

低消費電力化のための可変パイプライン

嶋田 創[†] 安藤 秀樹[†] 島田 俊夫[†]

近年のモバイルプロセッサにおいては、低消費電力と高性能の両立が求められている。そのため、近年では可変クロック周波数や可変電源電圧を用いている。しかし、プロセスの微細化によるリーク電流の増加によって、可変電源電圧は有効に使えなくなると考えられる。

我々はクロック周波数を低下させた時の性能低下の抑制と、消費電力削減の手法として、パイプラインレジスタのバイパスによるパイプラインステージの統合を提案する。これにより、クロック周波数を標準の4分の1に低下させた時、単純にクロック周波数を低下させた場合と比較して、最大限の見積もりで約24%の消費電力の削減と、性能低下を約2分の1に押さえることができることを確認した。また、0.18 μ m プロセスにおいて、電力遅延積を従来より約18%改善できることを確認した。

Pipelining with Variable Depth for Low Power Consumption

HAJIME SHIMADA,[†] HIDEKI ANDO[†] and TOSHIO SHIMADA[†]

In recent mobile processors, both low power consumption and high performance are demanded. Therefore, variable clock rate and variable supply voltage are used in current mobile processors. However variable clock rate will be hard to use in the future processes because of increasing leakage current.

We propose pipeline stage unification by bypassing pipeline registers for reduction of performance degradation and power consumption when clock rate is low. We confirm that our technique can reduce power consumption by approximately 24% and suppress performance degradation by approximately 50% compared with the conventional supply voltage reduction technique. We also confirm that power-delay product in program execution is improved by approximately 17% in 0.18- μ m process technology.

1. はじめに

これまでのプロセッサ技術において、大きく注目される技術は性能向上のための技術が大部分であった。しかし、近年ではモバイルコンピュータや組み込みシステムに高性能プロセッサを搭載するにあたり、消費電力と性能のトレードオフが問題となっている。近年のモバイルプロセッサでは、この問題に対し、クロック周波数とプロセッサの電源電圧を動的に変更することによって対処している^{1)~3)}。これらの現在の手法のうち、プロセッサの電源電圧の動的な変更という方法は、プロセス技術の進歩とともに有効に使えなくなると考えられる。理由は、この先のプロセス技術の進歩において、閾値電圧のスケーリングファクタが小さくなり、電源電圧と閾値電圧の差が少なくなると予想されるからである。

そこで、我々は新たな方法によるエネルギーの削減を提案する。近年のプロセッサにおける命令の実行は、多くのパイプラインステージに分割されており、多い

例では Intel Pentium4⁴⁾ の 20 段がある。このような多段のパイプラインは高いクロック周波数を達成するために必要であり、動的なクロック周波数の変更を用い、プロセッサの負荷が低い時に低いクロック周波数で動かすときには必要ない。逆に、クロック周波数が低ければ、多段のパイプラインでは分岐予測ミスペナルティやキャッシュアクセスレイテンシ等が大きいいため、同一のクロック周波数で、パイプライン段数の少ないプロセッサよりも、IPC (Instructions Per Cycle) は低下することになる。そこで、我々はプロセッサのクロック周波数を低下させた時において、パイプラインステージの統合を行うことによる IPC の向上と、クロックドライバの電力削減による消費電力の削減を提案する。これにより、あるプログラムの実行に要するエネルギーの削減になる。

2. 近年のプロセッサの消費電力削減手法と可変電源電圧の限界

2.1 近年のプロセッサの消費電力削減手法

Intel Mobile Pentium III-M, Intel XScale¹⁾, Transmeta Crusoe TM5400²⁾, AMD Mobile Athlon4³⁾ のような、近年のモバイルプロセッサでは、低消費電力

[†] 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

と性能を両立するため、プロセッサの負荷に応じてクロック周波数と電源電圧の動的な変更を行っている。プロセッサの消費電力は式 (1) で求まる。

$$P = pCV^2f \quad (1)$$

P : 消費電力
 p : スイッチングするトランジスタの割合
 C : 総キャパシタンス
 V : 電源電圧
 f : クロック周波数

上式よりわかるように、クロック周波数を低下させればそれに比例して、また、電源電圧を低下させればその 2 乗に比例して消費電力は低下する。実際に、この式通り、Crusoe TM5400 のクロック周波数 200MHz、電源電圧 1.1V 時の消費電力は、クロック周波数 700MHz、電源電圧 1.65V 時の約 13%となっている²⁾。

2.2 可変電源電圧による消費電力削減の限界

現在のモバイルプロセッサで主流となっている、可変電源電圧による消費電力の削減は、プロセスの微細化が進むと有効に使えなくなると考えられる。この技術は、電源電圧を標準の電圧とプロセッサが動作する下限の電圧の間で変化させるものである。トランジスタは閾値電圧以下では動作しないため、閾値電圧が可変電源電圧の下限と考えられる。プロセスの微細化が進むと、閾値電圧のスケーリングが悪くなるため、電源電圧に対する閾値電圧の割合が、従来よりも大きくなると予想される。

閾値電圧を大きく下げられない理由は、閾値電圧を下げることにより、リーク電流が大幅に増加することにある。リーク電流はトランジスタがオフの時に、ドレイン電流を完全に遮断できずに流れてしまう電流である。このリーク電流が発生する原因の一つに、ゲート電圧が閾値電圧以下であってもトランジスタのドレイン電流はわずかに流れるという点がある。これによるリーク電流を、サブスレッショルド電流と呼ぶ。

サブスレッショルド電流の減少率は式 (2) で表される。

$$\left[\frac{d(\log I_D)}{dV_G} \right]^{-1} = \left(\frac{k_B T}{q} \ln 10 \right) \left(1 + \frac{C_D}{C_{gox}} \right) \quad (2)$$

I_D : ドレイン電流
 V_G : ゲート電圧
 k_B : ボルツマン定数
 T : 絶対温度
 q : 電子の電荷
 C_D : 空乏層容量
 C_{gox} : ゲート容量

式 (2) 中における $\frac{k_B T}{q}$ の値は熱電圧と呼ばれ、その

大きさは絶対温度 300K において 0.026V である。これを、式 (2) に代入すると

$$\left[\frac{d(\log I_D)}{dV_G} \right]^{-1} = 0.06 \left(1 + \frac{C_D}{C_{gox}} \right) \quad (3)$$

となる。式 (3) より、一定のゲート電圧の減少に対して最もドレイン電流の減少率が大きくなるのは、

$$\frac{C_D}{C_{gox}} = 0 \quad (4)$$

の時、その時にゲート電圧が 0.06V 下がるごとに、ドレイン電流は 10 分の 1 となる。これより、閾値電圧が 0.06V 下がればサブスレッショルド電流は 10 倍になることがわかる。

以上より、プロセスの微細化が進むと、サブスレッショルド電流を減少させるために、閾値電圧は単純なスケーリングした値よりも上昇させることになると考えられる。このため、近年のプロセッサが利用している、電源電圧を上限と下限との間で動的に変更する手法は、変更可能な電圧の幅が小さくなることにより、効果的に利用することが難しくなると予想される。

3. 可変パイプラインの提案

我々はクロック周波数を低下させた時にパイプラインレジスタの動作を停止し、パイプラインレジスタへの入力信号を出力にバイパスさせ、ステージを統合することを提案する。本節では、パイプラインレジスタの動作を停止し、信号をバイパスさせることに必要な回路と、削減される消費電力を説明する。

3.1 可変パイプラインの説明

可変パイプラインを実現するには、統合を行うステージ間のパイプラインレジスタをバイパスする回路が必要になる。図 1 にマルチプレクサとバイパス回路を追加したパイプラインレジスタ部を示す。パイプラインレジスタの直前に設けられたマルチプレクサを用い、高いクロック周波数で動作中にはパイプラインレジスタに信号を入力して保持し、低いクロック周波数で動作しており、ステージ統合を行う時には、次のステージへと信号をバイパスさせる。

図 2 に可変パイプラインを実現するための、クロック分配線の変更を示す。クロックジェネレータより出力されるクロック分配線を、常時駆動するシステムとパイプラインステージ統合時には停止するシステムとの、2 系統に分割する。同時に、前述のパイプラインレジスタ部のマルチプレクサを切り替えるための信号線も追加する。なお、クロック周波数の低下の度合いによって、2 段のパイプラインステージの統合と 3 段以上のパイプラインステージの統合を切り替える場合、切り替えの段数に応じたクロック分配線と複数のマルチプレクサ切り替えの信号線が必要となる。

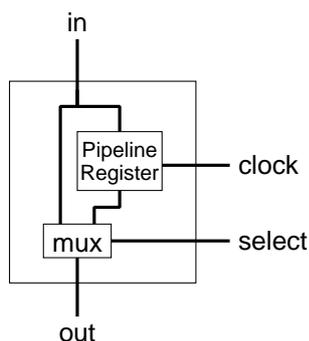
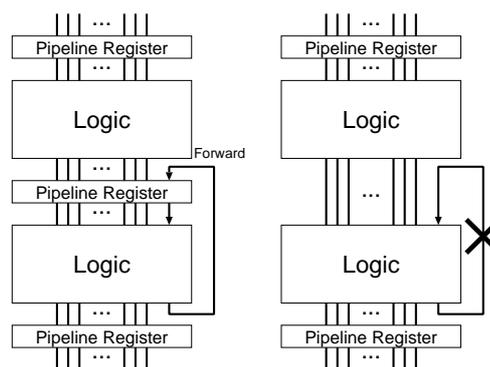


図1 バイパス回路を持つパイプラインレジスタ部



(a) (b)
図3 パイプラインステージを統合できない例

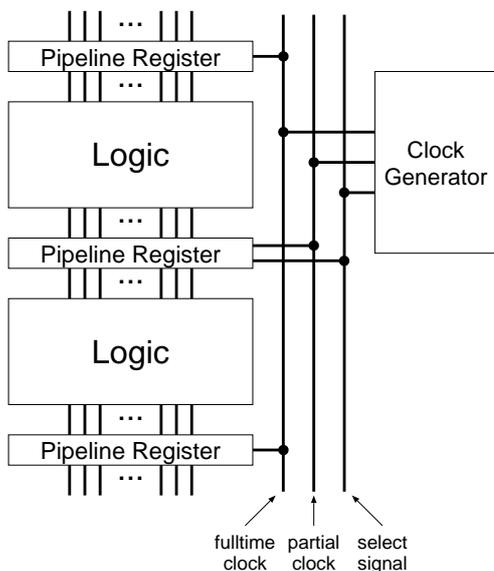


図2 クロック発生装置より複数系統のクロックを出力

クロック分配線を複数系統に分割する理由は、停止するパイプラインレジスタへのクロックの供給を、クロックジェネレータにおいて停止するためである。これにより、クロックの供給をパイプラインレジスタへの入力直前で停止する場合と比較して、パイプラインレジスタのスイッチングの電力だけではなく、クロック分配線をドライブすることによる消費電力も削減することが可能となる。

次にパイプラインステージの統合の制限について述べる。図3(a)のように、パイプラインステージの出力した値をフォワーディング等の目的で自身のステージの入力とする場合、パイプラインステージの統合はできない。なぜなら、パイプラインステージを統合すると図3(b)のような構成となり、これは論理的に誤りであるからである。

3.2 動作電力削減の見積もり

Compaq Alpha 21264 プロセッサをモデルとして消費電力の削減の見積もる。まず、今回の可変パイプ

ライン化によって削減対象となる、パイプラインレジスタ、クロック分配線等のクロックによって駆動される機構がプロセッサの何割を占めるかを調べた。文献5)によると、21264の消費電力の32%がクロックドライバで消費されている。パイプライン段数の削減の割合に応じてクロックドライバの消費電力が削減されると仮定すると、たとえば、20段のパイプラインを10段のパイプラインとした場合、

$$32\% \times \frac{10}{20} = 16\% \quad (5)$$

の消費電力を削減できる。

上記計算では、マルチプレクサとその信号線の消費する電力を考慮に入れていない。また、クロック分配線を複数系統用意することによって配線容量が増加し、単純な割合で消費電力を削減できないことが考えられる。これにより、実際には前述の計算ほど消費電力は低下しないと考えられる。上記の評価は時間がかかるため、今回は消費電力削減量の最大を見積もり、消費電力の削減量の精密な評価は、後の課題とする。

4. 評価

4.1 評価条件

SimpleScalar Tool Set⁶⁾中の out-of-order 実行シミュレータを用いて、パイプラインの段数を変化させた時の性能を測定した。命令セットはMIPS R10000⁷⁾である。ベンチマークプログラムは表1に示すように、SPECint95の8本を用いた。パイナリはGCC ver.2.7.2.3を用い、-O6 -funroll-loopsのオプションでコンパイルしたものを用いた。シミュレーション時間が過大にならないようにするために、関数の出現頻度をほぼ維持しつつ、実行命令数がおおよそ75M~450Mになるようにそれぞれのベンチマークプログラムへの入力を調整した。

シミュレーションにおいて仮定したプロセッサの構

表1 ベンチマーク・プログラム

ベンチマーク名	入力	命令数
compress95	bigtest.in	95M
gcc	genoutput.i	84M
go	2stone9.in	75M
jpeg	specmun.ppm	450M
li	train.lsp	183M
m88ksim	ctl.in	100M
perl	scrabbl.in	80M
vortex	vortex.in	80M

表2 プロセッサの構成

命令発行幅		8 命令
RUU		64 エントリ
LSQ		32 エントリ
メモリポート		8 ポート
機能ユニット	整数	8 個
	整数乗除算	4 個
	浮動小数点	8 個
	浮動小数点乗除算	4 個
TLB	命令	64 エントリ/4way
	データ	128 エントリ/4way
分岐予測	予測手法	gshare
	履歴長	6 ビット
	インデックス長	13 ビット
	BTB	2048 エントリ/4way
	RAS	16 エントリ
キャッシュ	L1 命令	64KB/32B ライン/1way
	L1 データ	64KB/32B ライン/1way
	L2	2MB/64B ライン/4way

表3 変更するレイテンシ(サイクル)

	条件 1	条件 2	条件 3
分岐予測ミスレイテンシ	20	10	5
L1 キャッシュヒットレイテンシ	4	2	1
L2 キャッシュヒットレイテンシ	16	8	4

成を表2に示す。表2にロード/ストアユニット数がないが、これは、SimpleScalarでは整数ユニットがロード/ストアユニットのアドレス生成器を兼ねているためである。また、可変パイプライン構成により、レイテンシが変化する部分については表3に示す。条件1は本評価での基準となるレイテンシである。条件2は可変パイプライン化により2ステージ統合を行った場合、条件3は4ステージ統合を行った場合のレイテンシである。

4.2 可変パイプラインによる性能低下抑制の結果

図4にクロック周波数とレイテンシを変更した時の性能低下を示す。グラフの横軸はベンチマークであり、各ベンチマークの3本の棒グラフはクロック周波数の倍率が左から1, 2分の1, 4分の1を表している。縦軸はクロック周波数の倍率1, 可変パイプライン化なしの性能を1として、性能を正規化したものである。

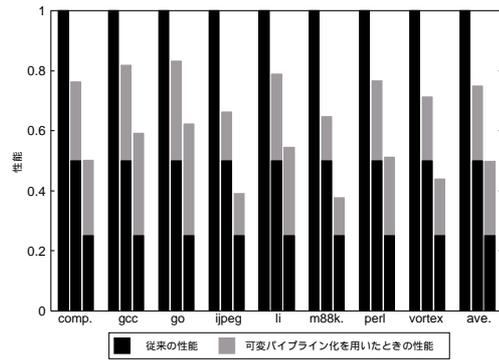


図4 性能低下の抑制

グラフの黒の部分は可変パイプラインなし時の性能であり、グラフの灰色の部分は、

$$fN = \text{一定} \quad (6)$$

f : 周波数の倍率

N : 統合するパイプラインの段数

となるように可変パイプライン化した時の性能である。そのため、2本目のグラフの灰色部のレイテンシは2ステージ統合を行った条件2に、3本目のグラフの灰色部のレイテンシは4ステージ統合を行った条件3である。

この結果より、クロック周波数を低下させた時に可変パイプライン化を行うことにより、性能の低下を抑制できることがわかる。従来では、クロック周波数を2分の1にした時には元のクロック周波数の時の2分の1まで性能低下していたが、可変パイプライン化によって、元のクロック周波数の約4分の3の性能低下に押さえることができる。さらに、元のクロック周波数を4分の1まで低下させた時には、従来では元のクロック周波数の4分の1まで性能低下していたものを、可変パイプライン化を行うことによって、元のクロック周波数の約2分の1の性能低下に押さえることができる。

4.3 電力遅延積の評価

可変パイプライン化がモバイルコンピュータの電池エネルギー節約に効果があることを示すため、電力遅延積による評価を行う。電力遅延積とは、あるプログラムを実行を完了するまでに必要とされるパワーを示すものであり、以下の式で表され、値は小さいほど良い。

$$Energy = PE_t \quad (7)$$

P : プロセッサの消費電力(式(1))

E_t : プログラムの実行時間

図5に0.18 μ mプロセスにおける電力遅延積による評価の結果を示す。グラフの横軸はベンチマークであり、各ベンチマークの5本のグラフは左からそれぞれ表4における値であり、可変電源電圧を用いた従来の

表 4 グラフの各項目の値

凡例	レイテンシ	電源電圧 (V)
クロック周波数 1 倍	条件 1	1.65
クロック周波数 2 分の 1 倍 (可変電源電圧を用いる)	条件 1	1.35
クロック周波数 2 分の 1 倍 (可変パイプライン化により 2 ステージ統合)	条件 2	1.65
クロック周波数 4 分の 1 倍 (可変電源電圧を用いる)	条件 1	1.1
クロック周波数 4 分の 1 倍 (可変パイプライン化により 2 ステージ統合)	条件 3	1.65

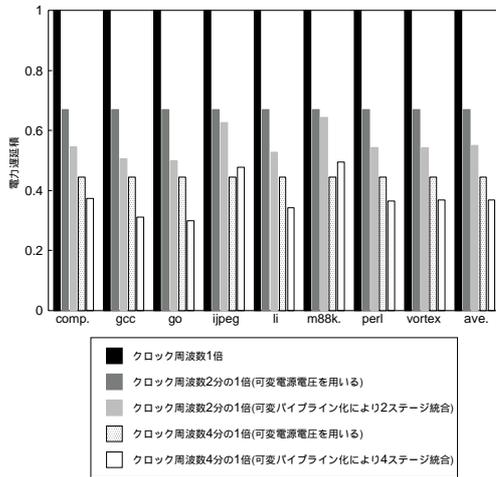


図 5 電力遅延積

性能と可変パイプライン化を行ったときの電力遅延積をそれぞれ示している。クロック周波数の倍率と電源電圧の関係は、Crusoe TM5400²⁾の取る値より決定した。

縦軸はクロック周波数 1 倍、電源電圧最大状態の電力遅延積を 1 として、性能を正規化したものである。

従来の可変電源電圧のみの場合、クロック周波数を 2 分の 1 にした場合、電力遅延積は平均で約 33% 改善されている。また、周波数を 4 分の 1 にした場合、電力遅延積は平均で約 56% 改善されている。これに対し、可変パイプライン化を行った結果では、クロック周波数を 4 分の 1 とした時の m88ksim と jpeg では従来よりも性能が悪くなっているが、それ以外のベンチマークでは性能は良くなり、全体で平均すると、クロック周波数 2 分の 1 時に平均で約 45% の改善、クロック周波数 4 分の 1 時に平均で約 63% の改善となり、従来よりも性能が良くなるのがわかる。従来からの改善率の増加率は、クロック周波数 2 分の 1 時に平均で約 18%、クロック周波数 4 分の 1 時に平均で約 17% となる。

5. 関連研究

プロセッサの消費電力の削減、可変パイプラインに関する関連研究を挙げる。

文献 8) では、動的にキャッシュと命令ウィンドウの

サイズを変更しながらクロック周波数を変更することを提案している。大容量キャッシュや大きなサイズの命令ウィンドウはクロック周波数を高めるのに障害となるが、プログラムによってはクロック周波数の高速化よりもキャッシュの大容量化や命令ウィンドウの拡大の方が有効なものもある。そこで、この論文では、プログラムの実行を監視しながら動的にキャッシュサイズ、命令ウィンドウサイズ、クロック周波数を変更し、性能を向上させることを提案している。

文献 9) では、動的に命令発行幅を変更して消費電力を削減することを提案している。この論文では、命令発行のための調停 / 選択機構とそれに続く機能ユニットが消費電力の大きな部分を占めていることに着目し、命令発行幅を拡大しても性能向上しない時にはその一部を停止させている。これにより、大きな性能低下なしに命令キューで 23%、実行ユニットで 13% の消費電力を削減している。

文献 10) では、予測精度の低い分岐命令がいくつか連続した場合、それより先の命令のフェッチを停止して消費電力を削減することを提案している。これは、予測精度が低い分岐より後の命令は、分岐予測ミスによってフラッシュされる可能性が高く、実行される可能性が低いことに着目している。このような、フラッシュされる命令のフェッチは無駄な電力の消費になるため、予測精度の低い分岐命令がいくつか連続した場合、その先の命令はフラッシュされる可能性が高いとして、フェッチを行わないようにしている。この方法により、大きな性能低下なしに分岐予測ミスしたパスからの命令のフェッチを 38% 削減し、それらの命令のフェッチに必要とされていた消費電力を削減している。

文献 11) では、パイプラインを流れるデータ量を削減することにより、消費電力を削減することを提案している。これは、パイプライン内に存在する値の大部分は小さく、上位ビットの大部分は 0 か 1 であり、下位ビットの符号拡張で表すとデータ量を大きく削減できることに着目している。符号拡張を行うことを示す 2,3 ビットの追加ビットのためのデータパスを付加しなくてはならないが、それらの機構によって増加した電力を上回る消費電力を削減できるため、各パイプラインステージにおいて、おおよそ 30 ~ 40% の動作を削減している。

6. ま と め

近年のモバイルプロセッサにおいては、低消費電力と高性能の両立が求められている。我々はクロック周波数を低下させた時の性能低下の抑制と、さらなる消費電力削減のため、パイプラインレジスタ停止によるパイプラインステージの統合による可変パイプライン化を提案した。パイプラインレジスタ停止によってクロック分配線とクロックドライバの消費電力が削減され、パイプラインステージの統合によって分岐予測ミスレテンシ、各メモリ階層のレイテンシが小さくなる。そのため、クロック周波数を4分の1とした時、約24%の消費電力の削減と、性能低下を約2分の1に抑えることができることを確認した。また、同じ条件で電力遅延積は、 $0.18\mu\text{m}$ プロセスにおいて約18%改善できることを確認した。

今後の課題として、消費電力の削減量の正確な見積もりがある。そのため、実際にどれぐらいの割合のパイプラインレジスタを停止させることができるかということと、追加した回路による消費電力の増加、複数のクロック供給システムを設けることによる消費電力の増加はどのぐらい影響するかを正確に見積もらなくてはならない。

謝辞 本研究の一部は、文部省科学研究費補助金基盤研究(C)(課題番号11680351)及び財団法人大川情報通信基金の支援により行った。

参 考 文 献

- 1) S.Leibson: XScale (StrongARM-2) Mustles in 1,000 MIPS at 900mW to 62MIPS at 10mW, Sleeps at 0.1mW, *Microprocessor Report*, pp. 1-5, September 2000.
- 2) 浅見直樹, 芳隕太郎, 枝洋樹: Crusoe の船出, 日経エレクトロニクス, No.765, pp.131-165, September 2000.
- 3) Advanced Micro Devices, Inc.: *Mobile AMD Athlon 4 Processor Model 6 CPGA Data Sheet*, 2001.
- 4) P.N.Glaskowsky: Pentium4(Partially) Previewed, *Microprocessor Report*, pp.1-4, August 2000.
- 5) M.K.Gowan, L.L.Biro, and D.B.Jackson: Power Considerations in the Design of the Alpha 21264 Microprocessor, In *proceedings of 35th Design Automation Conference*, pp. 726-731, June 1998.
- 6) D.Burger and T.M.Austin: The SimpleScalar Tool Set, Version 2.0, Technical Report CS-TR-97-1342, University of Wisconsin-Madison Computer Sciences Department, July 1997.
- 7) MIPS Technologies, Inc.: *MIPS R10000 Pro-*

cessor User's Manual, Version2, 1996.

- 8) D.H.Albonesi: Dynamic IPC/Clock Rate Optimization, In *Proceedings of 25th Annual International Conference on Computer Architecture*, pp. 282-292, June 1998.
- 9) R.I.Bahar and S.Manne: Power and Energy Reduction Via Pipeline Balancing, In *Proceedings of 28th Annual International Conference on Computer Architecture*, pp. 218-229, June 2001.
- 10) S.Manne, A.Klauser, and D.Grunwald: Pipeline Gating: Speculation Control For Energy Reduction, In *Proceedings of 25th Annual International Conference on Computer Architecture*, pp. 132-141, June 1998.
- 11) R.Canal, A.Gonzalez, and J.E.Smith: Very Low Power Pipelines using Significance Compression, In *proceedings of 33rd Annual International Symposium on Microarchitecture*, pp. 181-190, December 2000.