

## 低消費エネルギー化を目的とする可変電源電圧プロセッサを用いたITRONによる電圧制御アルゴリズム

大隈, 孝憲  
九州大学大学院システム情報科学府情報工学専攻

門前, 淳  
九州大学大学院システム情報科学府情報工学専攻

安浦, 寛人  
九州大学大学院システム情報科学府情報工学専攻

<https://hdl.handle.net/2324/7644>

---

出版情報：情報処理学会研究報告．SLDM, [システムLSI設計技術]．2000 (79), pp.9-15, 2000-09-07.  
情報処理学会  
バージョン：  
権利関係：

# 低消費エネルギー化を目的とする可変電源電圧プロセッサを用いた ITRON による電圧制御アルゴリズム

大隈 孝憲      門前 淳      安浦 寛人

九州大学 大学院 システム情報科学府 情報工学専攻

〒 816-8580 福岡県春日市春日公園 6-1

E-mail: {okuma, m\_atsusi, yasuura}@c.csce.kyushu-u.ac.jp

あらし:

動的に電源電圧を変化させ、その性能と消費電力を制御できる可変電源電圧プロセッサをリアルタイムシステムに用いる場合、OS によってタスク毎に電源電圧を制御することで低消費エネルギー化が図れる。電源電圧を下げると消費エネルギーを下げるができるが、電圧低下に伴ってプロセッサの動作周波数も低減するため、タスクの実行時間は長くなる。そのため、電源電圧を下げる時はリアルタイム制約を損なわないよう行う必要がある。

本稿では、タスクスケジューリングの際に、次に実行されるタスクを決定するだけでなく、そのタスクを実行するプロセッサの電圧も決定するアルゴリズムを提案する。このとき、すべてのタスクが時間制約を満たすという条件下で、システムの消費エネルギーを最小化する電圧が選ばれる。さらに、提案するアルゴリズムは、ITRON 仕様の OS に実装可能なアルゴリズムである。

キーワード: 低消費電力設計, 可変電源電圧プロセッサ, 実時間処理, タスクスケジューリング,  $\mu$ ITRON

## Voltage Scheduling Applied to the $\mu$ ITRON Operating System for a Variable Voltage Processor

Takanori OKUMA      Atsushi MONZEN      Hiroto YASUURA

Department of Computer Science and Communication Engineering, Kyushu University

6-1 Kasuga-koen, Kasuga, 816-8580, Japan

E-mail: {okuma, m\_atsusi, yasuura}@c.csce.kyushu-u.ac.jp

### Abstract:

This paper presents a voltage scheduling applied to the  $\mu$ ITRON OS for a variable voltage processor which can vary its supply voltage dynamically. On the variable voltage processor, task processing with lower supply voltage is able to reduce power consumption dramatically. But, lower voltage may violate time constraints of the real-time process. In this paper, we propose technique by which CPU time and supply voltage are simultaneously assigned each task to minimize power consumption. It is possible that our technique is implemented in  $\mu$ ITRON OS.

**Key Words:** Low Power Design, Variable Voltage Processor, Real-Time Scheduling,  $\mu$ ITRON OS

# 1 はじめに

組み込みシステムの多くは実時間処理を伴い、リアルタイム性への要求が厳しく高速化が望まれる。またその一方で、軽量のエネルギー源で多様な処理を長時間実行するシステムの実現に対する要求も高まっている。システムにリアルタイム性を持たせるためには、高い速度性能を持つ汎用のプロセッサを利用することが考えられるが、多くの場合、低消費エネルギー化に対する要求が満足されない。なぜなら、一般的にプロセッサのエネルギー消費とプロセッサの動作周波数との間にはトレードオフの関係があるためである。つまり、高速化と低エネルギー化を同時に満たすことは困難である。この問題を解決するために、文献 [?] において、電源電圧を動的に変更できる可変電源電圧プロセッサが提案されている。可変電源電圧プロセッサは以下の性質を持つ。

- プロセッサは電圧制御命令を持ち、アプリケーションやオペレーティングシステムがこの命令を使うことにより、プロセッサの電源電圧とクロック周波数を任意の実行ステップで変更できる。
- プロセッサは電源電圧の変更による回路遅延の変化に合わせたクロック周波数を発生する。

可変電源電圧プロセッサにより、制約時間が厳しく、短い時間で処理しなければならないタスクには、高い電圧で高速に処理を行ない、制約時間が緩く、高い処理能力を必要としないタスクには、低い電圧で処理し、消費エネルギーを抑えることができる。一般に消費エネルギーは電源電圧の 2 乗に比例する [?, ?] ので、低い電圧で処理を行なうことはエネルギー削減に大きく貢献できる。図??に例を示す。この例において電圧、周波数、エネルギーの関係を図中の表のように仮定する。この仮定の下で実行サイクル数が 1000M サイクルのタスクを制約時間 25 秒以内に完了しなければならないとき、5.0V で実行した場合と 4.0V で実行した場合とでは、両方とも制約時間を満たしているが、4.0V で実行した方がエネルギー消費が少なくなる。

電源電圧を下げると消費エネルギーを下げるができるが、電圧低下に伴ってプロセッサの動作周波数も低減するため、タスクの実行時間は長くなる。そのため、電源電圧を下げる時はリアルタイム制約を損なわないよう行う必要がある。プロセッサの電源電圧の制御はアプリケーションやオペレーティングシステムによって行なわれる。よって、アプリケーションが

Supply Voltage	4.0V	5.0V
Clock Frequency	40MHz	50MHz
Energy/Cycle	25nJ	40nJ

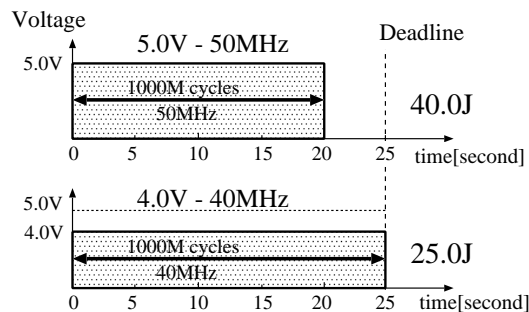


図 1: 電力削減の例

リアルタイム制約を満たすように電源電圧の制御を行なうためのコンパイラ技術や、オペレーティングシステム技術の確立が重要である。

文献 [?] において、筆者らは可変電源電圧プロセッサを用いた場合のオペレーティングシステムにおけるタスクスケジューリングのアルゴリズムを提案した。このアルゴリズムはタスクスケジューリングの際に、次に実行されるタスクを決定するだけでなく、そのタスクを実行するプロセッサの電圧も決定する。このとき、すべてのタスクが時間制約を満たすという条件下で、システムの消費エネルギーを最小化する電圧が選ばれる。電圧を動的に決定するアルゴリズムとして、あらかじめ到着時刻が分かっているタスクセットに対しては SD(Static order and Dynamic voltage allocation) アルゴリズムと、到着時刻があらかじめ分かっていないタスクセットに対しては DD(Dynamic order and Dynamic voltage allocation) アルゴリズムを提案した。

上記アルゴリズムは、電圧を低減させずに実行させた場合にリアルタイム制約が満たされているならば、電圧を低減させてもリアルタイム制約が満たされることを保証している。しかし、まだ実際の OS に対しての実装は行なわれていない。実際の OS として ITRON 仕様の OS を選択し、上記アルゴリズムを実装しようとした場合、アルゴリズムに対する仮定にいくつか問題があり、実装するのは困難であった。そこで本稿では、それらの問題点を明らかにし、ITRON 仕様の OS に実装し易いように改良した電圧制御アルゴリズムを

提案する。

本稿の構成は以下の通りである。第2章で以前提案した動的な電圧決定アルゴリズムについて簡単に述べる。第3章では、そのアルゴリズムを ITRON 仕様の OS に実装する上での問題点を挙げ、その解決策を議論する。第4章では、ITRON 仕様の OS に実装可能な電圧制御アルゴリズムを提案し、第5章で、その考察を行なう。最後に第6章でまとめと今後の課題を述べる。

## 2 動的な電圧決定アルゴリズム

本章では、文献 [?] で提案した、動的な電圧決定アルゴリズムについて説明する。このアルゴリズムは、対象とするシステムに対して SD アルゴリズムと DD アルゴリズムに別れるが、基本的な考え方は同じである。タスクの実行が、あるタスクから別のタスクへと実行が切り替わる際に、今から実行されるタスクの電圧を決定し、電圧をその値に変更する。その際、タスクの総実行時間は最悪のケースを考えて電圧を決定している。しかし実際にそのタスクを実行させた場合、予想終了時刻よりも早く終了する可能性が高いため、その分時間的な余裕(余裕時間)が生じる。今まで実行されていたタスクも、最悪ケースの実行時間を考慮に入れて決定された電圧で実行されていたが、上記の理由で余裕時間が生じるため、この余裕時間を次に実行されるタスクの実行時間として利用することで、そのタスクを実行させる電圧の値の低減を図っている。もちろんこのときも、今から実行されるタスクの実行時間は最悪ケースであることを考慮に入れて電圧が決定される。そうすれば、また余裕時間が生じ、タスクの実行が切り替わるごとに電圧の低減を図ることが可能となる。さらに、SD アルゴリズムでは、すべてのタスクの到着時刻があらかじめ分かっているため、今から実行されるタスクよりも後で実行されるタスクの予定終了時刻がそのデッドライン時刻と比較して余裕があれば、その時間も先に利用して電圧の低減を図っている。

### 2.1 アルゴリズムの構成

電圧決定アルゴリズムでは、以下の3つのステップで電圧を割り当てるタスクの開始時刻と終了予定時刻を求め、その時間内に終了するような電圧をタスクに割り当てている。

#### Step1: CPU Time Allocation

すべてのタスクを最大電圧で実行すると仮定して CPU 時間を割り当てる。この時、各タスクの実行サイクル数は最悪ケースで考える。このときの決定された各タスクの start time, end time を保持しておき、後のステップで利用する。また、各タスクに対してデッドライン時刻と end time との差分も保持しておく。これは SD アルゴリズムにのみ利用される。

#### Step2: End Time Prediction

次に実行されるタスクの終了予定時刻を決定する。DD アルゴリズムの場合、Step1 で決定されたこのタスクの end time が終了予定時刻である。さらに、SD アルゴリズムの場合は、後に実行されるタスクに対して、Step1 で求めたデッドライン時刻と end time との差分の最小値を終了予定時刻に加算する。

#### Step3: Start Time Assignment

次に実行されるタスクの開始時刻は、前に実行されたタスクの終了時刻によって変動する。タスクの実際の実行サイクル数は入力データによってバラツキがあり、最悪の場合より早く終了することの方が多い。つまり前に実行されたタスクが早く終了すればするほど、次に実行するタスクの開始時刻を早めることが可能である。

## 3 ITRON に実装する上での問題点

本章では、上記アルゴリズムを ITRON 仕様の OS に実装する上で、問題となる点をあげ、その解決策を述べる。

### 3.1 待ちの問題

上記アルゴリズムにおいて、タスクの順序を決定するアルゴリズムとして、Earliest Deadline First(EDF)[?, ?] が採用されている。EDF とはタスクをデッドラインの早い順に実行するアルゴリズムである。つまり、デッドラインの早いタスクが優先度の高いタスクとなる。現在実行されているタスクよりも高い優先度を持つタスクが実行可能になった場合、その実行はより高