

可変電圧プロセッサを用いたシステムレベルの 電力最適化手法

石原 亨 安浦 寛人

九州大学 大学院 システム情報科学研究科
〒 816 福岡県春日市春日公園 6-1

E-mail: {ishihara, yasuuru}@c.csce.kyushu-u.ac.jp

携帯型情報機器の普及とともに、高性能かつ多機能な LSI チップを低電力で動作させることが重要となっている。さまざまな低消費電力化手法が提案されているが、高速化と低消費電力化を両立するテクノロジーが非常に少ないのが現状である。本稿では高速化と低消費電力化を両立させる手法として、電源電圧を変更できるプロセッサを用いてアプリケーション毎にパフォーマンスと電力のトレードオフを最適化する手法を提案する。本最適化手法では、プログラムがアプリケーションプログラム中に時間制約を明示的に記述することにより、コンパイラが静的にアプリケーションの実行に最適な電圧を決定する。実験の結果アプリケーションによっては時間制約を満たして消費電力を 1/10 にできることが分かった。

キーワード：低消費電力設計，電力管理，CMOS マイクロプロセッサ，リアルタイム処理

Power Optimization with Variable Voltage Processor

Tohru ISHIHARA and Hiroto YASUURA

Department of Computer Science and Communication Engineering Graduate School of
Information Science and Electrical Engineering Kyushu University
6-1 Kasuga-koen, Kasuga-shi, Fukuoka 816 Japan

In this paper, we propose system level power optimization technique which realizes low power processing for a set of applications. The proposed technique finds an optimal supply voltage which minimize energy consumption for each application program under time constraint. The time constraint is explicitly specified by a programmer in the source programs. The most important feature of the proposed technique is that a compiler optimizes the operating voltage for each application program statically. For the application programs whose required performances are less than 1/3 of the maximum performance of a CPU, experimental results show that 85% power saving can be obtained with ten variable voltage levels compared with a system which use one fixed supply voltage.

Key Words : Low power design, Power management, CMOS microprocessor, Real time processing

1 はじめに

集積回路技術の向上にともなう、一つのVLSIチップに集積される素子数は指数関数的に増大し、1000万を超える素子を搭載したVLSIチップも開発されている。この間にCMOS LSIの消費電力は3年で4倍の割合で増え続けている。CMOS LSIの発熱は、さらなる高集積化あるいは高速化を制限する最大の要因となっている。さらに、携帯型情報機器の爆発的な普及を背景に、軽量のエネルギー源(バッテリーや電池など)で多様な処理を長時間実行するシステムが非常に重要となっている。

CMOSの発熱と携帯機器の普及を引金にして、高速化のみを追求する時代から低電力化と高速処理を両立させるテクノロジーを要する時代となった。今後システムLSIが性能を増進させつつ低電力化を実現するために、アプリケーションの要求性能に合わせて最低限必要な電力で、与えられた処理を実現することが重要である。

アプリケーションの要求を直接満足させるために、我々のグループではプログラム中にプログラマが要求する演算精度や時間制約を明示的に書き込むことのできるシステムを提案している[1, 2]。さらに筆者は、プログラム中に指定された要求に応じて電力と性能を変化させることのできるプロセッサアーキテクチャPower-Proを提案している。Power-Proアーキテクチャの最大の特徴は、プログラム中に指定された要求に応じてプロセッサの電源電圧とクロック周波数を動的に変化させることである。

本稿ではPower-Proアーキテクチャを使って、プログラムの処理に必要なエネルギーを大幅に削減するための最適化問題を提案し、実用上の有効性について検討する。

2 諸準備

2.1 電圧と遅延時間の関係

CMOS(Complementary Metal Oxide Semiconductor)は、電力消費が他のトランジスタに比べて非常に小さいことから、現在の集積回路を構成する要素の中で必要不可欠となっている。CMOSにおいては、信号の伝搬遅延時間を τ とおくと、 τ と電源電圧 V_{DD} の間には一般的に式(1)の関係が成り立つことが知られている[3]。

$$\tau \propto V_{DD} / (V_G - V_{th})^\alpha \quad (1)$$

V_{th} : しきい値電圧 V_G : ゲート電圧

α は、 V_G が高い領域でのキャリアの速度飽和を表す値で、2から1までの値をとる。短チャネル化により α の値は2から1に近づきつつある。式(1)の関係は、しきい値電圧より大きい範囲で電源電圧を変化させる時、遅延時間は電源電圧にほぼ反比例することを示している。さらに、CMOSがスイッチする時に消費される電力は、式(2)で表される。

$$P = C \cdot f \cdot V_{DD}^2 \quad (2)$$

f : ゲートのスイッチング回数

C : ゲートの負荷容量 V_{DD} : 電源電圧

式(2)は電圧削減が、低消費電力化に与えるインパクトが大きいことを示している。

2.2 低電圧化技術

低消費電力化に対する要求に加えてデバイスの信頼性向上への強い要求にも関わらずこれまで素子の微細化に比べて電源電圧がスケリング(比例縮小)されなかったのは主に以下に挙げる理由による。

1. 電源電圧は既存システムとの互換性から簡単には下げられない。
2. 電源電圧の低下は回路動作速度の低下を招くことから電源電圧はなるべく下げたくない。

1.の問題はレベルシフタを挿入することにより機能的には解決されているといえる。以下では2.に示した低電圧化による回路遅延の増加を解決する方法を検討すると共に本稿で提案するシステムレベルでの電力最適化手法について述べる。

2.2.1 しきい値電圧の最適化

単純に電源電圧を下げると、回路遅延は増加するが電圧に合わせてしきい値電圧も下げてやると動作速度は劣化しない(式(1))。ところが、しきい値電圧の低下はサブスレッショルド電流の増加を引き起こす。しきい値電圧 V_{th} とサブスレッショルド電流 I_{sub} の関係を式(3)に示す。

$$I_{sub} = I_0 \cdot 10^{-\frac{V_{th}}{\Delta V_{th}}} \quad (3)$$

I_0 は定数で、トランジスタがONした瞬間の電流を表している。サブスレッショルド電流 I_{sub} とはゲート電圧がしきい値以下の時に流れる電流である。これはゲート電圧0Vでどれくらいのリークがあるかを見る目安になる[3]。サブスレッショルド電流はしきい値の低下に対して指数関数的に増加するということが注目すべき点である。

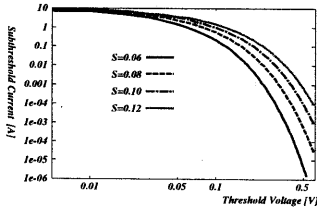


図 1: しきい値電圧とサブスレッショルド電流の関係

図 1 にしきい値電圧とサブスレッショルド電流の関係を示す。ただし、 $I_0 = 1.0[mA]$ 、回路規模は 200 万トランジスタと仮定した。プログラムの動作状態に合わせてしきい値電圧を調節することによりエネルギー消費と遅延時間のトレードオフを最適化することができる。

2.2.2 可変電圧プロセッサ

低電圧化による回路遅延の増加の問題は、微細化技術の進歩により部分的に改善されているが、回路遅延と消費電力のトレードオフを考慮してシステムを最適化することでさらに改善することができる。我々はアプリケーションの要求性能の違いを利用して Power-Delay トレードオフを最適化できるプロセッサアーキテクチャ: *Power-Pro* を提案している [2]。Power-Pro アーキテクチャが備える機能を以下に示す。

- プロセッサは電圧制御命令を備えており、その命令を使って電圧制御レジスタに値をセットし、プロセッサの電源電圧とクロック周波数を制御することができる。
- プロセッサは CPU と同一の die にリングオシレータを搭載しており、クロック周波数の変化が回路遅延特性の変化に追従する機能を持つ。

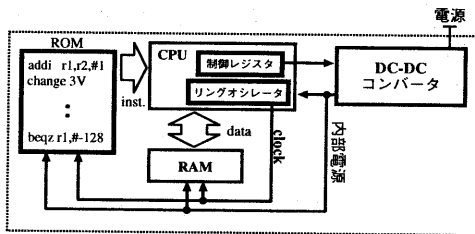


図 2: 電圧切り替え機能を持つアーキテクチャ

アーキテクチャの構成を図 2 に示す。低電力の DC-DC コンバータは必須の機能である。

本稿で述べる電力最適化手法は、Power-Pro アーキテクチャの電圧切り替え機能を前提としている。

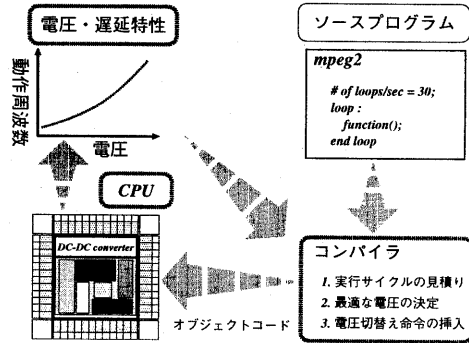


図 3: 目標とする最適化システム

2.3 実現しようとする電力最適化システム

我々が最終的に実現しようとするシステムを図 3 に示す。消費電力とパフォーマンスを最適化するフローは以下の通りである。

- プログラムは明示的に時間制約を記述したソースプログラムを用意し、システム設計者は以下の 3 つを提供する。
 - 可変電圧プロセッサ
 - プロセッサの電圧・遅延特性
 - コンパイラ
- コンパイラはプロセッサの電圧・遅延特性とプログラムの指定した時間制約から、本稿で述べる最適化法に基づいて最適な電圧を決定し電圧設定命令をオブジェクトコードとして出力する。
- プロセッサは電圧設定命令に指定された電圧でプログラムを実行する。

本システムによりプロセッサのプログラマビリティが向上し、プログラムを書き換えるだけで様々なアプリケーションの性能要求を必要最小限の電力で満足させることができる。本最適化システムは、VT-CMOS[4] などの技術に対しても同様の応用が可能である。

3 電圧割り当て最適化問題

3.1 アイデア

本稿で提案する最適化問題の基本概念は以下に示す作業を行うことによりプログラムの処理に必要なエネルギーを最小化することである。

- 与えられた複数の処理が1つのチップ上で逐次的に行われる時、時間制約が厳しい処理には高い電圧を割り当て、時間制約が緩い処理には低い電圧を割り当てる。
- 与えられた複数の処理がある制約時間内に実行される時、負荷(チップの負荷容量×平均稼働率)の重い処理には低い電圧を割り当て、負荷の軽い処理には高い電圧を割り当てる。

以下では、最適化の実現例を簡単な例を用いて説明する。説明を簡単に行うために、ターゲットとするシステムに以下の仮定を与える。

- プロセッサは以下に示す2種類の実行モードを自由に切替えて処理を行うことができる。以下の電源電圧とクロック周波数の関係は式(1)に従って決定した。式(1)のしきい値電圧 V_{th} は、ここでは $0.7[V]$ とした。
 - 電源電圧=1.8V, 動作周波数 = 25MHz.
 - 電源電圧=5.0V, 動作周波数 = 100MHz.
- DC-DC コンバータのエネルギー消費と電圧の切替えにかかる時間的オーバーヘッドは無視できるとする。
- 制約時間は 50 [sec].
- 表1に示す処理 Job-1, Job-2, Job-3 が与えられて、プロセッサで逐次的に実行される。 $Cycle_j$ と CL_j はそれぞれ、各 Job の実行サイクル数、負荷を表す。

表 1: 各プログラムに対する仮定

Jobs	$Cycle_j$ [cycle]	CL_j [nF]
Job-1	1×10^9	2.0
Job-2	10×10^9	0.8
Job-3	15×10^9	0.4

与えられた処理の実行方法には大別して以下の3種類が考えられる。

Case-1 電源電圧を常に $5[V]$ として動作させた時:
 $2.0 \times 25 + 8.0 \times 25 + 6.0 \times 25 = 400[J]$.

Case-2 電源電圧 V_{DD} を動的に変化させることができ、各プログラムの負荷の違いを考慮せずに電源電圧を割り当てた時:
 $2.0 \times 25 + 8.0 \times 25 + (3.2 \times 3.24 + 2.8 \times 25) = 330.368[J]$.

Case-3 電源電圧 V_{DD} を動的に変化させることができ、各プログラムの負荷の違いを考慮して電源電圧を割り当てた時:

$$2.0 \times 3.24 + (5.6 \times 3.24 + 2.4 \times 25) + 6 \times 25 = 234.624[J].$$

図4中の Case-1~3 は以上の Case-1~3 に対応する。

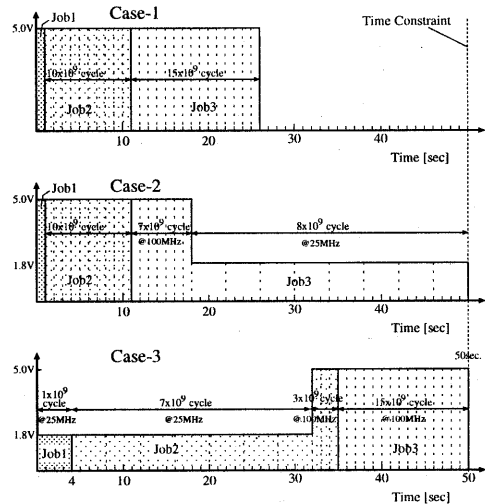


図 4: 時間制約の下での電圧割り当て問題

性能要求の違いだけでなく負荷の偏りを考慮して電圧の割当を行うことによりエネルギー消費を削減できることが分かる。

3.2 問題定義

本節では、電圧割り当て最適化問題の定義と定式化を行う。以下で説明する最適化問題は、十分に高速なプロセッサが用意されていて、電圧を下げてでもリアルタイム制約を満たすことができる時にどのように電圧を下げればエネルギー消費を最小にするかを決定する問題である。

問題の定式化に使う用語の定義を以下で行う。

- N プロセッサで処理される Job の数.
- L プロセッサが変更可能な電圧レベルの数. プロセッサは L 種類の電源電圧をダイナミックに使い分けることができる.
- $Step_j$ Job j の実行ステップ数. 1 ステップはプロセッサのクロックサイクルよりは十分大きい (例えば $1 \text{ step} = 10^6 \text{ cycle}$).
- CL_j Job j の負荷 (チップの総負荷容量 × Job の稼働率).

- T 時間制約. 与えられた Job は, 時刻 T 以前にすべての処理を終えなければいけない.
- v_i レベル i の電源電圧の値 ($1 \leq i \leq L$).
- f_i 電源電圧が v_i の時のクロック周波数.

$$f_i = \frac{k \cdot (v_i - V_{th})^{1.3}}{v_i} \quad k \text{ は定数}$$

- x_{ij} Job j が, 電圧レベル i で実行される実行ステップ数.

与えられた Job をすべて実行するのに必要なエネルギー E は, (4) で表すことができる. 時間制約は式 (5) である.

$$E = \sum_{j=1}^N \sum_{i=1}^L CL_j \cdot x_{ij} \cdot v_i^2 \quad (4)$$

$$0 \leq x_{ij} \leq Step_j, \quad \sum_{i=1}^L x_{ij} = Step_j$$

$$\sum_{j=1}^N \sum_{i=1}^L \frac{x_{ij}}{f_i} \leq T \quad (5)$$

電圧割り当て最適化問題は, 以下に示す問題として定義できる.

“ $N, L, CL_j, Step_j, T, v_i, f_i$ が与えられた時 E を最小にするような x_{ij} の集合を見つける問題.”

本章で述べた問題は整数線形計画法で解くことができる. 問題を解くのに必要な計算時間は $N \times L$ 変数の整数線形計画問題の解法時間に依存し, 実用的な N と L のサイズ (10~50 程度) に対して現実的な時間内で解くことができる. 整数線形計画問題の目標関数は式 (4), 制約条件は式 (5) である.

3.3 電圧割り当てに関する性質

本章で定義した電圧割り当て最適化手法は, プログラムの負荷に偏りが無い時には以下に示す性質がなりたつ.

性質 1 負荷に偏りのないプログラムが, ある電圧 v_{just} で処理され, ちょうど時間 $T (> 0)$ 以内にその処理を終えた場合, 電圧 v_{just} が与えられた処理に必要なエネルギーを最小にする唯一の電圧である. つまり複数の電圧を切替えて使ってもエネルギーは小さくならない.

証明 V_1, V_2, V_{th} ($0 < V_{th} < V_1 < V_2$) および $Cycle$ が与えられた時, $V_1 < v_{just} < V_2$ および $0 < x < Cycle$ を満たすあらゆる v_{just} と x に対して,

$$\frac{x \cdot V_1}{(V_1 - V_{th})^{1.3}} + \frac{(Cycle - x) \cdot V_2}{(V_2 - V_{th})^{1.3}}$$

$$= \frac{Cycle \cdot v_{just}}{(v_{just} - V_{th})^{1.3}} \quad (6)$$

の条件の下で,

$$V_1^2 \cdot x + V_2^2 \cdot (Cycle - x) > v_{just}^2 \cdot Cycle \quad (7)$$

となることを示す.

式 (6), (7) の左辺は, 電圧 V_1 で x サイクル, 電圧 V_2 で $(Cycle - x)$ サイクル実行した時の実行時間とエネルギー消費量を表し, 右辺はそれぞれ全サイクルを電圧 v_{just} で実行した時の実行時間とエネルギー消費量を表している.

$$T = \frac{x \cdot V_1}{(V_1 - V_{th})^{1.3}} + \frac{(Cycle - x) \cdot V_2}{(V_2 - V_{th})^{1.3}}$$

$$= \frac{Cycle \cdot v_{just}}{(v_{just} - V_{th})^{1.3}}$$

$$f(T) = V_1^2 \cdot x + V_2^2 \cdot (Cycle - x)$$

$$g(T) = v_{just}^2 \cdot Cycle \quad \text{とおくと.}$$

$$f(T) - g(T) < 0$$

$$\left(\frac{Cycle \cdot V_1}{(V_1 - V_{th})^{1.3}} < T < \frac{Cycle \cdot V_2}{(V_2 - V_{th})^{1.3}} \right) \quad (8)$$

となることを導くことができる.

$$\frac{df(T)}{dT} = \text{定数} < 0, \quad \frac{d^2g(T)}{dT^2} > 0$$

となることから, $f(T)$ および $g(T)$ と T の関係は図 5 に示す通りとなる.

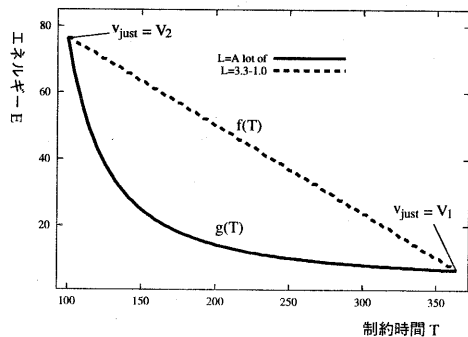


図 5: $V_{th} = 0.6, V_1 = 1.0, V_2 = 3.3$

証明終り

性質1は、プログラムの負荷に偏りが無い場合には、プログラムの実行中に電圧を変更しなくてもエネルギー消費を最小にする電圧の割り当てが見つかることを示す性質である。図6の例は、プログラムの負荷が一定であるため、例えば2.5[V]と5.0[V]の電圧を切替えて割り当てるより、常に4.0[V]で実行した方がエネルギー消費が小さいことを示している。

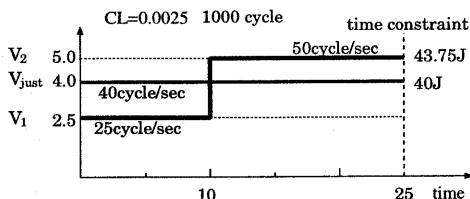


図6: 負荷に偏りが無い場合の電圧割り当て

性質1は、制約時間内に負荷の異なる複数のJobを処理する時には成立しない。Jobの負荷に偏りが大きい方が偏りが小さい時よりもエネルギー消費を小さくできる可能性があることを4.2節で述べる。

4 アプリケーションに関する考察

4.1 性能要求の違いを利用した電力削減

プロセッサに要求される性能はアプリケーションによって様々(図7)であり、アプリケーション毎に専用チップを設計すると設計にかかるコストや時間が膨大になる場合がある。設計資産を流用してTATを短縮するためには汎用のプロセッサを幅広い用途に応用する方法が考えられる。ところが、汎用プロセッサでは、性能要求を満足できないアプリケーションや消費電力が増大してしまうアプリケーションが存在する。アプリケーションの要求を満足し、かつなるべく多くの設計資産を様々なアプリケーションに流用するためには必要最小限の設計変更だけでシステムを実現することが重要である。

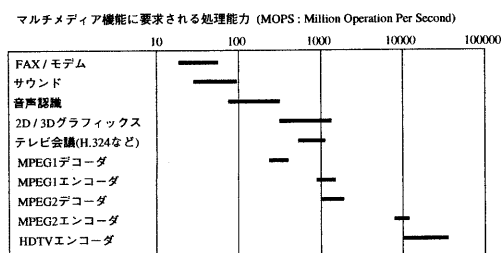


図7: メディア処理に要求される性能の違い

我々は本稿で述べた最適化手法を適用することによりアプリケーションの要求性能の違いを吸収して、処理に必要な電力を削減するという応用を検討している。同様に、複数のアプリケーションを一つのシステム上で逐次的に実行しようとする場合にもアプリケーション起動時に最適な電圧を選択することにより大幅に電力を削減することができる。図8では、制約時間内に処理を終え、かつエネルギー消費がもっとも小さくなるように電圧を選択した場合のエネルギー消費を、選択可能な電圧の種類を変えて比較している。図8に示した例ではプログラムの実行中には電圧を変更せずに起動時に選択した唯一の電圧でプログラムを実行することを仮定している。

3.3V から1.0V までの任意の電圧を選択できる場合は、時間制約を2倍にすると処理に必要なエネルギーを1/3にできる。また、この場合3.3Vと1.0Vの2種類しか選択できない場合と比較して、エネルギー消費が最大1/10になる場合がある。

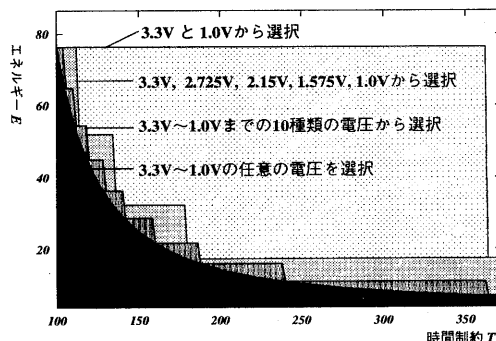


図8: 電力最適化による電力削減効果

図8の結果から、様々な性能要求を持つ複数のアプリケーションを一つのシステムで処理しようとする場合、あるいはプロセッサに汎用性を持たせても電力を増加させたくない場合には図9に示すような回路構成が適切であると言える。図9の構成では、チップ

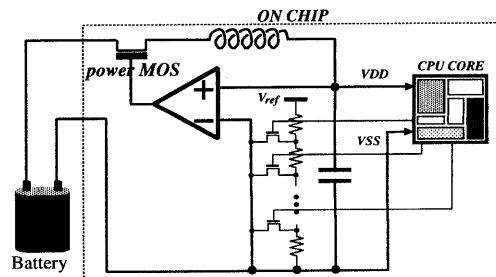


図9: 電力最適化を実現する回路構成(1)

ブ内に設けた MOS スイッチを CPU で制御することにより多くの (10 種類以上の) 参照電圧の中からアプリケーションに最適な唯一の電圧を選択し、電力削減を可能にする。電圧の切替えにかかる時間的オーバーヘッドは大きい (m[sec] のオーダー) がアプリケーションの起動時にのみ電圧を切替える場合にはプログラムが連続して実行される時間と比較して無視できる時間である。

4.2 負荷の偏りを利用した電力削減

本節では、アプリケーションの起動時に電圧を切替えるだけでなくプログラムの実行中に負荷に応じて電圧を割り当てるという応用について述べる。負荷に応じて電圧を割り当てる時の基本的なコンセプトは、

“負荷 (チップの負荷容量 × 平均稼働率) の重いプログラムには低い電圧を割り当て、負荷の軽いプログラムには高い電圧を割り当てる” である。

プログラム実行時の負荷に偏りがある場合には、プログラムの実行中に電圧を切替えて最適化することにより電力が削減できる例を表 2 に示す 4 種類のケースで説明する。

表 2: Job の実行サイクル数と平均負荷

cases	$Cycle_j(\text{job1,job2})$	$CL_j(\text{job1,job2})[\text{nF}]$
case-1	$(1.0, 1.0) \times 10^9$	(10, 10)
case-2	$(1.0, 1.0) \times 10^9$	(4, 16)
case-3	$(1.0, 1.0) \times 10^9$	(2, 18)
case-4	$(1.0, 1.0) \times 10^9$	(1, 19)

表 2 の例では、負荷の偏り方が異なる 4 種類のケースを想定している。実行サイクル数と負荷の総和はすべてのケースで同じである。実験は、3.3V~1.0V の範囲内の任意の 2 つの電圧を起動時に選択し、2 種類の電圧をプログラムの実行中に切替えて最適化することを想定して行った。したがって、プロセッサは任意の種類電圧を選択できるがプログラムの実行中に切替えられる電圧は 2 種類である。

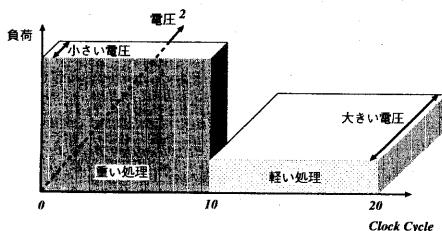


図 10: 負荷の偏りを考慮した電圧割り当て

プログラムの実行に必要なエネルギー消費量の最適化は、実行時間の制約を満たして図 10 に示した立方体の総体積を最小化することにより実現される。

表 2 のそれぞれのケースに対して時間制約を変化させた時のエネルギー消費の変化を図 11 に示した。

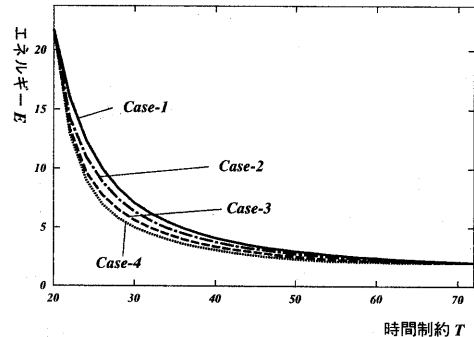


図 11: 負荷の偏りと電力削減効果の関係

図 11 の結果は、電圧の割当てを最適化することにより負荷の偏りが大きい方が偏りが小さい時に比べてエネルギー消費を小さくできることを示している。Case-1 は、負荷が一樣である時に、電力を最小にする唯一の電圧でプログラムを実行した時のグラフ (図 8 に示したグラフ) と全く同じになる。負荷の偏りがもっとも大きい Case-4 の電力削減率を図 12 に示した。Case-4 の電力削減率は式 (9) により算出した。

電力削減率 (%) =

$$\frac{\text{Case-1 の電力量} - \text{Case-4 の電力量}}{\text{Case-1 の電力量}} \times 100 (\%) \quad (9)$$

Case-4 では最大 30% の電力削減が見込まれることが確認できる。同様に Case-3 では最大 22%, Case-2 では最大 11% 程度の電力削減が期待できることを確認した。本手法は負荷の偏りが静的に予測できるアプリケーションに対しては効果的である。

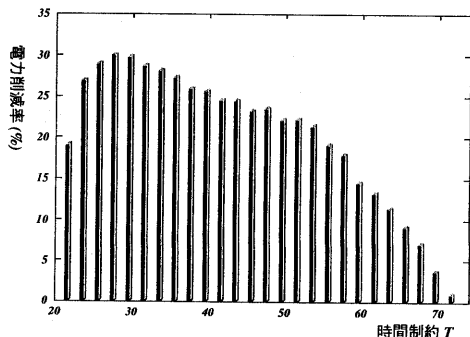


図 12: Case-4 の制約時間と電力削減率の関係

本節で述べた電圧割り当ては、起動時に一定範囲内の任意の2種類の電圧を選択し、さらにその2種類の電圧をプログラム実行中に切替えるという方法により実現される。起動時に電圧を切替える場合には電圧切替の時間的オーバーヘッドはそれほど重要な問題にはならないが、プログラムの実行中に電圧を切替える場合には非常に高速な電圧切替が必要である。筆者らはただか2~3種類程度の電圧を高速に切替える場合には、あらかじめ安定化された電圧の中からMOSスイッチを使って最適な電圧を選択するという方法を考えている(図13)。

アナログスイッチ(74HC4066)と8bitのプロセッサを使って実際に図13に示す回路を構成し電圧を切替える実験を行ったところ、電圧の切替え時間は50n~100[nsec], アナログスイッチによる電圧降下は消費電流が100[mA]の時におよそ0.1[V]であった。さらに10[nF]程度のバイパスコンデンサで、電圧切り換えのノイズを吸収でき、8bitプロセッサ上で簡単なプログラム(最大公約数を求めるプログラム)が20[MHz]で正常に動作することを確認した。

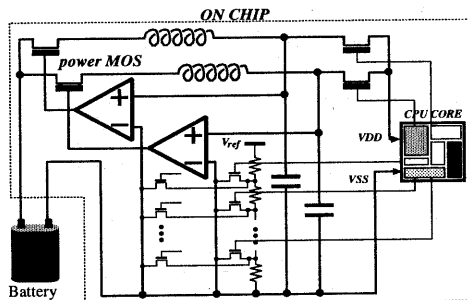


図 13: 電力最適化を実現する回路構成(2)

5 おわりに

本稿では、アプリケーションプログラムの要求する性能および実行中の負荷の違いを利用してプログラムの実行に必要なエネルギー消費量を最適化する手法を提案した。電圧の選択方法には大きく分類して以下の4種類が考えられ、これらの中から用途に応じて回路構成を決定するのが最良であると思われる。

1. 電圧は常に一定である(既存システム)。
2. アプリケーションの起動時に多数(3種類以上)の電圧の中から1種類の電圧を選択しその電圧でプログラムを実行する。

3. アプリケーションの起動時に多数の電圧の中から2~3種類の電圧を選択し、プログラム実行中にさらに電圧を切替える。
4. アプリケーションプログラムの実行中に多数(10種類以上)の電圧の中から1種類の電圧を選択し電圧を切替える。

特定用途のLSIには、仕様設計時に電圧を最適化することにより1.の構成で低電力化できるが、汎用チップあるいは多機能チップとして使用する場合には電力が増大する可能性がある。2.の構成では、アプリケーションの性能要求によっては1.の回路構成と比べて電力を最大1/10にできることを示した。3.の構成ではプログラムの負荷の偏りが大きい場合には2.の構成と比較してさらに30%の電力削減効果が期待できることを示した。4.の構成は、電圧切替が非常に高速かつ低電力、低コストの電源が必要になる。複数の電流源をオンチップ化することは现阶段では実用的でないといえる。

今後は電源をオンチップ化した時の問題と、プロセッサの電圧を変更することでアプリケーションプログラムのリアルタイム性を保証するコンパイラ技術を検討し、アナログ回路とデジタル回路およびコンパイラを含めたシステムを実現する予定である。

参考文献

- [1] 富山宏之, 井上昭彦, 清水友人, 神原弘之, 安浦寛人. “ハードウェア/ソフトウェア・コデザインのためのビット数指定言語 Valen-C とその処理系”. 信学技報 DA96-82-7, 1996 12月.
- [2] 石原亨, 甲斐康司, 安浦寛人. “マイクロプロセッサにおけるアーキテクチャレベルの低消費電力化手法”. 信学技報 VLD96-72, 1996 12月.
- [3] 菅野 卓雄 監修・飯塚 哲哉 編. CMOS超LSIの設計. 培風館, 1996.
- [4] T. Kuroda and T. Sakurai. “Threshold-voltage control scheme through substrate-bias for low-power highspeed CMOS LSI design”. *Kluwer J. of VLSI signal processing, special issues on technologies for wireless computeiog*, 1996.
- [5] A. J. Stratakos and S. R. Sanders R. W. Brodersen. “High-Efficiency Low-Voltage DC-DC converter for Portable Applications”. In *Proc. of IWLPD*, pages 105-110, Apr. 1994.
- [6] Shiro Sakiyama et al. “A lean power management technique : The lowest power consumption for the given operating speed of LSIs”. In *Proc. of Symposium VLSI Circuits*, pages 99-100, 1997.
- [7] Jui-Ming Chang and Massoud Pedram. “Energy Minimization Using Multiple Supply Voltage”. In *Proc. of International Symposium on Low Power Electronics and Design*, pages 157-162, 1996.