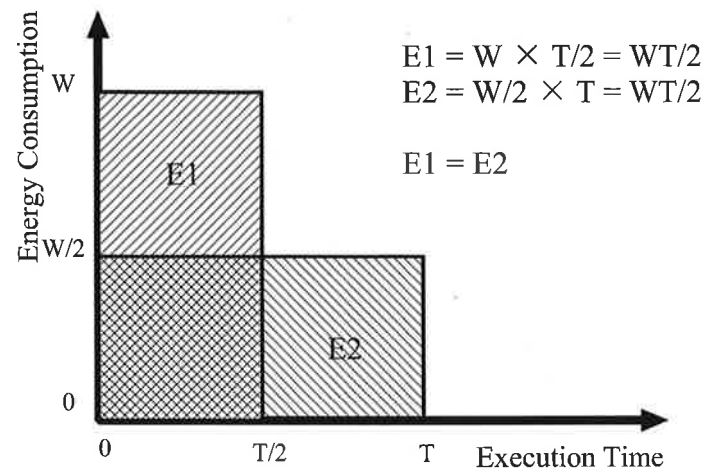


図 5.14: 実行時間と消費電力

るように消費エネルギーは動作周波数に依存しない。周波数を高くして実行時間を小さくすると消費電力が増大し、逆に周波数を低くして消費電力を小さくすると実行時間が増大してしまうからである。これが消費エネルギーと消費電力の違いである。

5.2 電力遅延積の定義

本研究では、どちらがより低消費エネルギーと高性能の両立を達成出来ているかを評価するために電力遅延積を用いる。電力遅延積は消費エネルギー E と実行時間 T を用いて以下の式によって求められる。

$$\text{電力遅延積} = E * T$$

この電力遅延積が小さいほど低消費電力と高性能の両立が達成できている。電力遅延積では周波数に依存しない消費エネルギーに周波数に依存する実行時間をかけているため、消費エネルギーの指標とは異なり、周波数に依存する値となる。また、電力遅延積は消費電力に実行時間の2乗をかけても求めることができる。そのため、消費電力と性能について評価する場合は性能の方が重要視された指標となるが、今回は消費エネルギーと性能についての評価であるので電力遅延積を用いた。

6 評価

6.1 電力遅延積の評価

本節では750～1の2進数のうち1のビットがいくつあるかを効率的に求めるプログラムを用いて閾値の組み合わせ別の電力遅延積の評価を行った。図6.15、図6.16は共に高速モードのときの値を1と正規化した。

図6.15に結果を示す。閾値は、HItoLOWが高速モードから低消費電力モードへの切り換えの際の値で、この値よりロード・ストア命令数が大きいときに切り換えを行う。LOWtoHIが低消費電力モードから高速モードへの切り換えの際の値で、この値よりロード・ストア命令数が小さいときに切り換えを行う。図6.17はモード切り換えのタイミングチャートを示

している。Mode は現在のモード (00:高速, 10:マイグレーション, 11:低消費電力) を示しており, Total はロード・ストア命令数の総和を示している。例として HItoLOW の値は 13, LOWtoHI の値は 6 とすると図 6.17 のようなモードの遷移を行う。図 6.16 に図 6.15 で得られた最も良い電力遅延積と高速モード・低消費電力モードとの比較した結果を示す。

評価結果を見ると, HItoLOW の値が小さく, また LOWtoHI の値が大きいときに電力遅延積の変動がよく見られる。これは低消費電力モードで動作割合が増えているからである。その中で HItoLOW の値が 1, LOWtoHI の値が 8 のときに電力遅延積が最も小さくなっている。この閾値の組み合わせは文献 [5] の値とは異っているが, それは本研究で用いるプロセッサとは構成が大きく異っているからである。また, 高速モード・低消費電力モード単体との比較を行った結果, 電力遅延積において改善が見られることがわかった。

6.2 ハードウェア規模の評価

本節では Rohm0.18 μm CMOS テクノロジで Synopsys Design Compiler を用いて細粒度切り換えコントローラとオーバーヘッド削減手法実現にかかるハードウェアの面積の評価を行った。本研究室で用いられている VSP

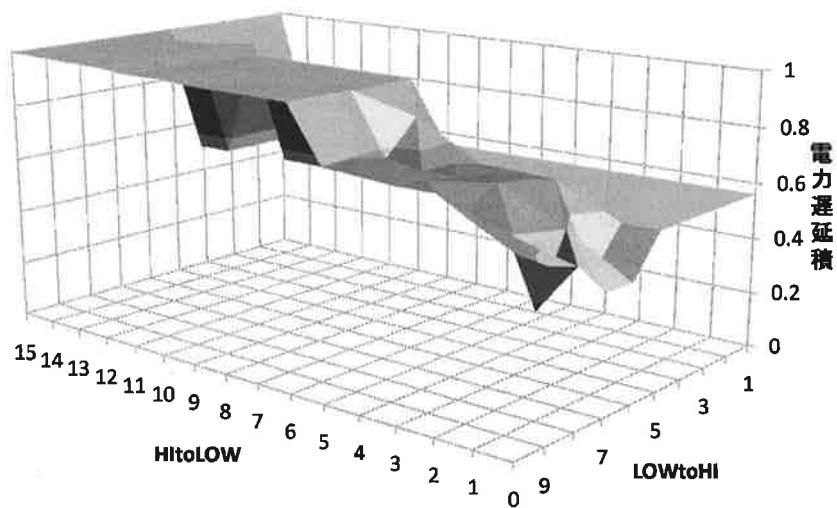


図 6.15: 閾値別電力遅延積

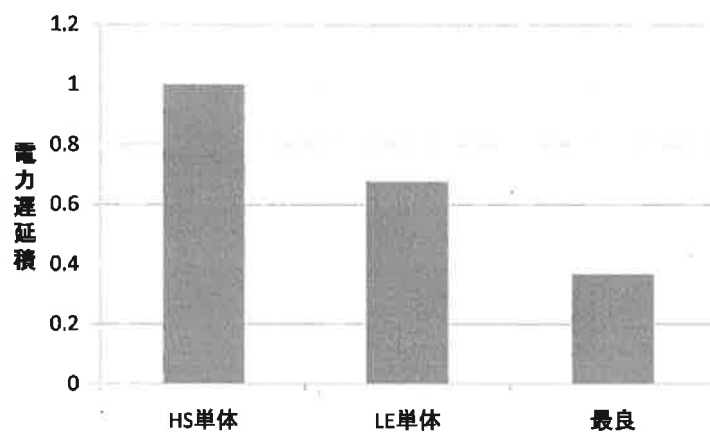


図 6.16: モード別電力遅延積

Mode	00	00	00	00	10	11	11	00	00	
Total	1	2	11	12	13	14	7	6	5	6
HItoLOW	13		13				13			
LOWtoHI	6		6				6			

図 6.17: タイミングチャート

プロセッサと、VSP プロセッサに細粒度切り換えコントローラとオーバーヘッド削減手法のためのハードウェアを追加したものと細粒度切り換えコントローラの比較を行った結果を表 6.2 に示す。また VSP プロセッサ全体に対してのコントローラ単体と追加ハードウェア全ての面積割合を表 6.3 に示す。

表 6.2: 面積結果

	コントローラ単体	VSP プロセッサ	VSP プロセッサ+コントローラ
面積	9680.8	1357550.6	1369024.3

表 6.3: 面積割合

	コントローラ単体	コントローラ+オーバーヘッド削減手法
面積	0.7131 %	0.8452 %

評価結果を見ると、細粒度切り換えコントローラはプロセッサと比較すると無視できるほどの面積であることがわかった。また、オーバーヘッド削減手法を実現するためのハードウェアも無視できるほどであることがわかった。そのため、細粒度切り換えコントローラとオーバーヘッド削減手法を実現するためのハードウェアを VSP プロセッサに組み込んでもプロセッサに与える影響は小さい。

7 まとめ

本研究では，MIPS R3000 命令互換の VSP プロセッサに細粒度切り換えコントローラを実装し，評価を行った．

細粒度切り換えコントローラとオーバーヘッド削減手法を実現するためのハードウェア量を合計してもプロセッサ全体の 1%にも満たない程小さいものであることが明らかとなった．また，電力遅延積で評価を行った結果，閾値が HItoLOW が 1，LOWtoHI が 8 のとき最も良い結果が得られ，その値は各モード単体に比べても良い結果であった．

今後の課題として，電子回路シミュレータである NanoSim を用いてより詳細な評価を行う．その後，LSI 設計を行い，実チップでの評価を行う必要がある．

謝辞

本研究を行うにあたり，多くの助言をいただきました近藤利夫教授，大野和彦講師，並びにご指導，ご助言いただきました下さいました佐々木敬泰助教に深く感謝いたします．また，様々な局面にてお世話になりました計算機アーキテクチャ研究室の皆様にも心より感謝いたします．

参考文献

- [1] 市川 裕三, 佐々木 敬泰, 弘中 哲夫, 北村 俊明, 近藤 利夫, “可変パイプラインを用いた低消費エネルギープロセッサの設計と評価“, ACS, 2006.
- [2] 市川 裕二, 佐々木 敬泰, 弘中 哲夫, 谷川 一哉, 北村 俊明, 近藤 利夫: 可変パイプラインを用いた低消費エネルギープロセッサの設計と評価, 情報処理学会論文誌 (コンピューティングシステム), Vol.47, pp.231-242, (2006 年 5 月) .
- [3] Yuji Ichikawa, Takahiro Sasaki, Tetsuo Hironaka, Kazuya Tanigawa, Toshiaki Kitamura, and Toshio Kondo: A Design of Prototype Low Energy Processor by Variable Stages Pipeline Technique, Proc. of International Technical Conference on Circuits/Systems Computers and Communications (ITC-CSCC2005), Vol.2, pp.561-562 (2005 年 7 月) .
- [4] Yao, J., et al., “ A dynamic control mechanism for pipeline stage unification by identifying program phases, ” IEICE Trans. on Information and Systems, vol. E91-D, pp. 10101022, 2009.

- [5] Sasaki, T., Nomura, K., Nakabayashi, T., Ohno, K. and Kondo, T.:
‘Fine Grain Controller for Variable Stages Pipeline Processor’ International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2010), July, 2010, pp. 748-751.
- [6] 嶋田 創, 安藤 秀樹, 島田 俊夫, “低消費電力化のためのパイプライン”, 情報処理学会研究報告, ARC, 2001.
- [7] 嶋田 創, 安藤 秀樹, 島田 俊夫, “パイプラインステージ統合: 将来のモバイルプロセッサのための消費エネルギー削減技術”, 先進的計算基盤システムシンポジウム SACSI2003, 2003.
- [8] Jun YAO, Hajime SHIMADA, Yasuhiko NAKASHIMA, Shin-ichiro MORI, Shinji TOMITA “ An EDP Study on the Optimal Pipeline Depth for Pipeline Stage Unification Adoption,”, Information Processing Society of Japan (IPSJ), 2006.
- [9] Jun YAO, Shinobu MIWA, Hajime SHIMADA, Members, and Shinji TOMITA, “A Dynamic Control Mechanism for Pipeline Stage Unification by Identifying Program Phases“, IEICE TRANS. INF. & SYST., Vol. E91-D, pp.1010-1022, APRIL 2008.