修士論文

題目

可変パイプライン段数アーキテクチャへのパワーゲーティング適用に関する研究

指導教員

近藤利夫 教授

平成23年度

三重大学大学院 工学研究科 情報工学専攻
計算機アーキテクチャ研究室

田中将輝 (410M514)

内容梗概

近年，携帯電話やノートパソコンなどのモバイル端末に使用されるプロセッサでは，性能向上に伴う消費エネルギーの増加が問題となってきている．消費エネルギーが多いと，発熱，故障のトラブルや，バッテリー持続時間の減少の原因となり得るため，高性能を維持しつつ低消費エネルギーを実現する手法が様々提案されている．

その手法の一つとしてこれまでにパイプライン段数を動的に変更する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．可変パイプライン段数プロセッサは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ことにより，高性能かつ低消費エネルギーを実現している．

一般にプロセッサで消費されるエネルギーは動的消費電力と静的消費電力に大別できる．動的消費電力はトランジスタのスイッチングによって消費されるエネルギーである．一方，静的消費電力はトランジスタの漏れ電流 (リーク電流) によって引き起こされ，トランジスタのスイッチングに関係なく，消費されるエネルギーで，リーク電力ともいう．

VSP は動的消費電力の削減には有効であるが，近年プロセス技術の微細化に伴い増加傾向にあるリーク電力には有効でない．

そこで本研究では，VSP が少段構成で動作する際に消費するリーク電力を削減するために，VSP 中のハイブリット分岐予測器内の一部の履歴テーブルにパワーゲーティングを適応的に適用する手法を提案し，大幅な性能低下の回避とリーク電力削減を行う．その結果，高性能かつ低消費電力を指し示す指標である電力遅延積において最大 21.3%の性能改善ができた．

三重大学大学院　工学研究科

Abstract

Recently, specially in mobile processors, the increase of energy consumption by enhancing performance becomes serial problem. it causes decrease of the battery continuance, fever trouble or trouble of note PC when energy consumption is increased. So achievement of both low energy and high-performance is demanded. Therefore various low energy and high-performance techniques have been proposed.

Variable Stages Pipeline (VSP) and Pipeline Stage Unification (PSU) are proposed as one of low energy and high-performance techniques. VSP changes the number of the pipeline stages dynamically and frequency. When load of the processor is high, VSP are increased the number of the pipeline stages and operated at high frequency. When load of the processor is low, VSP are reduced the number of the pipeline stages and operated at low frequency, VSP realizes high efficiency and low energy consumed in this way

Although the VSP is effective in dynamic power, it is ineffective in leakage power. This paper applies power gating technique to a part of hybrid branch predictor in the VSP to reduce leakage power and prevent performance degradation. According to evaluation result,proposal techinique can improves 21.3% energy-delay-products.

三重大学大学院　工学研究科

目 次

図 目 次

表 目 次

1　はじめに

近年，プロセッサの性能向上に伴う消費エネルギーの増加により，低消費エネルギーと高性能の両立が要求されている．消費エネルギーの増加は発熱量の増加やバッテリー持続時間の減少につながるためである．

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，高性能かつ低消費エネルギーを実現する手法が検討されている．その中の手法の一つにDVFS（Dynamic Valtage and Frequency Scaling) [1] と呼ばれる手法が提案されている．この手法は，電源電圧と周波数を動的に変更する手法であり，プロセッサの負荷が高い時には電圧と周波数を高くし，負荷が低いときには性能があまり必要でないため電圧と周波数を低くすることにより，高性能かつ低消費エネルギーを実現する手法である．DVFSでは負荷の監視にOSを利用し，待ちプロセスの数などをもとに負荷を算出している．DVFSは，現在のIntel社やAMD社などのプロセッサに導入されている．しかし，DVFSは将来的に消費エネルギー削減効率の低下が予想されている．なぜなら近年CMOSの電源電圧は低下の一途をたどっており電源電圧の下げ幅は小さくなっているためである．

そこで，電源電圧に依存しない低消費電力化手法として，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に応じてパイプライン段数と周波数を動的に切換えることで高性能と低消費電力化の両立を実現する手法である．一般にパイプラインレジスタへのクロック供給のためのエネルギーは膨大であるため，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エネルギーを実現している．またパイプライン段数が少段になることによりデータ依存や分岐による待ちサイクルが低減されるため，分岐予測器など一部の回路が不必要となる．このような回路へのクロック供給を止めることでも低消費エネルギーを実現している．また，可変パイプライン段数プロセッサは電源電圧に依存しないため，将来的に効果が期待される．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 StagePipeline) [3] [4] [5] が提案されている．

一般にプロセッサで消費されるエネルギーは動的消費電力と静的消費電力に大別できる．動的消費電力はトランジスタのスイッチングによって消費されるエネルギーである．一方，静的消費電力はトランジスタの漏れ電流 (リーク電流) によって引き起こされ，トランジスタのスイッチ

ングに関係なく，トランジスタ数と回路に電圧のかかっている時間に比例して消費されるエネルギーで，リーク電力ともいう．

近年，回路の微細化にともなって，トランジスタのゲート絶縁膜の薄膜化，閾値電圧の低下が進んでおり，リーク電力が増加し深刻化している．最近のプロセッサで用いられている 45nm プロセスや、それ以降のプロセスではリーク電力が無視できないほど増大し，消費電力全体におけるリーク電力の占める割合が半分程度にまでなる．

前述の通り，VSP では少段構成で動作する際に分岐予測器等の一部の回路の動作を停止させることで動的電力を削減している．しかし，従来の VSP ではリーク電力への対策がなされていないため，動作停止中もリーク電力を消費し続けている．この分岐予測器とは条件分岐命令が分岐するか否かを予測するための回路であり，性能に大きな影響を与えるため，そのハードウエア規模も大きく，リーク電力を大きく消費している．

そこで本研究では，VSP が少段構成で動作する際に消費するリーク電力を削減するために，VSP の分岐予測器に対して，リーク電力削減に有効な技術であるパワーゲーティングを局所的に適用する手法を提案する．パワーゲーティングとは使用していない回路への電源供給を一時的に遮断することでその回路のリーク電力の削減を行う技術である．しかし電源供給遮断により回路内で保持されているデータは破壊されてしまう．そのため分岐予測器内の履歴テーブルに単純にパワーゲーティングを適用すると大幅な性能低下を招く恐れがある．そこで本研究ではハイブリット分岐予測器内の一部の履歴テーブルにパワーゲーティングを適応的に適用し，大幅な性能低下の回避と，リーク電力削減を目指す．その結果，高性能かつ低消費電力を指し示す指標である電力遅延積において最大 21.3%の性能改善ができた．

2　可変パイプライン段数アーキテクチャ

2.1　パイプラインプロセッサに関する概括

命令を段階的に分けて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)，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．パイプラインプロセッサの動作を図 2.1 に示す．図 2.1 では Fetch を F，Decode を D，Register Read を R，Execution を E，Write Back を WB 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の処理は異なった回路で処理されるため，1 クロックで段階ごとに並列に動作させることができる．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．一般的な商用プロセッサを高性能なプロセッサにするため，パイプライン段数をできるだけ増やし，高周波数で動作させたいが、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．そのため，現在では約 10 段程度のパイプラインプロセッサが採用されている．

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切換えることにより，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) [7] [8] [9]，DPS (Dynamic Pipeline Scaling) [10] [11] などの方式が提案されている．また，これらの可変パイプライン段数プロセッサはパイプライン段数に対して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はないプロセッサと同等である．

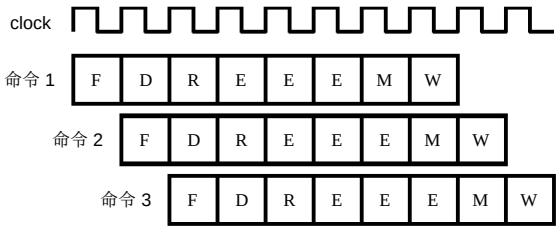

図 2.1: パイプラインプロセッサの動作

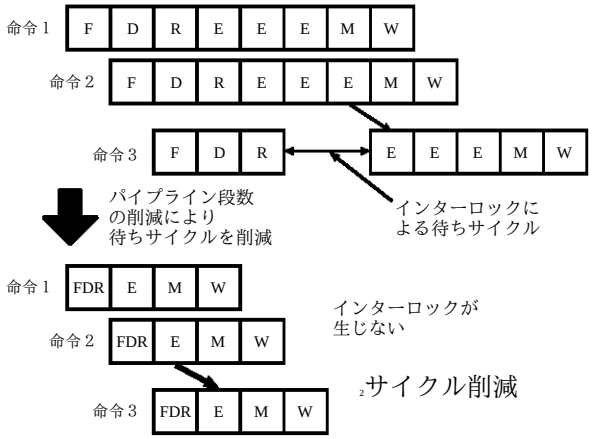

図 2.2: データ依存待ちサイクルの低減

- 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エネルギー低減に繋がる
- 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（図 2.2，図 2.3）．図 2.2，図 2.3 ともに水平方向が時間軸である．

図 2.2 において，多段パイプライン段数時の状態で命令を実行中に命令 2 と命令 3 に依存関係がある場合には，命令 3 の実行を命令 2 の処理結果が得られるまで待つ必要がある．しかし，少段パイプライン段数時の状態では命令に依存関係がある場合において，実行に必要となるクロック数が 1 クロックであるため待ち時間は発生しない．図 2.3 においても，少

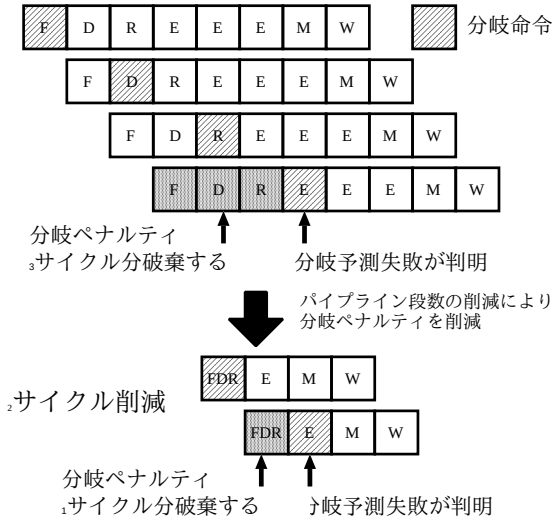

図 2.3: 分岐予測ミスペナルティ低減

段パイプライン段数時ではパイプライン段数が少ないために分岐予測ペナルティ削減ができる，分岐予測に関しては 4 章で詳しく説明する．

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 種類以上へ切換可能であるが，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は図 2.4 の構成となっている．高速モードはプロセッサの性能を最大限に活かすために，パイプライン段数が多く，高周波数のモードであり，本論文ではパイプライン段数は 9 段とする．低消費電力モードは消費エネルギーを抑えるために，パイプライン段数が少なく，低周波数のモードであり，本論文ではパイプライン段数は 3 段とする．

モード別のアーキテクチャの様子を図 2.5，図 2.6 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 2.5，図 2.6 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

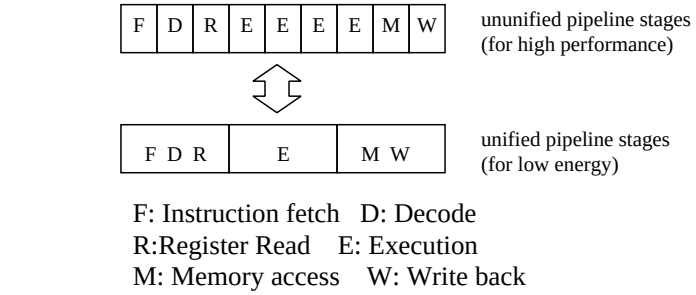

図 2.4: モード別パイプライン段数

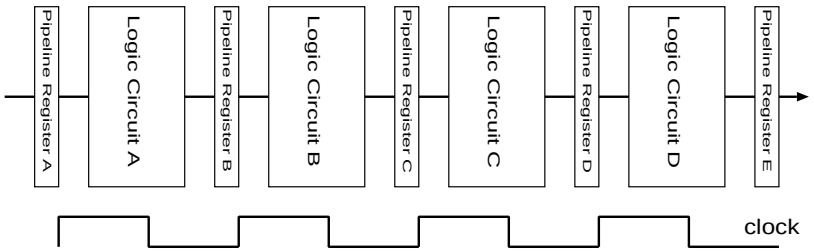

図 2.5: 高速モード

低減が実現される．

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 で保存するのかを選択するようにしている．図 2.7 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 2.7 の回路は，図 2.5 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

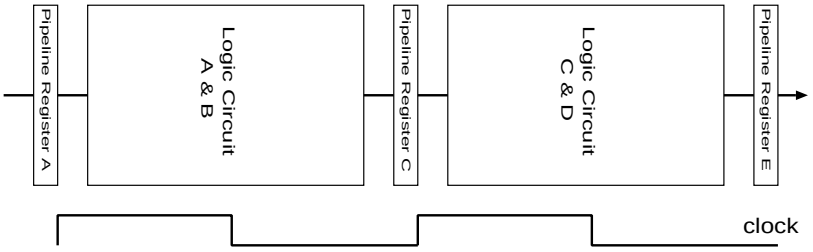

図 2.6: 低消費電力モード

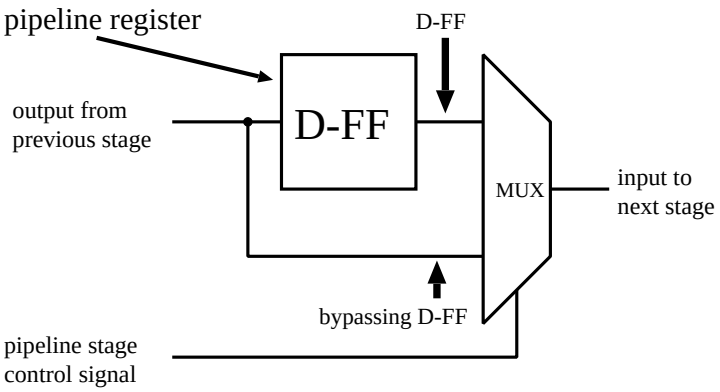

図 2.7: D-FF+MUX

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 2.7 の回路は挿入されない．

2.4 VSP について

VSP は，PSU と同様にパイプライン段数を切り換える手法であるが VSP ではパイプラインレジスタの一部に LDS-cell (Latch D-FF Selector-cell) という特殊なセルを用いグリッチと呼ばれる無駄な信号の変化の発生を抑制することでさらに消費電力を削減している．

また，VSP では多段のパイプライン構成時を HS (High Speed) モード，少段のパイプライン構成時を LE (Low Energy) モードと呼んでおり，その特徴を以下に示す．

HS モード：

- LDS-cell は通常のパイプラインレジスタとして動作する．
- gshare 分岐予測器を搭載しており，レジスタ間接をのぞく無条件分岐は分岐予測器において 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．

LE モード：

- 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制御依存やデータ依存によるインターロックが発生しない．
- 分岐予測器は使用しないので停止させている．
- 分岐予測器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エネルギーを削減することが出来る．

2.4.1 LDS-cell

VSP の重要な技術として，LDS-cell (図 2.9) 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，高速モード時で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で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 2.10 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 2.9 のようなマスター・スレーブ型 D-FF

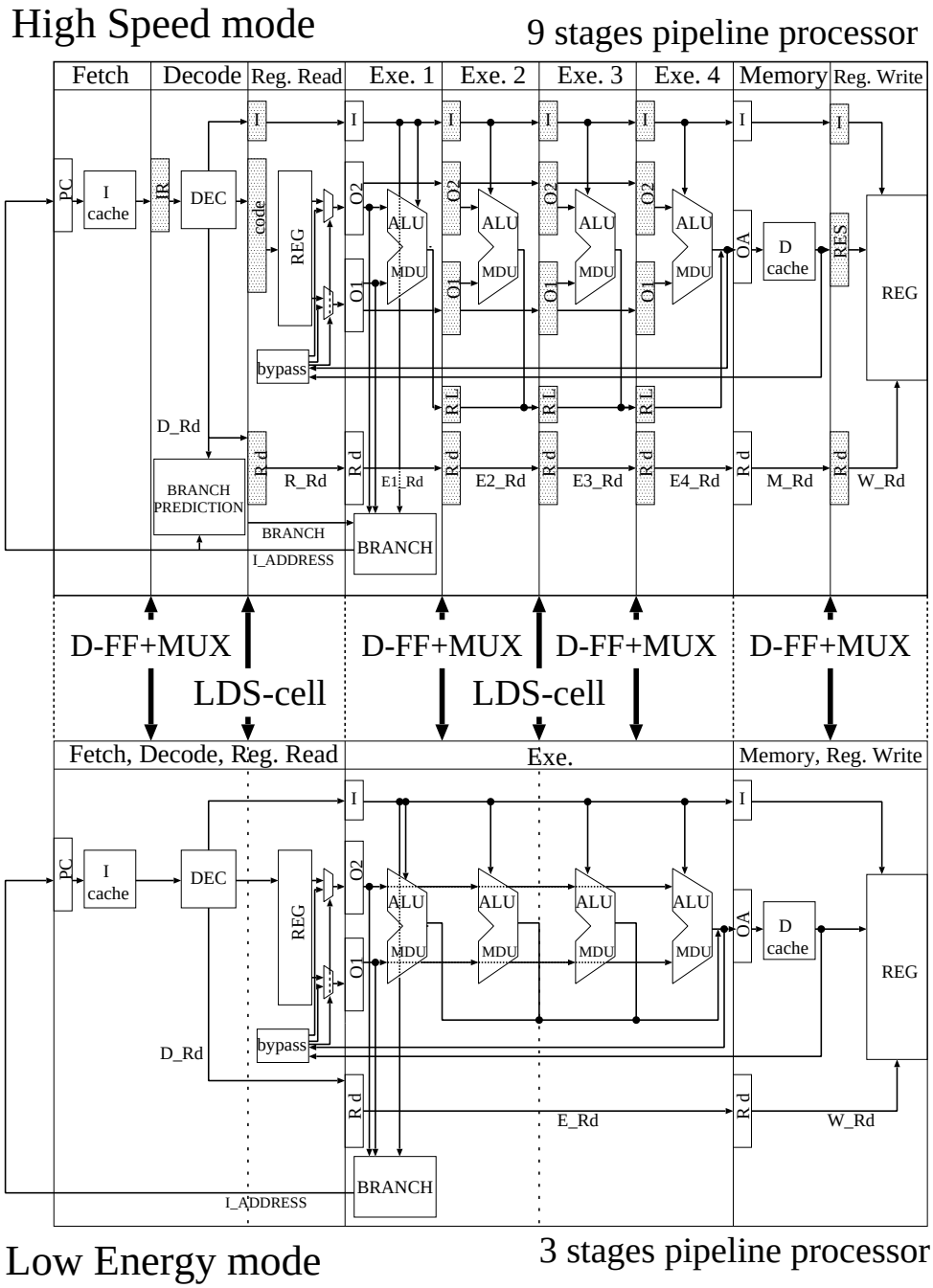

図 2.8: VSP プロセッサの構成図

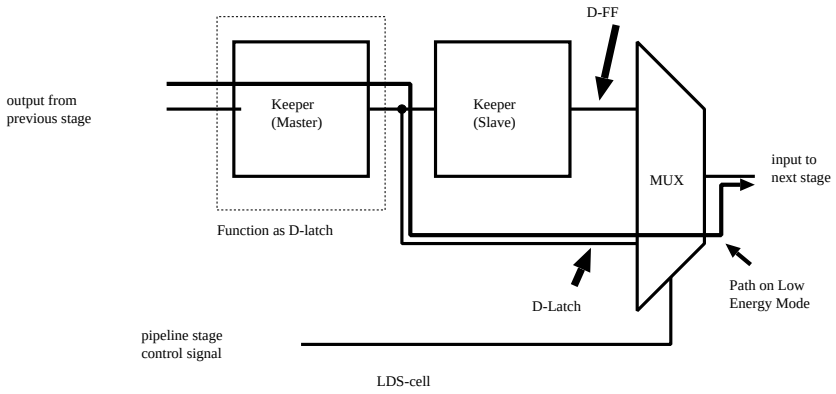

図 2.9: LDS-Cell

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．

2.4.2 VSP におけるリーク電力削減の必要性

すでに述べたように，VSP では LE モード時に，使用しないユニットへのクロックを停止させて消費電力を削減している．しかし，この電力削減効果は動的な消費電力に対してのみ有効であり，動作停止中もリーク電力は消費し続ける．そこで本研究では，LE モード時に使用しない箇所のリーク電力を削減することで更なる低消費エネルギー化を試みる．

VSP において LE モード時に使用せず，リーク電力を削除可能と考えられる箇所として以下のような部分が挙げられる．

- パイプラインステージ統合時に使用しなくなるパイプラインレジスタや LDS-cell の一部
- パイプラインステージ数が少段になる事により分岐による待ちサイクルが低減され不必要となる分岐予測器

また，表 2.1 は MIPS R3000 互換 VSP プロセッサにおけるリーク電力削減可能箇所とその割合を示している．本研究ではハードウエア量が多く，大きなリーク電力削減効果が見込める分岐予測器に注目し，低電力化を試みる．

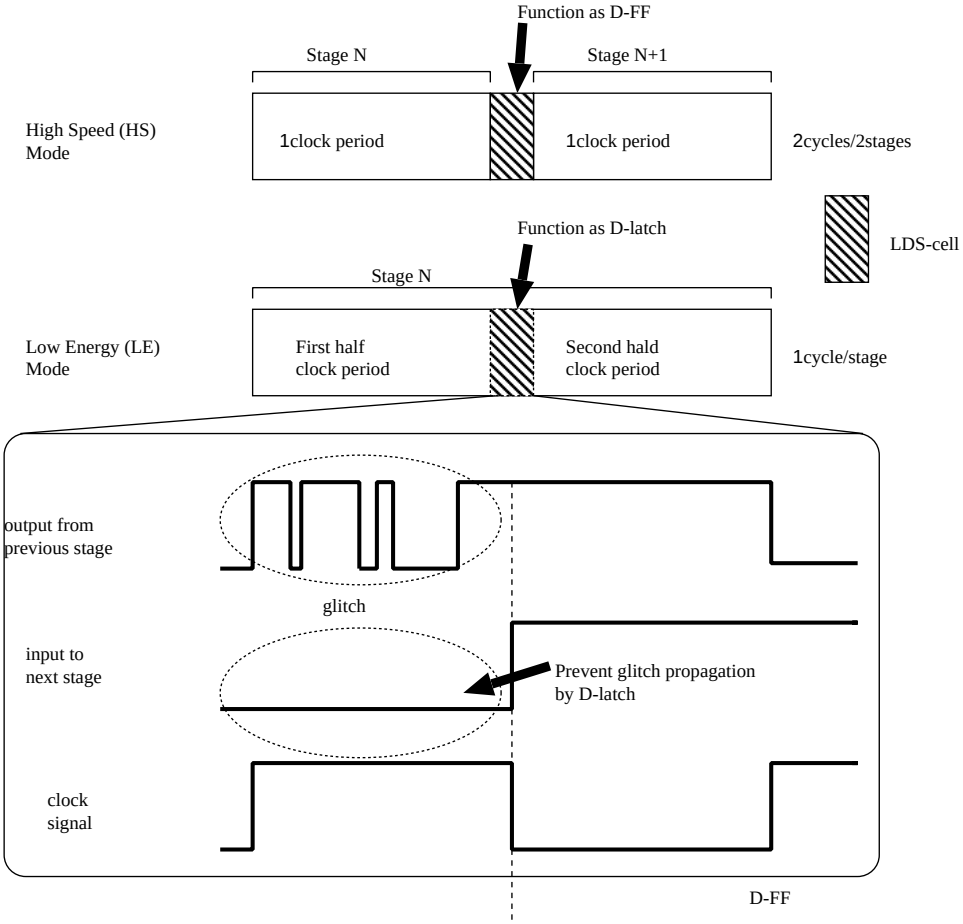
Function as D-FF
Stage N
Stage N+1
High Speed (HS)
Mode
1clock period
1clock period
2cycles/2stages
LDS-cell
Function as D-latch
Stage N
Low Energy (LE)
Mode
First half
clock period
Second hald
clock period
1cycle/stage
output from
previous stage
glitch
input to
next stage
Prevent glitch propagation
by D-latch
clock
signal
D-FF

図 2.10: グリッチ緩和

表 2.1: リーク電力削減可能箇所のハードウェア規模

削減可能箇所	プロセッサ全体に占める割合
パイプラインレジスタ	6%
分岐予測器	33%

2.4.3 VSP のモード切換

本節では VSP のモード切換制御を行うコントローラについて述べる．これまでに，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 VSP 向け細粒度切換コントローラを提案されている [6]．現在の VSP ではこの細粒度切換コントローラを用いてモード切換制御を行っている．

細粒度切換コントローラではプロセッサ内の状態を直近数十サイクル観測することで負荷変動を検出している．またその切換頻度は数百サイクルに 1 度の短い周期で行っている．そのため，前述のような LE モード時のリーク電力削減を細粒度で行う必要がある．

3　低消費電力化技術

3.1　CMOS トランジスタの消費電力

本節でまず，CMOS トランジスタの消費電力について述べる．CMOS トランジスタで構成されたプロセッサの消費電力 P は次の式で表す事が出来る．

$$P = St * C * V^2 * G + IleakV \tag{1}$$

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，$Ileak$ はリーク電流であり，第 1 項が動的消費電力，第 2 項がリーク電力を表している．

また，リーク電流はゲートリークとサブスレッショルドリークに大別でき，それぞれの様子を図 3.11 に示す．ゲートリークとはゲート酸化膜の薄膜化により本来流れないはずのゲート・基盤間に流れるリーク電流である．このゲートリークはゲート酸化膜に高誘電率の素材を用いる等のデバイス技術によりその削減が行われている．一方サブスレッショルドリークは閾値電圧以下でもソース・ドレイン間に流れてしまうリーク電流であり，次の式で表される．

$$\text{サブスレッショルドリーク} \quad = exp(-VthQ/kT) \tag{2}$$

Vth は閾値電圧，Q は電子の電荷，k はボルツマン定数，T は絶対温度を表している．現在の LSI で消費されるリーク電力では，このサブスレッショルドリークが支配的である．

近年の微細プロセスでは閾値電圧 Vth が低下傾向にある．これによりリーク電力が指数関数的に増加しており問題となっている．そのためこのリーク電力を削減する手法が様々研究されている．

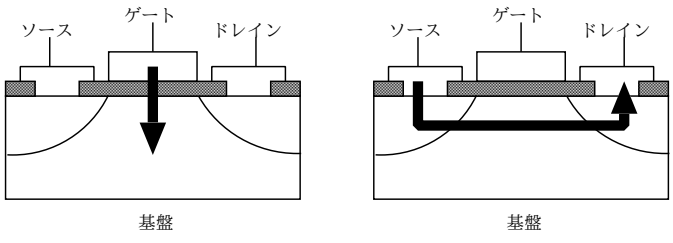

図 3.11: ゲートリーク・サブスレッショルドリーク

3.2 関連研究

3.2.1 低電力化のための基本アプローチ

動的消費電力は電源電圧の 2 乗に比例するため，回路レベルの低電力化手法では電源電圧を下げる方法が広く用いられている．ところが電源電圧を下げるとトランジスタの性能が低下してしまい，遅延時間が増大するというトレードオフが存在する．

一方，リーク電力に対しては，電源電圧を下げる方法に加えトランジスタの閾値電圧を上げる方法が効果的であることが知られている．閾値電圧は，前節の式で示したようにサブスレッショルドリーク電流の発生に大きく影響している．閾値を高めればサブスレッショルドリーク電流の発生を抑えることができるが，閾値電圧を上げると性能が低下するというトレードオフが存在する．以上のことを踏まえ次節以降でリーク電力削減の為に広く使われている手法を紹介する．

3.2.2 DVFS

現在の代表的な低消費電力化手法の 1 つである DVFS (Dynamic Valtage and Frequency Scaling) [1] は，動的に電源電圧と動作周波数を変化させることで消費電力を低減する．DVFS は動的消費電力削減を目的とした手法であったが，電源電圧が動的に下がる事によりリーク電力削減に対しても有効な手法である．

しかし，DVFS では数百命令単位で消費電力を削減することはできない．なぜなら，DVFS では電源電圧を変更する際，回路の充放電による電力オーバーヘッドが発生し，このオーバーヘッドが大きいためである．そのため，DVFS では切換の粒度は 1 万〜10 万サイクルのオーダーに設定する必要があり [12]，細粒度にモード切換を行う VSP とは相性が悪いと考えられる．

3.2.3 Multi-VDD

Multi-VDD 技術は，回路中で高性能が必要な部分にだけ従来の電源電圧を使い，さほど速い動作速度が必要とされない部分には低い電源電圧を使うという手法である．論理合成を使って生成された回路では，クリティカルパス上のゲートは全体の 10〜30%程度といわれており，かなり

のゲートを低電圧にすることができる．しかし，電源電圧の異なるブロックに信号を送出する場合には，レベルシフタと呼ばれる電圧振幅を変換する回路を挿入する必要がある．また，VSP プロセッサでは分岐予測器がクリティカルパスとなっているため，本研究でリーク電力の削減対象としている分岐予測器への適用には向いていないといえる．

3.2.4 Multi-Vth

Multi-Vth は，Multi-VDD 技術と同じように高性能が必要な部分にだけ低閾値電圧を使い，さほど速い動作速度が必要とされない部分には高閾値電圧を使うという手法である．標準的な設計手法としては，高 Vth のセルと低 Vth のセルの両方をセルライブラリの中に用意しておき，論理合成の実行段階で，クリティカルパス上のゲートには低閾値電圧のセル，クリティカルパス以外の部分には高閾値電圧のセルを割り当てる．

しかし，Multi-VDD 技術と同様に VSP プロセッサでは分岐予測器がクリティカルパスとなっているため，本研究でリーク電力の削減対象としている分岐予測器へ適用には向いていないといえる．

3.3 パワーゲーティング

パワーゲーティング (以下 PG と呼ぶ) は、回路内の使用されていないブロックの電力を一時的に遮断することでそのブロックのリーク電力の削減を行う手法である．電源供給の制御は，電源 (VDD) またはグランド (Ground) と，PG を適用する回路ブロックとの間にトランジスタを挿入する事で行う．このとき用いられるトランジスタはパワースイッチと呼ばれ，通常の CMOS に比べて閾値電圧が高いものを用いている．それによりリーク電流の発生も抑制している．図 3.12 にパワースイッチとして NMOS を用いて行う footer 型の PG の様子を示す．パワースイッチが ON の場合には，回路ブロックのグランド側端がグランドと同電位になり回路ブロックは通常の動作を行うが，リーク電力は消費され続けてしまう．一方パワースイッチが OFF の場合，仮想グランド線はグランドから切り離され，仮想グランドに電荷が溜まっていく．その結果回路ブロック内の電位差が小さくなり，リーク電力が削減される．しかし，回路ブロック内の電位差がなくなる事で DFF やラッチで保持していたデータが破壊されるという弊害が存在する．

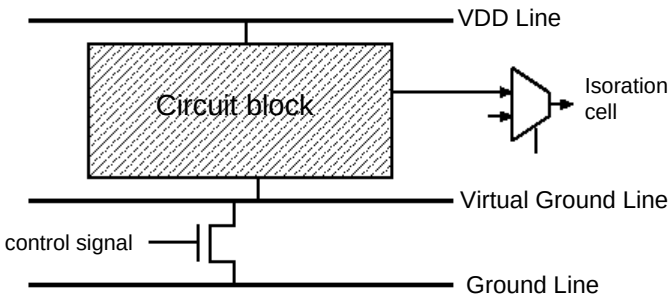

図 3.12: パワーゲーティングの概要

PG は電源供給を完全に遮断することから，他の手法よりもリーク電力削減に関して効果が高い．また，動作していない回路セルの電源を数 μ 秒という短時間で ON/OFF し、電源供給を遮断するというようなことも行われており [14]，数 100 サイクル単位で細粒度にモード切換を行う VSP とも相性が良いと考えられる．そこで本研究ではリーク電力削減の手法として PG に注目する．

4　分岐予測器

4.1　分岐予測器概要

分岐予測器とは，条件分岐命令が分岐成立か不成立かを予測することでパイプラインプロセッサの性能を向上させるためのプロセッサ内のユニットである．パイプラインプロセッサではプログラム中に条件分岐命令が現れた場合，条件式が演算実行ステージで処理されるまで次にフェッチするべき命令がわからないため，その間パイプラインをストールする必要があるが，分岐予測を行うことで待ちサイクル無しに実行をし続けることができる．もし予測が外れていた場合は分岐命令以降の命令を全て破棄し正しい命令で実行し直す必要がある．このオーバーヘッドは分岐予測ミスペナルティと呼ばれる．パイプライン段数が増加や，スーパースカラプロセッサのように並列度が増すにつれ分岐予測ミスペナルティは大きくなるため，その予測精度は性能に大きく影響する．

4.2　動的分岐予測

分岐予測の方法として様々な手法が提案されているが，最近のプロセッサでは、プログラム実行中の振る舞いから、動的に分岐方向を予測する動的分岐予測をハードウエア実装するのが主流である．その予測方法はプログラム実行中に各条件分岐命令が分岐成立だったか不成立だったかの履歴を観測し，分岐成立が続いているようであれば次も分岐成立するだろうという考えによるものである．分岐の履歴は，一般には 2 ビット程度の飽和カウンタを 1 エントリとする履歴テーブルにより記憶され，そのテーブルはプログラムカウンタの下位 n bit をインデックスとして索引される．飽和カウンタは分岐命令が実際に分岐成立 (taken) したか不成立 (not taken) だったかにより値がインクリメントもしくはディクリメントされる．2 ビットの飽和カウンタの様子を図 4.13 に示す．2 ビットの飽和カウンタの場合，値が 3 になると+1 しても 3 に留まり、値が 0 になると-1 しても 0 のままという動作をする．そしてその値が 2 以上ならば分岐成立と予測する．図 4.14 に 2bit 飽和カウンタを用いた分岐予測器の様子を示しており，このような構成の物は 2bit カウンタ予測，または bimodal 分岐予測 [15] と呼ばれる．

分岐予測器のその多くは大きな履歴テーブルによって構成されており，多くのプロセッサにおいて分岐予測器は大きなハードウエア規模を占める．

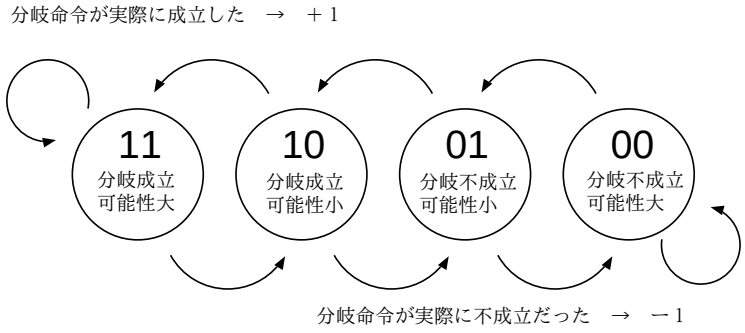

図 4.13: 2bit 飽和カウンタ

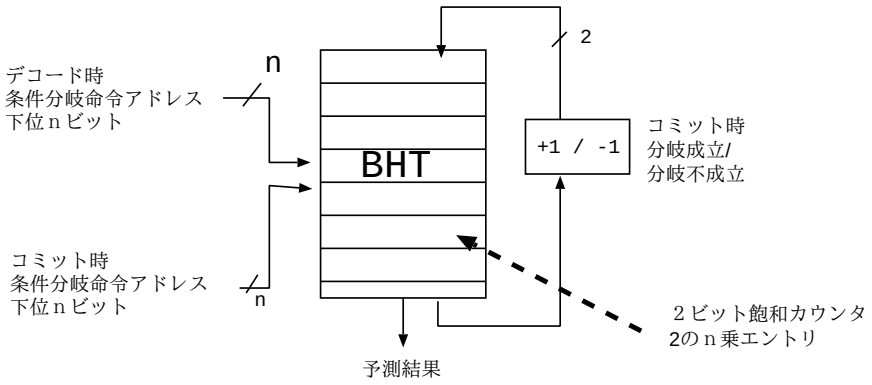

図 4.14: bimodal 分岐予測器のブロック図

現在実装されている VSP においても分岐予測器の占めるハードウエア量は大きく，MIPS R300 互換 VSP プロセッサではプロセッサ全体の 33%を占めている．また，VSP では LE モード時に分岐予測器を使用しておらずクロック供給を止め動作を停止させているが，その動作に関わらずリーク電力は消費され続けてしまっている．そのため本研究ではこの分岐予測器へ PG を適用することで低消費電力化を試みる．

4.3 従来 VSP での分岐予測器とその問題点

従来の VSP では，gshare 型 [16] の分岐予測器を用いていた．gshare 型は図 4.15 に示すように，直近の分岐命令 m 個が実際に分岐したか否かを記憶している GR(Global Register) と呼ばれる m ビットのシフトレジスタ

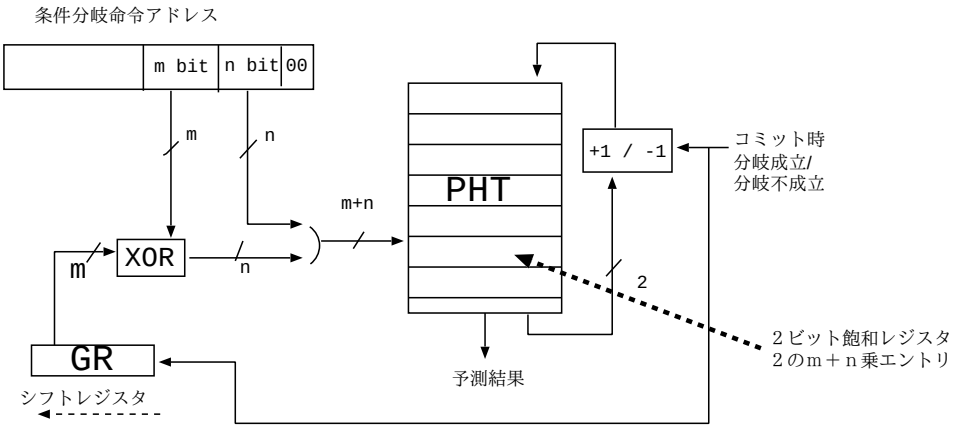

図 4.15: gshare 分岐予測器のブロック図

と，bimodal 型と同じように各分岐命令が分岐する可能性が高いかどうかを記憶している 2bit 飽和カウンタテーブル PHT(Pattern History Table) によって構成される．そしてこの PHT は，プログラムカウンタの下位 n bit と，GR に記憶されている履歴 m bit を排他的論理和をとったものをインデックスとして索引される．

このような分岐予測器に対して単純に PG を適用してしまうと LE モード時に電源供給が遮断され，PHT や GR 等の内部で保持していた各分岐命令の予測履歴が破壊されてしまう．それにより HS モード復帰後に分岐予測ミスが多発し大きな性能低下が予想される．

また，従来用いていた gshare 型の分岐予測器は，最近のプロセッサで多く採用されているハイブリッド分岐予測器と比較するとその予測精度は低いことが明らかとされている [16]，そこで本研究ではより予測精度が高いハイブリッド分岐予測器をターゲットとして PG の適用を検討した．

4.4 ハイブリッド分岐予測器

ハイブリッド型の分岐予測器の構成を図 4.16 に示す．ハイブリッド分岐予測器は 2 つの異なる型の分岐予測器と，そのどちらの予測結果を採用するかを選択する履歴テーブルによって構成される．

ハイブリッド分岐予測器は並列に 3 つの予測値を使う．すなわち，bimodal 分岐予測や gshare 分岐予測等の異なる種類の分岐予測器と、分岐命令毎にその二つのどちらを選択するための選択テーブルである．3つめ

の選択テーブルは 2bit 飽和カウンターテーブルを用いたテーブルで構成され，2 つの分岐予測器の予測結果が食い違ったときに，予測が正しかった方の分岐予測器を指す可能性が高くなるように値が更新される．

このハイブリッド分岐予測器は 3 つのテーブルサイズの合計と同じサイズのテーブルを持つ bimodal 型や gshare 型と比較してもより高い予測精度を得られることが既に実証されている [16]．そのため，多くの商用プロセッサではハイブリッド分岐予測器あるいはハイブリッド分岐予測器をされに改良したものを用いている．

VSP でももちいており，本研究で用いるハイブリッド分岐予測器は 2 つの分岐予測器に bimodal 型，gshare 型を用い，bimodal 型，gshare 型，選択テーブルの 3 つのテーブルのエントリー数の比が 1:2:1 となる構成とした．この構成は文献 [16] で示されている最善の構成である．

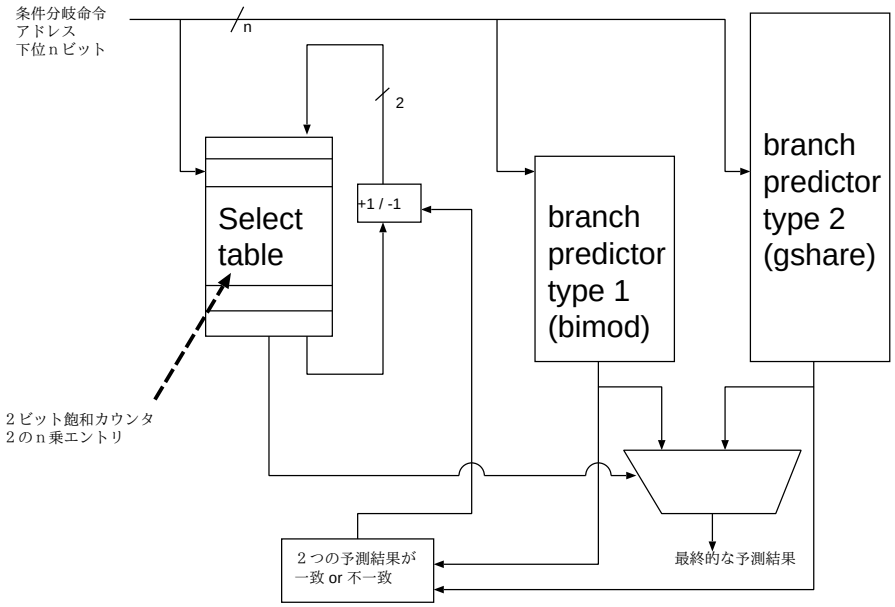

図 4.16: Hybrid 分岐予測器のブロック図

5　提案手法

本研究では、VSP におけるハイブリッド型の分岐予測器に対し，その一部に PG を適用する事で大幅な実行時間増加を回避し、リーク電力を削減する手法を提案する．

本手法ではハイブリット分岐予測器内にある 2 つの分岐予測器の内の片方と，予測結果を選択するテーブルに対し PG を適用する．図 5.17 の斜線部分が PG 適用部分に相当する．また，本手法で 2 つの分岐予測器の内の PG を適用する分岐予測器 (図中の type2) は今回の構成でハードウエア量を大きくした gshare 型の方とした．

これにより LE モード時に適用箇所分のリーク電力削減ができる．しかし，LE モード時には PG を適用した分岐予測器 type2 と予測結果を選択するテーブル (図中の Selector table) の履歴は破壊されてしまう．そして，HS 復帰直後に履歴が全くない状態の type2 の予測結果を用いてしまうと性能低下を招く恐れがある．そこで，本手法では HS モード復帰時に Selector table の各エントリーを PG 適用していない方の分岐予測器 (図中の type1) を選択するような値に初期化する．その結果，破壊された履歴の再学習が進むまでの間は type1 の履歴を用いて予測が行われるため大幅な性能低下を回避することが出来る．また再学習が進み type2 のヒット率が向上すると Selector table は type2 も選択するようになり自動的に本来のハイブリット分岐予測器の動作を行うようになる．

また，PG を適用した回路をスリープ状態からアクティブ状態にウェイクアップさせる際には PS の駆動と仮想的なグランド線に溜まった電荷を充放電することによるオーバーヘッドが発生する．PG を適用する回路の規模が大きい場合，仮想的なグランド線から電荷を放電し回路が正常に動作するまでに遅延が発生する．この遅延が図 5.18 のように 1 クロックサイクルを超える場合はプロセッサをストールさせる必要があり，このクロックサイクルが時間的なオーバーヘッドとなる．しかし，本手法ではウェイクアップ直後には本質的に PG 非適用の分岐予測器 type1 しか使用していない．そこで，ウェイクアップから遅延分だけ遅らせて図中の MUX2 を切換ることでプロセッサをストールさせる必要がなくなり，オーバーヘッドを隠蔽できると考えられる．

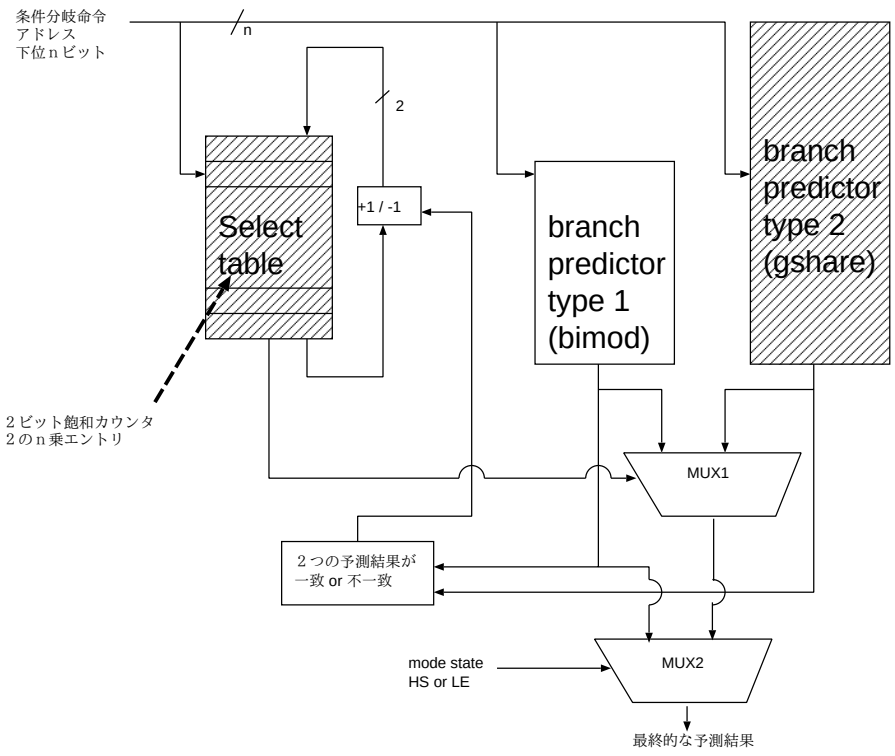

図 5.17: 提案手法のブロック図

6 性能評価

6.1 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 6.1.2 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．

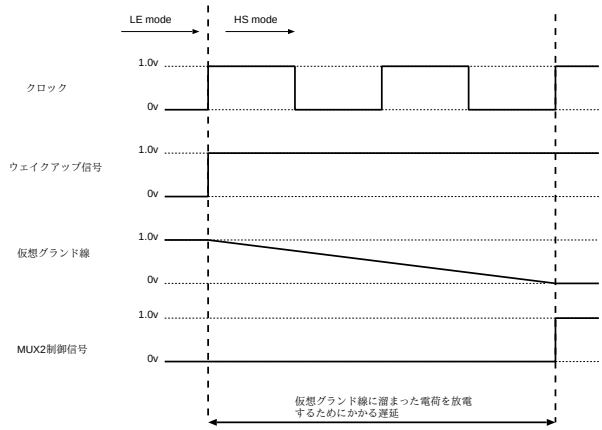

図 5.18: 時間的オーバーヘッド

6.1.1 消費エネルギーの定義

すでに章の式 1 で示した CMOS のプロセッサの消費電力 P は単位時間あたりに電流がする仕事量であり，消費電力に実行時間 T をかけたものが消費エネルギーとなる．消費エネルギー E は以下の式によって求められる．

$$E = P * T = (StCV^2G + IleakV) * T \tag{3}$$

動的消費電力は周波数を高くして実行時間を小さくするとが増大し，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ため，図 6.19 のように動的な消費エネルギーは動作周波数に依存しない．これまでの旧世代のプロセスではリーク電力が動的電力に比べ無視できる程小さかったが，微細化しリーク電力が顕著にあらわれるようになった最近のプロセスではリーク電力の分だけ動作周波数に依存するようになった．

6.1.2 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．

$$\text{電力遅延積} = E * T \tag{4}$$

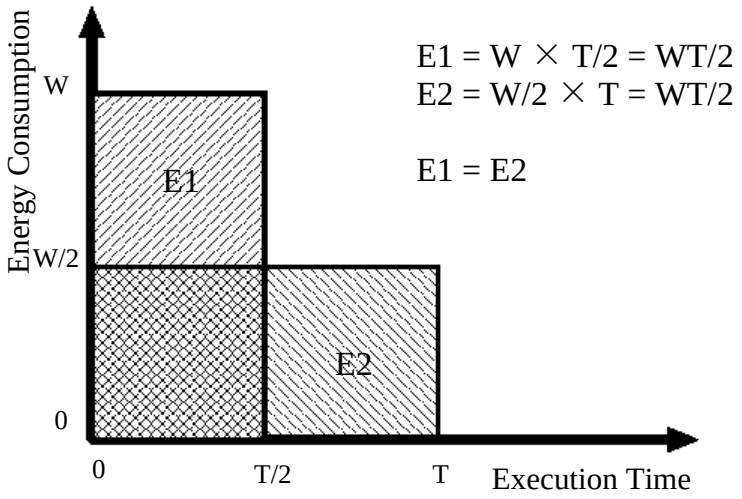

図 6.19: 実行時間と消費電力

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．

6.2 実行時間評価

まず，分岐予測器に対し PG を単純に適用した場合、また本手法を用いて PG を適用した場合に PG 非適用の VSP と比較しどれほどの性能低下を招くか見積もるために、実行時間評価を行った．本評価では SimpleScalar Tool Set 内の out-of-order 実行シミュレータをベースに分岐予測器部分を従来の gshare 型からハイブリッド型に改良し，PG の適用は LE モード時に適用回路部分の保持データを破壊することで疑似的に実装した．命令セットは SimpleScalar PISA であり，SPECint2000 の 5 本をベンチマーク・プログラムとして用いた．実行は最初の 20 億命令をスキップした後，20 億命令を測定に用いた．

SPEC2000 を用いて評価した結果を図 6.20 に示す．左側の系列が分岐

表 6.2: ベンチマーク

ベンチマーク名	処理内容
gzip	圧縮
vpr	FPGA の配置配線
bzip2	圧縮
parser	文字列処理
gcc	C 言語コンパイラ

表 6.3: simple scalar のプロセッサ構成

分岐予測器構成	
selector table	1entry=2bit 2048 entry
branch predictor type 1	bimodal 型 1entry=2bit 2048 entry
branch predictor type 2	gshare 型 GR: 8bit PHT: 1entry=2bit 4096 entry

予測器全体に PG を単純に適応した場合の実行時間を，右側の系列が本手法を用いて PG を適用した場合の実行時間を表している．また，それぞれの結果は通常の VSP の結果を 1 として正規化してある．分岐予測器全体に PG を単純に適応した場合平均 20.4%の実行時間増を招いてしまっているが，本手法を用いて PG を適用することで平均 5.6% の実行時間増におさえることが出来た．分岐予測自体がプログラム依存の性質が強いため各ベンチマークプログラムによってその効果にばらつきはあるが，全てのベンチマークプログラムにおいて本手法を用いることで実行時間増をかなり抑えることがわかった．

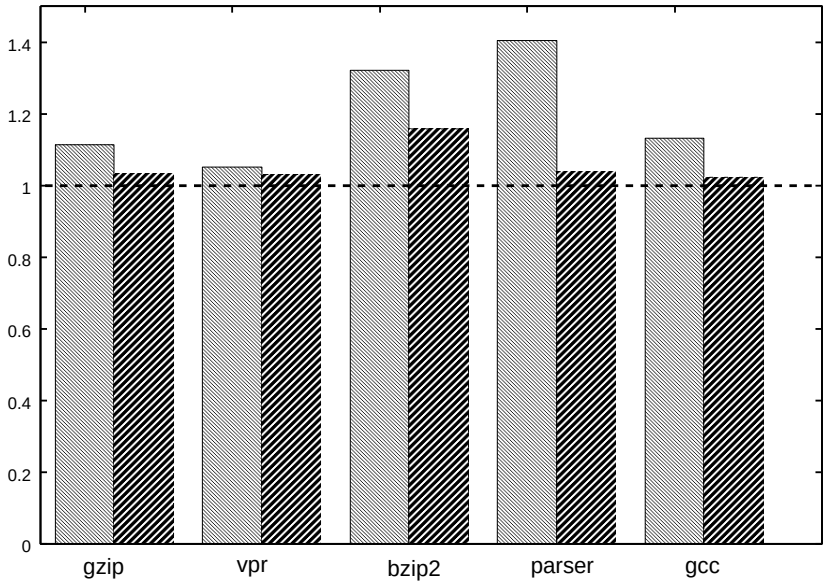

図 6.20: PG 適用による性能低下の見積もり

6.3 電力評価

6.3.1 評価方法

本節では PG 非適用の場合と本手法を比較するため実行時間，消費エネルギー，電力遅延積の評価を行う．前節で実行時間評価を行うために用いた Simple Scalar はスーパースカラ，アウトオブオーダ実行のアーキテクチャであった．しかし，電力評価を行うためには verilogHDL 等のハードウエア記述言語を用い実装する必要があり，現状ではまだ上記のよう

なアーキテクチャで VSP が実装されていない．そこで本研究の評価では，MIPS R3000 互換のシングルパイプライン，インオーダ実行の VSP プロセッサをベースに本手法を適用し実装を行った．

また，ROHM0.18μm テクノロジーを使用し，論理合成は Synopsys 社の DesignCompiler を用いて行った．消費電力の評価には Synopsys 社の PrimetimePX を用い，ベンチマークプログラムは MiBench[17] にて配布されているものの中から，整数の 2 乗根を求める int sqrt，long 型の変数中で 1 のビット数を数える bit count，文字列をクイックソートする quick sort を用いた．quick sort のアルゴリズムには Newlib[18] のものを利用した．また，ベンチマークで使用するデータは PrimeTime PX の計算時間を考慮し，数 10 万命令程度で終了するように調整し，シミュレーション時間の都合上，上記 3 つのプログラムで評価を行った．プロセッサの構成を表 6.4 に示す．

表 6.4: MIPS R3000 互換 VSP プロセッサの構成

パイプライン構成	
HS mode	9 stage
LE mode	3 stage
動作周波数	
HS mode	100MHz
LE mode	25MHz
分岐予測器構成	
selector table	1entry=2bit 256 entry
branch predictor type 1	bimodal 型 1entry=2bit 256 entry
branch predictor type 2	gshare 型 GR: 4bit PHT: 1entry=2bit 512 entry

評価に使用したライブラリが 0.18μm のプロセスであるため，最新プロセスのように顕著にリーク電力が現れない．そこで，リークエネルギーに関してはベースとなった MIPS R3000 プロセッサの動的消費電力、つま

りVSPを常時HSモードで動作させた時の動的消費電力をもとに，リーク電力が全消費電力の50%程度に相当するよう値を仮定し，そのリーク電力を用いて評価を行った．この場合、LEモード時の動的消費電力はHSモード時の4分の1以下になるので，LEモード時のリーク電力の比率はHSモードに比べ大きくなる．

また，Synopsys社のDesignCompilerを用い論理合成した際に得られたパワーゲーティング適用箇所のハードウエア量(プロセッサ全体の25%に相当)を元に，LEモード時にはリーク電力の25%が削減されるとして，モデル式化して評価を行った．

ベースプロセッサの消費電力から仮定したリーク電力をLP，PG非適用のVSPの実行時間をT_{nomal}とすると，PG非適用のVSPのリークエネルギー$LeakE_{nomal}$は式5で表される．

$$LeakE_{nomal} = LP \times T_{nomal} \tag{5}$$

また，本手法適用のVSPの実行時間をT_{pg}，HSモードで動作した実行時間の割合をTR_{HS}とすると，PG適用箇所であるプロセッサ全体の25%のリーク電力はHSモード時にしか消費されないので，本手法適用のVSPのリークエネルギー$LeakE_{pg}$は式6で表される

$$LeakE_{pg} = LP \times T_{pg} \times (0.25TR_{HS} + 0.75) \tag{6}$$

6.3.2 評価結果

評価結果を図6.21に示す．グラフ中の系列は左からそれぞれ実行時間，消費エネルギー，電力遅延積を示しており，PG非適用のVSPの結果を1として正規化してある．今回実装したVSPではモード切換えの負荷変動検出をメモリアクセス回数をカウントすることで行っており，HSモードからLEモードへ，LEモードからHSモードへ切換わる際の閾値をそれぞれパラメータとして与えている．本評価では従来のVSPにおいて電力遅延積が最善となった時の閾値を用いて評価を行った．

結果，すべてのベンチマークプログラムにおいて性能が改善されており，電力遅延積において最大21.3%改善ができた．この結果は，多くのベンチマークプログラムにおいてLEモードで長く動作したことによりPG適用による低電力効果がかなり大きくなった事によるものと考えられる．また，モード切換の頻度も少なかったことにより本手法適用による実行時間増加も最大0.21% に留まった．

このような傾向の結果が得られたのは，用いたベンチマークプログラムが短かった事，VSP のモード切換がプログラムの負荷を最適に追従出来ておらず細粒度に行われていなかった事が原因と考えられる．今回実装に用いた切換コントローラは Simple Scalar 上でスーパースカラ，アウトオブオーダ向けに提案されたものであったが，シングルパイプライン，インオーダ向けのコントローラも提案されている．そのコントローラを用いる事で今回評価に用いた qsort 等のプログラムでも細粒度にプログラムの負荷を追従できることが明らかになっている．このコントローラを実装に用いることで今回の評価結果とはまた異なる傾向の結果が得られると考えられる．

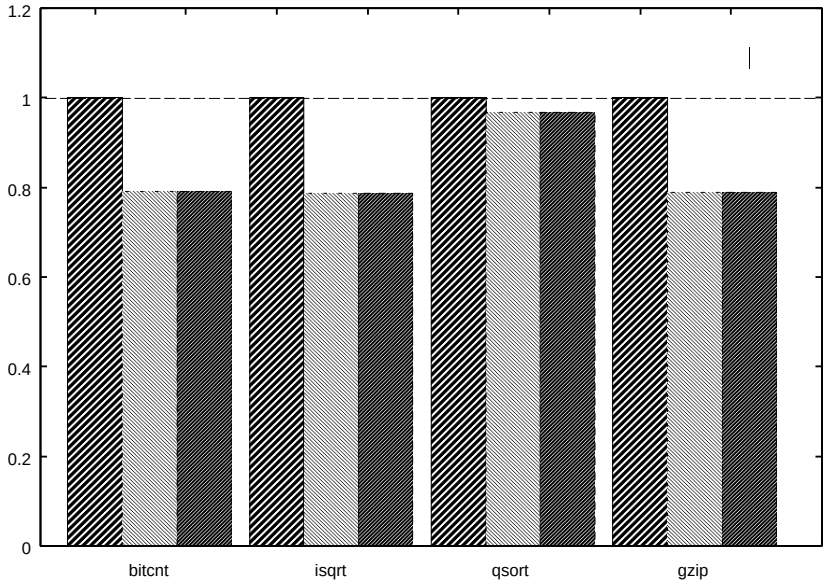

図 6.21: 実行時間，消費エネルギー，電力遅延積

6.4 オーバーヘッド評価

6.4.1 オーバーヘッドと BET

第 5 章で述べた通り，PG を適用した回路をスリープさせたりウェイクアップさせたりする際には，PS の駆動と仮想的なグランド線に溜まった電荷を充放電することによるオーバーヘッドが発生する．モード切換時

の電力消費の様子を図 6.22 に示す．一般に PG を適用する回路の規模が大きい場合，スリープイン時オーバーヘッド電力，ウェイクアップ時オーバーヘッド電力が大きくなる．また，スリープインしてからリーク電力が下がりきるまでの遅延や，ウェイクアップしてから仮想的なグランド線から電荷を放電し回路が正常に動作するまでの遅延も大きくなる．

一回のモード切換により発生するオーバーヘッド電力をペイ出来るまでにかかるスリープモードのサイクルは BET(Break Even Time) と呼ばれており，最短で数 100 サイクル程度の細粒度で頻繁にモード切換を行う事を想定している VSP ではオーバーヘッドの大きさ，BET の長さが重要となる．

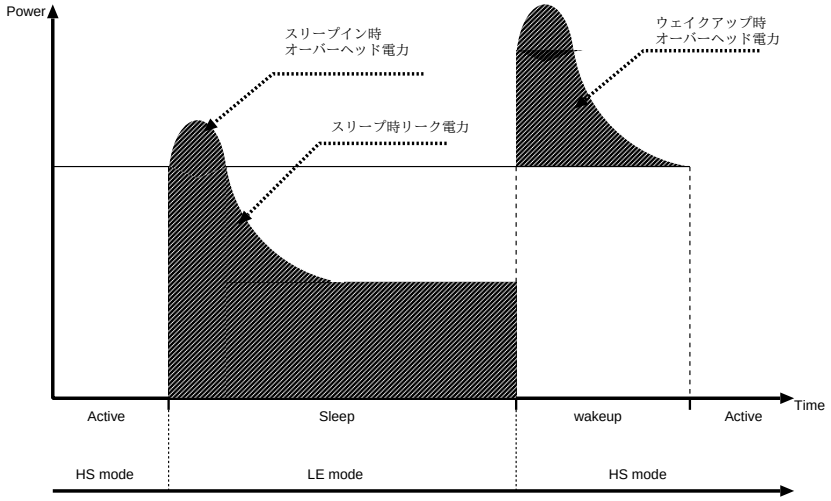

図 6.22: モード切換時の電力消費の様子

6.4.2 概算評価

本研究ではまだ最新の微細プロセスによる VSP 実装や, PG を行う上で必要となるパワースイッチのような特殊なセルの物理設計まで行えていない．前節で述べた本研究の評価はオーバーヘッドを無視した評価となっているため，本節でオーバーヘッドの評価を行う．

本研究で PG 適用対象である分岐予測器の内部はほぼ履歴テーブルで占められている．そこで本評価では VDEC により提供されている Renesas 社の 40nm プロセスを用いた SRAM 回路生成ツールを使用し，生成され

たSRAM回路の電源ライン分の配線容量を概算することでオーバーヘッドを評価した．

本手法でPG適用するgshare分岐予測器と選択テーブルのエントリー数がそれぞれ512エントリー，256エントリーであり，それぞれのSRAMの配線容量を計算し合計したところ2.1pFとなった．この配線容量の電源ラインに対しパワースイッチを通して電荷を充放電する電力がオーバーヘッドとなるため図6.23のような回路で評価を行うことが出来る．

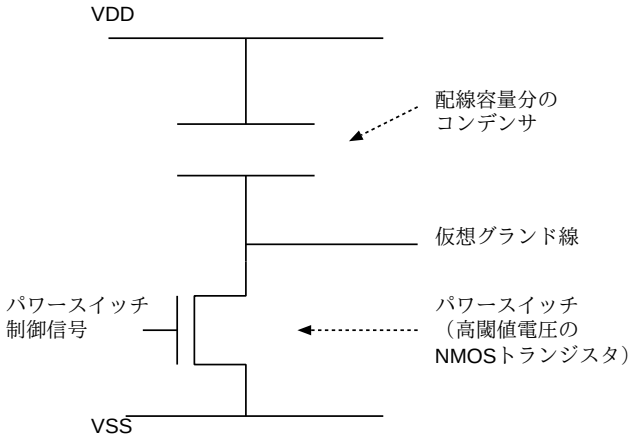

図 6.23: オーバーヘッド評価回路

最小のインバータで用いられているゲート長のNMOS 1個をパワースイッチとして評価したところ，ウェイクアップ時の遅延は37.6nsとなった．HSモード時に100MHzの動作周波数で動作する本研究でのVSPでは約4サイクル分に相当する．この遅延はパワースイッチの数やゲート長に反比例して短くなることに加え，第5章で述べたように遅延の隠蔽も可能なことから十分小さいと言える．また，オーバーヘッド電力も21.0×10^{-13}とかなり小さなものとなった．生成したSRAM回路自体のリーク電力の測定が不可能なためBETの算出はできなかったが，オーバーヘッド電力の小ささからBETが短くなることは明らかである．

このような小さなオーバーヘッドに留まった理由としては本評価にもちいたPG適用回路が想定していたよりも小さかったことによるものと考えられる．これは論理合成後のランダムロジックでプロセッサ全体の25%あった分岐予測器のハードウエア量が，SRAMに最適化されることでかなり小さくなったからである．

7　おわりに

7.1　まとめ

本研究では高性能かつ低消費エネルギーを実現するためにVSPの分岐予測器に対しPGの適用を試みた．その適用をハイブリット分岐予測器内の一部の履歴テーブルに適応的に行うことで，履歴破壊による大幅な性能低下を抑制する手法を提案し，実行時間，消費エネルギー，電力遅延積においてPG非適用のVSPプロセッサと比較評価を行った．その結果，電力遅延積において最大21.3%の性能改善が得られた．また，パワースイッチ駆動によるオーバーヘッドもかなり小さなことが明らかとなった．

7.2　今後の展望

本研究では電力評価の際，リーク電力をかなり大雑把に仮定した上で評価を行ったが，正確に評価する為に今後は最新の微細プロセスを用いて実装しリーク電力を含めた評価を行う必要がある．

また本研究では分岐予測器に対してPGを適用した時の低電力効果と性能低下を評価するためにPG適用箇所を分岐予測器に限定して評価を行っている．しかし，VSPには分岐予測器以外にもパイプラインレジスタやLDS-Cellの一部等LEモード時に動作が停止しPG適用可能な箇所が存在する．そして，上記のような部分以外でもクリティカルパスではなく高速動作が必要でない箇所に対してはMulti Vth等を用いる事で更なるリーク電力削減が可能であり，本手法及びVSPの有用性が高まると考えられる．

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] J. Pouwelse, K. Langendoen, H. Sips, “Dynamic Voltage Scaling on a Low-Power Microprocessor”, Proc. of The 7th ACM Int. Conf. on Mobile Computing and Net- working (Mobicom), pp .251-259, July 2001

[2] Min Yeol Lim, Vincent W.Freeh, “Determining the Minimum Energy Consumption using Dynamic Voltage and Frequency Scaling”, IEEE, 2007.

[3] 市川 裕二，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”, 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, May 2006

[4] 中林 智之，佐々木 敬泰，大野 和彦，近藤 利夫，“クロック系消費電力に着目した可変段数パイプラインプロセッサの低電力化 “，電子情報通信学会論文誌，Vol. J94-D，No. 4，pp. 646-656，April 2011

[5] Tomoyuki Nakabayashi, Takahiro Sasaki, Kazuhiko Ohno and Toshio Kondo, “Design and Evaluation of Variable Stages Pipeline Processor Chip”, Proc. of Int. Symposium on Information and Automation, November 2010, (In press)

[6] Takahiro Sasaki, Kazumasa Nomura, Tomoyuki Nakabayashi, Kazuhiko Ohno and Toshio Kondo “Fine Grain Controller for Variable Stages Pipeline Processor” Proc. of the 25th International Technical Conference on Circuits/Systems, Computers and Communications, pp.748-751,2010

[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”，2003 年先進的計算基盤システムシンポジウム SACSI2003, pp. 283-290，May 2003

[8] 嶋田創，安藤秀樹，島田俊夫，“パイプラインステージ統合と DVS の併用による消費電力の削減（省電力方式）”，情報処理学会論文誌（コンピューティングシステム），Vol. 48，No. SIG3(ACS17)，pp. 75-87，2007

[9] Jun Yao, Shinobu Miwa, Hajime Shimada and Shinji Tomita, “A Dynamic Control Mechanism for Pipeline Stage Unication by Identifying Program Phases”, IEICE Transactions on Information and Systems, Vol. E91-D, pp. 1010-1022, 2009

[10] Jinson J Koppanalil, Prakash Ramrakhyani, Sameer Desai, Eric Rotenbergm, Anu Vaidyanathan：A Case for Dynamic Pipeline Scaling, ACM 2002

[11] Efthymiou， A. and Garside，J. D. : Adaptive Pipeline Depth Control for Processor Power-Management， Proc. of Int. Conf. on Computer Design 2002，pp.454-457,（2002）.

[12] Atsuki Inoue, “Design Constraint of Fine Grain Supply Voltage Control LSI”, Proc. of the 16 th Asia and South Pacific Design Automation Conference (ASP-DAC 2011), pp. 760-765, January 2011

[13] J. Tschanz, et al,“ Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage, ”IEEE Journal of Solid-State Circuits, vol.37, no.11, pp.1396-1402, Nov. 2002.

[14] 大久保直昭　他, “細粒度動的スリープ制御による動作時リーク電力低減手法”, 情報処理学会 DA シンポジウム 2006, pp.199-204,July 2006

[15] J. E. Smith,“A study of branch prediction strategies.”, In Conference proceedings of the eighth annual symposium on Computer Architecture, pp.135-148, 1981

[16] S. McFarling,“ Combining Branch Predictors ”, WRL Technical Note TN-36, Digital Equipment Corporation, June 1993

[17] MiBench. http://www.eecs.umich.edu/mibench/

[18] Newlib. http://sourceware.org/newlib/