

論文 / 著書情報
Article / Book Information

題目(和文)	マイクロプロセッサにおける動作時消費電力低減方式の研究
Title(English)	
著者(和文)	十山圭介
Author(English)	Keisuke Toyama
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:乙第4071号, 授与年月日:2012年6月30日, 学位の種別:論文博士, 審査員:前島 英雄
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:乙第4071号, Conferred date:2012/6/30, Degree Type:Thesis doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

東京工業大学
学位論文

マイクロプロセッサにおける
動作時消費電力低減方式の研究

十山 圭介

平成 24 年 6 月

内容梗概

1990年代後半の組込み機器に向けた高性能マイクロプロセッサの開発によって、ハンドヘルドコンピュータや携帯電話など様々な製品へデジタル民生機器が広まってきた。今後、携帯電話やデジタル家電、車載機器などさらに多くの情報機器へマイクロプロセッサが組み込まれ、それぞれがネットワークで接続されて応用されるようになる。多機能で高度なデジタル化が進み、そこで使われるマイクロプロセッサに高性能化と低消費電力化の要求が高まってくる。これらの機器に使用されるマイクロプロセッサシステムを設計・開発するにあたっては、性能・価格・消費電力のバランスをうまくとることが重要で、低消費電力性と高性能性を両立させることが不可欠であるが、消費電力と性能はトレードオフの関係にあって同時にその要求を満たすことは容易ではない。システム全体として消費電力を低減するために、デバイスや回路、アーキテクチャ、ソフトウェアといった多くの技術分野にわたって密接に連携して対策を進めることが必要である。

本研究は、そのような連携の中でソフトウェアからのアプローチによってハードウェアが備えている電力制御機構を適切に動作させて、消費電力（消費電力量）を低減する方式について考察することを目的としている。

このため、プロセッサの動作周波数（ F ）と電源電圧（ V ）を協調して低減させる周波数－電圧協調制御（Frequency-Voltage Cooperative Control : FV 制御）方式を提案した。 F と V の 2 乗とが電力低下に寄与するため、プログラム動作時のプロセッサ消費電力量を大幅に削減できる。本研究では、あらかじめ立てた実行スケジュールと実際の実行時間との差分をフィードバックさせて動作周波数を設定するアルゴリズムを用いたフィードバック型周波数－電圧協調制御（FFV 制御）方式を実装し、その

効果を考察した。

まず、FV 制御の原理を述べ、これによる消費電力量の削減効果を検討・評価して妥当な動作周波数および電源電圧を決定するルールを与えた。理想的な消費電力量の値に対する削減度の劣化を解析的に評価することにより、FV 制御システムを実装するために用意すべき動作周波数群を決定した。適用する LSI チップが動作する最小の電源電圧を考慮して評価し、プロセッサの最高動作周波数からその 1/2、さらにその 1/2、… となる系列で周波数を採択して行くことが有効であることを確認した。

次に、この FV 制御ソフトウェアを実装した。制御を行う処理単位としてプログラムコードを分割した区間（セグメント）を用い、携帯電話のリファレンスプラットフォームとなる実機上で実装し、評価・検証した。

実装では、フィードバック型の FV (FFV) 制御として動作周波数（と電源電圧）を決定し、制御対象のプログラムに反映させている。この方式は、最悪の時間で実行されるとしてあらかじめ立てたスケジュールと実際の処理時間との差分をセグメントごとに採って処理負荷を見積もり、その負荷に応じて動作周波数を変更するものである。これにより MPEG-4 と MP3 デコーダ両者が同期して並行動作するプログラムで、この動画・音声再生アプリケーション全体の消費電力量が 28% 程度まで低減できることを実機で確認した。

今後、組込みシステムにおいてもマルチプロセッサが潮流となると考えられる。マルチプロセッサでは並列処理による低消費電力化が図られているが、さらに FFV 制御を適用し、負荷に応じて要素プロセッサごとに要求される動作周波数を低下させてアプリケーション実行に最適な動作周波数と電源電圧を設定することにより、動作時の消費電力量を一層低減できると期待できる。

そこで、FFV 制御方式を並列処理の要素プロセッサで実行されるスレッドに適用する。要素プロセッサそれぞれが、実行履歴から見積もった処理負荷によって、動作周波数と電源電圧を制御するマルチコア FFV 制御方式を提案した。この方式は画像や音声のような周期的でデータ依存性の少ないメディア処理に適しており、プロセッサ台数の増加や高負荷に対して有効であること、事前にプロファイルなどを取得する必要のないことが利点である。

シミュレーション結果によれば、プロセッサ台数ならびに各プロセッサ内で処理負荷の変動に応じて制御することで、シングルプロセッサでの実行と比較して 70% 以上の消費電力量低減効果を得ることができる。また、負荷の見積もりとしてプログラムの実行履歴を用いることの妥当性を示した。セグメントごとに最適な動作周波数と電源電圧を設定する FFV 方式の消費電力制御は、低消費電力のスリープや電源遮断の動作モードを併用することにより、オンチップ・マルチプロセッサでのマルチメディアアプリケーションに対して、高い省電力効果を発揮できると考えられる。

以上のように、本研究では、実行時の負荷状況に応じて動作周波数と電源電圧を協調して低下させ、マイクロプロセッサの動作時の消費電力量を低減するソフトウェア制御方式を提案し、シングルプロセッサおよびマルチプロセッサに対して実装と評価を行った。

目次

内容梗概	i
図一覧	vi
表一覧	ix
第1章 序論	1
1.1 低消費電力プロセッサシステムの背景	2
1.2 省電力化へのアプローチ	5
1.3 従来技法	7
1.4 本研究における低消費電力化技術の特徴と位置づけ	9
第2章 FV 制御によるプロセッサ動作時消費電力量低減方式の考察	17
2.1 はじめに	18
2.2 周波数－電圧関係による消費電力量の表現	20
2.3 離散 FV 制御での削減損失の評価	23
2.4 電力損失の評価	25
2.5 設計ルール	29
2.6 実プロセッサへの適用	31
2.7 まとめ	32
第2章の付録 基礎定理の証明	34
第3章 周波数－電圧協調型電力制御方式のフィードバック予測による適用	43
3.1 はじめに	44

3.2	FV 制御のシステム適用	
	–ソフトウェアによる周波数–電圧制御の解法	45
3.3	実験結果	52
3.4	Cool-Start 方式による低電力効果の向上	56
3.5	まとめ	56
	第 3 章の付録 FFV 制御評価用組込みシステムと評価パラメータ	59
第 4 章	マルチプロセッサにおける動作時消費電力量の低減方式	69
4.1	はじめに	70
4.2	マルチプロセッサアーキテクチャと動作時の消費電力制御	72
4.3	マルチプロセッサにおける動作時のプロセッサ消費電力制御方式	76
4.4	評価	89
4.5	マルチプロセッサにおける DVFS 制御の関連研究	108
4.6	まとめ	109
第 5 章	結論	119
	謝辞	125
	本研究に関する論文・発表文献	127
	本研究に関するその他の論文・発表文献	129

図一覽

1.1	マイクロプロセッサにおけるアプリケーションの展開（[9]による）	3
1.2	プロセッサの動作周波数と消費電力の動向	4
1.3	消費電力低減技術の分類	8
2.1	MPEG-4 デコーダにおける負荷変動の例	18
2.2	動画再生における負荷と動作モード例	19
2.3	F - P 関係のモデル	22
2.4	周波数離散化の影響	24
2.5	F_m の変化を考慮した電力損失評価	30
2.6	周波数離散化による電力量増加	35
2.7	三つ以上の周波数を組合せて使用する場合の状況	37
3.1	MPEG-4 デコーダの処理プロセス	47
3.2	MPEG-4 デコード動作における FV 制御	49
3.3	システム構成	53
3.4	FFV 制御実行時の電圧・電流波形	55
3.5	実験システムのタスク構成	61
3.6	可変電源電圧回路	62
3.7	評価用システムの外観	63
3.8	SH-Mobile 1 の Shmoo プロット（[12]による）	64
4.1	マルチプロセッサでの消費電力量低減方式の概要	71
4.2	マルチプロセッサ構成	72

4.3	VMU による動作周波数・電源電圧の制御	74
4.4	動作周波数 - 電源電圧の関係	75
4.5	並列処理による動作時消費電力量の低減	79
4.6	並列実行時の時間スロットに対する実行プログラムの割当て	80
4.7	シングルプロセッサでの FFV 制御による処理プロセス	81
4.8	独立モデルの動作イメージ	84
4.9	同期ポイントによるプログラムの領域分割	85
4.10	チェックポイントでの動作周波数決定処理	87
4.11	スレーブプロセッサでの FFV 制御による処理の様子	88
4.12	仮定した処理負荷モデル	90
4.13	処理負荷を考慮した並列制御による消費電力量の低減	93
4.14	一斉モデルでプロセッサの負荷に偏りがある場合の消費電力量低減 効果	95
4.15	一斉モデルと独立モデルでの消費電力量低減効果の比較	96
4.16	オーバーヘッドも考慮したマルチコア FFV 制御による消費電力量の 低減効果	97
4.17	評価用動画データ負荷変動 (600 フレーム分)	99
4.18	MPEG のデータ構造とプロセッサへのスライスの割当て	100
4.19	プロセッサ台数によるマルチコア FFV 制御の消費電力量低減効果	101
4.20	処理負荷の増加による消費電力量低減効果	102
4.21	チェックポイントの個数による消費電力量低減効果の比較	103
4.22	プロセッサ数の動的な変更による消費電力量の最小化	107

表一覧

2.1	電力損失の具体値	28
2.2	Crusoe への適用例	32
3.1	シミュレーションによる電力評価	51
3.2	トレースデータによる電力評価	52
3.3	フレーム内での <i>FV</i> 変更回数	54
3.4	評価用システムのパラメータ	60
4.1	動作周波数・電源電圧の組合せ	76
4.2	FFV 制御を使用した動作周波数・電源電圧の組合せ（初期設計）	91
4.3	並列処理オーバヘッドを考慮した消費電力量低減効果の変動	94
4.4	FFV 制御を使用した周波数・電源電圧の組合せ（実プロセッサ）	98
4.5	コア当り 1 フレーム処理での動作周波数・電源電圧の変更回数	104
4.6	デッドライン制約を超えたセグメント/フレームの出現率	105

第 1 章

序論

1.1 低消費電力プロセッサシステムの背景

1.1.1 組込みシステムの動向

1990 年代初頭から 8 ビットの制御用マイクロプロセッサを搭載した電子手帳や家庭用 FAX などが登場し、次いで高性能な組込み機器に向けたマイクロプロセッサの開発によって、ハンドヘルドコンピュータや家庭用ゲーム機、携帯電話、DVD など様々な製品へとデジタル民生機器が広まってきた [3–5]。今後、情報通信社会の発展により携帯電話やデジタル家電、車載機器などさらに多くの情報機器へマイクロプロセッサが組み込まれ、それぞれがネットワークで接続されて多方面に応用されるようになる [6, 7]。

特に、携帯電話や PDA (Personal Digital Assistant) に代表されるモバイル情報端末の進歩は顕著であり、携帯電話では重量やサイズが小さくなっているにもかかわらず音楽や動画再生が行われ、さらにスマートフォンとして PDA の機能を包含し、パーソナルコンピュータ (PC) とほぼ同等にウェブサイトが閲覧できたり、電子決済サービスが実現されたりして複雑な処理を行うようになってきている。組込み用マイクロプロセッサに用意されるアプリケーションの品揃えは図 1.1 のように広がって行き、高精細な三次元動画やリアルタイムの音声認識といった現在は PC で実現されているような高度なメディア処理の実現も目前である [4, 9, 10]。

処理が高度になれば、それに応じて処理を行うエンジンであるマイクロプロセッサの消費電力も増大する。そこで、次世代のモバイル情報端末をはじめとする組込み機器への搭載を目指して、高度なメディア処理を高速かつ低消費電力で実行できるデバイスやプロセッサ、システムが提案され検討されている [1, 2, 4–6]。

このような次世代の組込みシステムに要求される要素は次の三つである。

- **低価格**：民生用であるために様々なコストを抑えられることが重要である。このためには、既存のソフトウェアなどの資産を活用できる互換性があることも重要である。

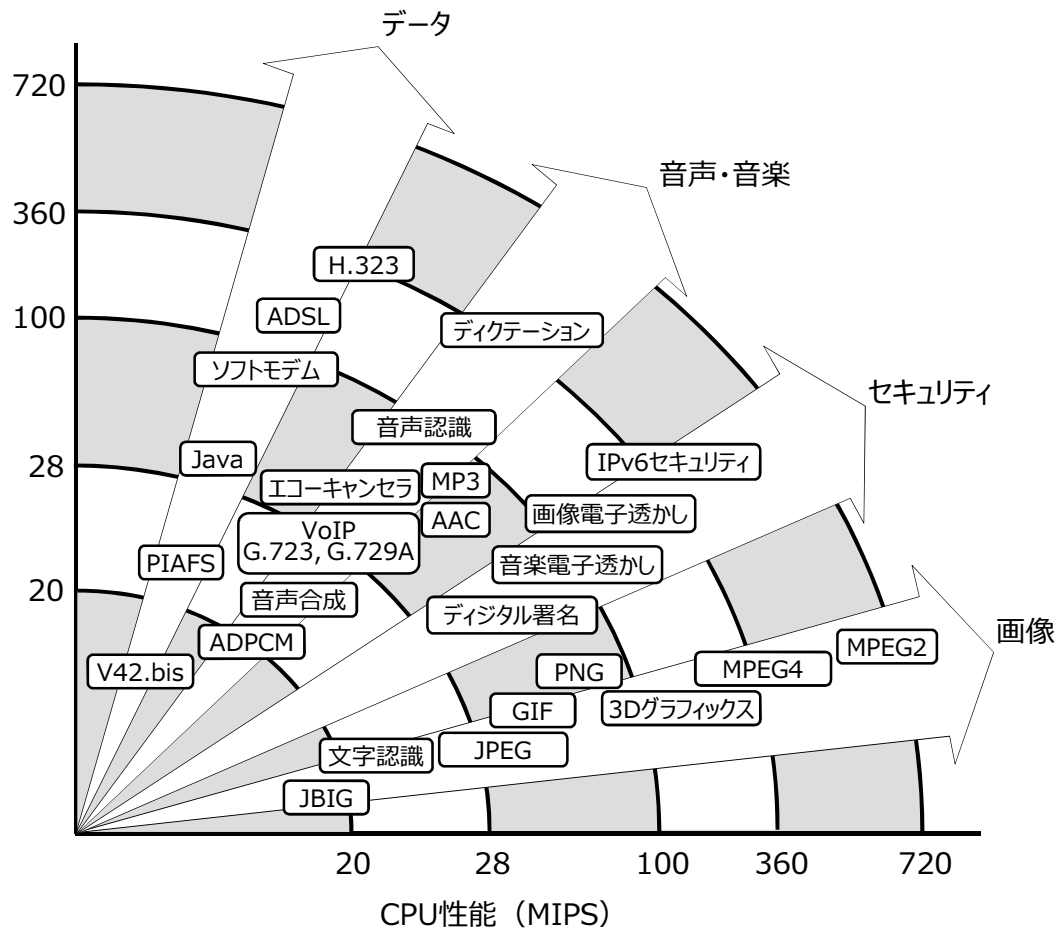


図 1.1 マイクロプロセッサにおけるアプリケーションの展開 ([9] による)

- 高性能：今後の組込みシステム上でのアプリケーションとして、グラフィックスやサウンドなど大量のマルチメディア処理、また音声認識や自動翻訳といった高度なメディア処理が挙げられる。これらをリアルタイムで行えるような高い処理性能が要求される。
- 低電力：携帯電話をはじめとする組込みシステムにおいては、コンパクトなサイズでありながら高機能な処理を行うことが必要である。消費電力が大きいとバッテリーの持続時間が短くなるという問題が発生する。また、消費電力が大きいと発熱のために実装コストが上がり、価格が上がるという問題も出てくる。携帯電話用のマイクロプロセッサでは動作時に 100~200mW 程度の消費電力

とする技術が求められている [8].

現在の組込みシステムの主要な適用先は、モバイル情報端末などの情報通信機器やDTVなどのデジタル家電に対するものであるが、今後ユビキタスコンピューティングの時代となり、様々な機器にマイクロプロセッサが多数存在するようになる。図1.2に示すようにプロセッサは、組込み向けや、ワークステーション (WS)・パーソナルコンピュータ (PC) 向けの用途に応じて分類されるが、それぞれ低消費電力性と高性能性が追求されて行き、従来の制御用マイコンやPC、サーバでは実現が困難な高性能・低電力・低価格を同時に満たす携帯電話などをターゲットとする組込み向けのマイクロプロセッサ (図中矢印の方向) が強く望まれている [4, 8]。そこでは、ソフトウェア資産の互換性を保ちながら性能を向上させ、それと背反して環境に優しく低消費電力を実現する設計がさらに重要になってくる。

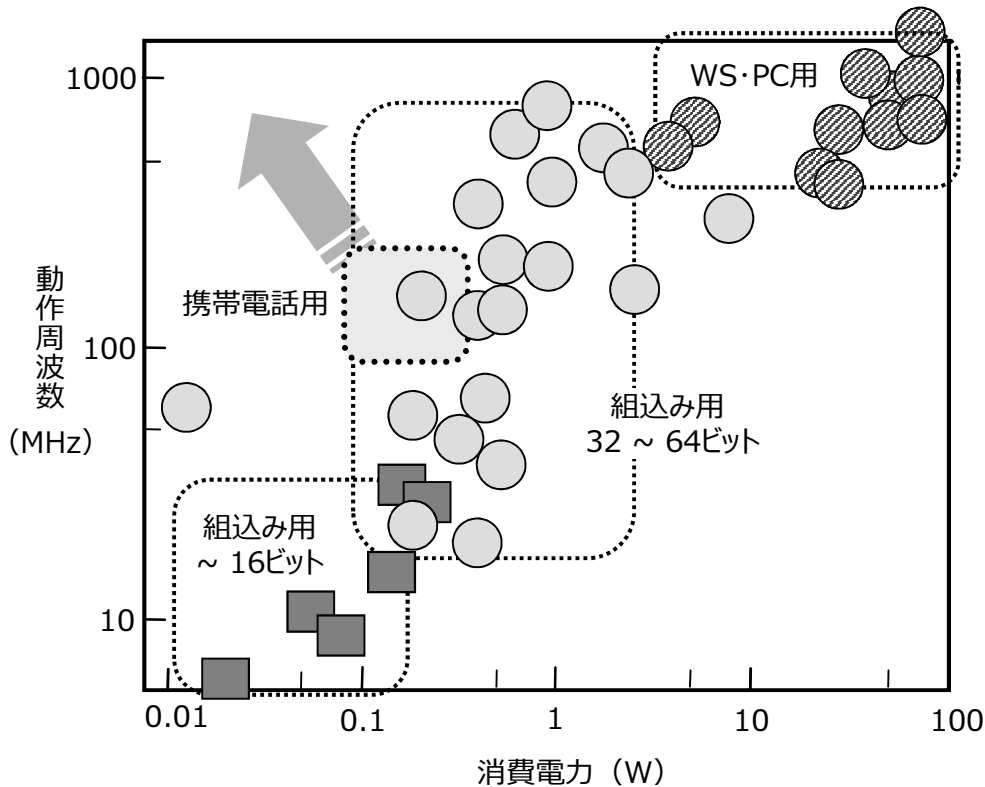


図 1.2 プロセッサの動作周波数と消費電力の動向

1.1.2 本研究の目的

ユビキタス情報社会の進展とともにネットワーク接続される機器が爆発的に増加し、それぞれの機器で実行される機能も非常に高度で複雑なものになってくると予想される。これにともなって、一般の PC や WS なども含めてプロセッサ消費電力の増加が問題視されてきており、特に携帯型情報機器への応用においてマイクロプロセッサの消費電力を低減することが大きな技術課題となっている。

これを解決するためには低消費電力と高性能を同時に満足させることが不可欠であり、携帯機器への適応を考えると小面積というニーズも加わる。しかしながら、低消費電力性と高性能性はトレードオフの関係があり、同時にこの要求を満たすことは容易ではない。また、微細化によって電力密度が増し、そのため消費電力は増え続けている。この状況を打破するため、様々な分野で低消費電力化技術の研究が盛んに行なわれている。

システム全体として消費電力を低減するためには、デバイスや回路技術、アーキテクチャ、ソフトウェアといった多くの技術分野にわたって密接に連携して進めることが必要である。本研究は、そのような連携の中でソフトウェアからのアプローチによってハードウェアが備えている電力制御機構を適切に動作させて、消費電力を低減させる方式について考察することを目的としている。

以下の節で、上述の技術分野での消費電力低減技術について概略を述べる。

1.2 省電力化へのアプローチ

CMOS 回路の消費電力 P は通常次式で与えられる：

$$P = \underbrace{\alpha \times f \times C \times V^2}_{\text{第一項}} + \underbrace{P_s}_{\text{第二項}} + \underbrace{P_l}_{\text{第三項}} \quad (1.1)$$

ここで、 α は回路のスウィッチング確率、 f は動作周波数、 C はゲートの負荷容量の総和、 V は電源電圧であり、式 1.1 の第一項は充放電電流による消費電力で、第二項 P_s 、第三項 P_l は、それぞれ貫通電流、リーク電流による消費電力である。

電源電圧が比較的高い場合は第一項の充放電電流による消費電力が全体の消費電力

の大部分を占めており，この式より CMOS 回路の低消費電力化の一般的なアプローチとして，(1) 電源電圧の低減，(2) 負荷容量の削減，(3) スイッチング確率の低減，(4) 動作周波数の低減，の四つが考えられる．これらに関する技術を次で簡単に述べる．

(1) 電源電圧の低減

電源電圧は消費電力低減のための大きな要因である．単純に電源電圧を下げると CMOS 回路の遅延時間が増加して動作速度が遅くなるが，電源電圧と閾値電圧をともに下げることにより回路遅延を増大させずに電力を削減できる．しかし，リーク電流の増加や動作速度のばらつきといった別の問題が生じる．これらの問題を解決する回路設計技術として，MTCMOS 技術や VTCMOS 技術などが提案されている [11, 12].

本研究では，動作速度との関係を基にソフトウェアによって電源電圧を制御することを目標とする．このためには電源電圧の設定範囲が大きくなるような設計技術が望ましい．

(2) 負荷容量の削減

負荷容量の削減については SOI のような半導体技術や設計時の様々な工夫が主となる．負荷容量の大部分が配線容量で占められている最近の LSI において，小面積化は負荷容量を縮小することになるので，小面積化することで高速化と低消費電力化を実現できることになる．

(3) スイッチング確率の低減

プロセッサの回路部分のうちクロックが消費する電力は全体の 30~50% に達し，スイッチング確率を減らしてこれを削減する手段としてゲートドクロック手法が幅広く用いられている [5, 8]. これは，フリップフロップのクロック入力を抑止することにより，出力変化を抑止して組合せ回路のスイッチングを抑えるものである．

また，メモリの参照回数を抑制するために，内部 RAM を置いて利用したり，メモリやキャッシュのアクセスを最適化することも行われる．

(4) 動作周波数の低減

動作周波数を低減するためには、従来からもあるように、高速実行を実現するアクセラレータなど専用演算器を使用することや目的プログラムの最適化を実施することが有効である。

これらの他、アーキテクチャの面からは、プロセッサの機能として低消費電力モードが用意される。これは、プロセッサに供給されるクロックを停止させて非常に低い消費電力となるモード（スリープモード）を備えるものである。例えば、情報端末ではキー入力などを待っているだけの状態でプロセッサが最大の動作周波数で稼働する必要はない。このようなプロセッサの待ち状態でスリープモードに遷移することで、平均消費電力を低減することができる [8]。

上記の技術のうちの多くは回路やデバイス、アーキテクチャといったハードウェアの領域だけで実現されるものであるが、動作周波数に関連するものやメモリ・キャッシュに関連するものは対応するハードウェア機能をソフトウェアから制御して実現できる。これらの技術を制御・実現される技術領域で分類すると以下の図 1.3 のようになる。

1.3 従来技法

本節では、動作周波数と電源電圧を制御してプロセッサの動作時消費電力を削減する従来技法について概説する。

(1) Intel SpeedStep, Enhanced Intel SpeedStep [13]

Intel SpeedStep は Intel 社の Mobile Pentium III 以降に搭載された省電力機能である。AC アダプタから電源が提供されている時はプロセッサを高動作周波数・高電源電圧で動作させ、バッテリー動作に切替わると低動作周波数・低電源電圧で動作させ省電力化して動作時間を延長するものであり、バッテリー駆動モードでは AC 駆動モードのおよそ 8 割程度の動作周波数と電源電圧で動作さ

	電源電圧の 低減	動作周波数の 低減	負荷容量の 削減	スイッチング 確率の低減
プロセス 回路	MTCMOS, VTCMOS 技術		SOI Low-K	
アーキテクチャ 論理方式 設計手法	電圧 ワイドレンジ 動作	専用演算器 アクセラレータ	小面積化	ゲートド ック 大容量RAM
ソフトウェア	電源電圧の制御		最適化による 高速実行	メモリ配置 の最適化

図 1.3 消費電力低減技術の分類

せる。また、プロセッサのアイドル状態を利用してこの2つの状態を切り替え、実行時に低電力化を実施できる。

さらに、この拡張版である Enhanced Intel SpeedStep が導入され、負荷に応じてプロセッサの動作モードを変更し、多段階に動作周波数と電源電圧を設定する機能が追加された。

(2) LongRun, LongRun2 [14–17]

Transmeta 社の Crusoe プロセッサは LongRun と称する技術を備えている。Intel SpeedStep が供給電源の種類によって動作周波数と電源電圧を変更しているのに対して、LongRun では処理負荷に応じてこれを変更し、プログラム実行時の消費電力を低減する。Crusoe では Code Morphing System (CMS) と呼ぶソフトウェアで、目的プログラムの x86 命令を動的に Crusoe 自身の VLIW 命令に変換しながら実行するが、その中で実行中のプログラム特性に応じて処

理負荷を認識し、それに応じて動作周波数と電源電圧を変更して消費電力を低減する。

また、プロセッサ内の回路からのリーク電流を制御する機能を含めた LongRun2 [15] 技術が同社から発表されている。これは、基板バイアスによって閾値電圧を制御してリーク電流による電力消費を抑制するものであり、Efficeon プロセッサに搭載されている。近年の半導体製造プロセスの微細化によってリーク電流による電力は無視できない問題になっており、LongRun2 ではこれを減少させることで電力消費を抑えるとしている。

(3) Foxtan [18, 21]

Itanium 2 プロセッサに搭載される予定であった Intel 社の Foxtan 技術は、動的にプロセッサの動作周波数と電源電圧とを切替える技術で、基本的には SpeedStep と同じであるが、SpeedStep が平均消費電力を下げるために使われるのに対して、Foxtan ではプロセッサの消費電力枠を制御するために使われる。プロセッサの消費電力が規定の枠を超えたり温度が既定値を超えたりした場合に、自動的に動作周波数と電源電圧が低減される。

同様の電力制御機構は、ARM の IEM (Intelligent Energy Manager) [20] や AMD の PowerNOW! [19] などでも見られるものである。

1.4 本研究における低消費電力化技術の特徴と位置づけ

低消費電力化のためには、前節で述べたようなプロセス技術や回路方式、アーキテクチャなどハードウェア面での改良に加えて、アプリケーションの動作に必要な処理をいかに効率良く実行するか、また、必要な低消費電力ハードウェア機構の設定をいつ行うか、というソフトウェア面での技術が重要になりつつある。コードチューニングや最適化コンパイラによって目的プログラムで実行される命令数を削減したり、キャッシュのアクセスを制御したりする最適化技法も低消費電力化の観点から捉えることができる。実行される命令数が削減されれば、結果的にプロセッサの稼働時間を短くすることができ、同じ処理を行うのに動作周波数を低減することができて消費電力の低減につながるからである。例えば MPEG-4 の符号化処理では積和演算が多く

実行されるが、この部分に DSP を用いることによって動作周波数の低減を図り、消費電力を低減できる [8].

本研究では、ソフトウェアによる消費電力制御手法の中で、プロセッサの動作周波数を積極的に低下させて電力削減を図るものを提案し、考察する。通常、プログラム実行時のプロセッサの処理量（負荷）は、それが処理するデータに依存して変動する。提案する方式は、プログラム動作時におけるこの実行負荷の変動を利用して、実行負荷が低いすなわち必要な処理量が小さい場合に、プロセッサの動作周波数および電源電圧を同時に低下させ、さらに、処理時間のデッドラインを守ることで、プログラム実行時間での電力量（消費エネルギー）の観点から低消費電力化を図るものであり、本研究において「周波数－電圧協調型電力制御：Frequency-Voltage Cooperative CPU Power Control」（以下「FV 制御」）方式と呼ぶ。

FV 制御は、DVS (Dynamic Voltage Scaling) や DVFS (Dynamic Voltage Frequency Scaling) [22, 26], V_{dd} Hopping [23] などとも呼ばれ、消費電力が動作周波数と電源電圧の両者に相関することに着目したプロセッサの消費電力制御方式である。特に電源電圧は 2 乗で消費電力に影響するため、動作周波数のみを低下させるのに比べて効率良く低消費電力化を実現できる [22–31]。現在これは、前述の LongRun や Enhanced SpeedStep によって利用されつつある技術であるが、以下のような状況であり、ソフトウェア主導の方式としてはまだ完成されていないと考えられる。

まず、Enhanced Intel SpeedStep では、プロセッサの負荷状況を OS のレベルで監視しており、そのためタスク全体のような比較的粒度の大きい処理単位で負荷を捉えることになりがちである。DVD の再生などでは、そのタスク内の様々な部分で再生されるデータに依存して負荷が変動するが、そのような負荷変動に正確には追従して制御できないと考えられる。

一方、LongRun では CMS によって Enhanced Intel SpeedStep よりも小さな粒度で処理負荷を監視できると考えられる。しかし、この方式の対象は PC 用であってコード変換の CMS のために使用するメモリ量も多くなり、組込み用途には適していないと考えられる。またアプリケーション実行時間のデッドラインの保証については考慮されていないようである。

組込みシステム用のマイクロプロセッサで FV 制御の効果を十分に引き出すためには、実行されているプログラムが積極的に関与してより粒度の小さい単位で処理量を見積もらなければならない。このためには下記二つの技術課題を解決することが必要である。

(1) 動作周波数と電源電圧の決定方式

FV 制御を効率良く行うために、動作周波数と電源電圧を決定する方式を確立する。理論上は、十分多数あるいは連続的に変化させることが可能な動作周波数と電源電圧とを用意すれば最高の効率（最小の消費電力量）で処理を行うことができるが、コスト高となり実用性に欠ける。実用的なシステム設計を行うためには、効率とのトレードオフを明確にして設定する動作周波数と電源電圧を決定する必要がある。

(2) 処理量の把握

また、FV 制御を効率良く行うために、プロセッサが実行すべき処理量を処理開始前に正確に把握することが必要となる。このようなことは、特定用途に専用されるプロセッサでは可能な場合もあるかもしれないが、多種類の処理をこなす汎用プロセッサでは困難である。近年では携帯機器のマイクロプロセッサにメディア処理やユーザインタフェース機能など多様な機能が搭載される傾向にあり、いかにして実行すべき処理量を把握するかが大きな技術課題となる。

本研究では、ソフトウェアから見たプロセッサの動作時消費電力量の制御について、上に挙げた FV 制御方式実現のための課題の解決法を与えて考察し、その方式を提案する。

以降、本研究ではソフトウェアによる低消費電力化の方式としてこの FV 制御を取上げ、第 2 章でその基本的な性質を明らかにし、方式をマイクロプロセッサに適用するにあたっての消費電力量削減効率の評価と低消費電力動作のために必要となる動作周波数群を得るアルゴリズムを述べる。第 3 章において、この FV 制御で、動作しているアプリケーションの負荷特性を反映させるフィードバック型のアルゴリズム

(フィードバック型 FV 制御) を提案する。そして、携帯電話を想定した実アプリケーションにおいて、そのアルゴリズムを用いて処理量を把握し、シングルプロセッサで稼動するものに適用した例について述べる。

また今後、組込みプロセッサシステムにおいてもマルチプロセッサが用いられるようになる [32–35]。組込みシステムでマルチプロセッサ構成をとる主要な目的の一つが低消費電力化であり、プログラムを並列実行することに提案する FV 制御方式を併せることによって、さらに消費電力を低減させる効果を得ることが期待できる。第4章において、マルチプロセッサでの動作時のプロセッサ消費電力量の低減方式について、第3章で与えたフィードバック型 FV 制御をそれぞれの要素プロセッサに適用したものをシミュレーション結果によって評価・考察する。

第 1 章の参考文献

- [1] 前島 英雄：“マイクロプロセッサの高性能化、高機能化”，電子情報通信学会 2000 年エレクトロニクスソサイエティ大会講演論文集 (2)， pp. 191–192 (2000-9)
- [2] 前島 英雄：“RISC + マルチ VLIW アーキテクチャ融合マイクロプロセッサ”， STARC シンポジウム 2001 講演予稿集， pp. 50–59 (2001-9)
- [3] 内山 邦男：“デジタル民生用マイクロコンピュータにおける低電力設計”， 電子情報通信学会論文誌 C, Vol. J83-C, No. 6, pp. 447–453 (2000-6)
- [4] 内山 邦男：“デジタル民生機器向けマイクロコンピュータの高性能化および低消費電力化に関する研究”， 東京工業大学 博士論文 (2001-3)
- [5] 内山 邦男：“高性能・低消費電力マイクロプロセッサ”， 日立評論 Vol. 87, No. 5, pp. 89–94 (2005-5)
- [6] 武田 英次， 矢野 和男， 鈴木 敬：“ユビキタス情報社会の実現に向けた半導体技術”， 日立評論 Vol. 84, No. 10, pp. 5–10 (2002-10)
- [7] (独) 新エネルギー・産業技術総合開発機構：“電子・情報技術分野の技術ロードマップ” (2005-4)
- [8] 入江 直彦， 十山 圭介， 西井 修， 中澤 拓一郎：“ユビキタス情報時代の超低消費電力 LSI 技術 アーキテクチャとシステムソフトウェア”， 日立評論 Vol. 84, No. 10, pp. 17–20 (2002-10)
- [9] 大場 信弥， 小島 昇， 富永 浩安， 久保 英二， 新村 篤：“ネットワークを支える

- マイコン用ミドルウェア – テレビ電話システムにおける例 –, 日立評論 Vol. 83, 10 月増刊号, pp. 31–36 (2001-10)
- [10] 桜井 貴康 他編: “システム LSI – アプリケーションと技術 –, (株)サイエンスフォーラム (1999-7)
- [11] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada: “1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS”, *IEEE Journal of Solid-State Circuits*, Vol. 30, No. 8, pp. 847–854, Aug. 1995.
- [12] T. Kuroda, T. Fujita, S. Mita, T. Nagamatu, S. Yoshioka, K. Suzuki, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai: “A 0.9V 150MHz 10mW 4mm² 2-D Discrete Cosine Transform Core Processor with Variable-Threshold-Voltage Scheme”, *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 11, pp. 1770–1779, Nov. 1996.
- [13] S. Gochman, R. Ronen, I. Anati, A. Berkovits, T. Kurts, A. Naveh, A. Saeed, Z. Sperber, and R. C. Valentine: “The Intel® Pentium™ M Processor: Microarchitecture and Performance”, *Intel Technology Journal*, Vol. 7, Issue 2, pp. 21–36, May 2003.
- [14] 浅見 直樹, 芳尾 太郎: “Crusoe の船出 ポストパソコン時代のマイコン像”, 日経エレクトロニクス 2000.3.13 (no.765), pp.131–165 (2000-3)
- [15] M. Fleischmann: “LongRun™ Power Management: Dynamic Power Management for Crusoe™ Processors”, Transmeta Corporation, Jan. 2001.
- [16] K. Krewell: “TRANSMETA GETS MORE EFFICEON: Transmeta Delivers New Core, Code-Morphing Software”, *Microprocessor Report*, October 27, Oct. 2003.
- [17] D. R. Ditzel: “Power Reduction using LongRun2 in Transmeta’s Efficeon Processor”, Spring Processor Forum Presentation, May 2003.

- [18] J. Wei: “プロセッサの動作周波数とアプリケーション・パフォーマンスを引き上げる Foxtan* テクノロジー” (日本語参考訳), *Technology@Intel Magazine*, pp. 3–5 (2005-9)
- [19] Advanced Micro Devices, Inc.: “AMD PowerNow!TM Technology Dynamically Manages Power and Performance”, (Informational White Paper), Nov. 2000.
- [20] ARM Ltd.: “Intelligent Energy Manager (IEM) Hardware Control System in the ARM1176JZF-S Development Chip”, Application Note 172, Nov. 2006.
- [21] R. McGowen, C. A. Poirier, C. Bostak, J. Ignowski, W. H. Parksand, and S. Nafziger: “Power and Temperature Control on a 90nm Itanium Family Processor”, *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 1, pp. 229–237, Jan. 2006.
- [22] T. D. Burd, T. Pering, A. Stratakos, R. W. Brodersen: “A Dynamic Voltage Scaled Microprocessor System”, *Digest of Technocal Papers: 2000 IEEE International Solid-State Circuits Conference*, pp.294–295, Feb. 2000.
- [23] 桜井 貴康: “電圧ホッピングによる SH4 搭載システムの低電力化”, 第 1 回 SH フォーラムシンポジウム (2000-4)
- [24] 川口 博, 張 綱, 李 誠洙, 辛 英洙, 桜井 貴康: “低電力実時間組込システムのための OS, アプリケーション, ハードウェア協調による CVS (Cooperative Voltage Scaling) と電圧ホッピング”, 電子情報通信学会技術研究報告 ICD2001-32, Vol. 101, No. 85, pp.59–65 (2001-5)
- [25] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos: “Data Driven Signal Processing: An Approach for Energy Efficient Computing”, *Proceedings of 1996 International Symposium on Low Power Electronics and Design Conference*, pp.347–352, Aug. 1996.
- [26] S. Lee and T. Sakurai: “Run-time Voltage Hopping for Low-power Real-time Systems”, *Proceedings of 37th Design Automation Conference*, pp.806-809, Jun. 2000.

- [27] 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”, 電子情報通信学会技術研究報告 CPSY96-84, Vol. 96, No. 426, pp. 57–64 (1996-12)
- [28] 石原 亨, 安浦 寛人: “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”, 電子情報通信学会技術研究報告 CPSY97-77, Vol.97, No. 347, pp. 65–72 (1997-10)
- [29] 石原 亨, 安浦 寛人: “可変電圧プロセッサを用いた省エネルギー化のための基本定理”, 電子情報通信学会技術研究報告 ICD98-45, Vol. 98, No. 66, pp. 69–76 (1998-5)
- [30] 大隈 孝憲, 石原 亨, 安浦 寛人: “可変電源電圧プロセッサに対するリアルタイムタスクスケジューリング手法”, 電子情報通信学会論文誌 C, Vol.J83-C, No.6, pp. 454–462 (2000-6)
- [31] T. Ishihara and H. Yasuura: “Voltage Scheduling Problem for Dynamically Variable Voltage Processors”, *Proceedings of 1998 International Symposium on Low Power Electronics and Design Conference*, pp. 197–202, Aug. 1998.
- [32] “シングルコアよりマルチコア”, 日経エレクトロニクス 2004.8.30 (no.896), pp. 96–121 (2004-8)
- [33] 枝廣 正人, 鳥居 淳, 酒井 淳嗣, 井上 浩明, 山端 潔, 石川 晋也, 伊藤 義行, 鈴木 悟, 西 直樹: “マルチコア向けソフトウェア・プラットフォームを開発し携帯電話機に適用”, 日経エレクトロニクス 2005.3.28 (No. 881), pp. 125–136 (2005-3)
- [34] 吉田 正康: “組み込み分野における「マルチプロセッサ」とは, — 多機能・低消費電力の要求にこたえるための技術要素と課題”, デザインウェーブマガジン, Vol. 10, No. 8, pp. 68–74 (2005-8)
- [35] A. Jerraya, H. Tenhunen, and W. Wolf: “Multiprocessor System-on-Chips”, *IEEE Computer*, Vol. 38, Issue 7, pp. 36–40, Jul. 2005.

第 2 章

FV 制御によるプロセッサ動作時消費
電力量低減方式の考察

2.1 はじめに

システム全体の電力消費の大きな要因であるプロセッサの動作時の消費電力を低減するため、ソフトウェア・ハードウェア両面から多様な手法が提案されている。その一つとして本研究で行う周波数-電圧協調型電力制御（FV制御）は、動画再生のように処理負荷がデータの状況によって変動するアプリケーションにおいて、動作速度が遅くてもよい場合に、プロセッサの動作周波数と電源電圧とを同時に低下させて消費電力を低減するものである [1-3]。FV制御では、動作周波数と電源電圧の2乗とが電力低下に寄与するため、プログラム動作時のプロセッサ消費電力を大幅に削減できると期待できる。

MPEG-4による動画再生プログラムの実行を例にとると、図2.1に示すような処理負荷の頻度分布となっており、平均では最大負荷の30%程度であるものの、80%程度である場合も僅かに存在する。そこで、このような動作時の出現頻度が低い高負荷の状況に備えるために余裕をもった、すなわち平均的には30~40MHzの動作周波数で十分なものを100MHz以上の動作周波数で動作させるという設計がなされている。

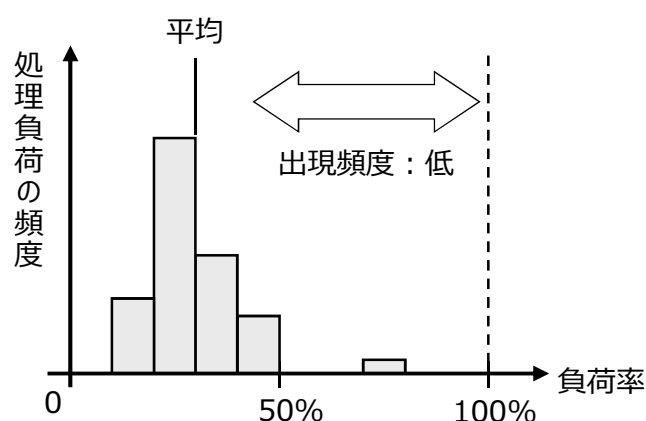


図2.1 MPEG-4デコーダにおける負荷変動の例

この動画再生の例を模式的に示すと、再生時の処理負荷は入力データである再生シーンによって変動し、この負荷が検出できれば、図2.2に示すようにシーンの変化

が大きい高負荷時には高速（動作周波数・電源電圧：高）モードで、シーンの変化が小さい低負荷時には低速（動作周波数・電源電圧：低）モードでプロセッサを動作させて消費電力を低減することができる。また、想定時間内に処理が終了した場合にプロセッサをクロック停止状態にすることも可能である。

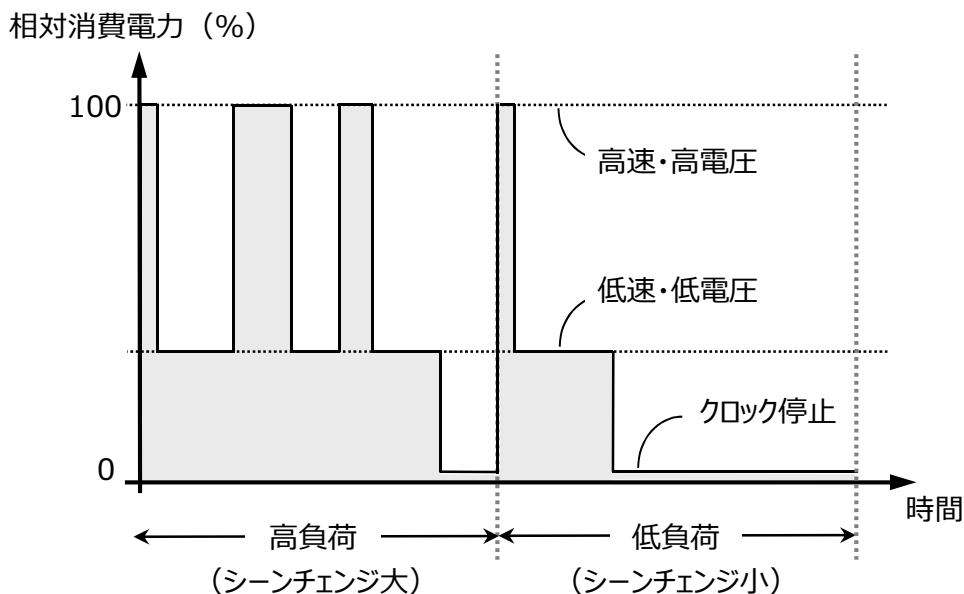


図 2.2 動画再生における負荷と動作モード例

このように、あらかじめ必要十分な負荷が分かっていたら、それに見合った動作周波数と電源電圧でプロセッサを動作させて、再生時間において理想的な消費電力量とすることが可能であるが、そのようなことは一般的には困難である。

この「負荷に見合った必要十分な動作」を得ることに対する実用的な解を得るため、本章では、FV 制御における最適な消費電力量のための動作周波数と電源電圧を得る方式について、その原理を示して動作周波数決定のルールを与える。このルールは、LSI チップの最小動作電圧を考慮して用意すべき動作周波数群を決定し、ルールの適用を打ち切る条件を明確化するものである。最近の低電圧動作のマイクロプロセッサ [11–14] では、用意する動作周波数は、最高の動作周波数に対して、もう 1 個追加すれば概ね十分であることが示される。

以降では、FV 制御を適用した場合のプログラム実行時におけるプロセッサの動作時消費電力量を評価するものとする。

2.2 周波数－電圧関係による消費電力量の表現

2.2.1 周波数－電圧制御の基本原則

FV 制御の基本的な性質が [4–6] により示されている。この要旨は以下の二つの定理で示される。(本章としてのこれらの証明は付録に与える。)

理想的な動作環境として、あるプロセッサの動作周波数 F および電源電圧 V がともに任意の値に設定できると仮定し、適当な V を決めた時 F は、その V において動作可能な最高周波数として一意的に決められるものとする。CMOS 回路の性質により、 F は V に対して単調増加であることができる。また、プロセッサが実行すべき処理について、その処理に必要なクロック数および処理を完了させるべき時刻（デッドライン時刻）があらかじめ分かっているものとする。

この仮定の下に、次の定理 1 が成り立つ。

定理 1

処理を最小の消費電力量でデッドライン時刻までに完了する最適な電源電圧 V_{ideal} が存在する。 V_{ideal} は処理がデッドライン時刻ちょうどに完了する最適な動作周波数 F_{ideal} に対応する電源電圧である。 V_{ideal} 以外の電源電圧による動作では消費電力量は最小にならず、 V_{ideal} 以外の電圧を複数個組合せて動作させる場合も同様に消費電力量は最小にならない。

現実のシステムにおいては、 F および V を任意の値で変化させることは回路規模や動作の安定性の点から製品レベルでの適用には無理があると考えられる。一方、 F を離散的に変えるには、使用する周波数の比が簡単な整数ならば数十ゲート規模の分周回路で実現可能であり、実際これは多くのプロセッサで既に実装されている。また電源電圧についても、数種類の固定電圧レギュレータを切り換える方式ならば、実用的な規模で実装できる。したがって、FV 制御は実用的には、数種類用意された

F と V を切り換えて使用する方式，すなわち「離散 FV 制御」になるものと想定される．この場合，次の定理 2 が成り立つ．

定理 2

V が離散的に複数個用意されている場合，前記 V_{ideal} に隣接する二つの電源電圧を用いた場合に消費電力量が最小になる．この場合にも二つの電源電圧を組合せた結果，処理がデッドライン時刻ちょうどに完了する場合に消費電力量が最小になる．

2.2.2 本研究における動作周波数－電源電圧制御の基本方針

前節で述べた二つの定理は FV 制御の基本となるものであるが，両者とも電源電圧を規定する形式で記述されているため，以下のような制約を生じている．

- (1) 現実のプロセッサにおいては，動作可能な電源電圧に下限 V_{min} がある． V_{min} においても動作周波数を低い方向に変化させることは可能であるから F と V の関係は 1 : 1 ではない．この場合， V をパラメータとすると V_{min} における F を一意的に表現できなくなり， F が低い側での制御方法が検討から漏れることになる．
- (2) アプリケーションでの処理量の多さは，通常その処理に必要なクロック数または実行時間，あるいはそれらから換算される動作周波数で表現される．これを電源電圧に換算するには何らかの F - V 関係が与えられている必要があるが，現実のプロセッサにおいて F と V の関係は，回路設計上の様々な条件に影響されるので，簡単に表現できるとは限らない．ソフトウェアで管理できるパラメータとしては処理量に関連することが必要であるので，このままでは実際のシステム設計に使いつらい．

この問題を回避するために，本研究では FV 制御の性質を検討するにあたって，動作周波数 F を主たる制御パラメータとして扱うことにする．さらに， V を陽に求めなくても良いように， F と消費電力 P との関係を論じることにする．現実のプロ

セッサにおける動作周波数 F と消費電力 P との関係 (F - P 関係) が与えられたものとして, F を制御することによる消費電力の増減を検討するものである. V は F を実現する最小電圧として決定できる.

2.2.3 動作周波数－消費電力 (F - P) 関係のモデル化

現実のプロセッサチップにおける動作周波数－消費電力の関係は, 前述の通り F - V 関係が回路設計上の様々な条件に影響されるので簡単なものではないが, 本研究では消費電力の検討を見通し良く行うため, これを γ 次曲線により近似するものとした.

まず, プロセッサの動作周波数－消費電力関係 (F - P 関係) を図 2.3 のとおりとして定めた.

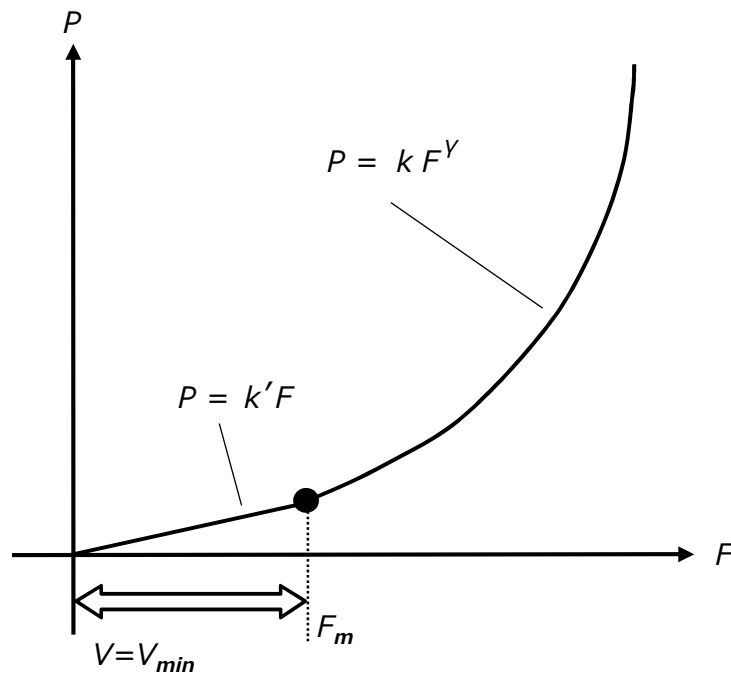


図 2.3 F - P 関係のモデル

これは, あるプロセッサの動作周波数 (F) を定めた場合に, 最低限必要な消費電力 (P) を示す曲線 (F - P 曲線) である. 現実の LSI ではこの曲線は多様なパラメータで規定されるものであろうが, これを式 2.1 によってモデル化することにした. こ

ここで、 γ はプロセッサチップの特性によって定まる定数であり、実際の LSI チップでは γ が 2 前後となる場合が多いようである。

$$P = \begin{cases} kF^\gamma & F \geq F_m \\ k'F & F \leq F_m \end{cases} \quad (2.1)$$

ただし k, k' は適当な比例定数

式 2.1 の F - P 曲線は二つの部分からなる。曲線の左半分は原点から始まる直線であり、右半分は次数 γ の代数曲線である。両者の継目となる周波数が F_m であり、これは当該 LSI チップの最低動作電圧 V_{min} における最高動作周波数である。 F_m より低周波数側では、電源電圧は一定となり消費電力は周波数に単純比例する。 F_m より高周波数側では、周波数の増加に応じて電源電圧も上昇するため、 F - P 曲線はカーブを描く。このカーブの曲がり具合を式 2.1 のパラメータ γ で代表させている。このモデルには物理的な裏付けは無いが、滑らかな代数曲線による近似であるため、現実の LSI に対して大きな誤差は出ないと考えている。

2.3 離散 FV 制御での削減損失の評価

使用できる F と V が離散的な値の組に限定される場合に、上で定めた F - P 関係がどのように変化するかを検討する。前述の定理により、何らかの基準によって最適周波数 F_{ideal} を実現するような動作周波数の組を求めれば、理想的な消費電力からある程度の損失をもって、これに近似できると期待できる。以下で、低減できる消費電力量に対する損失の評価を行って、この動作周波数の組の求め方を示して行く。

定理 2 によれば、定性的に F (および V) の選定に関して、二つの F をあまり離すと効率が劣化することになる。本節では離散 FV 制御において、 F の選定結果が電力効率にどのように影響するかを定量的に評価する。電力効率は、理想的な FV 制御を基準とした相対的な電力量の増加 (= 効率劣化) により評価するものとする。定理 1 により、最適な動作周波数 (電源電圧) での実行が効率の限界を与えるから、それを基準とすれば消費電力に改善の余地があるかどうかを評価できる。

2.3.1 周波数を離散化した場合の F - P 関係

図 2.3 および式 2.1 は，プロセッサの動作周波数（および電圧）を連続的に変化させられると仮定した場合の F - P 関係である．実際にこれらを連続的に変化させるには大規模で複雑なアナログ回路が必要となり実用的でない．そこで，動作周波数および電源電圧は離散化されている，すなわち，図 2.4 のように値 F_1, F_2, F_3 のみがクロック発生回路で発生でき，電源電圧は各々の動作周波数に対応する最も低い電圧が用意されるものとする．この場合， F - P 関係は図 2.4 の破線のように元の F - P 曲線を F_1, F_2, F_3 の点で線形内挿した折れ線状になる．折れ線と元の F - P 曲線とがどの程度乖離しているかを評価することにより，動作周波数を離散化することによる消費電力への影響を知ることができる．以下本章では，動作周波数を離散化しても十分な消費電力量削減効果が得られることを示す．

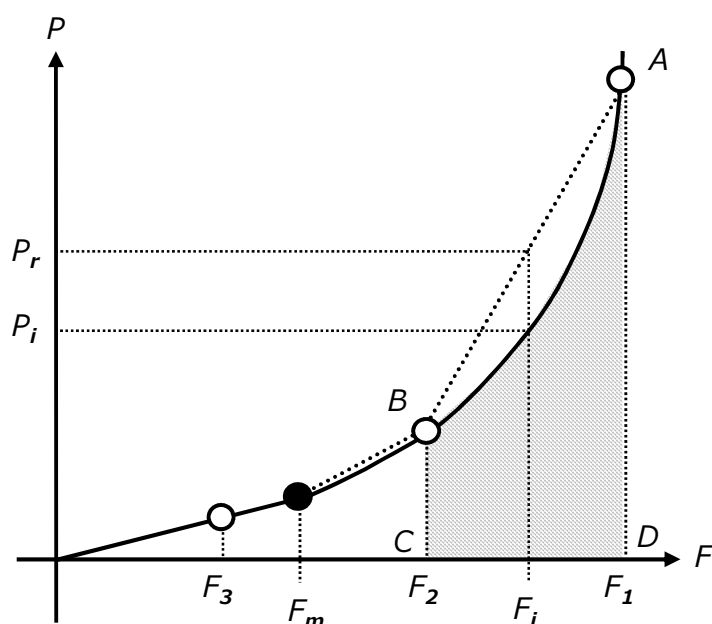


図 2.4 周波数離散化の影響

当該プロセッサが，あるプログラムをデッドライン時刻まで実行するために最適な最低の動作周波数 F_i が必要になると仮定する．もし動作周波数（および電源電圧）

を任意の値に設定できれば、図 2.4 の消費電力 P_i によって当該プログラムを実行することができ、この P_i が当該プログラムを実行するための最低消費電力となる。ここで動作周波数を離散化すると、 F_i に対応する消費電力は、折れ線に対応した値である P_r に増加する。これは、 F_i に相当する動作周波数を F_1 と F_2 の組合せで平均として実現すると動作周波数の減少に比例した消費電力の減少が得られることから明らかである。したがって、 P_r の P_i に対する比率を計算すれば、動作周波数を離散化した場合の、連続的に変化させられる場合すなわち最低消費電力量に対する電力損失が相対的に求められる。

2.4 電力損失の評価

以下の仮定および定義に基づいて、前記の電力損失を具体的に与える。まず、図 2.4 における最高周波数 F_1 は、当該プロセッサの最高動作周波数としてあらかじめ定められているものとする。次に、電力損失の式においては隣接する二つの周波数の比 F_1/F_2 , F_2/F_3 , ... を用いるものとし、この比の値を β とおく。

2.4.1 最大電力損失

前節で述べたとおり、動作周波数を離散化した場合のある動作周波数 F_i における消費電力量の損失（連続に変化させられる場合の最低電力量に対する相対値）は、デッドライン時刻まで実行するものとして図 2.4 における比 P_r/P_i として算出できる。この値を実際に計算すると以下のとおりとなる。

まず折れ線が F_m をまたがない場合（図 2.4 の $F_2 \sim F_1$ に F_i が存在する場合）は、直線 AB の方程式に F_1 , F_2 を代入することにより、

$$P_r = (kF_1^\gamma - kF_2^\gamma)/(F_1 - F_2) + (F_1 \cdot kF_2^\gamma - F_2 \cdot kF_1^\gamma)/(F_1 - F_2)$$

また、 $P_i = kF_i^\gamma$ であるので、 $\alpha = F_2/F_i$, $\beta = F_1/F_2$ で置き換えて、

$$P_r/P_i = (\alpha^\gamma(\beta - \beta^\gamma) + \alpha^{\gamma-1}(\beta^\gamma - 1))/(\beta - 1) \quad (2.2)$$

折れ線が F_m をまたぐ場合 (図 2.5 の $F_3 \sim F_2$ に F_i が存在する場合) は,

$$P_r/P_i = (\alpha^\gamma(K\beta - \beta^\gamma) + \alpha^{\gamma-1}(\beta^\gamma - K))/(\beta - 1) \quad (2.3)$$

$$\text{ただし } \alpha = F_3/F_i, \quad \beta = F_2/F_3, \quad K = (F_m/F_3)^{\gamma-1}$$

2.2, 2.3 両式において変数 F_i は, F_2 または F_3 との相対値 α として表現されている. 上記で式 2.2 は, 式 2.3 において F_m が区間の下端に位置した場合であり, したがって $K = 1$ と置いた場合に帰着される. よって以下では式 2.3 のみを検討する. 式 2.3 は特定の F_i に対する消費電力の損失である. したがって区間 $[F_3, F_2]$ での最大のこの電力損失は, F_i すなわち α の変化に対する式 2.3 の増減を調べることで算出できる. F_i の変域が $[F_3, F_2]$ であることに対応して, α の変域は $[1, \beta]$ となる. この区間での最大値は, 式 2.3 を α で微分することにより,

$$\alpha = (\gamma - 1)(\beta^\gamma - K)/\gamma\beta(\beta^{\gamma-1} - K) \quad (2.4)$$

において得られることが分かる. 式 2.4 を式 2.3 に代入することにより, 特定の β および K に対する最大損失が得られる.

2.4.2 平均電力損失

現実には最大損失を与えるような F_i のみが全実行時間に占めるようなプログラムだけが実行される場合はないと想定されるので, プログラムの実行が様々な F_i を持つ部分プログラム (これをプログラムセグメントと呼ぶ) から構成される場合の平均的な電力損失を評価する.

当該プロセッサが対象とする時間内で n 個のプログラムセグメントを実行し, 各々の実行に要する最低の動作周波数は $F_i(1), F_i(2), \dots, F_i(n)$ であるものとする. この場合, プログラム全体を完了するために必要な電力量は各プログラムセグメントに必要な電力量の和となるから, 電力損失は次の形で計算できる.

$$\sum^n P_r(F_i(\cdot))/\sum^n P_i(F_i(\cdot)) \quad (2.5)$$

この分母は動作周波数を連続的に変化させられるとした場合の電力であり、分子は動作周波数が離散的に用意されている場合の和である。ここで、 n が十分に大きく、かつ $F_i(\cdot)$ が区間 $[F_2, F_1]$ に一様に分布すると仮定すれば、分母は図 2.4 のハッチング部分、分子は台形 $ABCD$ の面積で置き換えられ、以下の式が得られる。

$$\sum P_r / \sum P_i = \frac{(\gamma + 1)(\beta^\gamma + 1)(\beta - 1)}{2(\beta^{\gamma+1} - 1)} \quad (2.6)$$

これは図 2.4 の区間 $[F_2, F_1]$ に対する評価であるが、区間 $[F_3, F_2]$ に対しては以下のとおりとなる。

$$\sum P_r / \sum P_i = \frac{\rho^{\gamma-1}(\gamma + 1)(1 + 1/\beta\rho^{\gamma-1})(1 - 1/\beta)}{(\gamma + 1)(1/\rho^2 - 1/\beta^2) + 2\rho^{\gamma-1}(1 - 1/\rho^{\gamma+1})} \quad (2.7)$$

$$\text{ただし } \rho = F_1/F_m$$

式 2.6 は、式 2.7 において F_m が区間の下端に位置した場合であり、 $\rho = \beta$ と置いた場合に帰着される。

この実行モデルは一つのプログラムの実行を複数の断片（プログラムセグメント）による区間に区切って、そのそれぞれの区間で要求される性能に応じてプロセッサの動作周波数を低下させてプログラム全体の実行性能を保証しながらプロセッサの消費電力量を削減するもので、本 FV 制御の基本的なモデルである。

2.4.3 具体的な例

■ F_m を固定した場合

表 2.1 は周波数比 β および F - P 関係のパラメータ γ を変化させた場合の前記の電力損失の計算結果である。

表の各欄において、上段は式 2.7 による平均電力損失を、下段は式 2.3 および 2.4 による最大電力損失をそれぞれ示す。数値は相対値であり、値 0 が損失なしを、値 10 は理想値に比べて 10% 余分に電力を消費することを示している。表 2.1 の (a) は F_m が評価対象の区間に含まれない場合で、同表の (b) は含まれる場合であり、代表

表 2.1 電力損失の具体値

(a) 区間 $[F_2, F_1]$

$\beta \backslash \gamma$	1.5	2.0	2.5	3.0
1.5	1 2	3 4	5 8	8 13
2.0	3 5	7 13	13 24	20 41
3.0	6 12	15 33	27 69	40 126

(b) 区間 $[F_3, F_2]$

$$F_m = (F_3 + F_2)/2$$

$\beta \backslash \gamma$	1.5	2.0	2.5	3.0
1.5	3 6	6 12	9 19	13 26
2.0	5 10	11 22	17 36	24 52
3.0	9 17	18 38	28 63	39 94

上側の行：平均の損失 (%)

下側の行：最大の損失 (%)

例として $F_m = (F_3 + F_2)/2$ の場合、つまり「継目」が区間の中点にある場合を示したものである。

実測によると実際のプロセッサでは、ほぼ $\gamma = 2.0$ であり、またプロセッサに用意されている動作周波数変更の機構を用いると $\beta = 2.0$ すなわち最大動作周波数の $1/2$ を用いて制御することが簡便であり、かつ効果が期待できるので、これが典型例と考えられる。表 2.1 の通り、(a) の場合には $\beta = 2.0$ とすれば、 $\gamma = 2.0$ の典型的な例に対して平均損失は 7% で、最大損失は 13% である。(b) においてはやや悪化するが、 $\beta = 2.0, \gamma = 2.0$ に対して平均損失 11% にとどまる。

■ F_m の変化を考慮した電力損失

次に F_m を変化させた場合の相対電力損失の値を図 2.5 に示す。ここで、周波数比 $\beta = F_1/F_2$ は 2.0 とした。また、 F_m は次式のパラメータ q を用いて示す。

$$q = (F_m - F_2)/(F_1 - F_2)$$

q は、解析の対象となる周波数範囲 $[F_2, F_1]$ において、 F_m が相対的にどこの位置に来るかを示すものであり、 $q = 0$ ($q = 1$) は $F_m = F_2$ ($F_m = F_1$) すなわち F_m が区間の下端 (上端) に位置することを示す。 $q = 1$ において相対電力損失が 1.0、すなわち損失なしとなるのは、この周波数範囲で当該プロセッサがすべて最低の電圧で動作しなければならず、FV 制御の実効的な意味がなくなることを示している。

図 2.5 のとおり、電力損失は $q = 0.3 \sim 0.4$ 付近で最大となり、最大値は典型例で示した値よりやや大きい。実用上重要となる $\gamma = 2.0$ 付近では、設計上問題となる差は生じない。

2.5 設計ルール

以上の解析と評価をもとに、プログラムの要求動作周波数に応じて F, V を制御して最適な消費電力量を得るための設計ルールを求める。

$\gamma = 2.0$ の典型的な LSI チップにおいて、設定すべき隣接する動作周波数を決定するのに $\beta = 2$ とした場合に削減できる電力に対する損失は、2.4.3 項の具体的な例で示したように 10% 前後である。したがって、最高動作周波数の $1/2$ で次の動作周波

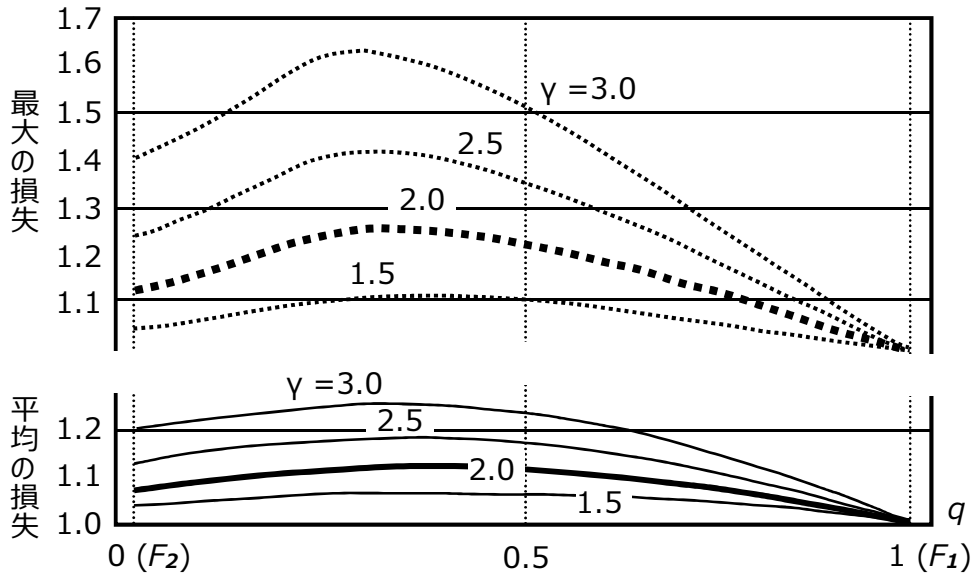


図 2.5 F_m の変化を考慮した電力損失評価

数とそれを実現する電源電圧を設定し、この二つを用いることにより、理想の場合の 10% 前後の損失でプログラム実行時のプロセッサ消費電力量を削減できる。電源電圧がさらに低減できるならば、さらに 1/2 で次の動作周波数（および、それを実現する電源電圧）と、1/2 の比率で動作周波数をとっていけば、最適なものから 10% 程度の損失で電力削減できる動作周波数の組が得られ、FV 制御における使用動作周波数を決定するルールは下記となる：

FV 制御システムの設計において、最高動作周波数 F_1 が与えられた場合、 F_1 以下の動作周波数は、

$$F_2 = 1/2 \cdot F_1, \quad F_3 = 1/4 \cdot F_1, \quad F_4 = 1/8 \cdot F_1, \quad \dots$$

と用意する。

上記で用意された動作周波数の組によって、最適となる最低の動作周波数 F_i での実行を実現してプログラムを実行すればよい。

現実の LSI チップにおいては図 2.3 のとおり $V = V_{min}$ の範囲で F - P 関係が原点

を通る直線となるから、この範囲では下端となる周波数 $F = 0$ と、高々もう 1 個の周波数が用意されれば十分である。すなわち上記ルールを適用して得られた動作周波数が F_m 以下ならば、さらに多くの周波数は不要でありルールの適用を打ち切ってよい。そこまでに得られた周波数群で制御し、さらに時間に余裕が生じると動作周波数 $F = 0$ とするのである。近年 LSI チップの低電圧化が進む結果、このアルゴリズムは $F_2 = 1/2 \cdot F_1$ を求めた時点で、LSI チップに与える電圧がそれより低下させられなくなり、打ち切りとなることが多いと思われる。言い換えれば、FV 制御のために追加する動作周波数は一つのみで実用上十分となる。

また、以上においては、ある時点でプログラムの実行を停止させて電力消費しないようにすること、すなわち周波数 = 0 がそのプロセッサで利用可能であることを前提としている。このためには実際にクロックを停止しても良いが、多くのプロセッサにはクロック停止と等価な低消費電力モード（スタンバイモードやスリープモードなど）が用意されているので、それを利用することができる。

2.6 実プロセッサへの適用

前節の設計ルールを実際のプロセッサに適用する。[7-10]

(1) SH-Mobile 1 [11, 15]

まず、この LSI チップの最高動作周波数 F_1 とそれに対応する電源電圧は、それぞれ 120MHz, 1.5V とした。パラメータ γ は 1.9 と求められ、ルールに従い FV 制御で使用する動作周波数は $F_2 = 60\text{MHz}$ で、これに対応する電源電圧は 1.25V となる。この電圧は同 LSI チップの動作保証の範囲外であるが、実験により正常に稼動することが確認されている。また、本設計においては 1.25V を V_{min} と見なすことにした。したがって $F_2 = 60\text{MHz}$ は F_m 以下であり、ルールの適用はここで打ち切られる。

以上の結果、二つの動作周波数 120MHz (1.5V), 60MHz (1.25V) と $F = 0$ に相当する「スリープモード」により、FV 制御を行うシステム設計が得られる。

(2) Crusoe [12, 13]

Transmeta 社の Crusoe は、FV 制御機構を内蔵したプロセッサとして先進的な位置付けにある。[12, 13] によれば同 LSI チップには 5~7 個の周波数-電圧の組が用意されており、ここでは 5 組のものを対象とする (表 2.2)。表 2.2 から $F_1 = 933$ MHz であり、 $F_2 = 467$ MHz が必要となる。これに対応する電源電圧は 1.35V が妥当である。 F_2 は F_m に達しておらず、 $F_3 = 1/2 \cdot F_2 = 233$ MHz が追加され、 F_3 では最低の電源電圧 0.9V による動作となるため、ここで打ち切られる。

以上の結果、3 組の動作周波数-電源電圧 (およびスリープモード) により、FV 制御が可能になる。

表 2.2 Crusoe への適用例

オリジナル [13] での 使用周波数・電圧		本方式による 使用周波数・電圧	
周波数 (MHz)	電圧 (V)	周波数 (MHz)	電圧 (V)
933	1.35	933	1.35
800	1.25	-	-
667	1.2	-	-
533	1.1	467	1.1
300	0.9	233	0.9

2.7 まとめ

マイクロプロセッサを低消費電力で動作させる手法として、動作周波数および電源電圧を実行時にソフトウェアから制御する「周波数-電圧協調型電力制御方式」が有望視されている。本研究では、まず実際のマイクロプロセッサの動作周波数-消費電力特性を関数で近似して、動作周波数と電源電圧を変化させることによる電力効率を

解析的に評価した。

この結果，理想的な消費電力量から 10% 程度の損失を容認すれば，提案する FV 制御方式で使用する動作周波数は，最高動作周波数から 2 : 1 の比で順次用意すれば良く，現実の LSI チップでは最高周波数の 1/2 となる周波数を 1 個だけ追加すれば概ね十分である，という設計ルールを得た．同ルールを用いて，実際のマイクロプロセッサである SH-Mobile 1 と Crusoe それぞれに向け，動作周波数と電源電圧を選択して消費電力量制御システムを設計することができた．

第 2 章の付録 基礎定理の証明

離散 FV 制御は、線分近似により F - P 関係を定めた仮想的なプロセッサに対して理想的（任意の F , V を使用可能）な FV 制御を行うことと等価である。この性質を用いて本文で述べた定理を証明する。

補題

適当なプロセッサにおいて、予め指定された 2 つの動作周波数 F_1, F_2 を適当な比率で混用することにより、平均動作周波数が F となったとする。この時の消費電力量を P とすれば、 P は F に対して線形に変化する。

この意味は以下のとおりである：

あるプロセッサの LSI チップにおいて、 F - P 関係が図 2.6 の曲線（点線）のように与えられたとし、使用可能な動作周波数が図中の F_1, F_2 のみであるとする。

この場合、一般の周波数 $F_i (\neq F_1, F_2)$ を用いてプロセッサを動作させることはできないが、 $F_2 \leq F_i \leq F_1$ ならば F_1, F_2 を適当な比率で混用することにより、平均動作周波数を F_i とすることができる。この場合、この区間における F - P 関係は、図において曲線上の 2 点 $A(F_1, P_1)$ と $B(F_2, P_2)$ とを結ぶ線分により表され、曲線を線分近似したのと等価である、というのが本定理の内容である。

なお P_1, P_2 は F_1, F_2 に各々対応した消費電力量である。

証明

総実行時間のうち、動作周波数 F_1 で実行した時間比率を $\delta (0 \leq \delta \leq 1)$ とすると、平均動作周波数 F は、

$$F = \delta F_1 + (1 - \delta) F_2 \quad (2.8)$$

であり、この場合の平均消費電力量 P は、

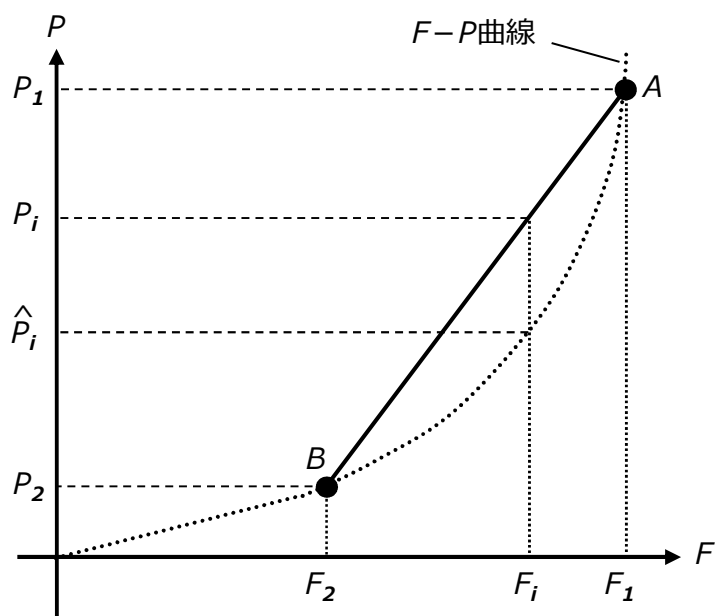


図 2.6 周波数離散化による電力量増加

$$P = \delta P_1 + (1 - \delta) P_2 \quad (2.9)$$

である。

式 2.8 において F_1, F_2 は定数であるから、 F は δ に対して線形に変化する。同様に式 2.9 において P_1, P_2 は定数であるから、 P は δ に対して線形に変化する。したがって、 δ を消去して得られる F と P の関係も線形となる。

証明終り

本文で述べた定理の証明を以下に与える。

定理 1

最適な電源電圧 V_{ideal} が存在する

本定理の前提として F - V 関係が $1:1$ に定まることを仮定している。このもとでは本定理は、最適な動作周波数 F_{ideal} が 1 つ定まることと等価である。またこの範囲においては、 V は一定ではない（真に単調増加する）から $\gamma > 1$ である。以上と補題とから、本定理を証明する。

証明

処理をデッドライン時刻ちょうどに完了する動作周波数を F_{ideal} 、また F_{ideal} に対応する消費電力量を P_{ideal} とすると、他の動作周波数 F による電力量は必ず P_{ideal} よりも大きくなるのが以下のとおり示される。

まず単一の動作周波数 F で処理を実行する場合を考える。 F が F_{ideal} 未満ならば処理を完了できないので条件に合わない。 F_{ideal} 以上の周波数については、 F - P 関係が単調増加関数であるから P_{ideal} が最小電力となる。

次に、二つの周波数 F_1, F_2 を組合せて処理を実行する場合を考える。補題によりこれは、線分近似によって F - P 関係を定めた仮想的なプロセッサにおいて、単一の周波数を用いて処理を実行することと等価である（図 2.6）。元のプロセッサの F - P 関係は $\gamma > 1$ の場合下に凸となるから、図において線分 AB は元の F - P 曲線より上方すなわち大電力側となり、 P_i と \hat{P}_i で示されるように、 P_i は最小電力量とはならない。

三つ以上の周波数を組合せる場合は、上記を繰り返し適用することに相当する。たとえば（図 2.7）のように、まず F_1 と F_2 により平均動作周波数 F_{12} で動作を行い、その後に第三の動作周波数 F_3 で動作させた場合、 F - P 関係は図の線分 CD で表現できる。この場合でも元の曲線が下に凸である限り、線分は曲線よりも上方となり最小電力とはならない。

証明終り

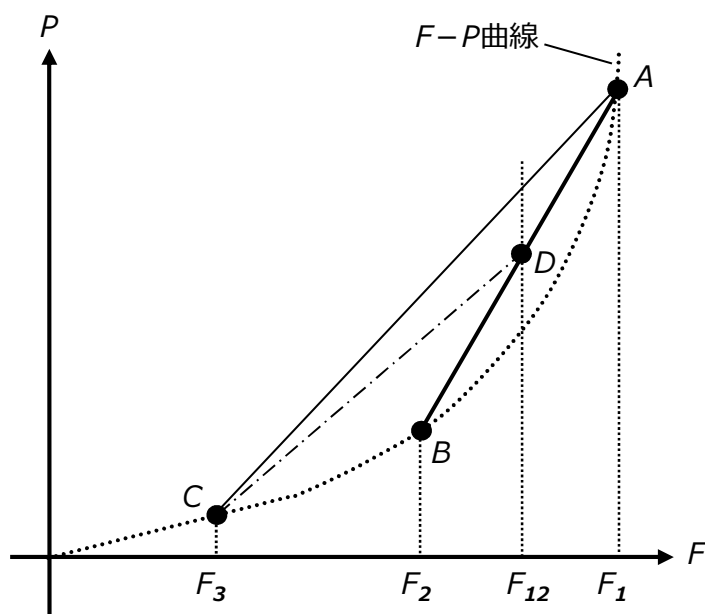


図 2.7 三つ以上の周波数を組合せて使用する場合の状況

定理 1 の系

スリープ制御では電力量最小にはならない。

処理をデッドライン時刻よりも早く完了した場合に、残りの時間はプロセッサをスリープモードのような低消費電力モードにすることにより消費電力を削減する方法をスリープ制御と呼ぶ。スリープ制御においては、処理はまったく行えない一方で消費電力は 0 とみなせるものとする。

証明

低消費電力モードは F - P 関係における $F = 0, P = 0$ の点に該当する。従ってスリープ制御は、二つの周波数 F_1 ($> F_{ideal}$ の範囲で任意) および $F_2 = 0$ を用いた離散 FV 制御と等価であるから、定理 1 により P_{ideal} よりも大きな電力を消費する。

証明終り

定理 2

V が離散的に複数個用意されている場合、前記 V_{ideal} を挟む二つの電源電圧を用いた場合に消費電力が最小となる。

ここで、 F - V 関係に関する前提は定理 1 におけるものと同じとする。したがって、「 V_{ideal} を挟む二つの電源電圧」は、「 F_{ideal} を挟む二つの動作周波数」と読み替えることができる。

証明

元のプロセッサの F - P 関係が下に凸であるから、線分近似する場合 F_{ideal} になるべく近い所に点をとる方が、線分と元の曲線との距離が小さい。図 2.7 の例で F_{12} が F_{ideal} であった場合、 B が C よりも F_{ideal} に近いので、線分 AB の方が AC よりも曲線に近い。

すなわち消費電力の増加が抑えられる。

証明終り

第 2 章の参考文献

- [1] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos: “Data Driven Signal Processing: An Approach for Energy Efficient Computing”, *Proceedings of 1996 International Symposium on Low Power Electronics and Design Conference*, pp.347–352, Aug. 1996.
- [2] T. Ishihara and H. Yasuura: “Voltage Scheduling Problem for Dynamically Variable Voltage Processors”, *Proceedings of 1998 International Symposium on Low Power Electronics and Design Conference*, pp. 197–202, Aug. 1998.
- [3] S. Lee and T. Sakurai: “Run-time Voltage Hopping for Low-power Real-time Systems”, *Proceedings of 37th Design Automation Conference*, pp.806–809, Jun. 2000.
- [4] 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”, 電子情報通信学会技術研究報告 CPSY96-84, Vol. 96, No. 426, pp. 57–64 (1996-12)
- [5] 石原 亨, 安浦 寛人: “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”, 電子情報通信学会技術研究報告 CPSY97-77, Vol.97, No. 347, pp. 65–72 (1997-10)
- [6] 石原 亨, 安浦 寛人: “可変電圧プロセッサを用いた省エネルギー化のための基本定理”, 電子情報通信学会技術報告 ICD98-45, Vol. 98, No. 66, pp. 69–76 (1998-5)

- [7] 十山 圭介, 三坂 智, 相坂 一夫, 在塚 俊之, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康: “CPU 消費電力削減のための周波数-電圧協調型電力制御方式の設計ルールとフィードバック予測方式による適用”, 電子情報通信学会和文論文誌 D-I 分冊, Vol.J87-D-I No. 4, pp. 452-461 (2004-4)
- [8] K. Aisaka, T. Aritsuka, S. Misaka, K. Toyama, K. Uchiyama, K. Ishibashi, H. Kawaguchi, and T. Sakurai: “Design Rule for Frequency-Voltage Cooperative Power Control and Its Application to an MPEG-4 Decoder”, *IEEE Solid-State Circuits Society 2002 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 216-217, Jun. 2002.
- [9] 相坂 一夫, 在塚 俊之, 三坂 智, 十山 圭介, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康: “周波数-電圧協調型電力制御における使用周波数群決定方法の提案 MPEG-4 デコーダによる検証”, 電子情報通信学会技術研究報告 ICD2002-37, Vol. 102, No. 234, pp. 13-16 (2002-7)
- [10] 相坂 一夫, 在塚 俊之, 三坂 智, 十山 圭介, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康: “周波数-電圧協調型電力制御におけるアルゴリズムとデザインルール”, 電子情報通信学会技術研究報告 ICD2002-72, Vol. 102, No. 274, pp. 49-52 (2002-8)
- [11] T. Yamada, M. Ishikawa, Y. Ogata, T. Tsunoda, T. Irita, S. Tamaki, K. Nishiyama, T. Kamei, K. Tatezawa, F. Arakawa, T. Nakazawa, T. Hattori, and K. Uchiyama: “A 133MHz 170mW 10 μ A Standby Application Processor for 3G Cellular Phones”, *IEEE International Solid-State Circuits Conference 2002 Digest of Technical Papers*, pp. 370-371, Feb. 2002.
- [12] Transmeta Corporation: “CrusoeTM Processor Model TM5500/TM5800 Version1.0 Data Book”, Feb. 2003
- [13] Transmeta Corporation: “CrusoeTM Processor Product Brief: Model TM5800”, Feb. 2003.

-
- [14] 浅見 直樹, 芳尾 太郎 : “Crusoe の船出 ポストパソコン時代のマイコン像”, 日経エレクトロニクス 2000.3.13 (no. 765), pp.131-165 (2000-3)
- [15] SH-Mobile1 (SH7290), 株式会社 ルネサス テクノロジ,
<http://japan.renesas.com/>

第3章

周波数－電圧協調型電力制御方式の
フィードバック予測による適用

3.1 はじめに

FV 制御（周波数－電圧協調型電力制御）を適用する処理単位としては、アプリケーション全体やタスクなどが考えられる [1, 2] が、プログラムを細分した部分（プログラムセグメント：以降単にセグメントと呼ぶ）ごとに制御すると、よりプログラムの実行特性を反映した電力制御が行えると考えられる。このため、本研究ではセグメントによる FV 制御の適用を中心に検討する [3, 4].

前章で解析的な考察により、消費電力を削減するために採用するプロセッサ実行時の動作周波数と電源電圧を決定する設計ルールを与えた。本章では、実際のアプリケーションへの FV 制御適用にあたって、動作周波数（と電源電圧）を変更するソフトウェア制御方式を示して、それを MPEG-4 デコーダに適用して実機上で動作確認し、望ましい消費電力の低減効果を得た結果を示す。

さらにこの方式が MP3 デコーダおよび、両者が並行タスクとして動作する環境でも想定どおりに機能し、低消費電力化できることを確認する。

3.1.1 ソフトウェアからの処理負荷の見積り

実際にプログラム実行時に、その負荷に応じて動作周波数を低減させるためには、何らかの形で負荷状況を（実行時にリアルタイムで）得る必要がある。そのようなことが可能なプログラムも存在するであろうが、一般には実行時のプログラムの挙動をその時点で正確に把握することはできず、何らかの形での推測を行うことになる。

Enhanced Intel SpeedStep [8] や LongRun [9] ではこれをプロセッサのパフォーマンスモニタなどを参照して推測することを想定している。また、ソフトウェアの処理によるものでは、動画再生において、二つの処理フレームで直前のものと現在のものとの画像のずれを計算して、この値と強い相関関係をもつ処理量の推測を行うものがある [10].

また、タスクやセグメントのようなプログラム区間に対してコンパイラのサポートと実行時プロファイルを基に処理量を予測する手法も提案されている [17–19].

本研究においては、本章で述べる「フィードバック型制御方式」によってこの見積りを行うこととした。この方式は、簡単な手順によって実行中に負荷状況をかなり正確に見積もれるものであり、アプリケーションに依存した計算を行うことなく、またパフォーマンスモニタによるものよりもアプリケーション実行特性の観点からは精度が高いと考えられる。

3.2 FV 制御のシステム適用

—ソフトウェアによる周波数—電圧制御の解法

低消費電力化に向けた機構としてハードウェアによるサポートは、前章で述べたことをまとめると、次の二つが必要になる。

(1) 可変動作周波数機構

プロセッサの仕様の範囲で動作周波数を任意の値に設定することは困難であるが、離散的な特定の値に設定することは簡単である。多くのプロセッサは基本的なクロックを逡倍することで動作周波数を得ている。この逡倍率はハードウェア的に変更できるプロセッサが多い。また、クロック停止と等価なモード（スリープモード）を備えること。

(2) 可変電源電圧機構

高効率なスイッチングレギュレータを用いて複数の電圧を用意すること。組込みシステムに向けたものではまだ普及していないが、PC などでは既に装備されているものもあり、組込みシステム用としての適用も可能である。

一方、それを制御するためのソフトウェアに要求されるものは、電力制御機能を備えたマルチタスク構造と処理負荷の認識機構である。携帯電話に搭載されるプログラムも含め、一般にプログラムはマルチタスクの形態で稼動する。プリエンティブなマルチタスク下で、プロセッサの省電力モード（スリープモード）と可変電源電圧機構を利用した協調型電力制御が実現できる道具立てが必要であり、

- 処理負荷に応じ、動作周波数と電源電圧を下げる制御

- プロセッサがアイドルになる動作状況での省電力モードへの移行

をマルチタスク上で実現することになる。

タスクに対してプロセッサの動作周波数および電源電圧を与えてスケジューリングする原理的な方式が [2] に述べられている。また、2次電池を最長利用するという観点でタスクをスケジューリングする手法が [11] にある。本研究では離散 FV 制御を用いて前章の結果を適用するため、一つのタスクの中で動作周波数と電源電圧を動的に変更・設定する方式を提案する。

3.2.1 フィードバック型 FV 制御

上記の前提のもと、前章までの解析と設計ルールを実際のプロセッサとそのプログラムに適用し、得られた動作周波数と電源電圧の組で、プログラムの負荷に応じた最適周波数 F_{ideal} での実行を近似して消費電力量を削減する。そのためにはソフトウェアからプログラムの負荷情報を与える必要がある。プログラムの負荷は実行データによって動的に変化するの、動作周波数の変更を動的に適切に行うため、何らかの形でプロセッサに対する負荷を予測しなければならない。フィードバック型 FV 制御（以降 FFV 制御と呼ぶ）は、プログラムの実時間性を前提としてあらかじめ実行スケジュールを立て、実際の動作時間とスケジュール時間との比較で、この予測を行うものである [4-7]。

FFV 制御では、制御されるプログラムが以下の性質を持っていることを前提とする。

- (1) 当該プログラムが実行を完了すべき限界時刻（デッドライン時刻）が分かっている。
- (2) 当該プログラムが実行を完了するために必要な処理ステップか時間がほぼ分かっている。

この条件のもとでは、MPEG-4 動画デコーダを例として、次の図 3.1 に示すような制御が可能となる。

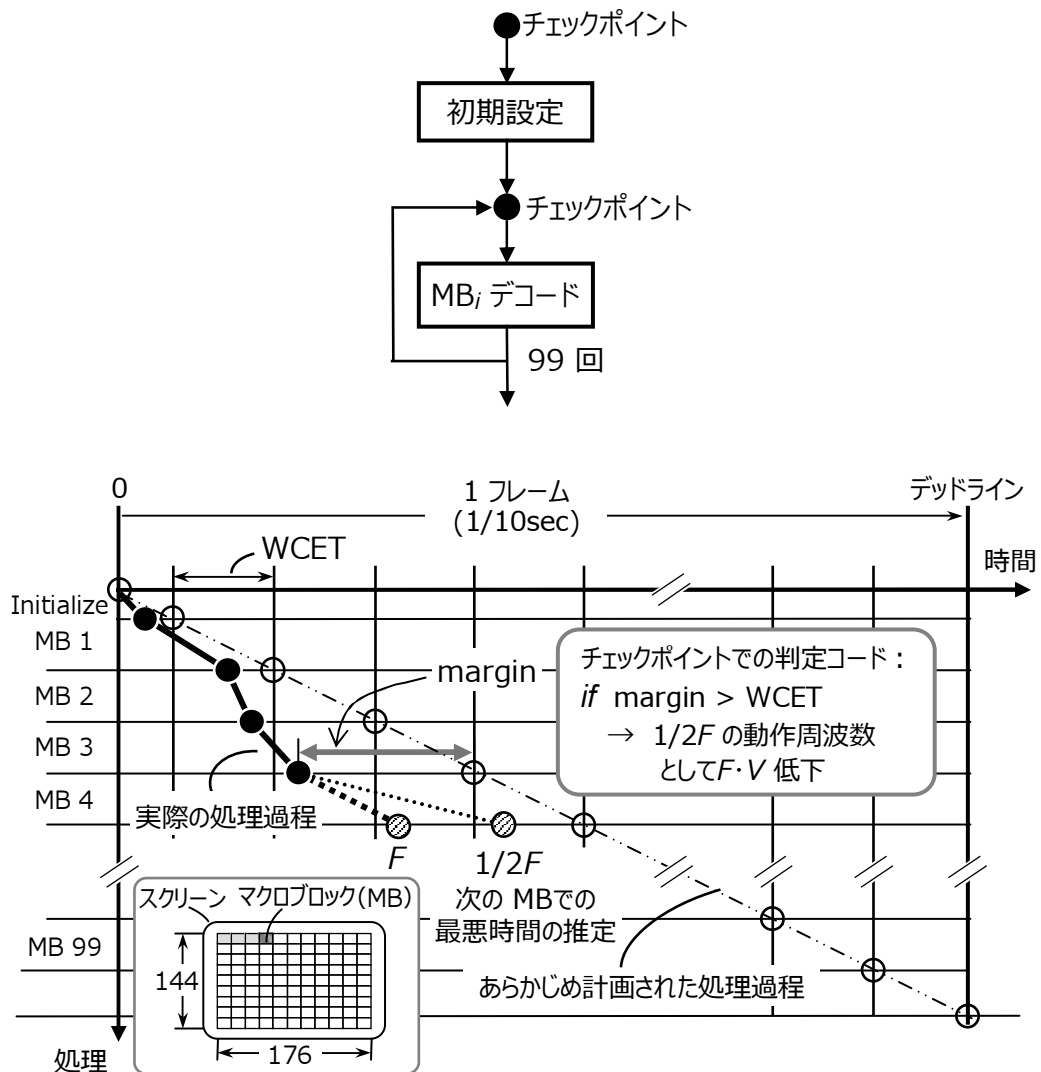


図 3.1 MPEG-4 デコーダの処理プロセス

まず、このデコーダの処理プログラムをいくつかの地点で区切る。この地点は原則的に当該プログラムの実行において必ず通過するものであり、この地点をチェックポイントとしてプログラムの進行状況を以下の手順で見積もることにする。隣接するチェックポイントに囲まれたプログラムの断片がセグメントになる。

MPEG 動画デコーダが、ある動画フレームの復号処理を開始したとする。復号処理はそのフレームの表示開始までに完了しなければならず、表示開始時刻がデッドライン時刻となる。復号を完了するためには初期設定処理およびマクロブロック復号

処理が必要である。マクロブロック (MB) は動画データを所定サイズ (16 × 16 画素) の小区画に分割したもので、MPEG (QCIF 画像) の場合 1 フレームは 99 個の MB からなる。図 3.1 のフローチャートに示すように、このプログラムの初期化処理時 (Initialize) および、99 個の各 MB_i の処理開始時のそれぞれにチェックポイントを置いて時刻を管理する。すなわち、各 MB_i の処理を前述のセグメントに対応させて FV 制御を適用するものである。図 3.1 の下側のチャートは、その処理の進捗状況を模式的に示している。

極めて処理量の多い動画データがデコーダに入力されたと想定すると、図中の二点鎖線のように、すべての MB_i に最悪の実行時間 Worst Case Execution Time (WCET) を費やして復号処理が実行される。この結果、最後の MB の復号を完了する時刻は図の対角線右端となるが、対象としている MPEG 動画デコーダにおいては、この時刻がデッドライン時刻を越えない、すなわち最悪時でも時間内に処理が終了するように設計されている。

多くの場合は処理量がそれほど大きくないため、処理は図中の太実線で示されるように WCET 以前に MB の復号が終了するよう進捗し、ある MB 復号処理ステップを開始する時点で、それまでの各 WCET の積算に対する処理時間の余裕を、図中の二点鎖線と実線との時間差 (margin) として求めることができる。

ここで、余裕が WCET よりも大きければ、次の処理ステップに通常の倍の時間を費やしてもフレーム全体として最悪ケースより遅れることはない。したがって、動作周波数を $1/2$ に落して実行しても安全であると判断できる。同様に、余裕が WCET の $2, 3, \dots, n$ 倍 よりも大きければ、動作周波数をそれぞれ $1/3, 1/4, \dots, 1/(n+1)$ に低下させて実行しても安全である。逆に余裕が少なければ、動作周波数を元に戻す必要がある。

この方法を用いると、処理の途中で最悪ケースとの比較評価が可能であり、余裕がなくなった時点で動作周波数を上げることにより、最悪の場合でもデッドライン時刻に間に合うような処理計画が立てられる。過去の処理時間の積算による次ステップでの余裕時間の評価とそれに基づく動作周波数の設定をフィードバック制御と呼ぶ。

3.2.2 MPEG-4 デコーダへの適用

上記のとおり，本設計ルールに従った FV 制御システムを試作して前章で挙げたプロセッサ SH-Mobile 1 に適用し，その動作をシミュレーションによって評価した．この方式によって，MPEG-4 デコード動作（図 3.2：図 2.2 の再掲）のような，負荷に基づく動作周波数・電源電圧の設定が可能となる．

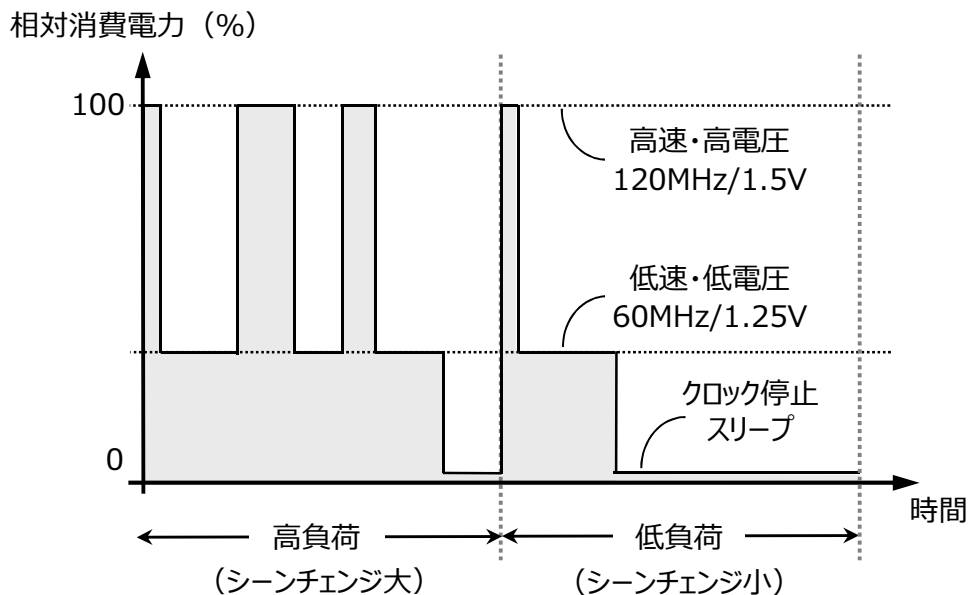


図 3.2 MPEG-4 デコード動作における FV 制御

2.6 節 (1) の結果により，120 MHz/1.5 V，60 MHz/1.25 V の二つの動作周波数・電源電圧の組を用いて，負荷に応じてデコード処理し，低周波数・低電圧での動作を行って更に実行時間が余る場合にはプロセッサをスリープモードとする．

制御対象のデコーダはすべてソフトウェアで実装されており，前述の余裕時間を計算・判定するプログラム上でのチェックポイント挿入点は，初期処理と各マクロブロックの処理時点となる．このチェックポイントによりプログラムを 100 個のセグメントに分割してそれらのセグメントごとに余裕時間を判定し，余裕がある場合に

Low 動作 (60 MHz/1.25 V) へ、余裕が無い場合に High 動作 (120 MHz/1.5 V) へと変化させることになる。

余裕時間は、動画表示のフレームレートを 10 fps として 1 フレーム全体の WCET が 100 ms で計算している。

このシステムに典型的な MPEG-4 動画ストリームデータ (平均負荷率 31.0%) を入力した場合の挙動を表 3.1 に示す。ここで平均負荷率とは、当該画像を正常に復号するためにプロセッサの最大性能の何 % が必要となるかを示すものである。表 3.1 では、例えばプロセッサ性能の 10~20% で復号できる動画フレームが全体 71 個中の 10 個であることを示している。この 10 個のフレームでは平均 1 回の動作周波数変更が起きており、実行時間のうち 8.76% が High 動作モード (120 MHz/1.5 V)、26.06% が Low 動作モード (60 MHz/1.25 V) で、残りの 65.18% がスリープモードで実行されたことを示している。各フレームで動作モードの占める割合にそのフレームの出現確率を乗じることにより、最終行の相対消費電力量* を求める。これにより消費電力量は FFV 制御が無い場合と比べて 27% に低減されると計算できる。ここでは Low 動作モードにおいて相対消費電力が High 動作モード時の 34.7% (周波数分: $1/2$, 電圧分: $(1.25/1.5)^2$)、スリープモード時の電力が 0 であるとして、この 71 フレーム分のシミュレーションデータから計算で電力量の削減率を得ている。

3.2.3 MP3 デコーダへの適用

さらに、MPEG-4 に対するものと同様同様の制御を MP3 デコーダに組み込んで、MP3 に対するこの FFV 制御の電力量削減効果を検証した。MPEG-4 ではマクロブロックの処理を単位としてセグメントに分割し制御したが、MP3 においても、そのようなプログラム構造に従って必ず通過する地点 (左右チャンネル処理およびグラニューール処理の開始地点) を選択し、8 個のセグメントに分割して動作周波数と電源電圧制御のためのチェックポイントを挿入している。MPEG-4 の場合と同様、120

* ここでの相対消費電力量は、全実行時間をスリープモードによる実行無しに High 動作で実行する場合を 100% とした比率である。スリープモードでの実行を考慮した場合、消費電力量は 80% に削減されると計算される。

表 3.1 シミュレーションによる電力評価

相対負荷 比率	フレーム数	FV 変更 回数	High モード (%)	Low モード (%)	スリープ (%)
0~10%	0	0	0	0	0
~20%	10	1.00	8.76	26.06	65.18
~30%	31	1.45	9.40	41.10	49.50
~40%	22	1.36	13.24	50.42	36.34
~50%	7	1.00	16.51	56.30	27.19
~60%	0	0	0	0	0
~70%	0	0	0	0	0
~80%	1	1.00	71.50	14.0	14.50
~90%	0	0	0	0	0
~100%	0	0	0	0	0
相対消費電力量 (%)			12.07	14.93	0

MHz/1.5 V, 60 MHz/1.25 V の動作周波数・電源電圧の組を用いて周波数・電圧の切換えを行い、さらに実行時間が余る場合にプロセッサをスリープモードとしている。

チェックポイントでの余裕時間計算と判定も MPEG-4 のものと同様である。MP3 の負荷率は 20% 程度と低く、またデータ処理の変動量は MPEG-4 に比べると小さい。試行データによると最大負荷と最小負荷とでは 3 : 2 程度の差を生じている。MP3 の場合、1 フレーム全体の WCET は音声出力の 1 フレーム分を考慮し、26 ms と設定している。

MP3 での消費電力量の評価は、次節に示す実験時のトレースデータを採取し、それによって計算で求めたものである。その 199 音声出力フレームによる評価データを表 3.2 に示す。大部分 (86%) のフレームでは 1 回の動作周波数および電源電圧変更である。すなわち、フレーム実行の最初だけ High 動作で、ほとんどが Low 動作あるいはスリープとなっているものと考えられる。

表 3.2 トレースデータによる電力評価

出現確率 (%)	<i>FV</i> 変更 回数	High モード (%)	Low モード (%)	スリープ (%)
85.9	1	2.45	34.3	63.25
1.0	2	14.36	85.7	0
13.1	3	7.35	44.1	48.55

このデータから、それぞれの動作モードでの消費電力が上述の MPEG-4 の場合と同じであるとして、MP3 において消費電力量は、FFV 制御が無い場合と比べて 15.7% に低減されると計算できる。

3.3 実験結果

上記の FFV 制御を実装するターゲットシステムとして、図 3.3 のシステム構成で (株) ルネサステクノロジ社製 SH-Mobile 1 [12, 20] 組込みのシステムボード ((株) 日立超 LSI システムズ社製 SolutionEngine [21]) を用い、これに図中 (1) の電源電圧制御回路を付加して二つの電源電圧に切換え可能とした。SH-Mobile 1 には可変電源電圧機構は備わっていないので、上記システムボードに装備されている電源レギュレータにも DC/DC コンバータの機能は備わっていない。

そこで可変電源電圧を実現するようシステムボードに変更を加え、電源レギュレータへの帰還電圧の分圧比を変えるアナログ回路を追加装着して、SH-Mobile 1 への電源電圧供給を切換える方法を採用した。

動作周波数の変更は、PLL 出力の分周比率を変更することで行う。設定する動作周波数は、前章での考察により最大周波数とその 1/2 (120 MHz/60 MHz)、電源電圧はそれらの速度を得る最小電圧 (1.5 V/1.25 V) である。

動作周波数と電源電圧を協調して設定するためのシステムコールを SetFV.Low (最大動作周波数の 1/2 に設定する) と SetFV.High (最大動作周波数に設定する) とし

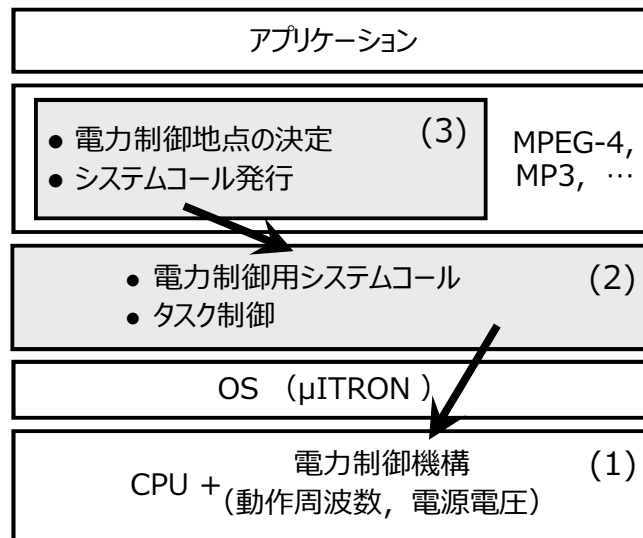


図 3.3 システム構成

て図 3.3 (2) のライブラリで実装し、OS (μ ITRON) 上のアプリケーションにおける音声つき動画再生のミドルウェアとして、MPEG-4 ビデオデコーダと MP3 デコーダを同期して並行に稼働させている。(詳細は付録に述べる。) 図 3.3 の (3) 「電力制御地点」が図 3.1 でのチェックポイントに対応している。

プロセッサがアイドル状態になる時にスリープモードを用い、プロセッサへのクロックの停止を行うため、タスク実行で最低優先度を持つ「スリープ移行タスク」を置き、デッドラインよりも早く処理が終わってアイドル状態になると、このタスクが起動してプロセッサをスリープモードに移行するようにする。MPEG-4 ビデオデコーダタスク、MP3 デコーダタスク各々が独立に自身の処理経過を把握し、この二つのタスク内部でそれぞれ FFV 制御を行っている。

表 3.3 (a) に、実験において MPEG-4 デコーダで処理された 188 フレームでの動作周波数 F と電源電圧 V の変更回数を示す。Sub-QCIF (128 × 96 画素) サイズ、10 fps の動画再生においては、74% のフレームにおいて 1 回の FV 変更であり、ほとんど Low モードの実行となっていることがわかる。10 回以上 FV 変更を行うフレームがあるが、これは負荷率が 50% 程度の場合に、動作周波数を低下させて実行

し、余裕時間がなくなりフィードバックによって動作周波数を上昇させるプロセスが繰り返されることを示している。

表 3.3 (b) は MP3 デコーダの 199 フレームでの FV 変更回数である。この表から分かるように、MP3 デコーダにおいては大部分のフレームが 1 回の FV 変更でよい。さらに、その変更は 2 セグメント目のものである。実際、この MP3 デコーダはすべて 60 MHz で稼動させても大丈夫なものである。(後述する方式を用いて、MP3 デコーダではすべて 60MHz での動作ができています。)

表 3.3 フレーム内での FV 変更回数

(a) MPEG-4

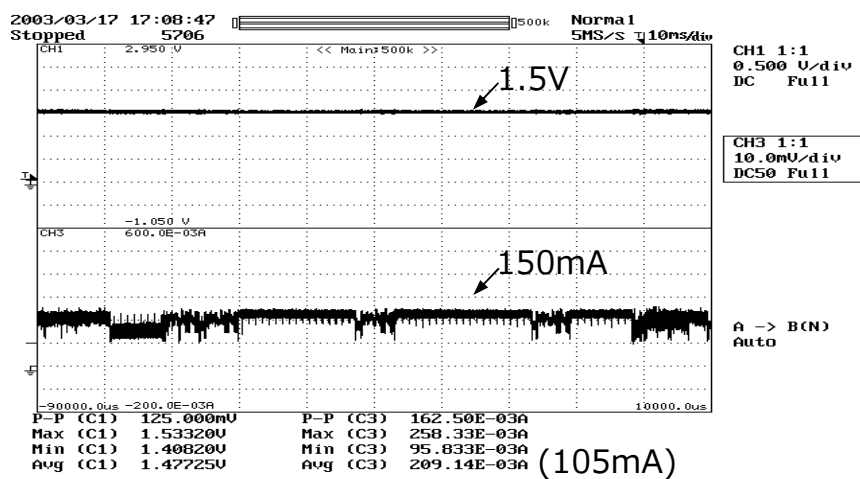
FV 変更回数	1	3	5	7	9	11	13	15	...	27
フレーム数	139	27	12	2	2	3	1	1	0	1

(b) MP3

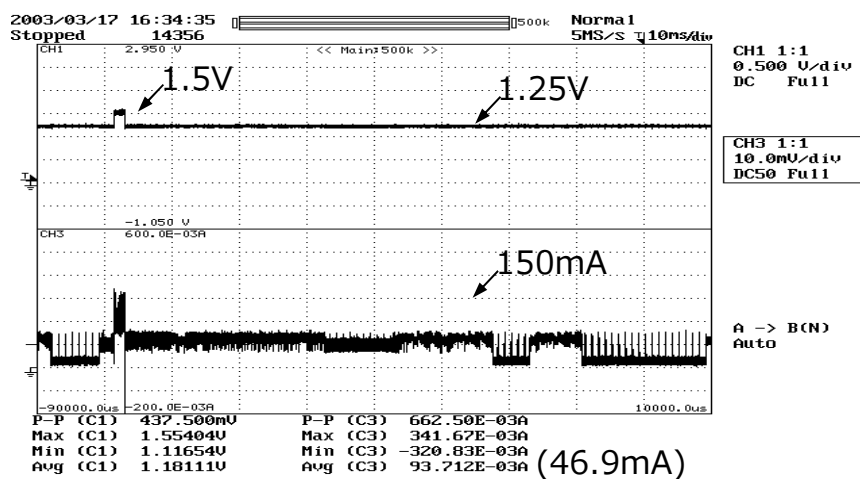
FV 変更回数	1	2	3
フレーム数	171	2	26

この実験システムにおいて、動画・音声再生時の電圧と電流の波形は図 3.4 に示すとおりである。図中のそれぞれのグラフで上側の波形が電源電圧を、下側が消費電流を示している。FFV 制御をしないもの (a) では平均電圧 1.48V、平均消費電流 104.6mA が得られ、平均消費電力は 230mW となる。これと比較して FFV 制御を行ったもの (b) では平均電圧 1.18V、平均消費電流 46.9mA が得られ、平均消費電力は 65mW となる。すなわち、FFV 制御により消費電力量を 28% に削減できている。同図 (b) に示されるように、大部分は Low モード (1.25V) での動作であり、High モード (1.5V) で動作する区間は非常に短いことが分かる。なお、消費電流が 0mA まで落ちている部分は、デコード処理がなくなりスリープモードに入っていることを

示すものである。†



(a) FFV制御なしの場合



(b) FFV制御ありの場合

図 3.4 FFV 制御実行時の電圧・電流波形

† スリープモードの部分に見られる等間隔のヒゲ状の波形は、OS によるタイマ割込みの処理を示すものである。

3.4 Cool-Start 方式による低電力効果の向上

ここまでの議論では、MPEG-4 や MP3 デコードの各フレームの処理において開始時点は High 状態の動作周波数・電源電圧であるとしてきたが、プロセッサの動作周波数が十分に（以下で示すように数～10 数 %）高い場合には、これを Low 状態もしくは初期状態で $1 \cdot \text{WCET}$ 分の余裕があるとして開始してもよい。（これを Cool-Start と呼ぶ。）

各セグメントが同じ時間で実行されるものとする、本 FFV 制御の場合 MPEG-4 デコーダで 1%、MP3 デコーダで 13% 程度余裕を見た設計ができていれば、2 セグメント目以降のフィードバック制御により WCET 以内での実行が保証される。本実験でもこの制御を行って Low 状態で開始し、MP3 デコーダは実質的にすべて Low 動作とできている。Low 動作時の消費電力は High 動作時の約 $1/3$ であるので、MP3 デコーダの場合では Cool-Start なしの制御に比べ約 6% 消費電力量が減少する。

表 3.3 を得たデータによると、MPEG-4 デコーダでは 1 回目の *FV* 変更は平均 19 セグメントで発生している。Cool-Start によりこれが 1 セグメント分割減できたとすると、1 回だけの *FV* 変更が支配的だとして 1% 強程度消費電力量の減少を見込める。Cool-Start は MP3 デコーダに適用したように分割セグメント数が小さい場合に、消費電力低減効果をあげる有効な方式である。

3.5 まとめ

本章では *FV* 制御システムを実際のシステム上で稼動するアプリケーションへの適用にあたって、動作周波数（と電源電圧）を変更するソフトウェア方式をフィードバック型 *FV* (FFV) 制御方式として示した。

前章で得た用意すべき動作周波数群の決定ルールに基づく *FV* 制御をフィードバックによる制御方式によって MPEG-4 デコーダと MP3 デコーダに適用して稼動させ、MPEG-4 デコーダ単独での消費電力量が 27% に、MP3 デコーダ単独での消費電力量が 15.7% に、それぞれ低減できることをシミュレーションによって評価した。また、

MPEG-4 デコーダと MP3 デコーダ両者が同期して並行動作するプログラムで、それらによる動画・音声再生プログラム全体の消費電力量が 28% に低減できることを実機実行での測定によって確認した。これらにより、本 FFV 制御方式が実用上有効であることが示せた。

従来の DVFS で提案されている動作周波数・電源電圧の変更スケジュールは、例えば MPEG の場合、1 フレームごとに変更するなど、ほぼタスクを単位としてなされるものであって、OS から制御されるものになる。また、処理の負荷は静的に（まえて）設定されるか、もしくは過去の履歴に基づいて想定される。これに対して、FFV 制御ではタスク内をさらにプログラムセグメントに分割し、アプリケーション内部から動作周波数と電源電圧を設定するので、より詳細な単位で制御可能であり、以降の進捗を予測して制御でき、負荷の変動にも追従できる点が特長である。

現在の最先端のマイクロプロセッサでは電源電圧が限界まで下がってきて、高電圧と低電圧との差が小さかったり低電圧に設定できないなどの状況が生じ、FV 制御そのままでは本章で適用したような電源電圧降下による大きな消費電力削減を得ることが困難になってくる。しかし、負荷に応じてプロセッサの速度を制御する方式自体は有効に機能する。例えば、閾値電圧の制御が可能なプロセッサにおいて、電源電圧を下げられなくなった場合に、FFV 制御と同様のアルゴリズムを適用して負荷の軽い区間で動作周波数を下げるとともに閾値電圧を高くする。それによってリーク電流を減少させて消費電力を低減することができる [13, 14]。さらに、電源電圧と閾値電圧を同時制御する方法も有効である [15]。

今後、リーク電流の問題がますます大きくなり、5 ページの消費電力の式 1.1 の第三項 P_l も主要となるが、これに対しても FV 制御が有効となると考えられる。

なお、動作周波数を下げるだけでは、消費電力は下げられるが、それに見合っただけで動作時間が増加するため、消費電力量（エネルギー）の観点では FV 制御の効果は得られない。式 1.1 では第二項 P_s で示すように貫通電流に依存する部分があり、これは動作周波数に比例するものなので、いくらかの消費電力量の削減効果を見込めるはず

であるが、上述の動画・音声再生の実験中に得たデータでは動作周波数のみを下げた場合、その部分での電流値は減少するものの制御対象区間での平均電流は、動作周波数を変化させない場合とほとんど変わらないか、多少（0.9%）の減少が見られるという程度であった。

第 3 章の付録 FFV 制御評価用組込みシステムと評価パラメータ

試作した可変電源電圧対応のシステムボード上で、FFV 制御の動画・音声動機再生マルチタスクプログラム「MP4 Player」を実装した実験について補足する。

消費電力低減効果を評価するための条件は、表 3.4 に記載のパラメータである。この評価条件は現在の携帯電話アプリケーションを想定した仕様および品質になるように設定しているものである。

制御対象のマルチタスク MP4 Player プログラムのタスク構成は図 3.5 である。MP4 File DeMux Task が入力ファイルをビデオとオーディオのビットストリームデータに分解し、それぞれが MPEG-4 Video Decoder Task と MP3 Decoder Task に渡されて復号され、フレームデータとして出力される。

タスクの優先順位は図の上方ほど高く、本実験ではオーディオ関連をビデオ関連より高く設定している。本文中で述べた「スリープ移行タスク」は図中の Power Save Task であり、最低優先度のタスクとして設定されている。

SH-Mobile 1 (SolutionEngine) では動作周波数を変更することは可能であるが、電源電圧を可変にする機構は備えていない。そこで、図 3.6 のように SH-Mobile 1 の CPU コア電源電圧端子およびデカップリングコンデンサ群の前段に可変電源電圧回路を追加実装した。

本文中に記載した動作周波数および電源電圧を制御するシステムコールによって、図 3.6 内の番号に対応して以下の操作が行われる。

表 3.4 評価用システムのパラメータ

使用ハードウェアボード		SH SolutionEngine	
制御対象アプリケーションプロセッサ		SH-Mobile 1	
電力モード		High	Low
電源電圧制御範囲		1.5V	1.25V
動作周波数制御範囲		120MHz	60MHz
電圧安定のための待ち時間		100 μ sec	
制御対象アプリケーション		設定項目	設定値
AV DeMux MP4 ファイル形式 (ISO/IEC 14496-1)	Video Decoder ・ MPEG-4/ ビジュアル シンプル プロファイル (ISO/IEC 14496-2)	ビットレート	32kbps
		フレームレート	10fps
		画面サイズ	Sub-QCIF (128 × 96)pixels
	Audio Decoder ・ MPEG-1/ オーディオ (ISO/IEC 11172-3) - Layer III ・ MPEG-2/ オーディオ (ISO/IEC 13818-3) - Layer III	ビットレート	32kbps
		サンプリング周波数	22.05kHz
		チャンネル	2ch.
合計		ビットレート	64kbps

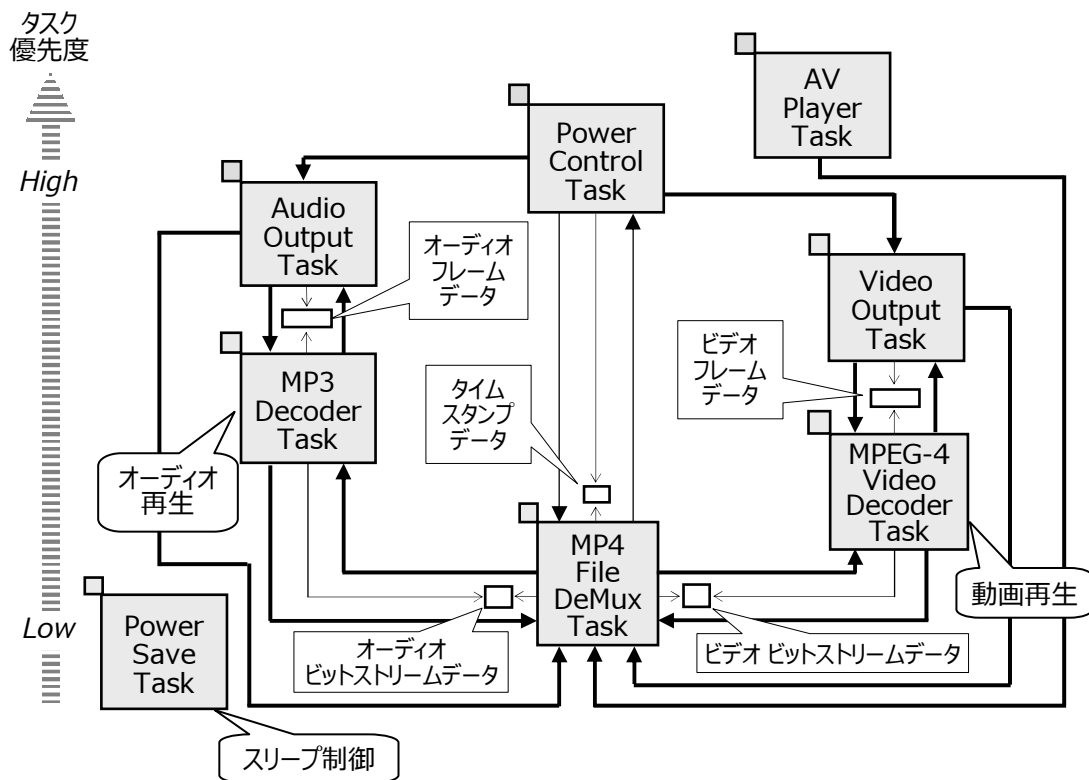


図 3.5 実験システムのタスク構成

- (1) SH-Mobile 1 の制御レジスタを用いて PLL 回路とアプリケーションプロセッサ (SH-Mobile 1) に印加する電源電圧を変更した場合の安定時間 (プロセッサ待機時間) をウォッチドッグタイマで設定する。
- (2) GPIO から可変電源電圧回路に対してアプリケーションプロセッサに印加する電源電圧を「低」または「高」に設定するよう on/off 信号を出力し、分圧比を切替える。
- (3) 制御レジスタによって動作周波数の分周比を「低」または「高」に設定する。

安定時間を考慮した電源電圧-動作周波数の切替え時間は $100\mu\text{sec}$ である。同じシーン (処理経過時刻) でトリガをかけて得られた 0.1sec 間の評価結果が、本文の図

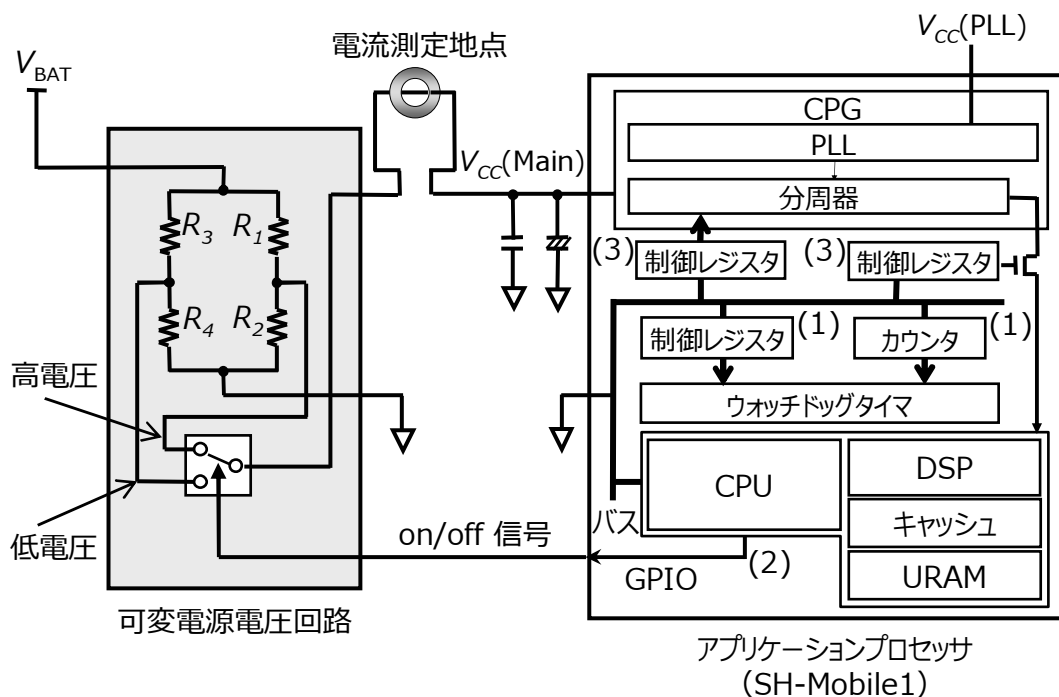


図 3.6 可変電源電圧回路

3.4 (a) (b) に示す二つの波形である。

評価用システムの外観は図 3.7 のとおりであり，向かって右側が可変電源電圧対応を施したシステムボード，左が波形測定のためのデジタルオシロスコープである。

なお，公開されている SH-Mobile 1 の仕様 [20] では，本章で低電圧として用いた電源電圧 1.25V は保証されていない。本来，このような電源電圧の変更を想定したプロセッサではなく，製品レベルで複数の電圧での動作を保証することがコストなどの点で合わないためであると考えられる。[12] による SH-Mobile 1 の Shmoo プロットを図 3.8 に示すが，同図の丸印で分かるように 1.25V/60MHz は十分な余裕をもった動作領域であり，本章の実験においても問題なく動作している。

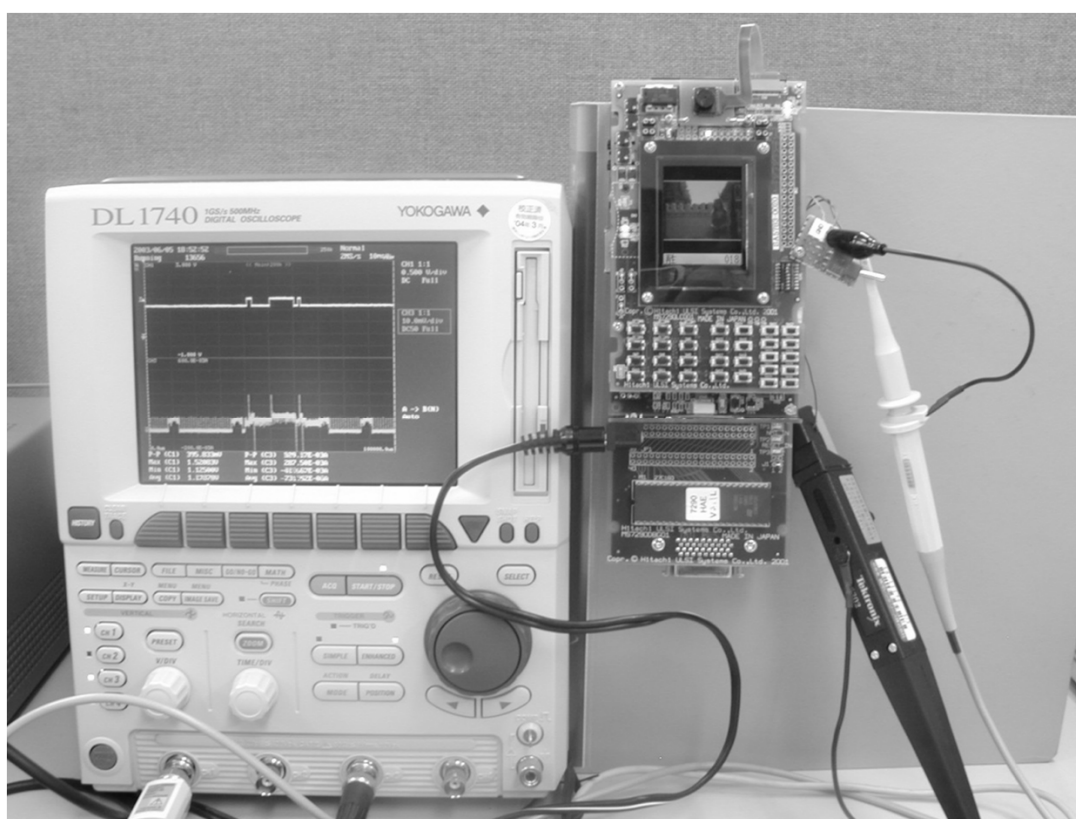


図 3.7 評価用システムの外観

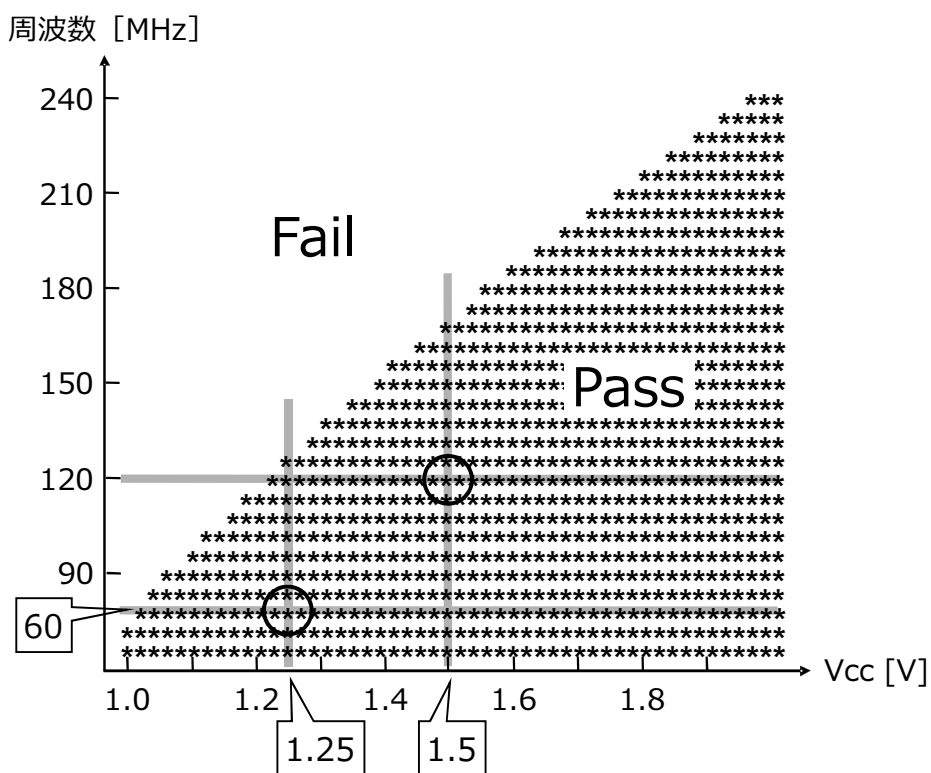


図 3.8 SH-Mobile 1 の Shmoo プロット ([12] による)

第 3 章の参考文献

- [1] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos: “Data Driven Signal Processing: An Approach for Energy Efficient Computing”, *Proceedings of 1996 International Symposium on Low Power Electronics and Design Conference*, pp.347–352, Aug. 1996.
- [2] T. Ishihara and H. Yasuura: “Voltage Scheduling Problem for Dynamically Variable Voltage Processors”, *Proceedings of 1998 International Symposium on Low Power Electronics and Design Conference*, pp. 197–202, Aug. 1998.
- [3] S. Lee and T. Sakurai:
“Run-time Voltage Hopping for Low-power Real-time Systems”, *Proceedings of 37th Design Automation Conference*, pp.806–809, Jun. 2000.
- [4] 十山 圭介, 三坂 智, 相坂 一夫, 在塚 俊之, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康: “CPU 消費電力削減のための周波数－電圧協調型電力制御方式の設計ルールとフィードバック予測方式による適用”, 電子情報通信学会和文論文誌 D-I 分冊, Vol.J87-D-I No. 4, pp. 452–461 (2004-4)
- [5] K. Aisaka, T. Aritsuka, S. Misaka, K. Toyama, K. Uchiyama, K. Ishibashi, H. Kawaguchi, and T. Sakurai: “Design Rule for Frequency-Voltage Cooperative Power Control and Its Application to an MPEG-4 Decoder”, *IEEE Solid-State Circuits Society 2002 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 216–217, Jun. 2002.
- [6] 相坂 一夫, 在塚 俊之, 三坂 智, 十山 圭介, 内山 邦男, 石橋 孝一郎, 川口 博,

- 桜井 貴康：“周波数-電圧協調型電力制御における使用周波数群決定方法の提案 MPEG-4 デコーダによる検証”，電子情報通信学会技術研究報告 ICD2002-37, Vol. 102, No. 234, pp. 13–16 (2002-7)
- [7] 相坂 一夫, 在塚 俊之, 三坂 智, 十山 圭介, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康：“周波数-電圧協調型電力制御におけるアルゴリズムとデザインルール”，電子情報通信学会技術研究報告 ICD2002-72, Vol. 102, No. 274, pp. 49–52 (2002-8)
- [8] S. Gochman, R. Ronen, I. Anati, A. Berkovits, T. Kurts, A. Naveh, A. Saeed, Z. Sperber, and R. C. Valentine: “The Intel® Pentium™ M Processor: Microarchitecture and Performance”, *Intel Technology Journal*, Vol. 7, Issue 2, pp. 21–36, May 2003.
- [9] M. Fleischmann: “LongRun™ Power Management: Dynamic Power Management for Crusoe™ Processors”, Transmeta Corporation, Jan. 2001.
- [10] 川上 健太郎, 大平 英雄, 深山 政幸, 吉本 雅彦：“フィードフォワード動的電圧制御による MPEG4 低消費電力化アルゴリズム”，電子情報通信学会技術研究報告 ICD2002-57, Vol. 102, No. 273, pp.67–72 (2002-8)
- [11] 中本 幸一, 辻野 嘉宏, 都倉 信樹：“実行順序制約を有するタスク群における2次電池の最長利用のためのスケジューリングアルゴリズム”，電子情報通信学会技術研究報告 CPSY99-128, Vol. 99, No. 726, pp.35–42 (2000-3)
- [12] T. Yamada, M. Ishikawa, Y. Ogata, T. Tsunoda, T. Irita, S. Tamaki, K. Nishiyama, T. Kamei, K. Tatzawa, F. Arakawa, T. Nakazawa, T. Hattori, and K. Uchiyama: “A 133MHz 170mW 10 μ A Standby Application Processor for 3G Cellular Phones”, *IEEE International Solid-State Circuits Conference 2002 Digest of Technical Papers*, pp. 370–371, Feb. 2002.
- [13] 野瀬 浩一, 平林 雅之, 川口 博, 李 誠洙, 桜井 貴康：“閾値ホッピング (V_{TH} -hopping) 手法を用いた低電圧・低消費電力プロセッサ”，電子情報通信学会技術

- 研究報告 ICD2001-33, Vol. 101, No. 85, pp. 67–73 (2001-5)
- [14] 桜井 貴康：“リーク電流はこう抑える (7) V_{th} 制御と V_{dd} 制御の併用でリーク電流削減効果を最大化”，日経エレクトロニクス, 2005.4.25 (no. 898), pp.121–126 (2005-4)
- [15] 宮崎 祐行, 小野 豪一, 清水 健央, 菅野 雄介, 河原 尊之：“超低電力システム LSI 向け 0.1V-CMOS 集積回路:低電力回路技術の課題と対策”，電子情報通信学会技術研究報告 ICD2002-60, Vol. 102, No. 273, pp. 85–90 (2002-8)
- [16] 三坂 智, 十山 圭介, 在塚 俊之, 相坂 一夫, 内山 邦男, 川口 博, 桜井 貴康：“マルチタスク実装マルチメディアに対する周波数-電源電圧協調型電力制御”，電子情報通信学会技術研究報告 ICD2003-40, Vol. 103, No. 216, pp.7–12 (2003-7)
- [17] D. Shin and J. Kim: “A Profile-Based Energy-Efficient Intr-Task Voltage Scheduling Algorithm for Hard Real-time Applications”, *Proceedings of International Symposium on Low Power Electronics and Design 2001*, pp.271–274, Aug. 2001.
- [18] CH. Hsu, U. Kremer, and M. Hsiao:
“Compiler-Directed Dynamic Voltage/Frequency Scheduling for Energy Reduction in Microprocessors”,
Proceedings of International Symposium on Low Power Electronics and Design 2001, pp.275–278, Aug. 2001.
- [19] A. Azevedo, I. Issenin, R. Cornea, R. Gupta, N. Dutt, A. Veidenbaum, and A. Nicolau: “Profile-based Dynamic Voltage Scheduling using Program Checkpoints”, *Proceedings of Conference on Design, Automation and Test in Europe 2002*, pp.168–175, Mar. 2002.
- [20] SH-Mobile1 (SH7290), 株式会社 ルネサス テクノロジ,
<http://japan.renesas.com/>
- [21] SH SolutionEngine (MSC7290P01), 株式会社 日立超 LSI システムズ,
<http://www.hitachi-ul.co.jp/SH-SE/>

第 4 章

マルチプロセッサにおける動作時消費
電力量の低減方式

4.1 はじめに

処理性能や信頼性，また，ソフトウェア開発効率の向上を狙いとしてマルチプロセッサへの対応が広がってきており，組込みシステムの分野においてもマルチプロセッサ構成が採り入れられるようになってきている [1, 4, 23–26]．パーソナルコンピュータやワークステーションの分野では，マルチプロセッサ化は主として並列化によって実行速度におけるピーク性能を追求するものであるが，組込みシステムにおいては，目標性能に対してプロセッサ 1 台当りに要求される演算性能を下げることによって消費電力を低減するという点での期待が大きい．

本章ではマルチプロセッサにおけるプロセッサの動作時消費電力量（エネルギー）を低減する方式を提案する．

近年，命令レベルよりも粒度の大きいスレッドレベルの並列性を利用して性能向上と消費電力削減を目指した研究開発が行われている．マルチプロセッサにおいては，このような並列処理に加えて，負荷に応じた制御を行って要素プロセッサに要求される動作周波数を低下させ，アプリケーション実行に最適な動作周波数と電源電圧を設定することによりプロセッサ動作時の消費電力量を低減することができる．前章までで述べたように，フィードバック型の周波数－電圧協調型電力制御（FFV 制御）によって動作周波数・電源電圧制御を適切に行うと，動作周波数と電源電圧の 2 乗とが電力低下に寄与するため，マルチプロセッサにおいてもプログラム動作時のプロセッサ消費電力量を大幅に低減できると期待される．

すなわち，マルチプロセッサにおける消費電力量低減の基本的な考えは，図 4.1 に示すように，プログラムを並列スレッドに分割して実行し，各プロセッサの負荷を下げて動作周波数および電源電圧を下げる（図中の (1))．さらに各要素プロセッサの中で，シングルプロセッサに対して行ったのと同じ FFV 制御を行う（図中の (2))．以上の二段階で制御するものとなる．

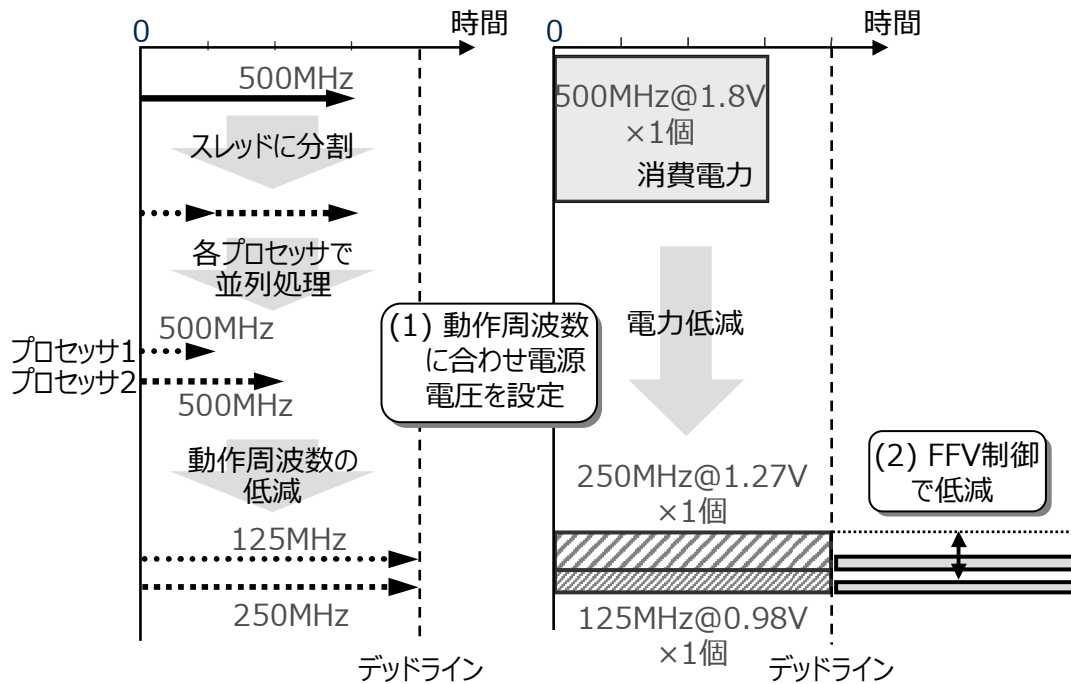


図 4.1 マルチプロセッサでの消費電力量低減方式の概要

本章では、FFV 制御方式を並列処理の要素プロセッサに割り当てられる動作周波数・電源電圧に適用することにより、柔軟で効果の高い動作時消費電力量の低減方式を実現する。第 3 章でも述べたが、FFV 制御によって消費電力量の削減効果を上げるためには対象プログラムがどれだけの負荷で動作しているかを正確に見積もることが必要となる。このため、プログラムを静的に解析する方式と動的に判定する方式がある。静的に解析する方式は実行時のオーバヘッドはないが、実行データに依存するものを正確に見積もることが困難である。一方、動的に判定する方式では実行状態は反映させられるものの、判定処理に要するオーバヘッドが懸念される。また、このときの負荷見積りの基準としては、同期地点までのプロセッサの実行時間が主である。

マルチコアの並列実行においてはこの見積りがより困難になると考えられ、前章で述べたシングルプロセッサでプログラム内にチェックポイントを挿入し、デッドラインまでの残り実行時間を予測することによって負荷を動的に計測する方式を本章で拡張する。処理の進捗と負荷を見積もるためのチェックポイントの設定、および消費電

力量を最小にする並列プロセッサ数の決定をマルチコアでの FFV 制御の課題とし、それらを解決する方式を提案する [5].

本章で提案した方式で、シミュレーションにより4台、8台並列時に60%から70%程度の消費電力量の低減効果を得ている.

4.2 マルチプロセッサアーキテクチャと動作時の消費電力制御

4.2.1 システムの構成

著者らの研究室ではアーキテクチャ面から高い動作時消費電力の低減効果を見込めるオンチップ・マルチプロセッサ構成を提案している [1-3]. そのアーキテクチャは、要素プロセッサ単位で動作周波数と電源電圧の制御を行って動作時の消費電力低減を見込めるマルチコア構成で、図4.2に示すように、一つのマスタプロセッサと複数のスレーブプロセッサをネットワークで結合し、それらのすべてを一つのチップ上で構成する. プロセッサ間の通信は、周波数・電圧管理ユニット: VMU (VLIW processors Management Unit) と呼ぶモジュールと共有メモリを用いて行う.

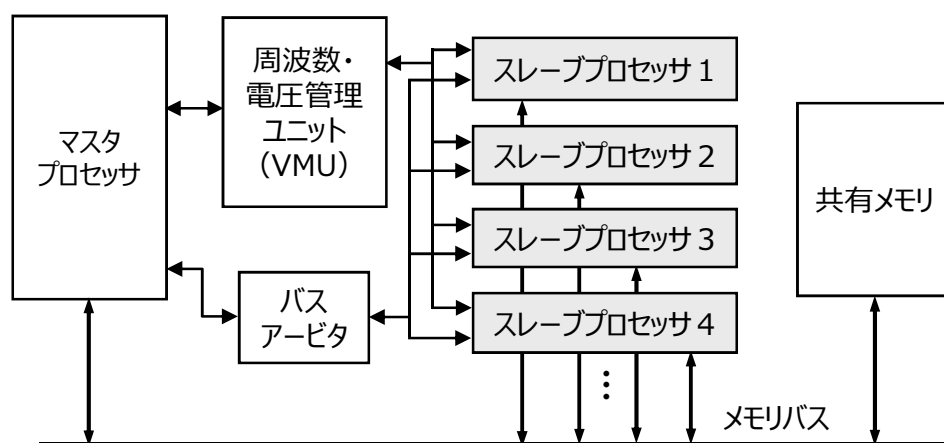


図4.2 マルチプロセッサ構成

マスタプロセッサには市場に流通している RISC プロセッサを用い、OS やデバイスドライバなどプロセッサアーキテクチャ依存のソフトウェア資源を有効活用することができる。他方、スレーブプロセッサにはマルチメディア処理などターゲットとなるアプリケーションに対して高い性能を示すプロセッサとして、VLIW アーキテクチャを用いた汎用プロセッサや DSP、あるいは SIMD 型の演算ユニットを備えた汎用プロセッサなどを用いることができる。これらのプロセッサは単体でプログラムの持つ命令レベル並列性を利用することで性能の向上を果たす。本アーキテクチャでは、このような命令レベル並列性に加えプロセッサを複数並列に動作させることで、より粒度の大きいスレッドレベル並列性を利用し計算性能の向上を図る。

プロセッサ間の制御は割り込み信号を用いて行う。マスタプロセッサ上で動作するスレーブプロセッサ管理ソフトウェアが VMU を操作することにより、各スレーブプロセッサに対して割り込み信号を出す。割り込みを受けたスレーブプロセッサでは例外ハンドラにより自身の管理ソフトを起動し、適切なタスクを実行する。スレーブプロセッサがマスタプロセッサに対して通知する場合も、VMU を介してマスタプロセッサに割り込む。さらに本アーキテクチャでは、複数のスレーブプロセッサに同時に割り込みをかける機構を実現してスレッド生成時のオーバヘッドを削減する。また、プロセッサごとに FFV 制御を可能にする機構として、VMU は図 4.3 に示す周波数・電圧管理ユニットを持ち、各スレーブプロセッサそれぞれが独立して動的に動作周波数と電源電圧を制御する。

周波数ユニットおよび電圧ユニットは、あらかじめ設定された複数の動作周波数および電源電圧の値を持ち、電源電圧を制御する可変電源回路は供給される電源ライン（電圧 V_0 ）を入力として三つの電源電圧（ V_1 , V_2 , V_3 ）を発生させる多電源回路と $V_0 \sim V_4$ のいずれかを選択するスイッチ、それを制御するデコーダから構成される。

4.2.2 プログラム動作時のプロセッサ消費電力量

プロセッサに要求される実行性能はアプリケーションの特性によって様々であり、常に最高速度が必要とされるわけではない。そこで、提案するマルチプロセッサにおいてアプリケーションをスレッドに分割して並列処理し、それぞれの実行に最適な動

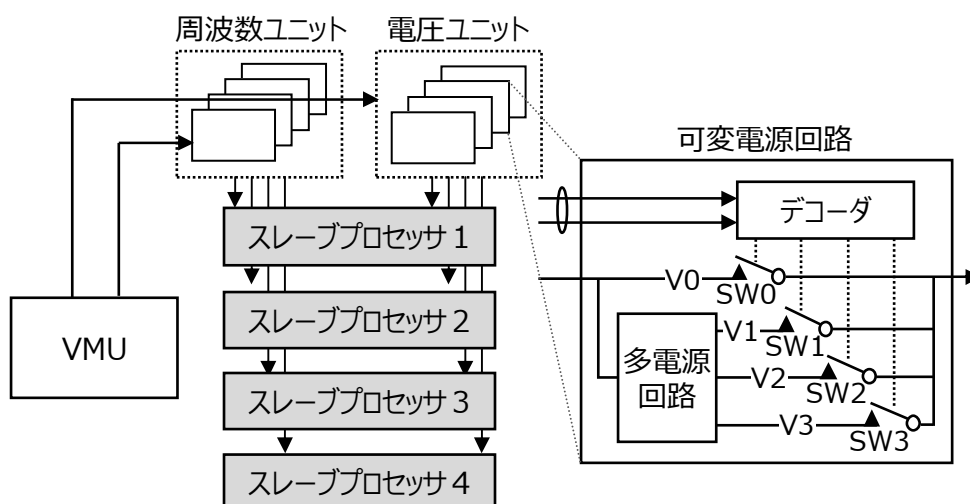


図 4.3 VMU による動作周波数・電源電圧の制御

作周波数を設定することにより、プログラム動作時の各プロセッサの消費電力量を低減することができる。制御されるスレーブプロセッサの消費電力特性として以下を想定した。

スレーブプロセッサの消費電力特性 (1)：初期設計

[1] で提案するマルチプロセッサシステムにおいて動作周波数と電源電圧の関係は、最高動作周波数 F_{max} がインバータの遅延時間に反比例すると仮定し、

- プロセッサ最高動作周波数 $F_{max} = 500\text{MHz}$ @ 電源電圧 $V_{dd} = 1.8\text{V}$
- プロセッサ 1 台の消費電力は 1W @ 500MHz

として、図 4.4 の通りとなっている。

このプロセッサでは、動作周波数の低下にしたがって電源電圧をほぼ線形に低下させることが可能であり、電圧の 2 乗の要因を大きく採れるため、高い電力削減効果が期待できる。このプロセッサにより、提案方式での電力量削減の傾向について予備的な評価を行った。

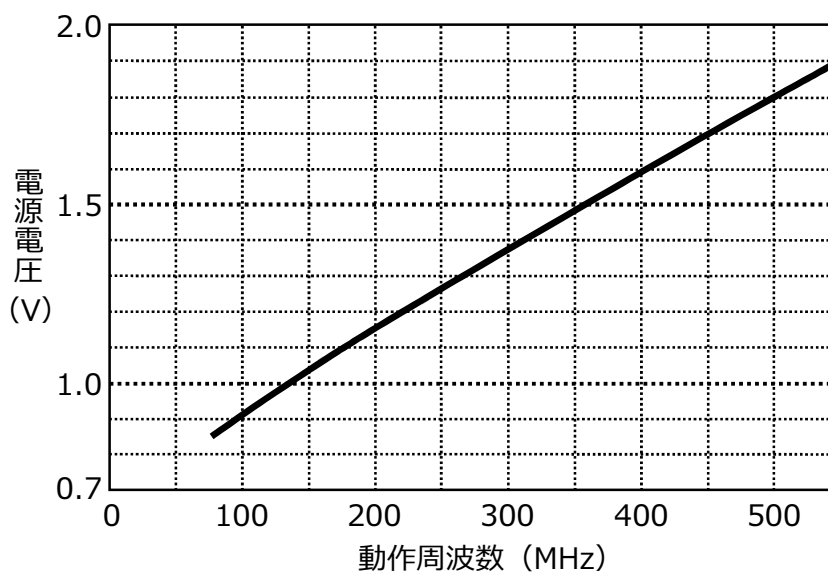


図 4.4 動作周波数 - 電源電圧の関係

スレーブプロセッサの消費電力特性 (2) : 実プロセッサの特性を反映した設計

提案アーキテクチャではスレーブプロセッサとして、既存の RISC プロセッサ SH [18]などを想定している。前章の評価でも用いたこのプロセッサ (SH-Mobile 1) の特性を基に、動作周波数や電源電圧を設定することにより、実際に即した評価を行うことができる。

4.2.3 並列化による「性能一定」下での消費電力量の低減

消費電力量の低減は、並列実行によって各スレーブプロセッサの実行速度を低下させ、それにとまって電源電圧を降下させることで実現できる。対象プログラムが全体としてデッドラインまでに処理を終えるよう処理時間を一定 (性能一定) として、並列化によって向上した計算性能を動作周波数および電源電圧の低減に利用するものである。並列化にとまない、提案アーキテクチャでは主にプロセッサ間の同期処理や並列化に伴う処理データの増加によるバスの待ちでオーバヘッドが発生するが、高い並列性が得られるプログラムでは、スレッド分割して性能一定になるように実行すれ

ば、プロセッサ台数の増加によっても処理するデータ量は変わらずにバスアクセスはほぼ一定にできる。本プロセッサが主な処理対象とする MPEG-4 などのようなマルチメディアプログラムは、このように命令レベル並列性とスレッドレベル並列性がともに潤沢に得られるアプリケーションであることが知られており、バスアクセスへの影響も小さいと考えられる。

4.3 マルチプロセッサにおける動作時のプロセッサ消費電力制御方式

初期設計のスレーブプロセッサの場合、基本となる動作周波数・電源電圧の値は表 4.1 の通りとなる。ここで、表中の動作周波数に対応する電源電圧はその動作周波数を実現する最も低い電圧である。実際の制御でこのすべての周波数・電圧の組を用いることは、制御機構のハードウェア/ソフトウェアでオーバヘッドがあり現実的ではないと考えられる。

表 4.1 動作周波数・電源電圧の組合せ

No.	動作周波数 [MHz]	電源電圧 [V]
1	500	1.8
2	450	1.7
3	400	1.59
4	350	1.48
5	300	1.38
6	250	1.27
7	200	1.15
8	150	1.04
9	100	0.91
10	50	0.81
11	クロック停止	-

シングルプロセッサで実行プログラムの負荷状況に応じて動作周波数と電源電圧を制御し、動作時の消費電力量を削減する手法が提案されている [6–12]. 前章で、アプリケーションプログラムを複数個の断片（セグメント）に分割し、セグメントごとに必要十分な速度と電圧で実行する FFV 制御方式を提案し、評価した [9, 11, 12]. これは、プログラムに挿入したチェックポイントによってセグメントを作り、あらかじめ設定した進行計画と実際の経過時間との差分を計測して負荷状況とし、それに基づいて動作周波数と電源電圧を適切に設定してプログラム実行時のプロセッサ消費電力量を低減するものである.

一方、マルチプロセッサにおける並列処理では、性能一定の条件下で全体の処理負荷を実効的なプロセッサ台数に合わせて低下させ、それに合わせて動作周波数・電源電圧を降下させて設定できる. この二つの手法を組み合わせることにより、並列プロセッサの台数に見合った動作周波数・電源電圧に低減し、さらに各プロセッサのスレッド内で負荷の動的な変化に従って、FFV 制御でセグメントごとに動作周波数・電源電圧を低減することにより、一層の低消費電力化ができると見込める.

並列実行で各プロセッサを独立に FFV 制御するには、チェックポイント挿入地点をプロセッサごとに決定し、処理の進捗を見積もることが必要になる. これに対して、代表データによって事前に実行プロファイルを作成して進行状況を把握する手法 [15] もあるが、本研究では、マルチコア FFV 制御方式として、実行履歴に基づく予測で各プロセッサが進捗状況を見積もる方式を提案する. FFV 制御が特に有効となるものは動画処理のようなマルチメディアアプリケーションで、処理するデータによる負荷の変動が大きなものであり、ここでもそのようなアプリケーションを対象とする.

また、FFV 制御では最高動作周波数を F_{max} として、動作周波数を $1/2 \cdot F_{max}$ にするところで大きく消費電力を下げるができるが、 $1/4 \cdot F_{max}$ 以降ではその効果は小さくなり [9], 台数が増加することでリーク電流の影響が大きくなるので、マルチコアにおいては、何台のプロセッサで並列動作させればよいかを決定することも必要になる.

4.3.1 スレッド分割による消費電力量の低減

低消費電力化の対象となる動画処理などでは、プログラムは並列性の高い処理単位（例えば MPEG-4 では規定による構造のマクロブロックやスライスの処理などが候補となる）に分割できる。また、並列化ソフトウェア技術の進歩により、コンパイラによってループ繰返しや基本ブロックのレベルでの並列化、およびこれらを組合せたプログラム全体での並列化が実現でき、4~8 並列程度ではプロセッサの台数にほぼ見合った並列化効果が得られている [13]。すなわち、各要素プロセッサに対してほぼ均等に負荷が分散できるようにスレッド分割されていると期待できる。

スレッド分割され、並列化効果を得ている場合の消費電力量を表 4.1 の動作周波数・電源電圧の組合せにより、1 台（並列化しない）がデッドライン時刻まで常に最高動作周波数 500MHz で動作する場合の消費電力量を 100% とした時の比率で見積もる。処理終了までの消費電力量として、周波数と電圧の 2 乗、時間、プロセッサ台数の積によって得るものとする。2 台のプロセッサで並列実行し理想的に 250MHz (1.27V) で実行できるとすると、実行に要する時間は同じであるとするので、相対消費電力量は 49.8%、3 台の場合、それぞれ 200MHz でデッドライン時刻まで実行するとして 49.0% など、図 4.5 のようになり、5 台までは並列化によって動作周波数と電源電圧が低減することで消費電力量を低減できる。一方、6 台以降では台数分の 1 という意味での最適な値を設定できず、高い動作周波数で実行させるので、消費電力量は増加することになる。

以下で、このマルチプロセッサにおいて、次を前提として消費電力量の低減効果を評価する。

前提： 対象プログラムに割り当てられた処理に要する時間スロットにおいて、並列動作するすべてのプロセッサがそのプログラムだけを実行する（図 4.6）。同図ではプログラム 1（例えば MPEG-4）に与えられた時間スロットではすべてのプロセッサが MPEG-4 だけを網掛け部分で実行し、MPEG-4 処理が終了または待機してそのスロットで空白部分が生じても別のプログラム 2（MP3 など）

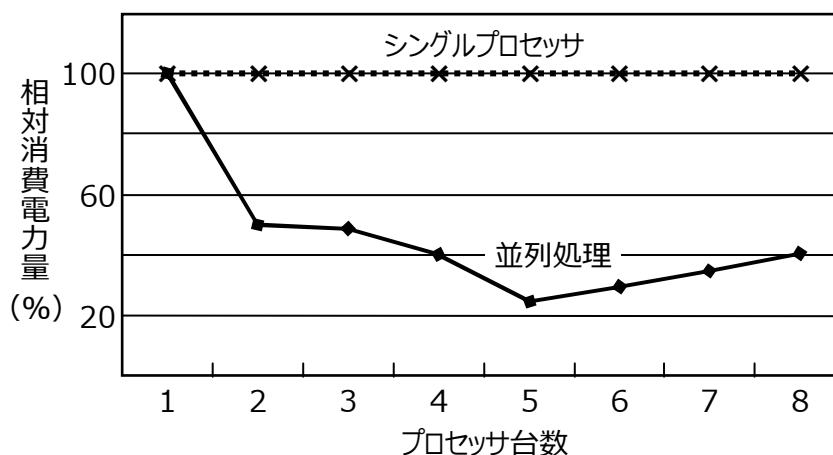


図 4.5 並列処理による動作時消費電力量の低減

が実行されることがないことを示す。

プロセッサは独立に動作周波数を変更するが、チェックポイントによる進行管理が適切であれば、例えば同期ポイントへ到達するまでに実行する動作周波数が変わるだけで、そこへの到達時刻は変化しないものとする。

4.3.2 電力制御アルゴリズム

対象となるプログラムの要求性能を見積もり、適切な動作周波数・電源電圧を設定することによってプログラム実行時の処理負荷に応じてプロセッサ動作に要する消費電力量を低減する。このためプログラム実行中にプロセッサの稼動状態を調べることになるが、マルチメディアプログラムなどのように周期的に駆動されて1周期の処理時間にデッドラインが存在するものに対しては、FFV制御で示すように、処理プログラムを適当な間隔のセグメントに分割してセグメントごとの実行時間の余裕を見積ってプロセッサの実行状態とする方式が簡便かつ有効である。

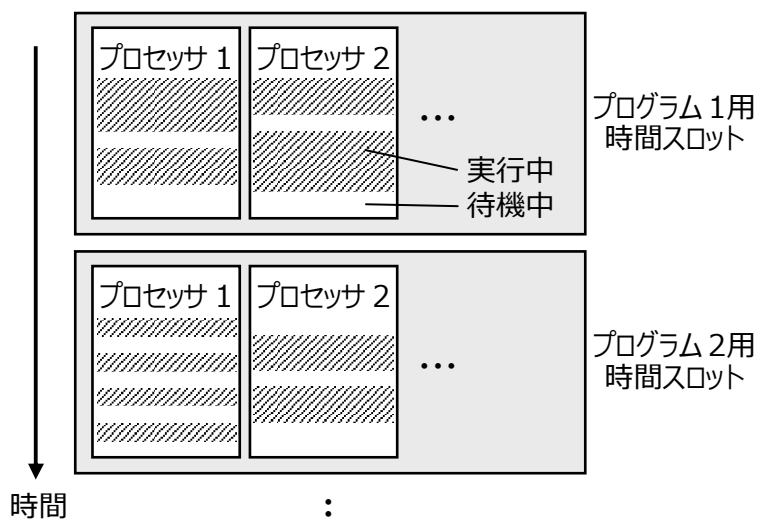


図 4.6 並列実行時の時間スロットに対する実行プログラムの割当て

■ シングルプロセッサでの FFV 制御方式とその課題

第2章で、著者らが [9] で示したシングルプロセッサにおける FFV 制御システムとして用意すべき動作周波数群と動作周波数の変更方式を述べた。それによれば、制御されるプログラムが

- (1) 1周期の実行を完了すべき限界時刻（デッドライン時刻）が分かっていること、
- (2) 1周期の実行完了のために必要な処理ステップ/時間がほぼ分かっていること、

を前提とする。

そして図 4.7（図 3.1 を再掲する）に示すように、プログラムをセグメントに分割し、各セグメントがほぼ同じ時間で実行されるとしてデッドラインとセグメント数から、セグメントごとに実行を完了すべき最悪の場合の時間：WCET（Worst case execution time）を設定する。デッドラインまで WCET 刻みで進行し、デッドラインちょうどで処理を完了する図 4.7 の二点鎖線（「あらかじめ計画された処理過程」）で示す最悪ケースの実行スケジュールをあらかじめ立てる。

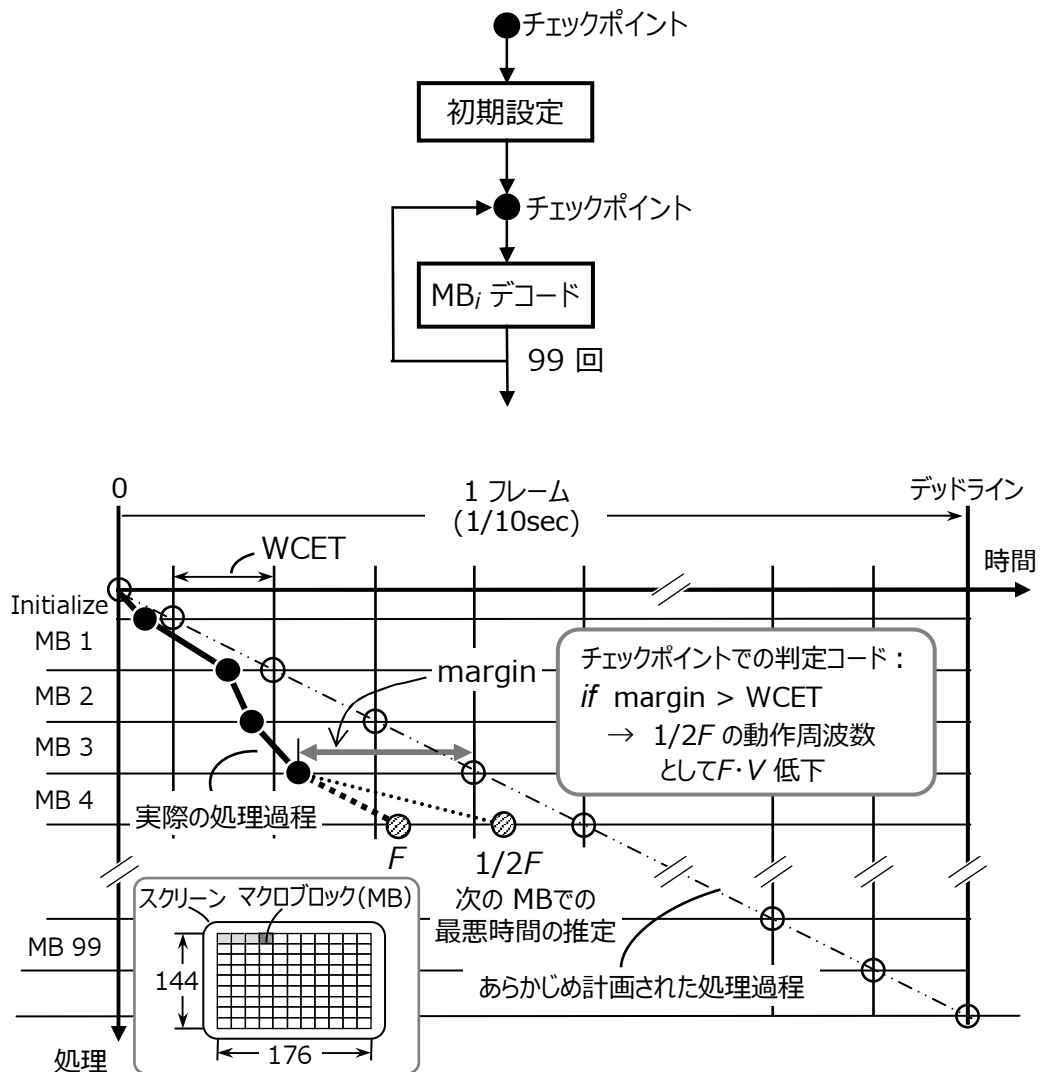


図 4.7 シングルプロセッサでの FFV 制御による処理プロセス

実際の処理においては一つのセグメントの実行に WCET の時間を要することは一般にはなく、処理の余裕時間 (margin) が累積されて行く。ある実行時点 (チェックポイント) で、この実行スケジュールに対する余裕時間が WCET 以上である場合に、動作周波数を $1/2$ ($3 \cdot \text{WCET}$ 以上である場合には $1/4$ 等) にして消費電力量を低減する。[9] では、変更のために採用すべき動作周波数は最高周波数 F_{max} から $1/2 \cdot F_{max}$, $1/4 \cdot F_{max}$, \dots と $1/2$ 倍ずつで降下させていけば十分であることを示している。

このシングルプロセッサにおける FFV 制御の課題として以下がある。まず、デッドライン時間を等分した WCET を基準とすると、処理に最大時間を要するとして余裕時間を判定するので、動作周波数低減の機会が少なくなることがある。また MPEG の処理ではマクロブロックという均質で細分された処理単位があり、マクロブロック群を自然なセグメントにすることができるが、例えば MP3 の処理などではそのような分割単位は簡単には設定できず、プログラムで制御できるよう実装することが困難になる。

さらに、この方式をマルチプロセッサでの並列実行に適用すると、スレッド分割の方法によってはプロセッサごとに実行されるセグメントの個数が異なる場合がある。シングルコア時には連続して実行されていたセグメントがスレッド内では必ずしも連続しないこともあり、スレッド分割に応じた進捗計算が必要で、プログラム設計が複雑になる。

■ 一斉モデルによる消費電力制御

上記 [9] の実行モデルはシングルプロセッサに対するものであるが、本研究のマルチプロセッサにおいても 4.3.1 項の前提により、対象プログラムに割り当てられた時間スロットにおいてすべての要素プロセッサがそのプログラムだけを実行するので、余裕時間をマルチプロセッサの系全体で管理される時間と考えると、シングルプロセッサの場合と同等になる。そこで、マルチプロセッサにおいて性能一定の条件で並列実行し、要素プロセッサが一斉に動作周波数・電源電圧を変更し、FFV 制御が可能なモデル（一斉モデル）を考えることができる。

シングルプロセッサの FFV 制御においては、余裕時間を判定するチェックポイントがプログラム中に挿入されて全体の時間経過を管理するので、並列処理においてもこれらチェックポイントの順序関係が守られることが必要である。実行スケジュールが逐次的にチェックされることによって、計算される余裕時間が妥当なものとなる。この一斉モデルでの FFV を用いたマルチプロセッサの電力制御アルゴリズムは次のようになる：

- (1) FFV 制御機構も含めた対象プログラムを並列化コンパイラなどによりスレッド分割する。余裕時間の判定が逐次に進行するよう、チェックポイントについて

てはシングルプロセッサでの順序を保存する制約を設ける。

- (2) 要素プロセッサに適用される初期動作周波数を定める。例えば2スレッドの場合には対象プロセッサの最大周波数の1/2, 4スレッドの場合には1/4とするなどである。
- (3) FFV制御によりプロセッサの動作周波数と電源電圧を変更する。(2)により与えられた周波数から開始して, WCET分の余裕が生じれば1/2, $3 \cdot \text{WCET}$ 分の余裕が生じれば1/4, と1/2ずつ低下させ, デッドライン時刻までに処理終了となったらスリープさせる。セグメントごとの判定ですべてのプロセッサの動作周波数・電源電圧を一斉に変更する。スリープについても, すべてのプロセッサが一斉にクロック停止状態に入るものとする。

ここでは, 4.2.3項に示す「性能一定」という前提であり, 並列化にともなうデータアクセスの量はシングル実行の場合とほとんど変化せず, バスの周波数はプロセッサ動作周波数の変更とは独立に設定されてアクセス要求が調停され, プロセッサ動作周波数の変化によってはバスやメモリの消費電力に変化がないことを仮定している。

■ 独立モデルによる消費電力量制御

一斉モデルでは, すべてのプロセッサが一斉に動作周波数と電源電圧を制御される。この方式は簡便であり, またシングルプロセッサ用に作成されたFFV制御用のプログラムを変更することもない。

しかし, 各プロセッサが独立して動作周波数と電源電圧の制御を受けると, より高い消費電力量削減効果が期待できる。対象とするマルチプロセッサシステムでは, スリーププロセッサは周波数・電圧管理ユニットによって所定の動作周波数と電源電圧で起動され, 各々の負荷に応じて独立にFFV制御され, 独立した動作周波数と電源電圧の制御が達成できる。この制御方式を独立モデルと呼ぶことにする。図4.8に模式的に示すように, 各プロセッサはそれぞれ他のプロセッサとは独立して動作周波数と電源電圧の変更を受ける。

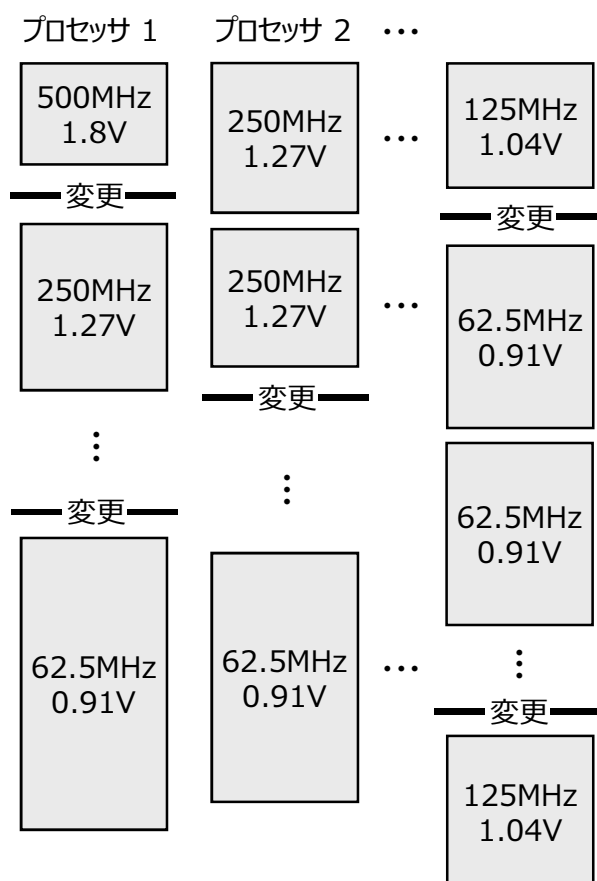


図 4.8 独立モデルの動作イメージ

■ 実行履歴に基づくマルチコア FFV 制御方式

本研究では独立モデルにおいて、シングルプロセッサで導入したチェックポイントの考えを拡張する形式で、スレーブプロセッサごとに独立に動作周波数・電源電圧を制御するマルチコア FFV 制御方式を提案し、考察する。

動作周波数・電源電圧の変更を各スレーブプロセッサの負荷状況によって行うので、スレーブプロセッサで稼動するスレッドの実行状況について前項で述べた課題を解決するため、以下のようにセグメントの実行時にプロセッサごとの実行時間を計測し、その履歴によって負荷と処理の進捗を見積もる。前章の MPEG-4 でもマクロブロックの実行時間はそれぞれデータにより異なるので、このように実行履歴によって負荷の変動を反映する方が、より尤もらしい進捗管理になると考えられる。

並列実行される被制御プログラムが初期周波数 F_{max} で周期的に駆動を受け、1 周期のデッドライン時刻 W が定められているものとする。並列化されたプログラムは、図 4.9 のように並列動作するスレッドの同期ポイントによって複数の領域に分けられる。各同期ポイントでのデッドラインが得られれば、同期ポイントによって分割された領域ごとにスレッドを対象にし、その中で FFV 制御を行うことができる。

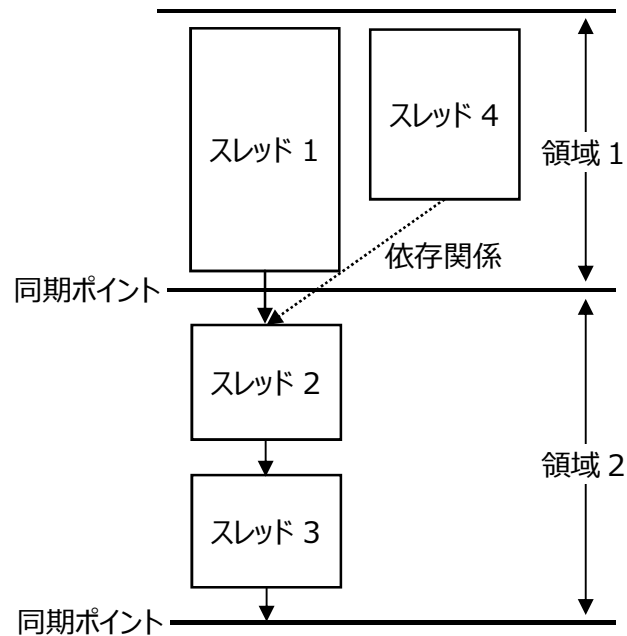


図 4.9 同期ポイントによるプログラムの領域分割

そこで、独立モデルにおいてスレッドが割り当てられるスレーブプロセッサごとの電力制御アルゴリズムを図 4.10 の処理フローに示すとおり、次のようにする：

まず、プログラムへの前処理として各要素プロセッサでのプログラム実行の進捗を見るため、プロセッサごとにプログラム（スレッド）を分割してスレッド実行時に必ず通過する地点を選んでチェックポイントを挿入する。これはプログラムの構造からみて必ず実行される地点であり、スレッドやループの入口、出口などが候補として挙げられる。連続するチェックポイントが一つのセグメントを構成し、複数のセグメントで 1 周期が構成される。チェックポイントの挿入地点は、各セグメントの実行時間

ができるだけ等しくなるように設定することが望ましい。

これによって、プログラムを N 個のセグメントに分割するものとする。 N は通常、数十程度である。スレッドの実行時、 k 番目のチェックポイントによって計測を開始し、 $k+1$ 番目のチェックポイントまでの実行時間を得る。これをセグメント k に対する実行時間とする。また、1周期の繰返しを実行するためのデッドライン時間が W として分かっているものとする。

次に、以下の手順で見積りを行い、セグメントでの動作周波数を決定する。

- (1) プログラムが一繰返し分実行される時、 k 番目のセグメント（セグメント k ）に対する実行時間 P_k が得られる。さらに全セグメントでの実行時間数の総和

$$T = \sum P_k$$

によって、 P_k を W で正規化した値が

$$W \times P_k / T$$

として得られる。これをセグメント k での最悪所要時間（WCET）の見積り値 w_k として処理の進捗を管理する。

- (2) 次の繰返し実行時にセグメント i で P_i を得る。 $w_i - P_i$ でそのセグメント i での余裕時間が得られる。それを 1 から $k-1$ まで総和し、セグメント k の開始時にこの総和 M を前回実行時の P'_k と比較して、

- $M + w_k < 2P'_k$ であれば、前回実行時の周波数のままで、
 - $M + w_k < 4P'_k$ であれば、そのセグメントを実行するための見積もり最悪時間に加えてさらに余裕として P'_k あるので、最低周波数でなければ動作周波数を前回周波数の $1/2$ に、
 - $M + w_k < 8P'_k$ であれば、同様に降下できる範囲で $1/2, 1/4$ に、
- …などと低減する。
- $M + w_k < P'_k$ であれば、前回周波数が最高周波数 F_{max} ならそのまま、動作周波数が $1/2 \cdot F_{max}$ 以下なら 2 倍にする。

動作周波数・電源電圧を変更する場合、スレーブプロセッサそれぞれが VMU に割込みを送り、引数で $1/2, 1/4, \dots$ と設定すべき動作周波数の倍数を指定する。電源電圧は、設定する動作周波数に対応して電圧ユニットで定める値が採られる。

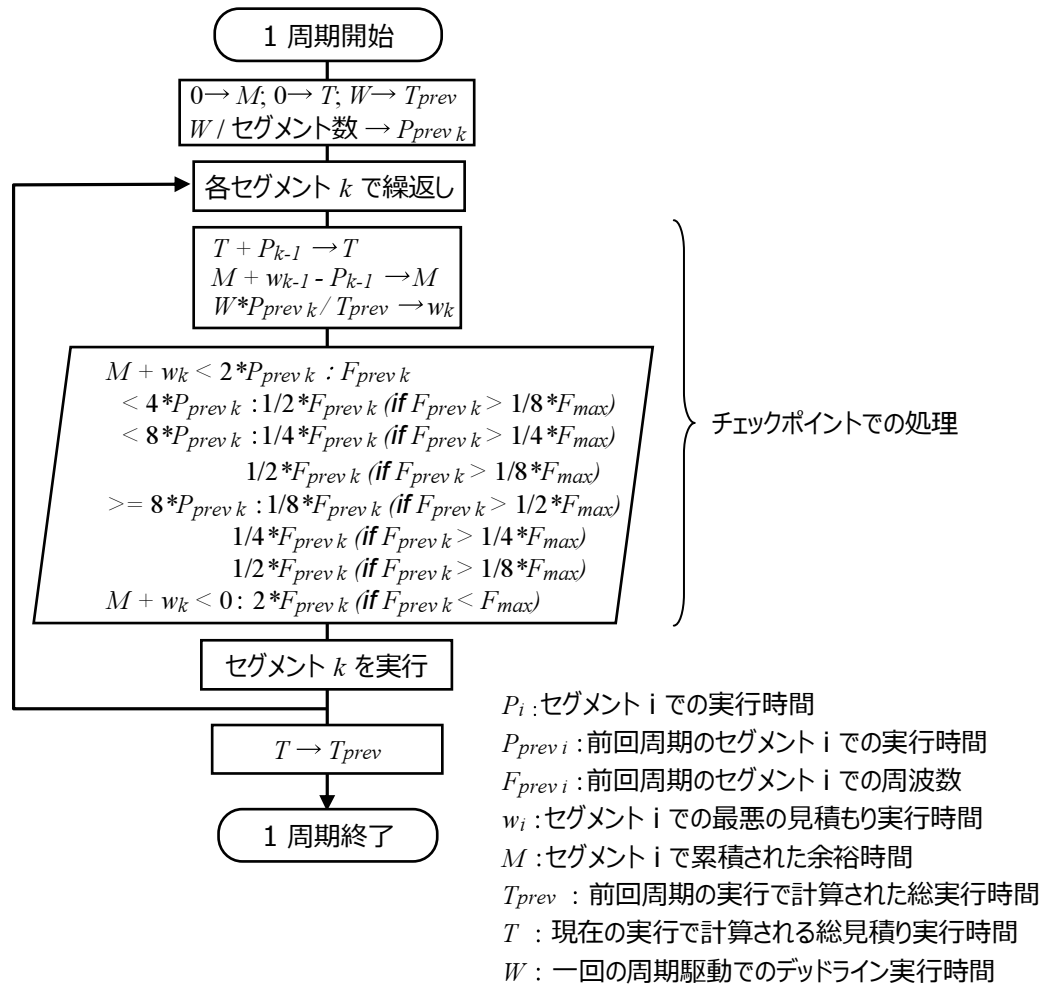


図 4.10 チェックポイントでの動作周波数決定処理

シングルプロセッサの場合、前章ではセグメントがほぼ等間隔であることを前提とし、それに基づいた WCET から余裕時間を算出している [9] が、このマルチプロセッサ FFV のアルゴリズムではチェックポイント間で $W \times P_k / T$ によって前回の実行状況からそのセグメントでの進捗度と余裕を見積もる形式としている。これは、全体のデッドライン W が既知であるとし、前回実行時のセグメントごとの所要時間に見

合った比率で処理が進行するとして余裕を見積もり，前回の実行時間と比較するものである．図 4.11 で，これに対応した処理プロセスの進捗状況を示す．なお，割込み処理も含め，この判定自体に要する命令実行時間と電圧変更時の安定に要する時間をオーバーヘッドとして余裕時間の計算において考慮している．

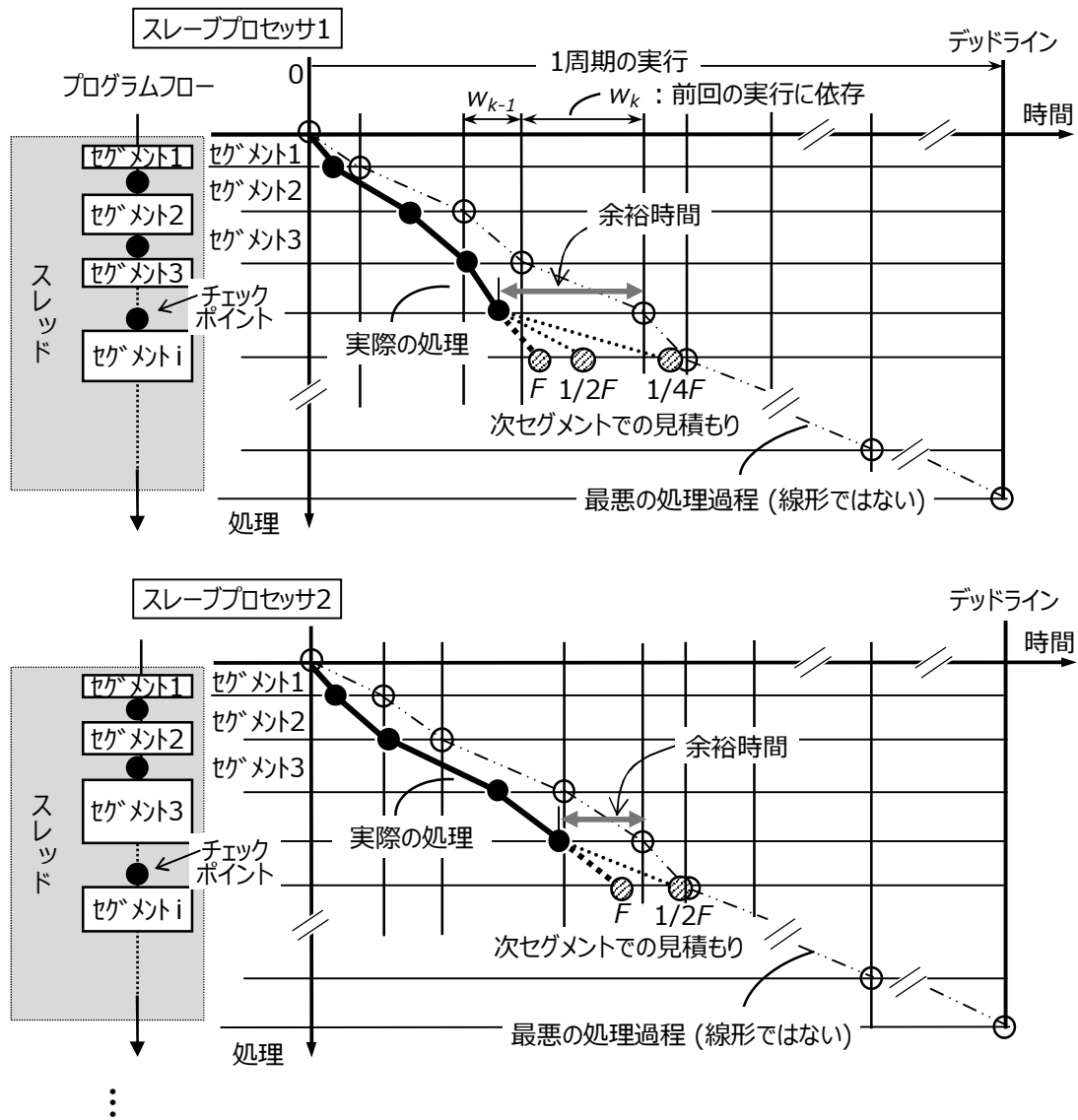


図 4.11 スレーブプロセッサでの FFV 制御による処理の様子

4.4 評価

前節のマルチコア FFV 制御を用いたマルチプロセッサにおける動作時のプロセッサ消費電力量の低減効果を評価する。

まず予備的な評価として、4.2.2 項 (1) の初期設計のスレーブプロセッサの特性により一斉モデルと独立モデルとで消費電力量の比較を行い、独立モデルを用いて前述の電力制御アルゴリズムを適用することが妥当であることを示す。

さらに、方式の詳細な評価を行うため、4.2.2 項 (2) のプロセッサをスレーブプロセッサとし、実際の特性を反映した消費電力量の変化を考察する。ここでは、プロセッサの低電力機構としてスリープや電源遮断での消費電力も考慮した評価を行う。なお、前章の評価では FFV 制御を行わない比較対象はスリープさせないとして実行するものであったが、以降本章では、いずれにおいてもスリープや電源遮断を行うものとして評価する。

4.4.1 予備評価

以下、4.2.2 項 (1) の要素プロセッサに対して、実験から得られた平均負荷（約 31%）をもとに設定した図 4.12 の処理負荷モデルを用い、シミュレーションによってプロセッサ 1 台が常に最高動作周波数 500MHz で稼動する場合を 100% とし、それに対する % 比率として消費電力量を評価している。なお、図 4.12 で相対処理負荷の 100% は、1 フレームに割り当てられた時間すべてを最高動作周波数で実行する場合に対するものである。

FFV 制御のため動作周波数を $1/2^n$ 倍ごとに設定できるものとする、本プロセッサでは表 4.2 のとおり、最大動作周波数 500MHz (1.8V) のプロセッサを 250MHz (1.27V), 125MHz (1.04V), 62.5MHz (0.91V), 31.25MHz (0.81V), クロック停止 (スリープ) として動作させることになる。

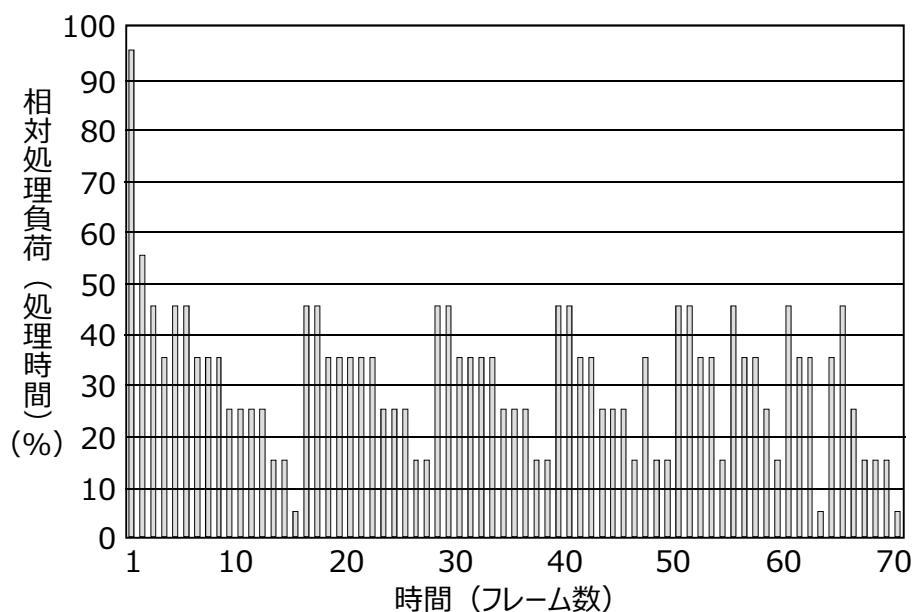


図 4.12 仮定した処理負荷モデル

MPEG-4 デコーダプログラムは、大きなシーンチェンジなど高負荷のデータに対応するため、一般にプロセッサの最高性能の半分以下の平均負荷を想定して実行できるように設計されており、最大処理性能を用いて処理しなければ出力に間に合わないフレーム（1 画面）は全フレームのごく僅かであるので、負荷の変動を有効に利用することができる。図 4.12 のモデルは実際の MPEG-4 動画デコードでの 71 フレーム分のトレースデータより得た負荷割合に基づいて仮定したものである。FFV 制御では累積して行く余裕時間によって動作周波数と電源電圧を変化させるため、このモデルが方式にとって特に有利になる訳ではない。この設定した負荷モデルに基づいて、フレームごとに処理の余裕時間を得て累積する。余裕時間に見合って動作周波数を設定し、それに対する電源電圧を得て消費電力量を求めている。

また、一般にはプログラムの並列化にともなって処理のオーバヘッドが生じ、共有メモリ型では主にプロセッサ間の同期処理によって発生する。オーバヘッドによるプロセッサ動作時の消費電力への影響を考慮するため、同期待ちがスピンドループによっ

て実現されているとして実質の処理量に対するオーバヘッドの比率を 10%, 20%, 30% とプロセッサの負荷に反映させ、それによって実行時間が増加するものとして消費電力量の変動を見る。

表 4.2 FFV 制御を使用した動作周波数・電源電圧の組合せ（初期設計）

No.	動作周波数 (MHz)	電源電圧 (V)
1	500	1.8
2	250	1.27
3	125	1.04
4	62.5	0.91
5	31.25	0.81
6	クロック停止	-

■ 比較の基準となる動作時消費電力量

動作時消費電力量の比較のため、以下の二つの場合をあげる。

(1) 最適周波数での動作の場合

シングルプロセッサの場合、図 4.4 による任意の値が採れるものとして、負荷 31% に対する最適な動作周波数は 155MHz であり、このとき消費電力量は 10.5% となる。2 プロセッサの場合、最適な動作周波数は 77.5MHz で消費電力量は 6.86% となる。4 プロセッサでは 6.24% である。8 プロセッサでの最適な動作周波数は 20MHz 以下であり、動作周波数の低下に見合った電源電圧降下を得られないため、消費電力量は 6.26% となり、6.24% が消費電力量の下限となる。これを図 4.13 の (1) に示す。ここでは消費電力量の下限を見るため、オーバヘッドはないものとしている。

(2) 並列化とスリープを行う場合

電源電圧が離散的に表 4.2 の値を採るものとする、消費電力は図 4.13 (2)

のとおりとなり、プロセッサ台数に応じて 500MHz から 250MHz や 125MHz に動作周波数を低下させた場合の電源電圧低下による消費電力量低減が得られる。ここでもオーバヘッドはないものとし、各プロセッサの負荷がそれぞれ 31% になって、割り当てられた動作周波数・電源電圧で動作した後にスリープする場合を想定し、2 台で 15.4%，4 台で 10.3%，8 台で 7.91% となる*。

■ FFV 制御を追加した場合の動作時消費電力量

マルチコア FFV 制御を行うと、MPEG-4 デコードなどのプログラムで余裕を持った処理が行われている部分に対し、各プロセッサ内でさらに動作周波数を下げて消費電力を低減できる。このプロセッサでは、500MHz～50MHz 程度で電源電圧が線形に低減でき、100MHz 以下に周波数を下げた場合にも電圧低下による消費電力の低減を期待できる。表 4.2 の動作周波数と電源電圧により、以下で一斉モデルと独立モデルについて評価する。なお、この条件で図 4.12 の負荷モデルによってシングルプロセッサで FFV 制御すると、消費電力量は 16.5% となる。

(1) 一斉モデルの場合

前提から各プロセッサにおける処理はほぼ等しく、それぞれの処理負荷の状況も単一のプロセッサの場合と同様として、負荷モデルにより各動作周波数での実行時間をシミュレーションすると、各プロセッサの負荷が等しく並列処理される理想的な場合（均等負荷）では、2 台の場合プロセッサが 250MHz で動作する時間は全体の 3.9%，125MHz で動作する時間は全体の 34.1% などとなり、1 台（500MHz）での実行時と比較して消費電力量を 11.2% にまで低減できる。同様に 4 台の場合は 7.85%，8 台の場合は 7.61% と図 4.13 の (3) として示すように低減できる。これは同図 (2) の並列処理＋スリープによる消費電力量低減効果に比べて大きなものである。

さらに、並列化にともなうオーバヘッドの比率を 10%，20%，30% とプロセッサ負荷に反映させて実行時間が増加するとし、消費電力量の変動を見る

* 4.3.1 項での消費電力量の低減にはスリープの効果を含めていないが、以降の評価ではスリープを考慮している。

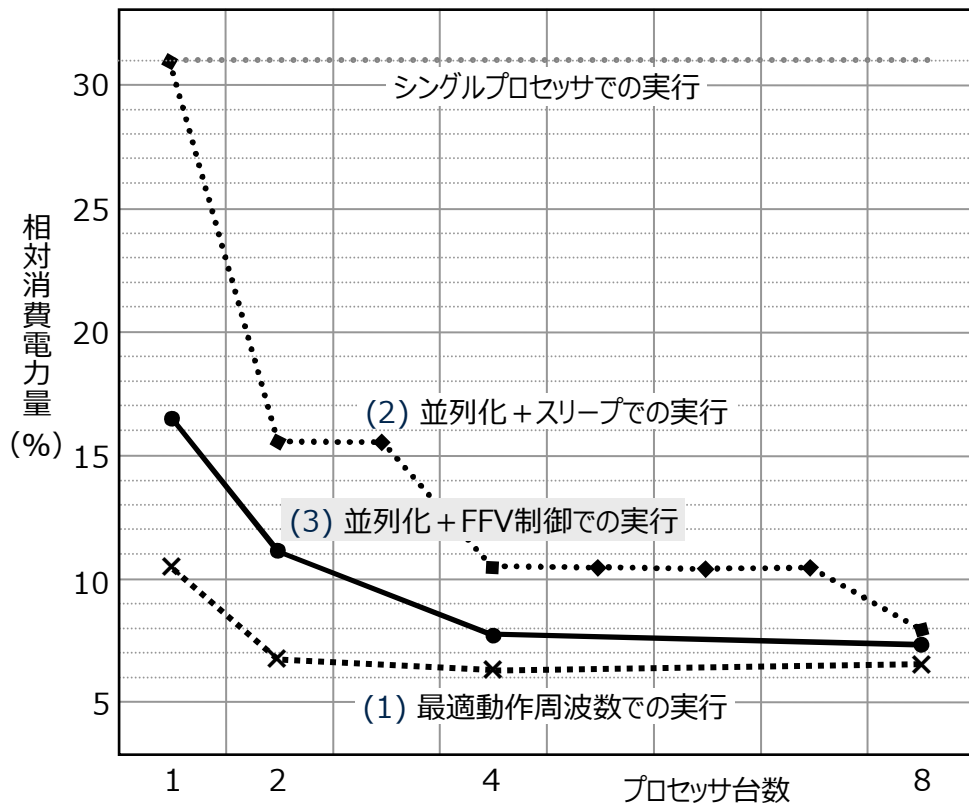


図 4.13 処理負荷を考慮した並列制御による消費電力量の低減

と、2台でオーバヘッド10%の場合14.2%など表4.3のとおりとなる。1台の場合に16.5%の消費電力量であるので、この表によるとほぼ30%までのオーバヘッドであれば、並列化とFFV制御によってプロセッサ動作時消費電力量の低減効果が得られると言える。30%のオーバヘッドでは4台時に3.1倍の高速化となるので、性能一定の下では妥当な台数効果が得られて動作周波数が低減でき、消費電力量の低減効果は低下するものの、台数に応じて向上することが分かる。

以上は各プロセッサに均等に負荷が分割されている場合であるが、負荷の偏りを考えると、プロセッサ2台の場合に、両者の負荷比率を7:3として消費電力量をシミュレーションすると18.0%となり、均等(5:5)の場合の60%

表 4.3 並列処理オーバーヘッドを考慮した消費電力量低減効果の変動

オーバーヘッド	プロセッサ台数		
	2 プロセッサ	4 プロセッサ	8 プロセッサ
なし	11.2%	7.85%	7.61%
10%	14.2%	9.35%	8.26%
20%	15.7%	10.3%	8.73%
30%	16.8%	10.9%	9.13%

程度劣化する。プロセッサ間の負荷に偏りがある場合の消費電力量を図 4.14 に示す。図 4.14 ではプロセッサ数が 2 台、4 台、8 台の場合に、均等な負荷と偏りのある場合をそれぞれ左右の棒グラフで示した。一斉モデルでは高負荷のプロセッサに全体の動作傾向が支配されるので、プロセッサ間の負荷の偏りが大きい場合に電力低減効果の劣化が激しく、4 台で 6:2:1:1 の負荷比率では均等負荷の場合に比べて消費電力量が 3 倍以上になる。

(2) 独立モデルの場合

図 4.15 の独立モデル部分の棒グラフに示すように、各プロセッサで独立に動作周波数・電源電圧を制御すると、プロセッサ 2 台の場合に両者の負荷比率を 7:3 とすると、シミュレーションから消費電力量は 13.8% となる。図 4.14 の棒グラフの「均等負荷」で示す理想的な均等の負荷比率の場合よりは消費電力量の低減効果が落ちるが、一斉モデルの場合の 1/3 程度の劣化ですむ。プロセッサ 4 台では比率を 6:2:1:1 とした場合で消費電力量は 13.5% となり、やはり独立モデルの方が負荷の偏りによる影響が小さくなっている。独立モデルではプロセッサ 8 台でも消費電力量低減効果はそれほど変わらず、負荷の偏りの影響が小さいといえる。プロセッサ台数が多いほど負荷の偏りの影響が大きくなるので、独立モデルで FFV 制御にすることが必要であると考えられる。

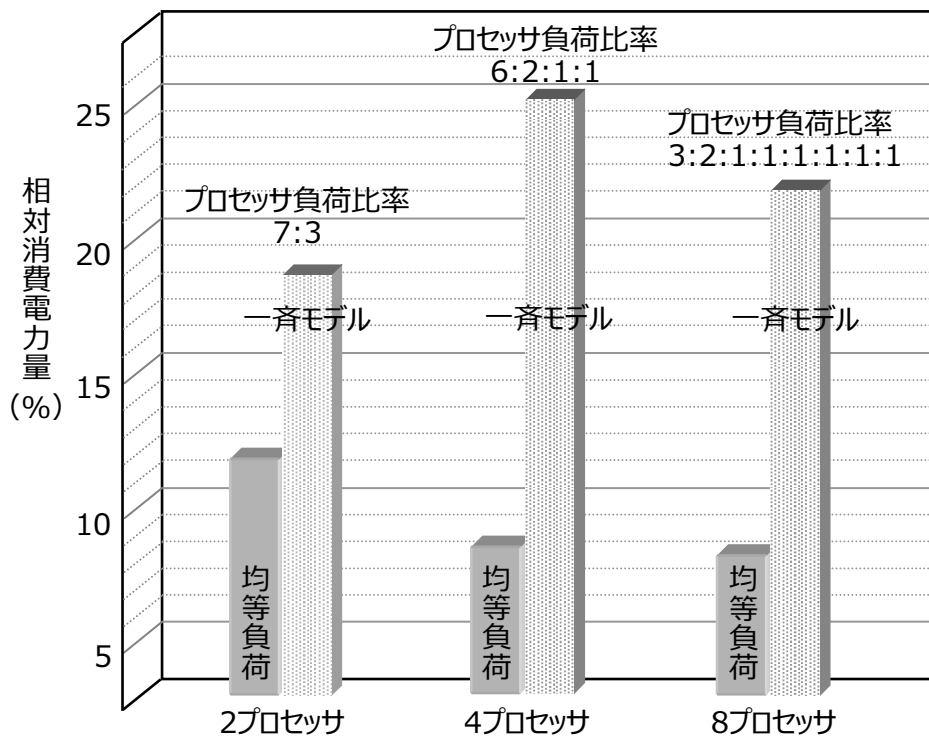


図 4.14 一斉モデルでプロセッサの負荷に偏りがある場合の消費電力量低減効果

4.4.2 マルチコア FFV 制御による消費電力量の低減効果

図 4.16 でオーバーヘッドを加味してマルチコア FFV 制御による消費電力量低減効果の傾向を示す。負荷は均等であり、並列化オーバーヘッドは 2 台時を 10%、4 台時 20%、8 台時 30% として見積もっている。

図 4.16 のグラフに示すとおり、この条件ではシングルプロセッサで、500MHz で稼働させた場合、平均負荷が 31% であればスリープ時の消費電力を 0 としてスリープ制御のみ (500MHz で動作し、終了後スリープ) では 31%、シングルプロセッサの FFV 制御で 16.5% への低減が限度である。これと比較してマルチプロセッサ構成では台数に応じて 10% 以下になるまで消費電力量を大きく低減できる。

マルチコア FFV 制御によるものでは、図 4.16 の (2) に示すように 2 台で 10% の

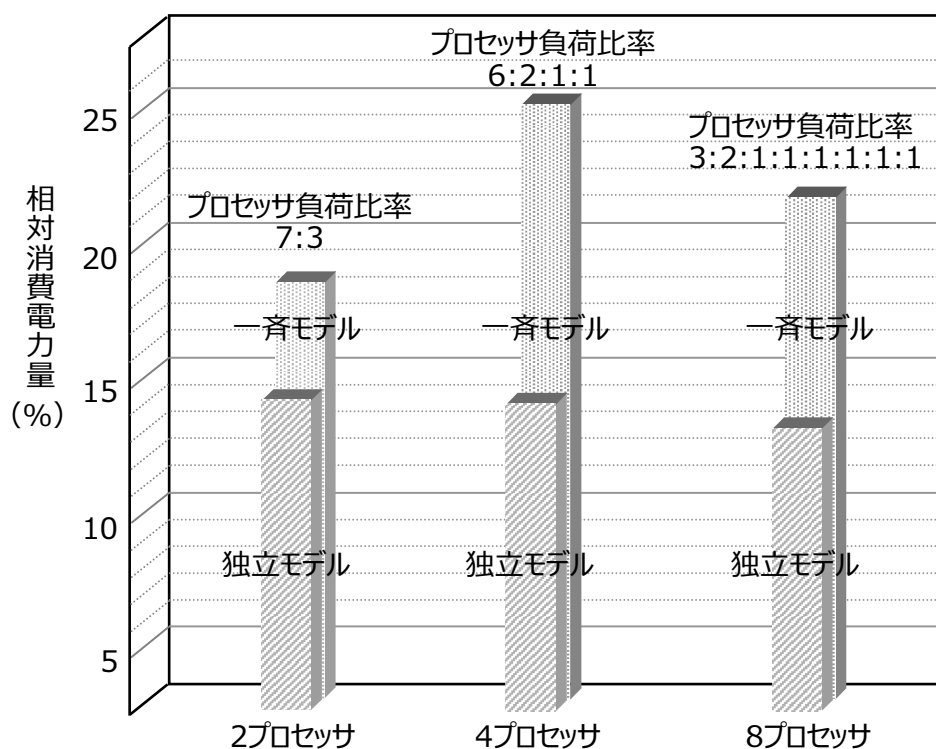


図 4.15 一斉モデルと独立モデルでの消費電力量低減効果の比較

オーバヘッドに対して消費電力量は 14.2%，4 台（オーバヘッド 20%）で 10.3%，8 台（オーバヘッド 30%）で 9.13% である。

一方，並列化+スリープによる方式では消費電力量は，図 4.16 の（3）に示すように，2 台でオーバヘッドが 10% に対して 17.0%，4 台（オーバヘッド 20%）で 12.5%，8 台（オーバヘッド 30%）で 10.3% となる。

このように，均等負荷の条件では，スリープのみを使用する制御と比較して，FFV 制御によって 2 台並列時に 16% 程度，4 台並列時には 18% 程度それぞれ消費電力量の低減効果が高くなり，図 4.16 の（1）で示すオーバヘッドを加味した最適な周波数での消費電力量に近くなる。8 台並列時では並列化+スリープとマルチコア FFV 制御では低減効果の差は 11% 程度になる。

また，マルチコア FFV 制御とシングルプロセッサとの比較では，4 台時で 66.8%，8 台時で 70.5% の低減となり，提案プロセッサのように電源電圧低下の余地の大きい

ものでは高い省電力効果が得られるといえる。

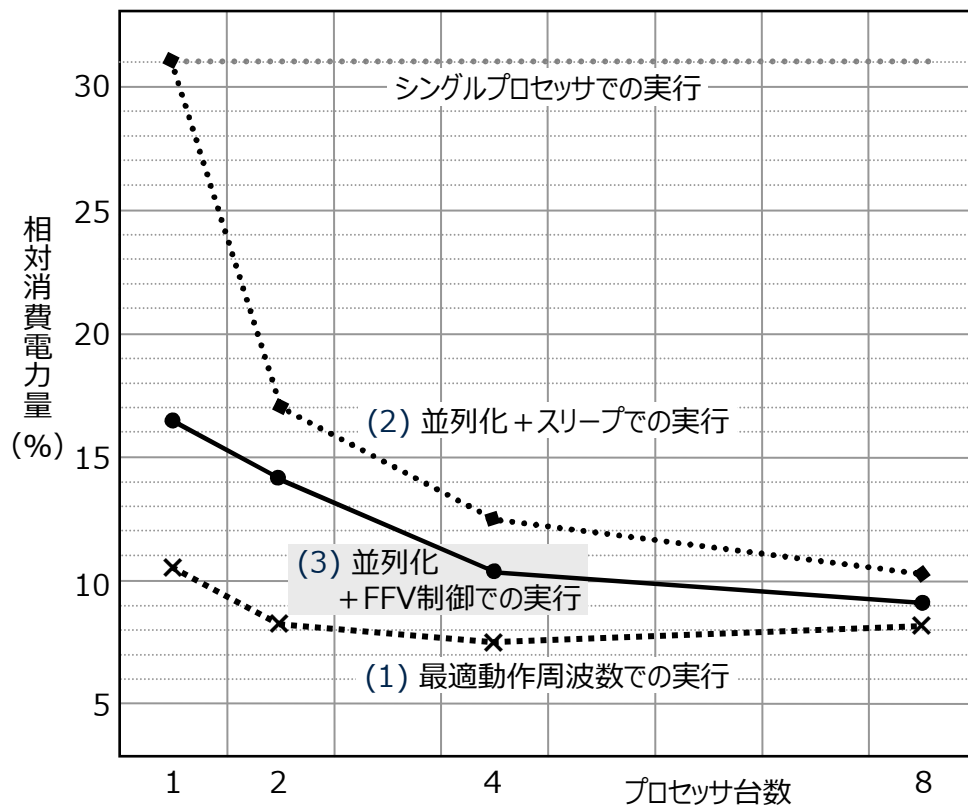


図 4.16 オーバヘッドも考慮したマルチコア FFV 制御による消費電力量の低減効果

4.4.3 実プロセッサデータによる詳細評価

予備評価による確認を踏まえて独立モデルを採用し、4.2.2 項のスレーブプロセッサの電力特性 (2) : 実プロセッサの特性を反映した設計により、マルチプロセッサにおける動作時の消費電力量の低減効果を評価する。スレーブプロセッサそれぞれが 240MHz 動作の場合の消費電力量を 100% とした時の比率で消費電力量を表す。

(i) 評価する消費電力量

動作周波数を $1/2^n$ 倍ごとに設定し、表 4.4 のとおり、最大周波数 240MHz (1.97V) のプロセッサを 120MHz (1.24V)、60MHz (1.0V)、30MHz (1.0V)、クロック停止

(スリープ) で動作させるものとして制御し、プログラム実行中のプロセッサ動作時間における消費電力量を求めて評価対象とする。[†]

ここで、[19]によるプロセッサ電力評価での240MHz時200mA以下(100mA程度)、スタンバイ時2.2mAという測定値に基づき、スリープ状態の動作時消費電力を240MHz時の1.5%であるとしている。

表 4.4 FFV 制御を使用した周波数・電源電圧の組合せ (実プロセッサ)

No.	動作周波数 (MHz)	電源電圧 (V)	消費電力 (W)%
1	240	1.97	100
2	120	1.24	19.8
3	60	1.0	6.4
4	30	1.0	3.2
5	クロック停止	1.0	1.5

(ii) 評価モデル

負荷変動を反映させるため、実験から得られた処理負荷データを用い、シミュレーションによってスレーブプロセッサ1台だけが最大動作周波数240MHzで稼動する場合に対する%比率として、スレーブプロセッサが1台、2台、4台、8台についてマルチコアでの消費電力量を評価する。

評価プログラムとしてはMPEG-4デコーダを用い、実際に640×384解像度の動画データを復号した600フレーム分の実行トレースデータから図4.17の負荷変動データを得た。それによると平均負荷は想定する最大負荷の39%、最小負荷は同じく31%となっている。

[†] 電源電圧が低下しない30MHzは本モデルのシミュレーションによる評価では意味がないが、第3章のまとめで述べた貫通電流の影響も想定して含めている。

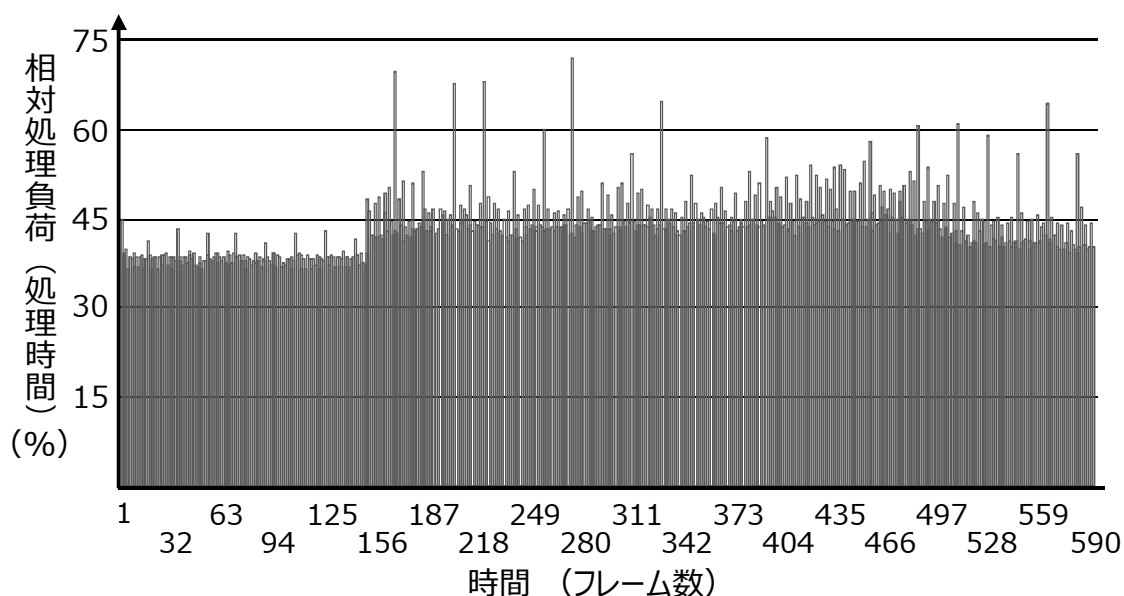


図 4.17 評価用動画データ負荷変動 (600 フレーム分)

MPEG では処理されるデータ構造 (図 4.18) の規定により, 1 画面を帯状に分割したスライス (Slice) と呼ぶ単位ごとに効率よく並列実行できることが知られている。スレーブプロセッサにスライスの処理を割り当てて行く [16, 17] ことで各スレーブプロセッサの処理量を平均化し, スライス間のデータ依存性がなく実行できて, 性能一定の条件で実行する場合には, プロセッサの台数に応じて動作周波数を低下させて消費電力量を低減できる。本章の評価モデルでは同図に示すように, 提案アーキテクチャの各スレーブプロセッサに連続するスライスの処理を順に割り当てて並列処理する形態としている。

■ 比較の基準となる動作時消費電力量

動作時消費電力量の比較のため以下の二つの場合をあげる。

(i) すべてのプロセッサを最高周波数で動作させる場合

各スレーブプロセッサがそれぞれのセグメントを最高動作周波数 240MHz で実行し, 1 フレームのデコード処理が終了した後スリープするものとする。この場合, ス

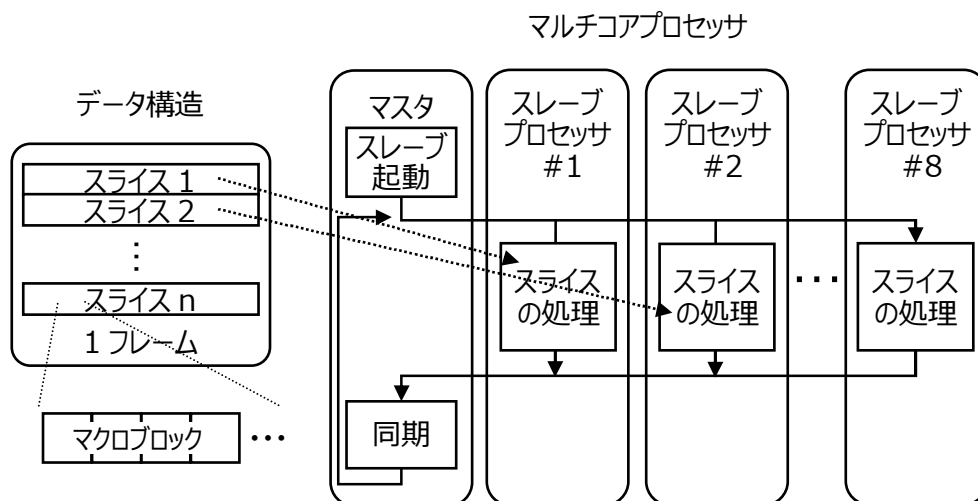


図 4.18 MPEG のデータ構造とプロセッサへのスライスの割当て

リープ時の消費電力がスレーブプロセッサ数の増加とともに増加するため、全体の消費電力量も図 4.19 (1) のように増加して行く。

(ii) 並列化台数に応じて周波数と電圧を低下させた場合

電源電圧が表 4.4 の値を採るものとして、並列化を行って動作周波数と電源電圧を低下させる。プロセッサ台数に応じて 240MHz から 120MHz や 60MHz に動作周波数を低下させた場合の電源電圧低下 (1.97V から 1.24V に低下など) により、実行時間一定 (性能一定) のもとで消費電力量は低下する。

スレーブプロセッサそれぞれが、並列化台数に応じて割り当てられた動作周波数・電源電圧で動作した後スリープする理想的な場合を想定すると、ここでの MPEG-4 デコーダの並列化では各プロセッサの負荷をほぼ均等にでき、またデータ依存もないので、図 4.19 (2) で示すとおり、2 台並列時で消費電力量は 42.9%、4 台で 33.5%、8 台で 41.8% となる。4 台時以降にはスリープ時の消費電力 (1.5%) の影響が大きくなるため消費電力量が増加する。これは、実行履歴を用いず、かつ同期ポイントだけで低電力のための制御を行う場合に相当すると考えられる。

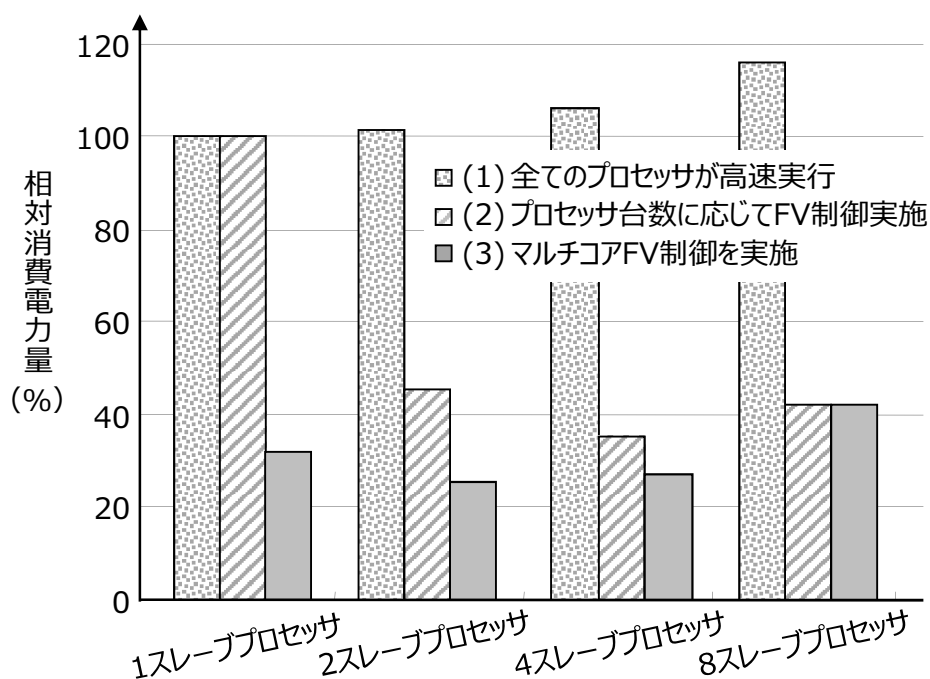


図 4.19 プロセッサ台数によるマルチコア FFV 制御の消費電力量低減効果

■ マルチコア FFV 制御を実行した場合の動作時消費電力量

マルチコアで FFV 制御を行ってさらに動作周波数と電源電圧を低下させる。スレーブプロセッサに対して FFV 制御を適用する場合、MPEG-4 デコーダでのセグメントの候補は、画像フレームを処理するスライス、もしくはマクロブロックであり、スレーブプロセッサごとにスライスやマクロブロック処理の入口にチェックポイントを挿入することになる。4.3.2 項のアルゴリズムに従って、図 4.17 の負荷に基づいてセグメントごとに処理の余裕時間を累積する。一つのセグメントで負荷が $x\%$ であれば、 $100 - x\%$ の余裕時間があるとし、求めた余裕時間に見合せて動作周波数を設定し、全体の消費電力量を得ている。

図 4.19 (3) に示すように、4 台までは消費電力量低減の効果がある。4 台超で低減効果がないのは、60MHz 動作で消費電力量としては低減されないことと、スリープ時の消費電力量が影響することによる。負荷が増加した場合の傾向を見るため、ト

レースデータから得た負荷を基準（標準負荷）とし，その1.5倍（中負荷），2倍（高負荷）として評価すると，図4.20（2）（3）に示すように，高負荷時にも台数効果が現れて消費電力量が低減できている．これらは前回の結果に基づく制御で適正な余裕時間の評価が行えることによるものと考えられる．

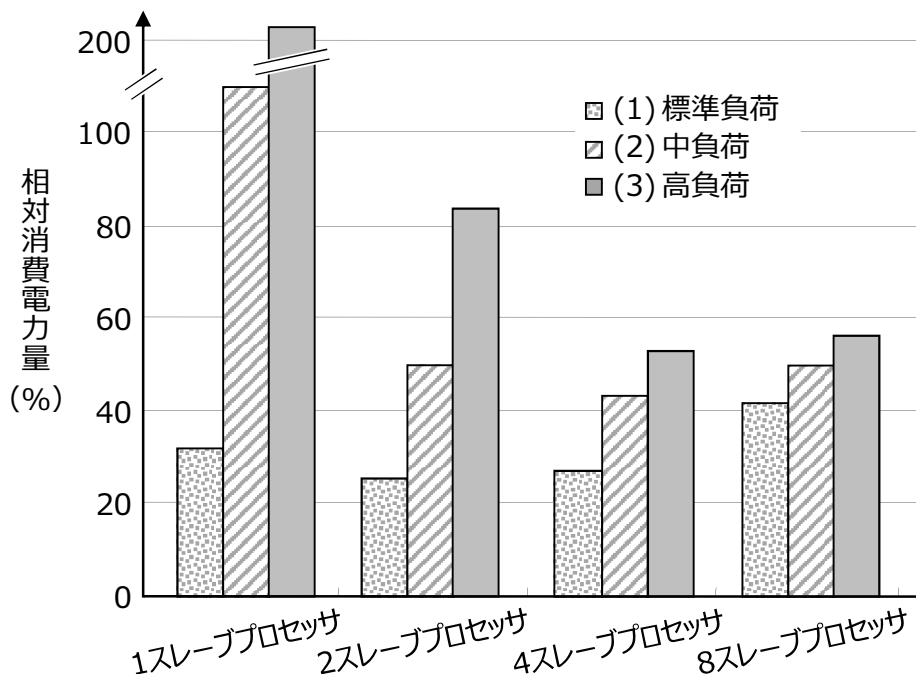


図 4.20 処理負荷の増加による消費電力量低減効果

また，プロセッサ台数に対する効果では，実行履歴を用いず，同期ポイントだけで制御を行う場合，図4.19（2）と比較して，2台並列時に27%，4台時には5%程度消費電力量を低下できており，同期ポイント以外で制御できるFFV制御の効果が出ていると言える．

さらにマルチコアFFV制御方式の効果を示すため，以下の観点で評価する．

(1) チェックポイントの数による効果の差

チェックポイント数が多いほど負荷に追従した制御が可能になると考えられるが，チェックポイントでの判定処理がオーバーヘッドになるので，各プロセッサでのセグメ

ントの個数による効果の差を調べるため、40個、20個、10個、1個のマクロブロックごとにそれぞれチェックポイントを入れて消費電力量の変化を見た。図4.21にその結果を示すが、40、20、10マクロブロックごとの判定で消費電力量はほとんど変わらない。

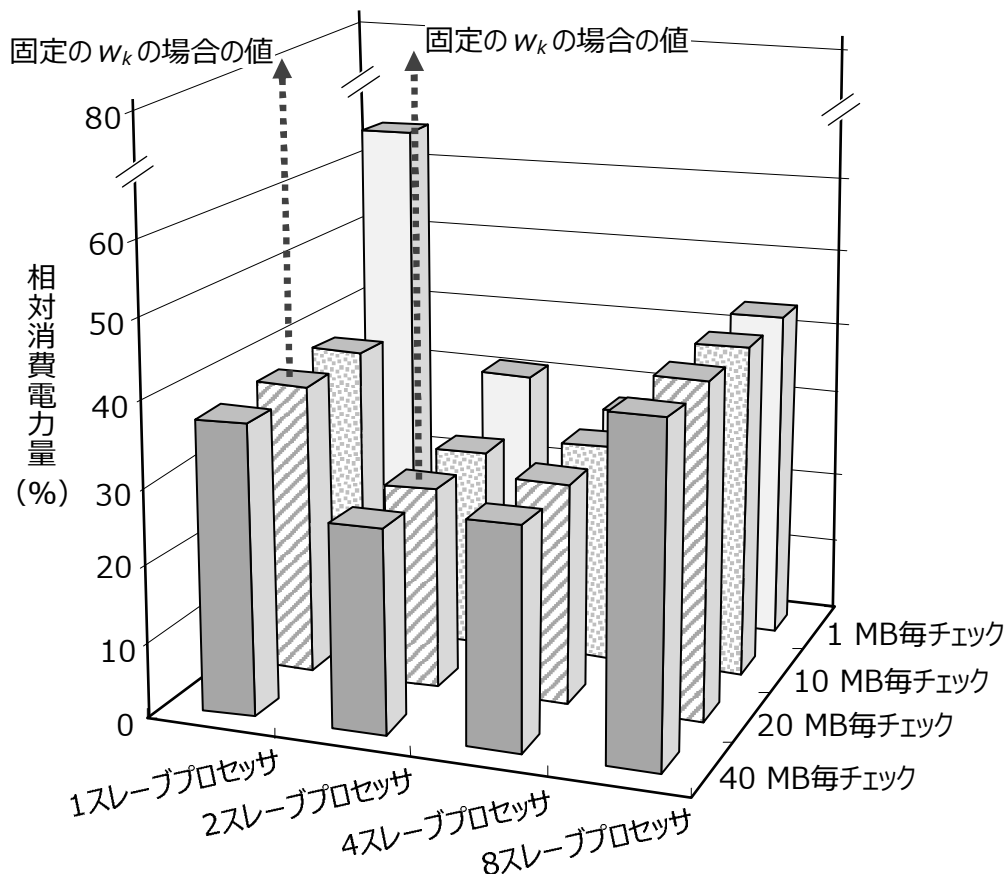


図 4.21 チェックポイントの個数による消費電力量低減効果の比較

これは、チェック自体に要するオーバーヘッドが数十命令程度で無視できるものであること、すでにプロセッサ台数の並列化によって負荷が下がっており、各プロセッサでオーバーヘッドを要する動作周波数・電源電圧変更の機会がほとんどなく、電圧設定での安定待ち時間（ $100\mu\text{sec}$ 程度）がほとんど影響しないことによる。1マクロブロックごとにチェックする場合は、マクロブロックそれぞれの負荷変動は比較的大きくて動作周波数を上げる場合も頻出し、オーバーヘッドが無視できなくなっている。

FV 制御 (DVFS) による最大の電力削減効果は F_{max} から $1/2 \cdot F_{max}$ に低下させるところであるが、マルチコア FFV 制御の場合、その効果を並列化によって得ることになるので、チェックポイント個数が少なくても電力量削減に与える効果が少なく、前回実行時間とデッドラインによる最悪時間を比較しているため、初回から低周波数で動作させることができるためと考えられる。本章での評価には 20 マクロブロックごとのデータを用いている。

(2) 実行履歴に基づく進捗管理方式の効果

前回の実行時間を用いてプログラムの進捗状況を管理するが、この方式と固定した WCET ごとの進捗を管理する方式 (シングルプロセッサで適用したもの) とを比較した。シングルプロセッサで適用したようにデッドライン時間をコアごとのセグメント個数で等分した固定 w_k を用いてシミュレーションすると、消費電力量は 1 コア時 79%、2 コア時 78.6% となり (図 4.21 の点線で示す)、1 コア時には約 21% 削減できるが削減量としては多くなく、また、2 コアではそれ以上の効果が出ていないことが示される。これに対してマルチコア FFV 制御では、その $1/2 \sim 1/3$ 以上となる削減効果を得ている。

表 4.5 コア当り 1 フレーム処理での動作周波数・電源電圧の変更回数

	1 コア	2 コア	4 コア	8 コア
平均上昇回数	8.4	4.6	0	0
平均降下回数	8.4	4.6	0	0
平均変更回数	16.8	9.2	0	0
最大変更回数 (上下)	15	9	2	1

また、提案方式でコアごとの動作周波数・電源電圧の変更回数を得ると表 4.5 のようになり、1 コアや 2 コアではチェックポイントの $1/3$ 程度で変更が発生しているが、4 コアや 8 コアではほとんど変更が見られない。これは、 $1/4$ や $1/8 \cdot F_{max}$ で開始すると、周波数低下で消費電力的に有利になることがないからである。4 コアや

8 コアで変更を行うものは、実行履歴のない第 1 フレームの最初のセグメントで一旦上昇した後、低下しているためである。このことは、実行履歴を用いない最悪値での進捗管理での性能が悪くなりがちであることを裏付けていると考えられる。

チェックポイントやフレームごとに余裕時間を求めると表 4.6 の標準負荷の欄のようになり、セグメント実行の見積り時間やフレーム処理のデッドラインをほとんど超えずに制御でき、実行履歴による評価が妥当であることが示される。

表 4.6 デッドライン制約を超えたセグメント/フレームの出現率

	1 コア	2 コア	4 コア	8 コア
標準負荷 (≈ 30%)	0.02%	0.03%	0.03%	0.05%
	0	0	0	1
1.5 倍負荷 (≈ 45%)	0.75%	0.27%	0.35%	0.56%
	10	1	1	11
2 倍負荷 (≈ 60%)	8.8%	3.2%	3.9%	5.4%
	71	20	48	92

上側の行：余裕時間が負となるセグメントの比率

下側の行：デッドラインを超過したフレーム数

(3) 消費電力量最小となるコア数での実行

動作周波数を降下させても電源電圧の降下には限度があり、リーク電流の影響もあるため、並列動作プロセッサの台数を増加させても消費電力量は下がらず、電力量を最小にできるコア数がある。ここでの評価では、平均負荷が 30% 程度で図 4.20 (1) に示すように実行プロセッサ数が 2 で最小電力となり、8 プロセッサ時の消費電力量は逆に大きくなる。また、平均負荷が大きくなって基準の 2 倍 (60% 程度) としたものは同図 (3) のように 4 コアが電力最小になる。動作周波数を 1/2 にする部分で最大の削減効果が見込めるので、 F と $1/2 \cdot F$ で制御する単純化したモデルを考えると、平均 50% の負荷では 2 セグメント実行して 1 セグメント分の余裕ができた以降

$1/2 \cdot F$ で実行できるが、負荷が 60% になると平均的に $1/2 \cdot F$ で実行できるセグメント数は 4 割程度に下がってしまい、それに伴って動作周波数・電源電圧変更によるオーバヘッドも増える。40% の負荷では $1/2 \cdot F$ で実行して余裕時間が増加するが、さらに $1/4 \cdot F$ にしても電力量の削減効果は出ない。また、30% 負荷で $1/2 \cdot F$ にして負荷が倍になると 60% の場合に帰着する。これらから 30% の負荷を目標に制御すると、消費電力量について効果的に動作周波数の変動を起こせると考えられる。

表 4.6 の 1.5 倍や 2 倍負荷の欄で余裕時間やデッドライン超過の発生を見ると、余裕時間がなくなるセグメント数は標準負荷 (30%) に対してそれぞれ 10 倍、100 倍になること、1.5 倍負荷では 2, 4 コアでフレームのデッドラインをほとんど超えないのに、2 倍負荷の場合はかなりの割合でそれを超えることが示されることから、30% から 45% を下回る負荷で制御することが妥当であると考えられる。

そこで、以下の手法により 30% 台の負荷に誘導するよう、論理的な分割数を最大の並列プロセッサ数に等しく 8 スレッドに分割し、その 8 スレッドを負荷に応じて稼働するプロセッサ数で振分けて並列実行すると消費電力量最小のコア数で実行できると考えられる。

- 割り当てるプロセッサ数を 2, 4, 8 に限ってプロセッサごとの実行スレッド数を同じにする
- どれかのプロセッサで負荷が 60% 超ならプロセッサ数を 2 倍にしてスレッド数を半分にする
- すべてのプロセッサで負荷が 40% 未満ならプロセッサ数を $1/2$ にしてスレッド数を倍割り当てる

とし、フレームごとの同期地点でそれまでの実行時間と余裕時間とによって負荷の値を得て、プロセッサとスレッドの割当てを上記によって設定する。スレッドへの実際のプロセッサ割当てはマスタプロセッサがスレッドライブラリを呼び出すことで行う。

負荷データを変化させてシミュレーションすると図 4.22 のようになり、基準とする標準負荷の 0.7 倍となる低負荷では 8 プロセッサで開始し、2 フレーム目に 4, 3 フレーム目に 2 と実行中のプロセッサ数が変化する。

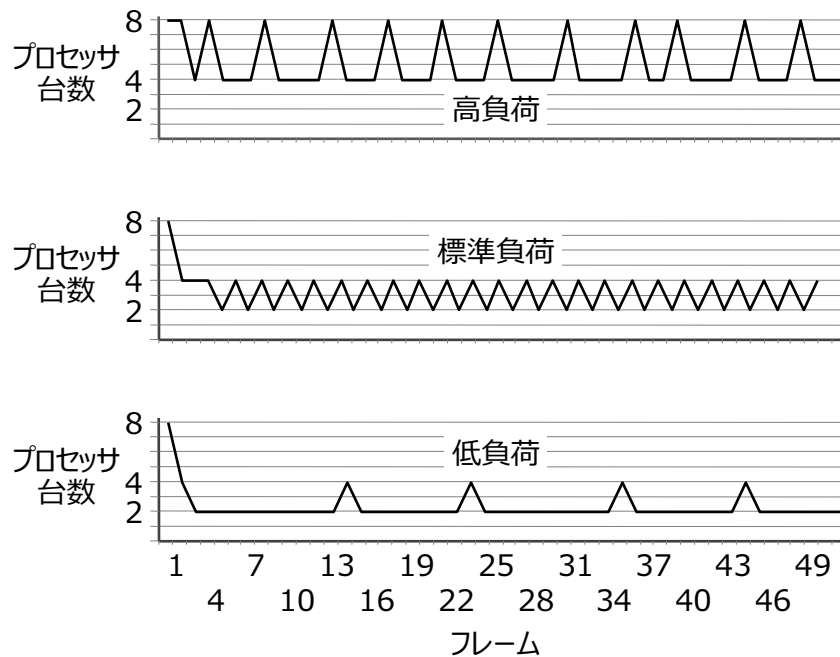


図 4.22 プロセッサ数の動的な変更による消費電力量の最小化

この場合、約 90% のフレームが 2 プロセッサで、10% が 4 プロセッサで動作するので、残りのプロセッサで MPEG-4 以外を実行できれば、MPEG-4 デコーダとしては 18.2% の消費電力量となる。同様に、標準負荷では 26.2% となり、ほぼ最適な消費電力量となるコア数で動作することが示される。

(4) 電源遮断による停止処理の実現

高い動作周波数のプロセッサでは、リーク電流のためスリープ時の消費電力量が 20% 程度 [20] にもなって、多くの台数のプロセッサがスリープを用いて FFV 制御することは現実的ではない。消費電力をより小さくできる電源遮断の機構を用いると、[21] によるように電源遮断実行時に、再開アドレスを設定してレジスタ等のコンテキスト情報を内蔵メモリに退避して復帰時のコヒーレンシ保証に備えキャッシュをパージし、復帰時はリセットによって再開アドレスから回復するようにすることで、スリープに替わる消費電力の少ない制御を実現できる。[21] では消費電力を最大時の 2.4%、キャッシュのパージも含めた遮断処理のオーバヘッドを数 μsec とできている

ので、本稿で 1.5% の消費電力としたスリープ処理としてこの電源遮断を適用できると考えられる。

キャッシュパージの実行時間は大きいですが、電源遮断は一周期の同期待ちでプロセッサごとに 1 回だけ生じるものであるため、本章の例のようなものでは、これによるオーバヘッドは問題にならないと考えられる。

4.5 マルチプロセッサにおける DVFS 制御の関連研究

マルチプロセッサで DVFS 制御を行うために負荷や残り実行時間を見積もる方式が提案されており、処理の進捗状況を見る地点をタスクやスレッドの同期ポイントとし、並列に動作する複数のスレッドがその同期ポイントで待ち合わせを行う時間を活用して各要素プロセッサの動作周波数・電源電圧を降下させる。

[29] では、繰返し処理において、並列動作するプロセッサそれぞれが繰返し判定のバリア同期ポイントに到達した時点で、他のプロセッサを待つストール時間に低動作周波数・電源電圧で動作させ、次の繰返しでの動作周波数と電源電圧を調整する。そのために、現在の繰返しに要した時間とストール時間を用い、その比率を次の負荷の見積りとする動的な解析を行っている。一方 [14] では、実行順序保証のためプロセッサ間の同期処理を行うマクロタスクをスケジューリングした時に得られる実行時間とマクロタスクの同期処理でのデッドライン時刻とから負荷を評価して、動作周波数と電源電圧を制御している。ここでの負荷の見積りは、コンパイラによって静的に推定されるものである。

両者ともプロセッサ間の同期処理を行う地点（同期ポイント）だけで動作周波数と電源電圧の制御を行うが、本方式では同期ポイントまでのプログラム区間を複数のセグメントに分割した地点ごとに、セグメントの実行時間を用いて進捗状況を見積もり、プロセッサそれぞれで同期処理とは無関係に動作周波数と電源電圧を制御している。

上記や本方式でのソフトウェアによる制御と異なり、[27] では実行中にハードウェアによって設定したインターバル間隔ごとにバス競合のサイクル数を得て、それがプロセッサ間で等しくなるように動作周波数と電源電圧を制御している。プロセッサが

備える性能監視機構から得られる情報を用いて性能予測を行い、OS のスケジューラによって動作周波数を制御する方式も提案されている [28].

また、負荷の評価について、特にマルチプロセッサ構成を意識していないが、コンパイル時に対象プログラムを静的に解析してプログラムの区間ごとの実行サイクルを見積もる方式について、[15] ではプロファイルを採取して動作周波数と電源電圧を決定しており、[22] ではシミュレーションによってトレースログを採取し、その履歴から最悪の実行サイクルをより正確に見積もっている。

バリアやマクロタスクの境界といったスレッドの同期ポイントだけで DVFS 制御を行う [14] や [29] に比べ、本方式では同期ポイント以外でも、スレッドをより小さな部分に分割したセグメントで、実行履歴によって進捗状況を予測して DVFS 制御するため、4.4 節に示すように消費電力量の削減効果を上げている。さらに、負荷に応じて割り当てるコア数を変更できることが利点である。

4.6 まとめ

要素プロセッサごとに電源電圧と動作周波数を制御するマルチコアアーキテクチャにおいて、以下のようにアプリケーションとその実行状況に応じて最適な動作条件を設定することによって動作周波数と電源電圧を制御して動作時の消費電力量を低減できる。

- アプリケーションをスレッドに分割して並列処理することで、各スレッドを実行するプロセッサを台数に見合った動作周波数と電源電圧で実行する。
- 加えて、各スレッドでプログラムをさらに分割したセグメントごとに実行履歴に基づく負荷状況に応じて必要十分な動作周波数と電源電圧で実行する。

このため、プログラム内に設けたチェックポイントによって動的に動作周波数と電源電圧を制御する方式をスレッドに対応させてマルチプロセッサに適用し、処理の進行情報を与えて各要素プロセッサが独立して DVFS 制御するマルチコア FFV 制御方式を提案した。本アルゴリズムは、画像や音声のような周期的でデータ依存性の少な

いメディア処理に適している。前回の負荷に基づいて余裕時間を見積もることにより、少ないチェックポイント数でプロセッサ台数の増加や高負荷に対して有効であることや、実行履歴で進捗を管理するためセグメントの区切りを比較的自由に採れ、マクロブロックのような単位がなくてもセグメント分割でき、同期待ちに制約されずに動作周波数・電源電圧の制御地点を採れることや事前にプロファイルなどを取得する必要のないことが本方式の利点である。

オンチップ・マルチプロセッサでプロセッサの停止状態として低電力のスリープや電源遮断機構を利用することにより、FFV 制御でマルチメディアアプリケーションに対して省電力効果を発揮できる。また、[9]での実行例や本章でのシミュレーションにおいて、FFV 制御によって動作周波数を低下させたために、処理全体としてデッドラインを超過することはなかった。

シミュレーション結果によれば、提案するマルチコア FFV 制御方式によって並列プロセッサの台数ならびに各プロセッサ内での処理負荷の変動に応じて DVFS 制御することで、シングルプロセッサでの実行と比較して 70% 以上の消費電力量低減効果を得ることができる。低周波数域では動作周波数の低減に見合った電源電圧低減が見込めず、消費電力量は低減できないので、4.4.3 項 (3) の方式で各プロセッサの負荷によって動的に実行プロセッサ数を 8 台、4 台、2 台と調節することで上記の低減効果が見込めることになる。

なお、以上の議論はプログラム全体がほぼ均一な負荷の処理に並列分割できることを前提としているが、例えば MPEG デコーダでは逆量子化のようなプログラムの中の高負荷な機能単位に着目して、それが効果的な負荷目標に収まるように機能分散してプロセッサ割付けを行うような制御も考えられる。

今後、一般的になるマルチプロセッサで各スレッドを実行するプロセッサの動作周波数と電源電圧を制御し、さらに、OS が管理するようなプロセッサの性能情報とも連携して各要素プロセッサ内でもセグメントごとに動作周波数と電源電圧を制御することにより、タスクやセグメントそれぞれに最適周波数と電圧を設定するマルチコア FFV 制御でマルチメディアアプリケーションに対して高い省電力効果を発揮できると考える。

関連研究で述べたように、並列実行される複数プロセッサにおいて、各プロセッサで独立に動作周波数・電源電圧を設定する制御手法が提案され始めており、FV制御のマルチプロセッサ適用への期待は高いといえる。

また、前章のまとめでも述べたように、閾値電圧を制御することなど [30, 31] によってリーク電流も FFV 制御で同様に低減でき、負荷に応じて動作周波数を下げて閾値電圧を上げることによって、リーク電流による消費電力を低減できる。

リーク電流の影響が増大することやプロセッサ動作の電源電圧自体が十分低くなって制御できる電圧レンジが減少することを考えると、多数の動作周波数を用いて FFV 制御を行うことでは消費電力量削減への効果を大きくできない。そこで、電源電圧と閾値電圧を組み合わせることで最高周波数動作とその 1/2 で FFV 制御し、スリープや電源遮断の制御を行って、マルチプロセッサによる並列処理と FFV 制御を相補作的に適用することが動作時消費電力量の低減にとって有効であると考えられる。

第 4 章の参考文献

- [1] 前島 英雄：“RISC + マルチ VLIW アーキテクチャ融合マイクロプロセッサ”，
STARC シンポジウム 2001 講演予稿集，pp. 50–59（2001-9）
- [2] 添田 真史，伊藤 智之，高橋 司，前島 英雄：“マスタ・スレーブ型マルチプロ
セッサのソフトウェア制御方式”，電子情報通信学会 エレクトロニクスソサイエ
ティ大会講演論文集，C-12-13，p.83（2004-9）
- [3] 石野 公一，高橋 司，前島 英雄：“マスタ・スレーブ型マルチプロセッサのプロ
セッサ間接続方式”，電子情報通信学会 エレクトロニクスソサイエティ大会講演
論文集，C-12-12，p.82（2004-9）
- [4] 内山 邦男：“ディジタル民生用マイクロコンピュータにおける低電力設計”，電
子情報通信学会論文誌 C，Vol. J83-C，No. 6，pp. 447–453（2000-6）
- [5] 十山 圭介，久保田 瞬，山本 修一郎，前島 英雄：“マルチコアでのメディア処理
におけるプロセッサ動作時消費エネルギー低減方式の提案”，映像情報メディア
学会誌，Vol. 66，No. 3，pp. J74–J84（2012-3）
- [6] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos: “Data Driven Signal Process-
ing: An Approach for Energy Efficient Computing”, *Proceedings of International
Symposium on Low Power Electronics and Design Conference 1996*, pp.347–352,
Aug. 1996.
- [7] T. Ishihara and H. Yasuura: “Voltage Scheduling Problem for Dynamically Variable
Voltage Processors”, *Proceedings of 1998 International Symposium on Low Power*

- Electronics and Design Conference*, pp. 197–202, Aug. 1998.
- [8] 大隈 孝憲, 石原 亨, 安浦 寛人: “可変電源電圧プロセッサに対するリアルタイムタスクスケジューリング手法”, 電子情報通信学会和文論文誌 C, Vol. J83-C, No. 6, pp. 454–462 (2000-6)
- [9] 十山 圭介, 三坂 智, 相坂 一夫, 在塚 俊之, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康: “CPU 消費電力削減のための周波数-電圧協調型電力制御方式の設計ルールとフィードバック予測方式による適用”, 電子情報通信学会和文論文誌 D-I 分冊, Vol.J87-D-I No. 4, pp. 452–461 (2004-4)
- [10] S. Lee and T. Sakurai: “Run-time Voltage Hopping for Low-power Real-time Systems”, *Proceedings of 37th Design Automation Conference 2000*, pp.806–809, Jun. 2000.
- [11] K. Aisaka, T. Aritsuka, S. Misaka, K. Toyama, K. Uchiyama, K. Ishibashi, H. Kawaguchi, and T. Sakurai: “Design Rule for Frequency-Voltage Cooperative Power Control and Its Application to an MPEG-4 Decoder”, *IEEE Solid-State Circuits Society 2002 Symposium on VLSI Circuits Digest of Technical Papers*, pp. 216–217, Jun. 2002.
- [12] S. Misaka, K. Toyama, T. Aritsuka, K. Uchiyama, K. Aisaka, H. Kawaguchi, and T. Sakurai: “Frequency-Voltage Cooperative Power Reduction for Multi-tasking Multimedia Applications”, *IEEE International Symposium on Low-Power and High-Speed Chips (COOL Chips VI)*, Vol. I, pp. 103–116, Apr. 2003.
- [13] 小高 剛, 中野 啓史, 木村 啓二, 笠原 博徳: “OSCAR チップマルチプロセッサ上での MPEG2 エンコードの並列処理”, 情報処理学会研究報告 2004-ARC-160, Vol. 2004, No. 123, pp. 53–58 (2004-12)
- [14] 白子 準, 吉田 宗弘, 押山 直人, 和田 康孝, 中野 啓史, 鹿野 裕明, 木村 啓二, 笠原 博徳: “マルチコアプロセッサにおけるコンパイラ制御低消費電力化手法”, 情報処理学会論文誌: コンピューティングシステム, Vol.47, No. SIG 12 (ACS

- 15), pp.147–158 (2006-9)
- [15] A. Azevedo, I. Issenin, R. Cornea, R. Guputa, N. Dutt, A. Veidenbaum, and A. Nicolau: “Profile-based Dynamic Voltage Scheduling using Program Checkpoints”, *Proceedings of Conference on Design, Automation and Test in Europe 2002*, pp.168–175, Mar. 2002.
- [16] 須賀 敦浩, 今井 賢, 田中 篤志, 早川 文彦, 多胡 真一郎: “FR1000 におけるチップマルチプロセッサアーキテクチャの紹介”, *情報処理 Vol.47, No.1*, pp. 24–28 (2006-1)
- [17] E. Iwata and K. Olukotun: “Exploiting Coarse-grain Parallelism in MPEG-2 Algorithm”, *Technical Report CSL-TR-98-771*, Sep. 1998.
- [18] T. Yamada, M. Ishikawa, Y. Ogata, T. Tsunoda, T. Irita, S. Tamaki, K. Nishiyama, T. Kamei, K. Tatezawa, F. Arakawa, T. Nakazawa, T. Hattori, and K. Uchiyama, “A 133MHz 170mW 10 μ A Standby Application Processor for 3G Cellular Phones”, *IEEE International Solid-State Circuits Conference 2002 Digest of Technical Papers*, pp. 370–371, Feb. 2002.
- [19] M. Ishikawa, T. Kamei, Y. Kondo, M. Yamaoka, Y. Shimazaki, M. Ozawa, S. Tamaki, M. Furuyama, T. Hirose: F. Arakawa, O. Nishii, K. Hirose, S. Yoshioka, and T. Hattori, “A 4500 MIPS/W, 86 μ A Resume-Standby, 11 μ A Ultra-Standby Application Processor for 3G Cellular Phones”, *IEICE Transactions on Electronics*, Vol.E88-C, No.4, pp.528–535, Apr. 2005.
- [20] M. Ito, T. Hattori, Y. Yoshida, K. Hayase, T. Hayashi, O. Nishii, Y. Yasu, A. Hasegawa, M. Takada, M. Ito, H. Mizuno, K. Uchiyama, T. Odaka, J. Shirako, M. Mase, K. Kimura, and H. Kasahara: “An 8640MIPS SoC with Independent Power-Off Control of 8 CPUs and 8 RAMs by an Automatic Parallelizing Compiler”, *IEEE International Solid-State Circuits Conference 2008 Digest of Technical Papers*, pp. 90–91, Feb. 2008.

- [21] M. Onouchi, K. Toyama, T. Nojiri, M. Sato, M. Mase, J. Shirako, M. Sato, M. Takada, M. Ito, H. Mizuno, M. Namiki, K. Kimura, and H. Kasahara: “Green Multicore-SoC Software-Execution Framework with Timely-Power-Gating Scheme”, *Proceedings of International Conference on Parallel Processing 2009*, pp. 510–517, Sep. 2009.
- [22] 立松 知紘, 横山 哲郎, 菊地 武彦, 富山 宏之, 高田 広章: “組込みシステムのタスク内 DVFS のための実行トレースマイニング”, 電子情報通信学会技術報告 VLD2008-128, Vol. 108, No. 478, pp. 11–16 (2009-3)
- [23] “シングルコアよりマルチコア”, 日経エレクトロニクス 2004.8.30 (no. 896), pp. 96–121 (2004-8)
- [24] 枝廣 正人, 鳥居 淳, 酒井 淳嗣, 井上 浩明, 山端 潔, 石川 晋也, 伊藤 義行, 鈴木 悟, 西 直樹: “マルチコア向けソフトウェア・プラットフォームを開発し携帯電話機に適用”, 日経エレクトロニクス 2005.3.28 (No. 881), pp. 125–136 (2005-3)
- [25] 吉田 正康: “組み込み分野における「マルチプロセッサ」とは, — 多機能・低消費電力の要求にこたえるための技術要素と課題”, デザインウェーブマガジン, Vol. 10, No. 8, pp. 68–74 (2005-8)
- [26] A. Jerraya, H. Tenhunen, and W. Wolf: “Multiprocessor System-on-Chips”, *IEEE Computer*, Vol. 38, Issue 7, pp. 36–40, Jul. 2005.
- [27] 近藤 正章, 中村 宏: “CMP 向け動的電源電圧・周波数制御手法”, 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. SIG 13 (ACS 19), pp. 260–269 (2007-8)
- [28] 金井 遵, 佐々木 広, 近藤 正章, 中村 宏, 天野 英晴, 宇佐美 公良, 並木 美太郎: “性能予測モデルの学習と実行時性能最適化機構を有する省電力化スケジューラ”, 情報処理学会論文誌: コンピューティングシステム, Vol. 49, No. SIG 2 (ACS 21), pp. 20–36 (2008-3)

-
- [29] C. Liu, A. Sivasubramaniam, M. Kandemir, and M. J. Irwin: “Exploiting Barriers to Optimize Power Consumption of CMPs”, *Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium 2005*, pp. 5–14, Apr. 2005.
- [30] 野瀬 浩一, 平林 雅之, 川口 博, 李 誠洙, 桜井 貴康: “閾値ホッピング (V_{TH} -hopping) 手法を用いた低電圧・低消費電力プロセッサ”, 電子情報通信学会技術研究報告 ICD2001-33, Vol. 101, No. 85, pp. 67–73 (2001-5)
- [31] 桜井 貴康: “リーク電流はこう抑える (7) V_{th} 制御と V_{dd} 制御の併用でリーク電流削減効果を最大化”, 日経エレクトロニクス, 2005.4.25 (no. 898), pp.121–126 (2005-4)

第 5 章

結 論

現在、マイクロプロセッサが使用される組込みシステムの主要なものは携帯電話や PDA に代表されるモバイル情報端末、DTV などのデジタル家電などであるが、今後ユビキタスコンピューティングの時代となり、様々な機器にマイクロプロセッサが多数搭載されて、そのそれぞれが高度な処理を行うようになる。多様な方面で高度なデジタル化が進み、そこで使用されるマイクロプロセッサへの高性能化と低消費電力化の要求がさらに高まる。これらの機器に使用されるマイクロプロセッサシステムを設計・開発するにあたっては性能・価格・消費電力のバランスをうまくとることが重要で、特に性能すなわち処理速度を保ったまま、あるいは向上させながら、それと背反して環境に優しく使用される電力量（エネルギー）を低減する意味で、低消費電力を実現する設計がさらに重要になってくる。

本研究ではこの問題に一つの解を与えるべく、組込みシステムにおけるマイクロプロセッサの動作時の消費電力量を低減する方式について、ソフトウェアから電力制御を行うという観点から検討している。消費電力量の削減効率を解析的に考察することによりプロセッサの動作周波数と電源電圧を協調して低減させる周波数・電圧協調制御（FV 制御）方式を提案し、さらにそれを、予め立てた実行スケジュールとの差分をフィードバックさせて適切な動作周波数を設定するアルゴリズムを用いたフィードバック型周波数・電圧協調制御（FFV 制御）方式として実装し、その効果を考察した。

第1章では低消費電力が要求される背景と低消費電力化へのアプローチや従来技術を概観し、ソフトウェアによる電力制御、すなわちアプリケーションの特性に応じてソフトウェアの側からプロセッサの動作時消費電力量を制御するという本研究の位置づけを述べた。

それに基づき、第2章でプロセッサ動作時に消費される電力量の削減方式について考察し、CMOS 回路の消費電力に大きな影響を与える電源電圧と動作周波数をソフトウェアから制御する FV 制御方式を対象として取上げた。そして、消費電力量低減のために有効な制御方式として、実行プログラムの負荷変動に応じて動作周波数と電源電圧とを動的に変更して最適な消費電力量を得るものについて、その原理と消費電力量の削減効果を検討・評価して、妥当な動作周波数および電源電圧決定のルールを定めた。ここでは、解析的な考察によって、最適な消費電力量から 10% 程度の効率

低下を許容することで FV 制御システムを実装するために用意すべき動作周波数群を決定するアルゴリズムを与えている, LSI チップに適用するアルゴリズムでは, 最小動作電圧を考慮して評価し, 最高周波数からその $1/2$, $1/4$, ... となる系列で動作周波数を採択して行くことが有効であることを示している. さらに, この動作周波数採択の系列を打ち切る条件を明確化し, 最近の低電圧で動作するマイクロプロセッサチップでは, FV 制御を行うために採択すべき動作周波数は最高周波数にその $1/2$ のものを 1 個追加すれば概ね十分であることを示した. また, このルールを適用して商用のプロセッサで容易に FV 制御が設計できることも述べている.

FV 制御を適用する処理単位にはアプリケーション全体やタスクなどがあるが, プログラムをその断片に細分したプログラムセグメント (セグメント) ごとに制御すると, よりプログラムの実行特性を反映した消費電力制御が行えると考えられる. そこで, 第 3 章ではセグメントを用いた FV 制御を検討し, 携帯電話のリファレンスプラットフォームとなる実機ボード上で実装し評価・検証した.

第 2 章で得た動作周波数と電源電圧を決定する設計ルールをフィードバック型制御 (FFV 制御) 方式として制御対象のアプリケーションプログラムに反映させた. FFV 制御では例えば, MPEG-4 の動画処理における処理の単位マクロブロックをセグメントとして 1 画面分を最悪時間で実行するとし, あらかじめ立てた実行スケジュールと実際の処理時間との差分によって処理負荷を見積もる. この FFV 制御方式を MPEG-4 デコーダにおいてモバイル向けの実プロセッサに適用した場合, 消費電力量が 27% に低減できることをシミュレーションにより評価した. さらに, この方式が MP3 デコーダでも想定どおりに機能して低消費電力化できることを確認し, MPEG-4 と MP3 デコーダ両者を同期して並行動作させるアプリケーションで, その動画・音声再生プログラム全体の消費電力量が 28% に低減できることを前述の実機で確認した. これらにより, 本 FFV 制御方式が, 同期して稼動する一般的なマルチタスクアプリケーションにおいて実用上有効であることを示している.

今後, 組込みシステムにおいてもマルチプロセッサが潮流となると考えられる. 第 4 章ではマルチプロセッサにおいて並列処理と負荷に応じた制御を行って要素プロセッサに要求される動作周波数を低下させ, アプリケーション実行に最適な動作周波

数と電源電圧を設定することにより，プロセッサ動作時の消費電力量を低減する方式について考察した。

第2章と第3章で示したように，動作周波数・電源電圧制御を適切に行うと，周波数と電圧の2乗とが電力低下に寄与するためプログラム動作時のプロセッサ消費電力量を大幅に削減できる．並列処理によって要素プロセッサに割り当てられる動作周波数・電源電圧を低下させ，さらに，各要素プロセッサ内のプログラムセグメント毎にFFV制御方式によって実行スレッドでの動作周波数・電源電圧を低下させることにより，柔軟で効果の高い動作時の低消費電力化が実現できると期待される。

このため，プログラム内に設けたチェックポイントによって動的に動作周波数と電源電圧を制御する方式を並列動作するスレッドに対応させてマルチプロセッサに適用し，処理の進行情報を与えて各要素プロセッサが独立してFFV制御するマルチコアFFV制御方式を提案した．この方式は画像や音声のようなマルチメディア処理に適しており，プロセッサ台数の増加や高負荷に対して有効であって，同期待ちなどに制約されずに動作周波数・電源電圧の制御地点を採れることや事前に実行プロファイルなどを取得する必要のないことが利点となる．動作周波数と電源電圧に加えて，制御対象としてプロセッサの停止状態である低電力のスリープや電源遮断機構を利用することにより，高い省電力効果を発揮でき，MPEG-4デコードでの実行データを用いたシミュレーションによって，提案する制御方式が有効に機能することを示し，4台，8台並列時に70%程度の消費電力量の削減効果が得られている。

以上のように，本研究では実行時の負荷状況に応じて動作周波数と電源電圧を協調して低減し，それによってマイクロプロセッサの動作時の消費電力量を低減するソフトウェアの制御とその実現方式を提案し，有効性を評価した．シングルプロセッサシステムと並列アーキテクチャを提案するマルチプロセッサシステムに対して，携帯電話を想定したアプリケーションにおいてFFVおよびマルチコアFFV制御方式を実装し，実機とシミュレーションで消費電力量の削減効果を示している。

これまでは，微細加工が進むにつれて電源電圧が下がり低消費電力化も達成できたが，現在の最先端のマイクロプロセッサでは電源電圧が限界まで下がってきており，またリーク電流が消費電力に占める割合が多くなるなど，電源電圧だけを対象とする

FV 制御方式そのままでは、第3章で適用したような電源電圧降下による大きな消費電力削減効果を得ることが困難になる。

しかし、負荷に応じてプロセッサの速度を制御する方式自体は有効に機能する場が存在する。例えば閾値電圧の制御が可能なプロセッサにおいて、電源電圧が限界値になって低下させられなくなった場合に、FV 制御と同様のアルゴリズムを適用して処理負荷の軽い区間で動作周波数を下げるとともに、閾値電圧を高くする。これにより、リーク電流を減少させて消費電力を低減することができる。今後、リーク電流の問題がますます大きくなっていくが、そのような状況でも FV 制御が有効で、電源電圧や閾値電圧、また電源遮断機構などを組み合わせて動作時消費電力量を低減することができると考えられる。

将来、さらに微細化したプロセスでマイクロプロセッサの高性能化を進めると消費電力の問題がこれまで以上に大きな壁となる。これを克服するためにはマイクロプロセッサの応用システムを良く把握し、ソフトウェア、アーキテクチャ、論理・回路など多方面から総合的に技術開発を進め、協調して検討することが必要となる。本研究で提案したマルチプロセッサ対応 FFV 制御方式がその一つとして利用されることを期待する。

謝 辞

本論文をまとめるにあたって、指導教官として東京工業大学 大学院総合理工学研究科 物理情報システム専攻 前島 英雄 教授のご指導とご鞭撻をいただき、辛抱強く数々の懇切なご助言を賜りました。また、同専攻 羽鳥 好律 教授、小林 隆夫 教授、佐藤 誠 教授、杉野 暢彦 准教授には、本論文の内容につきまして貴重なご討論とご助言を賜りました。ここに心から感謝の意を表します。

本研究は著者が株式会社日立製作所 中央研究所と社会人博士課程の学生として在籍した東京工業大学 大学院総合理工学研究科で行ったものです。

著者が日立製作所 中央研究所において、システム LSI 向けソフトウェアおよび本研究を行う機会とご指導、ご支援をいただきました歴代の中央研究所 所長である 科学技術振興機構 中村 道治 理事長、株式会社 日立総合計画研究所 武田 英次 研究顧問、日立製作所 西野 壽一 執行役常務、日立オートモティブシステムズ株式会社 福永 泰 CTO、日立製作所 日立研究所 小島 啓二 所長、日立製作所 中央研究所 長我部 信行 現所長の各氏をはじめ、歴代の同所センタ長である 日立ハイテクノロジーズ 田地 新一 執行役常務、日立ハイテクノロジーズ 木村 勝高 執行役、日立製作所 研究開発本部 尾内 享裕 本部長、日立製作所 中央研究所 中川 八穂子 シニアプロジェクトマネージャ、日立製作所 中央研究所 山足 公也 現センタ長の皆様に厚くお礼申し上げます。

著者所属部署の歴代の部長、ユニットリーダーである佐々木 勝朗 氏、日立製作所 中央研究所 矢野 和男 主管研究長、同 入江 直彦 企画室長、同 水野 弘之 チーフプロジェクトマネージャ、同 長田 健一 主任研究員、同 近藤 伸和 主任研究員、同 山田 哲也 主任研究員、同 菅野 雄介 主任研究員、日立製作所 情報・通信システム社 伊藤

雅樹 部長の各氏には本研究遂行において多大なるご支援をいただきました。

日立製作所 研究開発本部 内山 邦男 技師長には研究の様々な面や学位取得に関してご指導やご助言をいただき、また、本テーマの共同研究先であった東京大学 生産技術研究所 桜井 貴康 教授、神戸大学 工学部 川口 博 准教授に多くの有益なご指導とご討論をいただきました。

ここに記して皆様に深甚なる謝意を表します。

本研究の推進にあたっては、研究開始時に日立製作所 中央研究所 主任研究員であった相坂 一夫 氏をはじめとするユニット員 日立製作所 中央研究所 三坂 智 研究員、同 在塚 俊之 主任研究員、同 松本 孝浩 研究員、同 小高 俊之 主任研究員や同 小野内 雅文 研究員、ルネサスエレクトロニクス株式会社 茶木 英明 執行役員、同 館内 嗣治 主管技師、電気通信大学 大学院情報理工学研究科 石橋 孝一郎 教授、また、東京工業大学 大学院総合理工学研究科 山本 修一郎 助教や久保田 瞬 氏（現在、株式会社 NTT データ）をはじめとする前島研究室の皆様など、多くの方々に多大なご協力、ご援助をいただきました。関係の諸兄に深く感謝いたします。

最後に、常によき理解者で協力者であります妻 綾子に感謝します。

本研究に関する論文・発表文献

【学術論文】

- [1] 十山 圭介, 久保田 瞬, 山本 修一郎, 前島 英雄：
“マルチコアでのメディア処理におけるプロセッサ動作時消費エネルギー低減方式の提案”，
映像情報メディア学会誌, Vol. 66, No. 3, pp. J74–J84 (2012-3)
- [2] 十山 圭介, 三坂 智, 相坂 一夫, 在塚 俊之, 内山 邦男,
石橋 孝一郎, 川口 博, 桜井 貴康：
“CPU 消費電力削減のための周波数-電圧協調型電力制御方式の設計ルールとフィードバック予測方式による適用”，
電子情報通信学会和文論文誌 D-I 分冊, Vol.J87-D-I, No. 4, pp. 452–461 (2004-4)

【国際会議】

- [1] Satoshi Misaka, Keisuke Toyama, Toshiyuki Aritsuka, Kunio Uchiyama, Kazuo Aisaka, Hiroshi Kawaguchi, and Takayasu Sakurai:
“Frequency-Voltage Cooperative Power Reduction for Multi-tasking Multimedia Application”,
International Symposium on Low-Power and High-Speed Chips (COOL Chips VI), Vol. I, pp. 103–116, Apr. 2003.

【その他の発表】

- [1] 十山 圭介, 入江 直彦, 西井 修, 中澤 拓一郎:
“ユビキタス情報時代の超低消費電力 LSI 技術アーキテクチャとシステムソフトウェア”,
日立評論 Vol. 84, No. 10, pp.17–20 (2002-10)
- [2] 十山 圭介 :
“システム LSI の低電力化技術” (パネル討論),
電子情報通信学会集積回路研究専門委員会 (ICD) 第 6 回システム LSI ワーク
ショップ, p. 147 (2002-11)
- [3] 十山 圭介, 岡山 祐孝 :
“システム LSI –アプリケーションと技術 – 第 2 篇 システムの共通要素技術第 10 章 ソフトウェア”, pp. 269-280,
株式会社 サイエンスフォーラム (1999-7)

本研究に関するその他の論文・発表文献

【学術論文】

- [1] Keisuke Toyama, Satoshi Misaka, Kazuo Aisaka, Toshiyuki Aritsuka, Kunio Uchiyama, Koichiro Ishibashi, Hiroshi Kawaguchi, and Takayasu Sakurai:
“Frequency-Voltage Cooperative CPU Power Control: A Design Rule and Its Application by Feedback Prediction”,
Systems and Computers in Japan, Vol. 36, No. 6, pp. 39–48, Jul. 2005. John Wiley & Sons, Inc.
- [2] Tetsuya Yamada, Naohiko Irie, Takanobu Tsunoda, Takahiro Irita, Kenji Kitagawa, Ryohei Yoshida, Keisuke Toyama, and Motoaki Satoyama:
“A Hardware Accelerator for JavaTM Platforms on a 130-nm Embedded Core”,
IEICE Transactions on Electronics, Vol. E90-C, No. 2, pp. 523–530, Feb. 2007.

【国際会議】

- [1] Masafumi Onouchi, Keisuke Toyama, Toru Nojiri, Makoto Sato, Masayoshi Mase, Jun Shirako, Mikiko Sato, Masashi Takada,

- Masayuki Ito, Hiroyuki Mizuno, Mitaro Namiki, Keiji Kimura, and Hironori Kasahara:
“Green Multicore-SoC Software-Execution Framework with Timely-Power-Gating Scheme”,
Proceedings of International Conference on Parallel Processing 2009, pp. 510–517, Sep. 2009.
- [2] Tetsuya Yamada, Naohiko Irie, Takanobu Tsunoda, Takahiro Irita, Kenji Kitagawa, Ryohei Yoshida, Motokazu Ozawa, Toru Hiraoka, Saneaki Tamaki, Keisuke Toyama, Motoaki Satoyama, Osamu Nishii, Fumio Arakawa, Shinichi Yoshioka, Toshihiro Hattori, and Kunio Uchiyama:
“A Hardware Accelerator for Java™ Platforms on a Microprocessor Core for Mobile Applications”,
International Symposium on Low-Power and High-Speed Chips (COOL Chips VII), Vol. I, pp. 255–266, Apr. 2004.
- [3] Kazuo Aisaka, Toshiyuki Aritsuka, Satoshi Misaka, Keisuke Toyama, Kunio Uchiyama, Koichiro Ishibashi, Hiroshi Kawaguchi, and Takayasu Sakurai:
“Design Rule for Frequency-Voltage Cooperative Power Control and Its Application to an MPEG-4 Decoder”,
IEEE Solid-State Circuits Society 2002 Symposium on VLSI Circuits Digest of Technical Papers, pp. 216–217, Jun. 2002.

【研究会・大会論文、他】

- [1] 三坂 智, 十山 圭介, 在塚 俊之, 内山 邦男, 相坂 一夫, 川口 博, 桜井 貴康:
“マルチタスク実装マルチメディアに対する周波数-電源電圧協調型電力制御”,
電子情報通信学会技術研究報告 [集積回路] ICD2003-40, Vol. 103, No. 216,
pp. 7–12 (2003-7)

- [2] 入江 直彦, 山田 哲也, 近藤 雄樹, 石川 誠, 津野田 賢伸, 十山 圭介, 山田 孔司, 西本 順一, 館澤 健, 中澤 拓一郎, 川崎 郁也, 内山 邦男 : “次世代携帯電話向けアプリケーションプロセッサ技術”, 電子情報通信学会技術研究報告 [集積回路] ICD2002-173, Vol. 102, No. 525, pp. 53–58 (2002-12)
- [3] 相坂 一夫, 在塚 俊之, 三坂 智, 十山 圭介, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康 : “周波数-電圧協調型省電力制御におけるアルゴリズムとデザインルール”, 電子情報通信学会技術研究報告 [集積回路] ICD2002-72, Vol. 102, No. 274, pp. 49–52 (2002-8)
- [4] 相坂 一夫, 在塚 俊之, 三坂 智, 十山 圭介, 内山 邦男, 石橋 孝一郎, 川口 博, 桜井 貴康 : “周波数-電圧協調型電力制御における使用周波数群決定方法の提案と MPEG-4 デコーダによる検証”, 電子情報通信学会技術研究報告 [集積回路] ICD2002-37, Vol. 102, No. 234, pp. 13–16 (2002-7)
- [5] 近藤 伸和, 十山 圭介, 里山 元章, 北井 克佳 : “SuperH マイコンを用いたネットワーク コンピュータ”, 日立評論 Vol. 80, No. 7, pp. 37–40 (1998-7)

【特許】

- [1] 桜井 貴康, 辛 英洙, 川口 博, 相坂 一夫, 十山 圭介, 石橋 孝一郎 : “電力制御装置及び方法並びに電力制御プログラム”, 特開 2003-140787, 平成 13 年 11 月 6 日出願