

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

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^{††}

Frequency-Voltage Cooperative CPU Power Control: A Design Rule and Its Application by Feedback Prediction

Keisuke TOYAMA[†], Satoshi MISAKA[†], Kazuo AISAKA[†], Toshiyuki ARITSUKA[†],
Kunio UCHIYAMA[†], Koichiro ISHIBASHI[†], Hiroshi KAWAGUCHI^{††},
and Takayasu SAKURAI^{††}

あらまし 周波数-電圧協調型電力制御 (FV 制御) は, ソフトウェアの負荷情報を利用してハードウェアの動作条件を動的に制御するため, プログラム動作時の CPU の消費電力を低減する効果大きい. 本論文では, この FV 制御の方式と効果を解析的に導いて, 必要となる周波数決定のルールを得, フィードバック型のアルゴリズムを提案してそれによる FV 制御の実装について示した. 2 種の周波数と電圧によって MPEG-4 及び MP3 デコーダに適用し, 動作時の CPU 消費電力の 72% を削減している. 更に, FV 制御の開始時を低周波数とする Cool-Start 方式で電力削減効果が向上することを示した.

キーワード 周波数-電圧協調型電力制御 (FV 制御), フィードバック型制御, MPEG-4, MP3

1. ま え が き

LSI の集積規模が増大するに従って電力消費が設計における重要課題となりつつある. 特にモバイル応用では影響が大きく, 電力消費の大きな要因である CPU の消費電力低減のため, ソフト・ハード両面から多様な手法が提案されている. その一つとして, 動画のように処理負荷が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クロック周波数と電源電圧とを同時に低下させる「周波数-電圧協調型電力制御 (以下, FV 制御)」がある [1], [2]. FV 制御では, 周波数と電源電圧の 2 乗が電力低下に寄与するため大幅にプログラム動作時の CPU 消費電力を削減できる.

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, プログラムを細分した部

分 (プログラムスライス) ごとに制御すると, よりプログラム特性を反映した電力制御が行えるため, 我々はプログラムスライスへの FV 制御を中心に検討している [3], [4].

我々は, 解析的な考察により, 周波数と電圧を決定する設計ルールを与え [5], [6], それを MPEG-4 デコーダに適用した. 更にこの方式が MP3 デコーダでも想定どおりに機能し低電力化することを確認した.

2. 動的周波数-電圧制御の背景

2.1 基本概念

CPU の消費電力 P は,

$$P = \alpha \times C \times F \times V^2$$

α : スイッチング確率, C : 静電容量,

F : 動作周波数, V : 電源電圧

で表される. これらの要因を低減することが消費電力の削減につながり, 特に V を低減させると効果が大きい. 多値電圧制御可能な CPU において, F と V とを制御することによる電力消費に関する基本性質は

[†] (株) 日立製作所中央研究所, 国分寺市

Central Research Laboratory, Hitachi, Ltd., 1-280 Higashi-Koigakubo, Kokubunji-shi, 185-8601 Japan

^{††} 東京大学国際・産学共同研究センター, 東京都

Center for Collaborative Research, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8904 Japan

次のように示されている [2] .

- 処理を最小の消費電力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 .
 - CPU の電源電圧が離散的に複数個用意されている場合 , 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が最小になる .
- この性質を実際のシステム設計に適用するためには , 以下の点の考慮が必要である .

(1) CPU には動作可能な下限電圧が存在し , この下限電圧でも周波数を低くすることは可能である .

(2) 処理量をもとに電源電圧への影響を反映させるためには , $F-V$ の関係を表現できることが必要である .

このため , 我々は動作周波数 F を主たる制御パラメータとし , F と消費電力 P との関係で解析することにした .

2.2 周波数-電力関係のモデル化

まず , CPU の周波数-電力関係 ($F-P$ 関係) を図 1 のとおり定めた . これは , ある CPU 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 では曲線は多様なパラメータに左右されるであろうが , これを式 (1) によってモデル化した .

$$P = \begin{cases} kF^\gamma & (F \geq F_m) \\ k'F & (F \leq F_m) \end{cases} \quad (1)$$

ただし k, k' は適当な比例定数

式 (1) の $F-P$ 曲線は二つの部分からなる . 曲線の左半は原点から始まる直線であり , 右半は次数 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

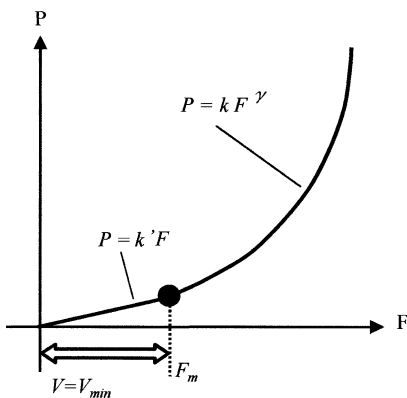


図 1 $F-P$ 関係のモデル
Fig.1 Modeling of $F-P$ relation.

り , これは当該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 (1) のパラメータ γ で代表させる . このモデルには物理的な裏付けはないが , 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 .

3. 離散 FV 制御

3.1 周波数を離散化した場合の $F-P$ 関係

図 1 及び式 (1) は , CPU の動作周波数 (及び電圧) を連続的に変化させられると仮想した場合の $F-P$ 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 , 周波数及び電源電圧は離散化されている , すなわち , 図 2 のように F_1, F_2, F_3 のみがクロック発生回路で発生でき , 電源電圧は各々の周波数に対応する最低動作電圧が用意されるものとする . この場合 , $F-P$ 関係は図 2 の破線のようにもとの $F-P$ 曲線を F_1, F_2, F_3 の点で線形内挿した折線状になる [1], [2] . 折れ線ともとの $F-P$ 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 , クロック周波数を離散化したことによる消費電力への影響を知ることができる . 我々は , クロックを離散化しても十分な消費電力削減効果を得られることを示す .

当該 CPU があるプログラムを実行するために理想的な最低周波数 F_i が必要になると仮定する . もし周波数 (及び電圧) を任意の値に設定できれば , 図 2 の

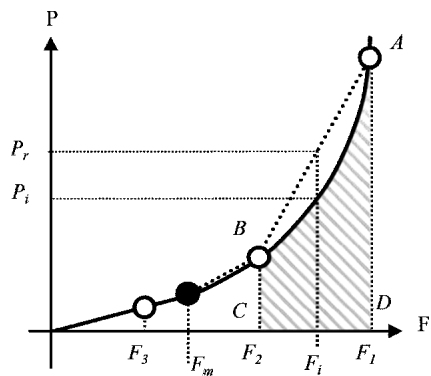


図 2 周波数離散化の影響
Fig.2 Influence of discrete frequency.

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電力となる。ここで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 に相当する周波数を F_1 と F_2 の組合せで平均として実現すると、周波数減少に比例した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電力に対する電力損が相対的に求められる。

3.2 電力損の評価

以下の仮定及び定義に基づいて、前記の電力損を具体的に与える。まず、図 2 における最高周波数 F_1 は、当該 CPU 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 $F_1/F_2, F_2/F_3, \dots$ を用いるものとし、この比の値を β とおく。

3.2.1 最大電力損

前節で述べたとおり、周波数を離散化した場合のある周波数 F_i における消費電力の損失（連続に変化させられる場合の最低電力に対する相対値）は図 2 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 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、

$$P_r = \frac{kF_1^\gamma - kF_2^\gamma}{F_1 - F_2} + \frac{F_1 \cdot kF_2^\gamma - F_2 \cdot kF_1^\gamma}{F_1 - F_2},$$

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、

$$\frac{P_r}{P_i} = \frac{\alpha^\gamma(\beta - \beta^\gamma) + \alpha^{\gamma-1}(\beta^\gamma - 1)}{\beta - 1} \quad (2a)$$

折れ線が F_m をまたぐ場合（図 2 の $F_3 \sim F_2$ に F_i が存在する場合）は、

$$\frac{P_r}{P_i} = \frac{\alpha^\gamma(K\beta - \beta^\gamma) + \alpha^{\gamma-1}(\beta^\gamma - K)}{\beta - 1} \quad (2b)$$

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

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 (2a) は、式 (2b) 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 $K = 1$ とおいた場合に帰着される。よって以

下では式 (2b) のみを検討する。

式 (2b) 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は、 F_i すなわち α の変化に対する式 (2b) 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 (2b) を α で微分することにより、

$$\alpha = (\gamma - 1)(\beta^\gamma - K) / \gamma\beta(\beta^{\gamma-1} - K) \quad (3)$$

において得られることが分かる。式 (3) を式 (2b) に代入することにより、特定の β 及び K に対する最大損失が得られる。

3.2.2 平均電力損

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 をもつ部分プログラム（これをスライスと呼ぶ）から構成される場合の平均的な電力損を評価する。

当該 CPU が対象とする時間内で n 個のプログラムスライスを実行し、各々の実行に要する最低周波数は $F_i(1), F_i(2), \dots, F_i(n)$ 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は各プログラムスライスに必要な電力の和となるから、電力損は次の形で計算できる。

$$\frac{\sum_{i=1}^n P_r(F_i(\cdot))}{\sum_{i=1}^n P_i(F_i(\cdot))} \quad (4)$$

この分母は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 の斜線部分、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。

$$\frac{\sum P_r}{\sum P_i} = \frac{(\gamma + 1)(\beta^\gamma + 1)(\beta - 1)}{2(\beta^{\gamma+1} - 1)} \quad (5a)$$

これは図 2 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。

$$\begin{aligned} \frac{\sum P_r}{\sum P_i} = & \frac{\rho^{\gamma-1}(\gamma + 1) \left(1 + \frac{1}{\beta}\rho^{\gamma-1}\right) \left(1 - \frac{1}{\beta}\right)}{(\gamma + 1) \left(\frac{1}{\rho^2} - \frac{1}{\beta^2}\right) + 2\rho^{\gamma-1} \left(1 - \frac{1}{\rho^{\gamma+1}}\right)} \quad (5b) \end{aligned}$$

表 1 電力損の具体値

Table 1 Example value for power loss ratio.

(a) Interval $[F_p, F_i]$		β	γ	1.5	2.0	2.5	3.0
1.5	1.5	1	3	5	8		
		2	4	8	13		
2.0	2.0	3	7	13	20		
		5	13	24	41		
3.0	3.0	6	15	27	40		
		12	33	69	126		

(b) Interval $[F_p, F_i]$ $F_m = (F_3 + F_j) / 2$		β	γ	1.5	2.0	2.5	3.0
1.5	1.5	3	6	9	13		
		6	12	19	26		
2.0	2.0	5	11	17	24		
		10	22	36	52		
3.0	3.0	9	18	28	39		
		17	38	63	94		

Upper row: Average loss (%)
Lower row: Maximum loss (%)

$$\text{ただし } \rho = \frac{F_2}{F_m}$$

式 (5a) は、式 (5b) において F_m が区間の下端に位置した場合であり、 $\rho = \beta$ とお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 (スライス) 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 CPU 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 CPU 消費電力を削減するもので、本 FV 制御の基本的なモデルである。

3.2.3 具体的な例

(1) F_m を固定した場合

表 1 は周波数比 β 及び $F-P$ 関係のパラメータ γ を変化させた場合の前記の電力損の計算結果である。表の各欄において、上段は式 (5b) による平均電力損を、下段は式 (2b) 及び (3) による最大電力損を示す。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す。表 1(a) は F_m が評価対象の区間に含まれない場合、表 1(b) は含まれる場合であり、代表例として $F_m = (F_2 + F_3) / 2$ の場合、つまり「継目」が区間の中点にある場合を示した。

実測によると実際の CPU ではほぼ $\gamma = 2.0$ であり、また CPU に用意されている周波数変更の機構を用いると $\beta = 2.0$ すなわち最大周波数の 1/2 を用いて制御することが簡便で効果が期待できるので、これが典型例と考えられる。表 1 のとおり、(a) の場合には $\beta = 2.0$ ととれば、 $\gamma = 2.0$ の典型的な例に対して平均損は 7%、最大損は 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

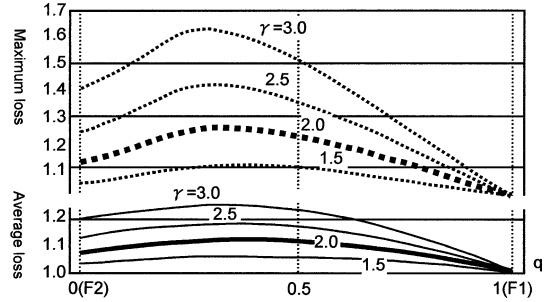


図 3 F_m の変化を考慮した電力損評価
Fig. 3 Power Loss estimation with F_m altered.

11%にとどまる。

(2) F_m の変化を考慮した電力損

次に F_m を変化させた場合の相対電力損の値を図 3 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 が相対的にどの位置にくるかを示すものであり、 $q = 0 (1)$ は $F_m = F_2 (F_1)$ 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 CPU がすべて最低動作電圧で動作せねばならず、FV 制御の実効的な意味がなくなること示している。

図 3 のとおり、電力損は $q = 0.3 \sim 0.4$ 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。

3.3 設計ルール

以上の解析と評価をもとに、プログラムの要求周波数に応じて FV を制御して最適な電力消費を得るための設計ルールを求める。

$\gamma = 2.0$ の典型的なチップにおいて、設定すべき隣接する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の損失は、3.2.3 で示したように 10% 前後である。したがって、最大周波数の 1/2 で次の周波数とそれを実現する電圧を設定し、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 CPU 消費電力を削減できる。電源電圧が更に低減できるならば、更に 1/2 で次の周波数 (及び、それを実現する電圧) と、1/2 の比率で周波数をとっていけば理想の 10% 程度の損失で電力削減できる周波

数の組が得られ、FV 制御における使用周波数の決定は下記のルールとなる：

FV 制御システムの設計において、最高周波数 F_1 が与えられた場合、 F_1 以下の周波数は、 $F_2 = 1/2F_1, F_3 = 1/4F_1, F_4 = 1/8F_1, \dots$ と用意する。

上記で用意された周波数の組によって、理想の最低周波数 F_i での実行を実現してプログラムを実行すればよい。

現実のチップにおいては図 1 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、ただかかもう 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周波数が F_m 以下ならば、更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。そこまで得られた周波数群で制御し、更に時間に余裕が生じると周波数 $F = 0$ とするのである。近年チップの低電圧化が進む結果、このアルゴリズムは $F_2 = 1/2F_1$ を求めた時点で、チップの電圧がそれ以上下げられなくなり、打ち切りとなることが多いと思われる。言い換えれば、チップに追加する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 $= 0$ がチップ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よいが、多くの CPU 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等）が用意されているので、それを利用することができる。

3.4 実 CPU への適用

前節の設計ルールを実際の CPU に適用する。

(1) SH-Mobile1 [7]

まずチップの最高動作周波数 F_1 及び対応する電源電圧は 120 MHz 及び 1.5 V とした。パラメータ γ は 1.9 と求められ、ルールに従い FV 制御で使用する周波数は、 $F_2 = 60$ MHz、これに対応する電源電圧は 1.25 V となる。この電圧は同チップの動作保証の範囲外であるが、実験により稼働が確認されている。また本設計においては 1.25 V を V_{min} とみなすことにした。したがって $F_2 = 60$ 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、二つの周波数 120 MHz, 60 MHz 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。

表 2 Crusoe への適用例

Table 2 Presumed frequency on Crusoe.

Original design [8]		This design	
F [MHz]	V [V]	F [MHz]	V [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) Crusoe [8], [9]

Transmeta 社の Crusoe は、FV 制御機構を内蔵した CPU として先進的な位置付けにある [8], [9] によれば同チップには 5~7 個の周波数-電圧の組が用意されており、5 組のものを対象とする（表 2）。表 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 1.35 V が妥当である。 F_2 は F_m に達しておらず、 $F_3 = F_2/2 = 233$ MHz が追加され、 F_3 では最低電源電圧 0.9 V 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 組の周波数-電圧（及びスリープモード）により、FV 制御が可能になる。

4. FV 制御のシステム適用

4.1 フィードバック型 FV 制御

前章までの解析と設計ルールを実際の CPU とそのプログラムに適用し、得られた周波数と電圧の組で、プログラムの負荷に応じた理想周波数 F_i での実行を近似して消費電力削減を実現する。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で、周波数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 [3]。

フィードバック型 FV 制御では、制御されるプログラムが以下の性質をも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/時間がほぼ分かっている。

この条件のもとでは、MPEG 動画デコーダを例とし

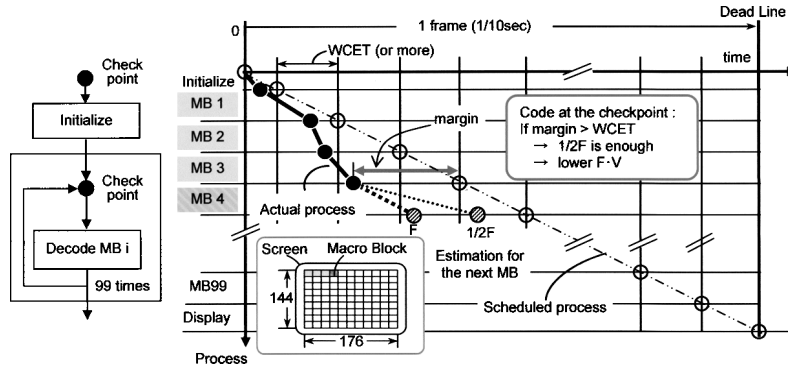


図 4 MPEG-4 デコーダの処理プロセス
Fig. 4 Decoding process of MPEG-4.

て図 4 に示すような制御が可能となる。

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及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 4 左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), 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 の処理を 3.2.2 のスライスとして FV 制御を適用する。図 4 右のグラフは、その処理の進捗状況である。

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 2 点鎖線のように、すべての MB に 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超えない、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 2 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

テップに通常の数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、実行クロック周波数を 1/2 に落としても安全であると判断できる。同様に、余裕が WCET の 2, 3, ..., n 倍よりも大きければ、周波数を 1/3, 1/4, ..., 1/($n + 1$) に落しても安全である。逆に余裕が少なければ、周波数をもとに戻す必要がある。

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周波数の設定をフィードバック制御と呼ぶ。

4.1.1 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、動作を評価した。3.4(1)の結果により、120 MHz/1.5 V, 60 MHz/1.25 V の周波数・電圧の組を用いて低周波数動作を行い、更に実行時間が余る場合 CPU をスリープモードとする。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 個のスライスに分割してスライスごとに余裕時間を判定し、余裕がある場合に Low へ、余裕がない場合に High へと変化させる。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 を 100 ms として計算している。

このシステムに典型的な MPEG-4 動画ストリーム

表 3 シミュレーションによる電力評価
Table 3 Simulated power reduction.

relative workload	#frames	#FV change	High mode (%)	Low mode (%)	Sleep (%)
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	14.00	14.50
- 90	0	0	0	0	0
- 100	0	0	0	0	0
relative power (%)			11.67	14.50	0

データ（平均負荷率 31.0%）を入力した場合の挙動を表 3 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 CPU の最大性能の何%が必要となるかを示すものである。表 3 では、例えば CPU 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 回の周波数変更が行われており、実行時間のうち 8.76%が High 動作（120 MHz/1.5V）、26.06%が Low 動作（60 MHz/1.25V）、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 relative power（相対消費電力）を求める。これにより消費電力は制御がない場合と比べて 26%に低減されると計算できる。ここでは Low 動作において相対消費電力が High 動作時の 1/3、スリープ時の電力が 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削減率を得た。

4.2 MP3 デコーダへの適用

MPEG-4 に対するものとほとんど同様の制御を MP3 デコーダに組み込んで、MP3 に対するこの FV 制御の電力効果を検証した。MPEG-4 ではマクロブロックの処理を単位としてスライスに分割し制御したが、MP3 でも、そのようなプログラム構造に従って必ず通過する地点（左右チャンネル及びグラニュール処理の開始地点）を選択し 8 個のスライスに分割して周波数、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 MHz/1.5 V、60 MHz/1.25 V の周波数・電圧の組を用いて周波数・電圧の切換を行い、更に実行時間が余る場合 CPU をスリープモードとする。

表 4 トレースデータによる電力評価

Table 4 Simulated power reduction by trace data.

Appearance ratio (%)	#FV change	High mode (%)	Low mode (%)	Sleep (%)
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

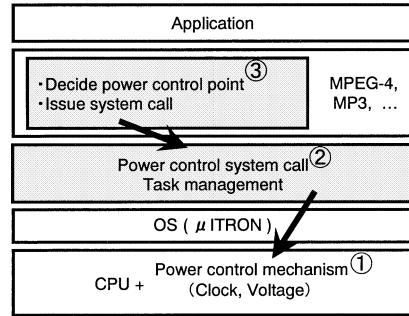


図 5 システム構成
Fig. 5 System architecture.

表 5 フレーム内での FV 変更回数

Table 5 No. of FV changes in frame execution.

#FV change	1	3	5	7	9	11	13	15	-	27
#frames	139	27	12	2	2	3	1	1	0	1

(a) MPEG-4

#FV change	1	2	3
#frames	171	2	26

(b) MP3

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 では次節に示す実験時のトレースデータを採取して電力評価した。199 音声出力フレームのデータを表 4 に示す。大部分（86%）のフレームは 1 回の周波数及び電圧変更であることが分かる。

このデータから、MP3 において消費電力は、制御がない場合と比べて 15.3%に低減されると計算できる。

4.3 実験結果

上記の FV 制御を実装するターゲットシステムとして、図 5 のシステム構成で日立製 SH-Mobile1 組み込みのシステムボード（SolutionEngine）を用い、これに ①の電源電圧制御回路を付加して二つの電源電圧に変更可能とした。周波数変更は PLL 出力の分周比率を変更することで行う。制御する周波数は、3. での考察に

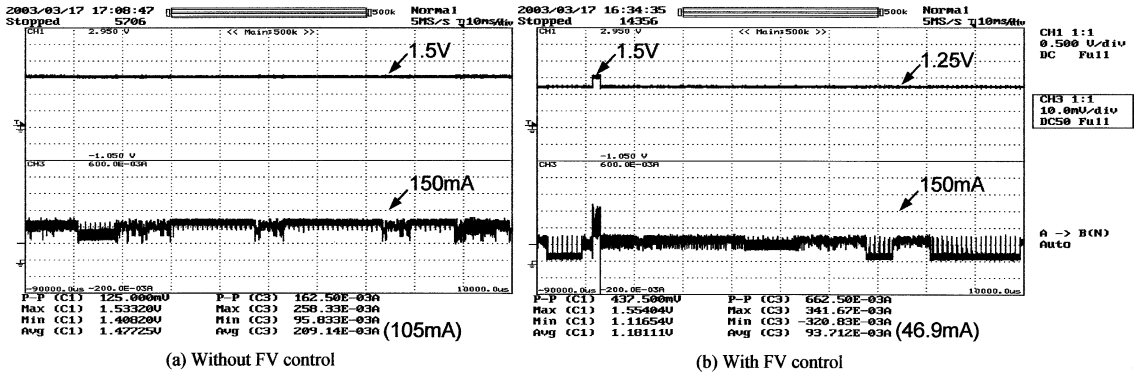


図 6 実行時の電圧・電流波形
Fig. 6 Voltage and current wave forms during execution.

より最大周波数とその 1/2 (120 MHz/60 MHz), 電圧はその速度を得る最小電圧 (1.5 V/1.25 V) としている。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 (1/2 周波数) と SetFV_High (最大周波数) として②で実装し, OS (μ ITRON) 上で MPEG-4 ビデオデコーダと MP3 デコーダを同期して稼働させている。図 5 の③Decide power control point が図 4 での Check point に対応する。

表 5(a) に MPEG-4 で処理された 188 フレームでの F, V の変更回数を示す。Sub-QCIF サイズ, 10 fps の動画再生においては, 74% のフレームが 1 回の変更であり, 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, 周波数を低下させて余裕時間がなくなりフィードバックによって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 5(b) は MP3 デコーダの 199 フレームでの FV 変更回数である。表から分かるように, MP3 においては大部分のフレームが 1 回の FV 変更でよい。更に, その変更は 2 スライス目のものである。実際, この MP3 はすべて 60 MHz で稼働させても大丈夫なものである。

動画・音声再生時の電圧, 電流波形は図 6 に示すとおりであり, FV 制御をしないもの (a) の平均消費電力 230 mW と比較して FV 制御を行ったもの (b) は 65 mW となり, 消費電力を 28% に削減できている。

4.4 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処理で開始時点は High 状態の周波数・電圧であるとしてきたが, CPU 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 状態で開始 (Cool-Start) してもよい。各スライスが同じ時間で実行されるものとする, 本 FV 制御の場合 MPEG-4 で 1%, MP3 で 13% 程度余裕をみた設計ができていれば, 2 スライス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い, MP3 は実質的にすべて Low 動作とできている。Low 動作時の消費電力を High の 1/3 として, MP3 の場合 Cool-Start なしに比べ 5.9% 消費電力が減少する。

表 5 を得たデータによると, MPEG-4 では 1 回目の FV 変更は平均 19 スライスで発生している。Cool-Start によりこれが 1 スライス分削減できたとすると, 1 回だけの FV 変更が支配的だとして 1.4% 消費電力の減少を見込める。Cool-Start は MP3 に適用したように分割スライス数が小さい場合に, 削減効果を上げる有効な方式である。

5. む す び

本論文では, FV 制御システムに用意すべき周波数の決定ルールを解析的な電力損評価により導いた結果, 周波数は 1/2 によるルールに基づいて用意すれば実用上十分であることを示した。更に, チップの最小動作電圧を考慮して詳細評価した結果このルールが有効であることを確認した。また, この 1/2 ルールの適用を打ち切る条件を明確化し, 最近の低電圧チップでは動作電圧のレンジが狭くなり, 最大周波数の 1/2 を実現する動作電圧以下の低減はできず, FV 制御のための低速周波数は最大周波数の 1/2 を 1 個追加すれば十分であることを示した。

更に、同ルールに基づくFV制御システムをMPEG-4, MP3 デコーダに適用して稼働させ、MPEG-4 デコーダの消費電力が26%に低減できること、MPEG-4 とMP3 デコーダ両者を同期して並行動作させるプログラムで、この動画・音声再生プログラム全体の消費電力が28%に低減できることをシミュレーションと実機で確認し、本FV制御方式が実用上有効であることを示した。

謝辞 本研究に関し、(株)ルネサステクノロジ 茶木氏、館内氏、近藤氏、徳永氏、(株)日立製作所 江崎氏、(株)日立超 LSI システムズ 柴田氏、茂木氏、三嶋氏、田村氏、並びに御討論、御支援頂いた関係各位に感謝致します。

文 献

- [1] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data driven signal processing: An approach for energy efficient computing," ISLPED, pp.347-352, 1996.
- [2] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," ISLPED, pp.197-202, 1998.
- [3] S. Lee and T. Sakurai, "Run-time voltage hopping for low-power real-time systems," DAC, pp.806-809, June 2000.
- [4] 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," 2002 Symp. VLSI Circuits, pp.216-217, June 2002.
- [5] 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," 信学技報, ICD2002-37, July 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アルゴリズムとデザインルール," 信学技報, ICD2002-72, Aug. 2002.
- [7] 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 133 MHz 170mW 10 μ A standby application processor for 3G cellular phones," ISSCC Digest of Technical Papers, pp.370-371, Feb. 2002.
- [8] Transmeta Corporation, "Crusoe Processor Model TM5500/TM5800 Data Book V1.0," Feb. 2003.
- [9] Transmeta Corporation, "Processor Product Brief Model TM5800," Feb. 2003.

(平成 15 年 4 月 17 日受付, 8 月 26 日再受付)



十山 圭介

1980 京都大学大学院情報工学専攻修士課程了, 同年(株)日立製作所入社。システム開発研究所, 中央研究所でコンパイル技術及び最適化コンパイラ技法, Java システム, システム LSI 向け低消費電力ソフトウェアの研究開発に従事。1987~1988 Copenhagen 大学 Research Associate。情報処理学会会員。



三坂 智 (正員)

1994 東京電機大学大学院情報通信工学専攻修士課程了。同年(株)日立製作所中央研究所入社。MPEG-4 標準化, 及び組込みマイコン用 Audio/Visual ミドルウェアの再利用化, 低消費電力化の研究開発に従事。



相坂 一夫

1982 東京大学大学院情報工学専攻修士課程了。1985(株)日立製作所中央研究所入社, 医療用画像システムの研究開発に従事。1998 よりシステム LSI 研究部にて, 組込みマイコン向けソフトの生産技術・省電力制御方式の研究開発に従事。現在,(株)日立製作所中央研究所企画室主任技師。



在塚 俊之 (正員)

昭 60 上智大・理工・電気電子卒。昭 62 同大学院博士前期課程了。同年(株)日立製作所中央研究所入社。主として音響, 音声情報処理, 及びシステム LSI の研究に従事。平成 3(株)ATR 視聴覚機構研究所研究員。現在,(株)日立製作所中央研究所主任研究員。



内山 邦男 (正員)

1976 東工大・理・情報科学卒, 1978 同大学院情報科学科修士課程了, 同年(株)日立製作所中央研究所に入社。中央研究所にて CAD, 小型メインフレーム, キャッシュ LSI の研究を経て, マイクロプロセッサの研究開発に従事。1985~1986 米国 Carnegie Mellon 大学客員研究員。第 30 回市村賞, 1999 年度 R&D100, 2000 年度科技厅長官賞受賞。2001 東工大博士(工学)。



石橋孝一郎 (正員)

1985 東工大・総合理工学研究科・博士課程了。博士(工学)。同年(株)日立製作所中央研究所入社以後、高集積 SRAM、低電力マイコン、低電力回路技術の研究開発に従事。2001 より(株)半導体理工学研究センター・設計技術開発部・低電力技術開発室長。1999R&D100 受賞。IEEE Senior Member.



川口 博

平 5 千葉大学大学院電子工学専攻了。工修。同年コナミ(株)入社、技術研究部においてアーケードゲーム機の設計開発に従事。主に大型体感ゲーム、ポリゴンゲームの設計開発を行う。平 8 より東京大学生産技術研究所技術官、現在助手。低消費電力 VLSI 設計の研究に従事。IEEE, ACM メンバー。



桜井 貴康 (正員)

昭 56 東京大学電子工学専攻博士課程了。工博。(株)東芝入社、CMOS メモリ、高速プロセッサ、世界初の MPEG2 用 LSI などを研究開発。昭 63 から平 2 まで U.C.Berkeley にて LSI CAD の研究。平 8 より東京大学教授。高速、低消費電力 LSI 設計の研究に従事。VLSI 回路シンポジウム委員長、ASPAC 副委員長ほか、ISSCC, CICC, DAC, ICCAD, FPGA workshop, ISLPED, TAU, ICVC などの国際学会の技術委員。IEEE AdCom メンバー、IEEE フェロー、IEEE distinguished lecturer。