

特徴の説明書

1. タイトル

75%電力節減可能な離散 FV 制御機構を有する低電力リアルタイム OS: μ ITRON-LP

2. 応募部門

ソフト設計資産部門

3. 分野

低消費電力技術

4. 所属機関

東京大学

5. 所属部署

生産技術研究所

6. 応募者名

川口 博

7. 連絡先住所

153-8505 東京都目黒区駒場 4-6-1

8. 電話番号

03-5452-6253

9. FAX 番号

03-5452-6632

10. E-mail アドレス

kawapy@iis.u-tokyo.ac.jp

11. 協力機関名あるいは協力者名

株式会社日立製作所, 相坂氏, 十山氏, 石橋氏.

日立米沢電子株式会社, 八巻氏.

研究費については日立製作所, 日本学術振興会.

12. 特徴の説明の要約

低電力型 μ ITRON

75%電力節減可能な離散 FV 制御機構を有する低電力リアルタイム OS: μ ITRON-LP

川口 博, *辛 英洙, 桜井 貴康

東京大学生産技術研究所

*IBM T. J. Watson Research Center

1. はじめに

バッテリーで駆動される携帯機器には低電力と高性能の両立が求められる。プロセッサの低電力化のためにクロック周波数と電源電圧を制御することによって電力の最適化を図る方法 [1]-[6] が提案されている。これらはクロック周波数 f を境界条件として与え、この条件下でどこまで電源電圧 V_{DD} を下げることができるか、という命題をハードウェアで解明している。すなわち所望の f を与えるとハードウェア帰還により V_{DD} がプロセッサに供給される。これに対しプロセッサの高速化にともないソフトウェアの重要度が増している背景からソフトウェアによる低電力設計手法 [7]-[14] も脚光を浴びている。

この論文では市販プロセッサを用いた組込システム上でのリアルタイム OS とタスクの協調による低電力化手法について述べる。昨年応募した電圧ホッピング [15]-[18] には OS は実装されておらず、そのため単一のタスクに対してのみ有効であったが、このリアルタイム OS への拡張と考えられる。 μ ITRON をベースにし、複数のタスクが実行される環境でもプロセッサの消費電力の最適化を実現できる。これを μ ITRON-LP と呼ぶ。

μ ITRON-LP は定電源電圧動作のオリジナル μ ITRON に比べ 1/4 以下にまで電力を節減できる。電力節減は実時間性を損なうことなく実現される。

2. μ ITRON-LP

2.1 アプリケーションスライシング

画像や音声のようなマルチメディアタスクの場合、タスクは特定の周波数 (例えば 30Hz や 44.1kHz) に同期して起動され、周期的である。周期的タスクはこの起動周期と起動周期より小さい値を持つ WCET (Worst Case Execution Time: 最悪実行時間) とを有する。WCET が起動周期より小さいことによって実時間性は維持される。また MPEG に代表されるようなアプリケーションでは、入力画像データによって処理速度が変化することが知られている。具体的には被写体の動きが顕著である場合や背景の変化が大きくなるに従って負荷が重くなる。しかしながら画像の変化が大きくなるに従って負荷が無限に大きくなるわけではなく、理論上の最大負荷が決まっている。このような理論上の最大負荷がタスクの WCET に対応する。実際のタスクでは最大負荷のデータを処理することはほとんどなく、大抵の場合には WCET 以前にタスクを終了し、時間余裕が発生する。またタスクの起動周期は WCET より長いので、最大負荷のデータを処理しても時間余裕が発生する場合もある。

μ ITRON-LP ではこのような時間余裕が存在することをタスクの実行中に検知して、プロセッサの f と V_{DD} を制御 (FV 制御) することによって、消費電力の最適化を目指す。そこでタスク内では電圧ホッピングと同様にアプリケーションスライシングを導入する。

いま単一のタスクのみが走っていると仮定し、アプリケーションスライシングを図 1 を用いて簡単に説明する。要点は現在時刻と次のスライスを実行すべき時刻までの余裕を調べ、最適な f を決定することである。タスクは N 個のスライス 1, 2, 3, ..., N に分割されている。各々のスライスの WCET と各スライスからスライス N (最終スライス) までの WCET は解析や測定を通じて取得される [19]。ここでスライス 2 について考える。

スライス 2 の WCET を t_2 、スライス 3 からスライス N までの WCET を t_{3-N} 、スライス 2 の開始からタスクの次の起動時刻までの時間を V_d とする。今回のタスクの実行を次の起動時刻までに行う必要があるため、タスクの次の起動時刻 V_d がタスクの締切となる。スライス 3 からスライス N までの処理が時間 t_{3-N} で確実に終わるので、スライス 2 を実行するための時間余裕 t_R は $V_d - t_{3-N}$ となる。したがってスライス 2 では理想的には f を基本周波数 f_{clk} の t_2/t_R 倍まで下げることができる。

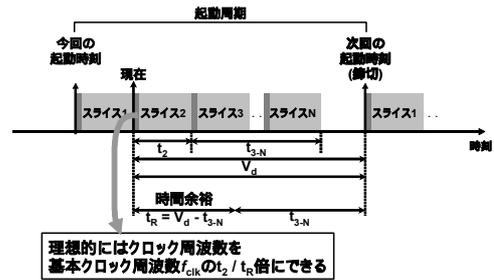


図 1. アプリケーションスライシング。

実際には f を任意の値に設定するのは困難であるため、 f_{clk} の t_2/t_R 倍を下回らない最小の $1/n$ 倍に離散的に設定される。例えば t_2/t_R が 0.48 の場合、 f は $f_{clk}/2$ となる。これによりプロセッサと外部デバイスとのインターフェイスの問題を回避できる。ここで問題は複数のタスクが走る環境で締切をどう提供するかである。以下にこれに対する解決策として μ ITRON-LP について述べる。

2.2 モデル

図 2 は μ ITRON-LP とその周辺との関係を示すモデルである。 μ ITRON-LP は起動時刻キュー、実行優先度キュー、FV ルックアップテーブルを有する。またタスクとのインターフェイスとしてスケジューラと特別なシステムコールハンドラが実装されている。

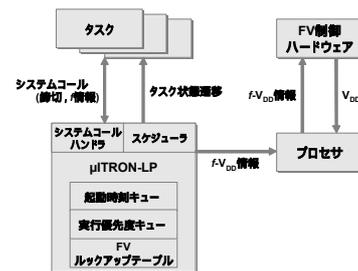


図 2. μ ITRON-LP のモデル。

起動時刻キューは全タスクを次の起動が行われる時刻の順番に保持し、タスクが起動または終了される度に更新される。ここで起動時刻キューの先頭のタスク (今後最も早く起動されるタスク) の起動時刻を第 1 起動時刻と呼ぶ。実行優先度キューは実行中と実行可能なタスクを実行優先度の高い順番に保持する。FV ルックアップテーブルは f と V_{DD} を最適化するための参照可能な関係を有する。

スケジューラは管理するタスクのスケジューリングを行い、後述のとおりタスク状態を遷移させる。システムコールハンドラはタスクとの間で情報の受渡しを行う。アプリケーションはシステムコールを用いて締切を得て、その締切からどのように f を変化させるかをアプリケーションスライシングに基づき計

算する。その後 f に関する情報を μ ITRON-LP にシステムコールで伝える。 μ ITRON-LP は得られた f に関する情報を用いて FV ルックアップテーブルからプロセッサに供給すべき V_{DD} に関する情報を得る。 f と V_{DD} 双方の情報はプロセッサの I/O を経由し FV 制御ハードウェアに伝えられ、プロセッサの f と V_{DD} が制御させる。

2.3 タスク状態

タスク状態は図 3 に示すとおりスケジューラによって遷移される¹。もしタスクが RUN 状態にあるならば、タスクは現在実行中であり、現在それがプロセッサを占有している。もしタスクが READY 状態にいるならば、タスクは実行可能であるが、より高い優先度を持った他のタスクが RUN 状態にあるためにプロセッサ割り付けを待機していることを意味する。前述どおり RUN 状態または READY 状態にあるタスクの実行優先度は高い順で実行優先度キューに保持されている。実行中のタスクが終了またはタスクが起動された場合、実行優先度キューは更新される。スケジューラは実行優先度キューの先頭のタスクをプロセッサに割り付ける。もしタスクが DORMANT 状態にあるならば、タスクの実行がすでに終了し、タスクが休止中であり、次の起動を待機している。スケジューラは単位時間ごとに起動時刻キューを調べ、休止中のタスクを起動させるか否かも判断する。タスクが起動された場合、スケジューラは実行優先度キューを更新し、現在実行中のタスクの優先度と起動直後のタスクの優先度とを比較する。起動直後のタスクの優先度が実行中のタスクの優先度より高い場合、実行中のタスクはプロセッサ待機を強要され、実行すべきタスクが入れ替えられる。入れ替えられたタスクは READY 状態となり、再びプロセッサ割り付けを待機することになる。

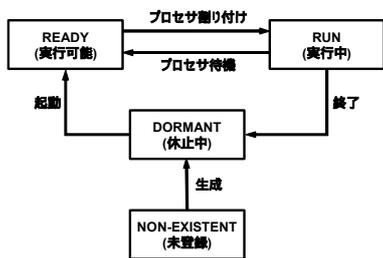


図3. タスク状態遷移。

2.4 締切取得

上記で述べた基本動作以外の μ ITRON-LP 特有の機能は前述どおり RUN タスクに締切を与えることと RUN すべきタスクがない場合にプロセッサをスリープモードにすることである。

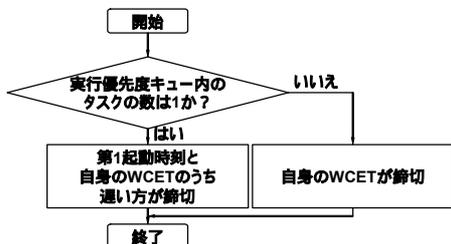


図4. 締切取得ルーチン。

図 4 に締切取得ルーチンを示す。実行優先度キューに存在する実行中のタスクの個数が 1 であるか否かをシステムコールで判断する。実行優先度キューに実行中のタスクしか存在しない場合、すなわち実行優先度キューに存在するタスクが 1 個である場合、プロセッサ割り付けを待機するタスクが他に存在しないことを意味する。その結果、タスクは少なくとも

第 1 起動時刻までプロセッサを独占することができる。このためこの第 1 起動時刻をシステムコールにより参照し、実行中のタスクの WCET (正確にはタスク全体の WCET から現在までの積算実行時間を減算し、現在の時刻を加算したもの、積算実行時間もシステムコールにより参照可能である。)と比較する。そして遅い方を締切とした場合でもタスク処理に関して影響がない。

一方、実行優先度キューに実行中のタスク以外のタスクが存在する場合、すなわち実行優先度キューに依存するタスクが 2 個以上ある場合、締切を実行中のタスクの WCET とすべきである。これによってシステムの実行確実性が維持される。

なおもし全てのタスクが DORMANT 状態にあり、RUN すべきタスクがない場合は第 1 起動時間までプロセッサをスリープさせる。

2.5 例

複数の周期的タスクの例²として図 5 に示すように実行優先度=1, WCET=6 単位時間, 周期=20 単位時間であるタスク A と実行優先度=2, WCET=12 単位時間, 周期=30 単位時間であるタスク B と実行優先度=3, WCET=2 単位時間, 周期=40 単位時間であるタスク C が存在するものとする。タスク A, B, C の各々の矩形はスライスを表し、1 スライスは 2 単位時間を有する。縦方向にクロック周波数を示し、横方向に時間経過を示す。

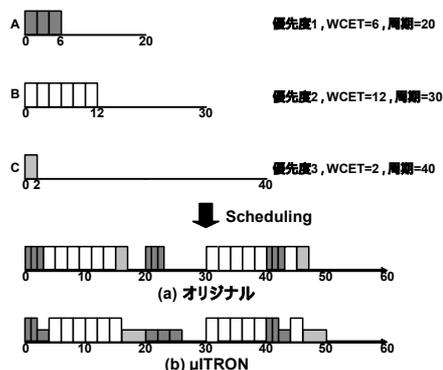


図5. タスク A, B, C のスケジューリングの例。(a)オリジナル。(b) f が 2 段階に制限されている場合の μ ITRON-LP。スライスの高さは f を表す。

タスク A, B, C の実際の実行時間は負荷に依存する。この例の場合、タスク A の負荷は 50% であると仮定し、実行時間を WCET の半分とする。またタスク B, C の負荷は 100% であると仮定し、実行時間を WCET とする。

消費電力の最適化を行わないオリジナルのスケジューリングでは図 5(a) に示すようになる。それに対して、クロック周波数として f_{clk} と $f_{clk}/2$ の 2 段階が設定される μ ITRON-LP でのスケジューリングは図 5(b) に示すようになる。

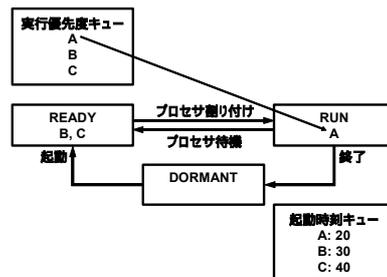


図6. 時刻 0 のタスク状態。

¹ μ ITRON には本来多くの状態があるが、 μ ITRON-LP では簡単化のために 4 状態のみを対象とする。

² この論文では理解しやすいように時刻は絶対値として表記されているが、 μ ITRON-LP では相対時間が採用されている。

このスケジューリングについて詳細に説明する。時刻 0 でタスク A, B, C が一斉に起動されると図 6 に示すように実行優先度が最も高いタスク A が実行される。

実行優先度キューにタスクが 2 以上存在するので、タスク A の 1 番目のスライスでは締切はタスク A の実行の開始からタスク A 自身の WCET が経過した後の時刻すなわち時刻 6 となる。2 番目のスライスから最後のスライスまでの WCET が 4 単位時間であるので、時間余裕は 2 単位時間となる。WCET が 2 単位時間であるスライスを 2 単位時間の時間余裕で実行しなければならないため、クロック周波数は f_{clk} となる。この f_{clk} の値はシステムコールとして μ ITRON-LP に提供され、FV ルックアップテーブルに基づいて V_{DD} が設定される。

タスク A の負荷は 50% であるため、本来の WCET では 2 単位時間を要した 1 番目のスライスが 1 単位時間で終了する。タスク A の 2 番目のスライスではタスク A 自身の WCET から 1 単位時間の積算実行時間を減算した 5 単位時間が締切までの時間となる。最終スライスの WCET が 2 単位時間であるので、時間余裕は 3 単位時間となる。2 番目のスライスの WCET も 2 単位時間であるので、2 番目のスライスも f_{clk} で実行される。

最後のスライスでは 1 番目および 2 番目のスライスと状況が異なる。タスク A において最後である 3 番目のスライスの実行が開始される時刻は 2 であり、同じく積算実行時間は 2 単位時間である。3 番目のスライスの実行の開始から締切までの時間はタスク A 自身の WCET の 6 単位時間から積算実行時間の 2 単位時間を減算した 4 単位時間である。この 3 番目のスライス以後にはタスク A において実行すべきスライスが存在しないため、3 番目のスライスの実行の開始から締切までの 4 単位時間をそのまま利用することができ、時間余裕は 4 単位時間となる。したがって、3 番目のスライスではクロック周波数を $f_{clk}/2$ に低減させることができ、これと同時に V_{DD} も制御することにより電力節減が可能となる。

タスク A は時刻 4 で終了し、同様にタスク B が時刻 4 から時刻 16 の間で実行される。

時刻 16 においてタスク C の 1 番目のスライスの実行を開始する。このときタスク C のみが実行優先度キューに存在するので、締切は第 1 起動時刻である次回のタスク A が起動する時刻 20 とタスク C の実行が開始される時刻 16 からタスク C 自身の WCET である 2 単位時間が経過した時刻 18 のうちの遅い方の時刻、すなわち時刻 20 となる。換言すれば、タスク C 自身の WCET に比べて第 1 起動時刻まで時間の方が大きい。これはタスク C の WCET よりも時間余裕が存在することを意味し、そのためタスク C は A や B の場合とは異なり、最初のスライスでもクロック周波数を $f_{clk}/2$ にすることができる。このような条件下では電力節減の効果がさらに増す。タスク C は時刻 20 で終了し、以下同様に理解できる。

3. 実装

図 7 にシステムを示す。ターゲットとして日立製作所製 SH-4 組込システム基板を選び、 μ ITRON[20] に準拠した日立製作所製 HI7750[21] を改造し、 μ ITRON-LP を実装した。具体的には優先度や開始アドレスなどのタスク固有の情報を持ったデータ構造である TCB (Task Control Block) に周期、次回起動時間、締切などを追加した。(改造の詳細は仕様書を参照のこと。)

アプリケーションとしては MPEG4 codec と FFT (Fast Fourier Transform) をマルチタスクとして実行した。

システムのブロック図を図 8 に示す。MPEG4 の入力生データとして H.263 標準画像の“carphone”をフラッシュ ROM に保存している。FFT は 4096 点を計算する。

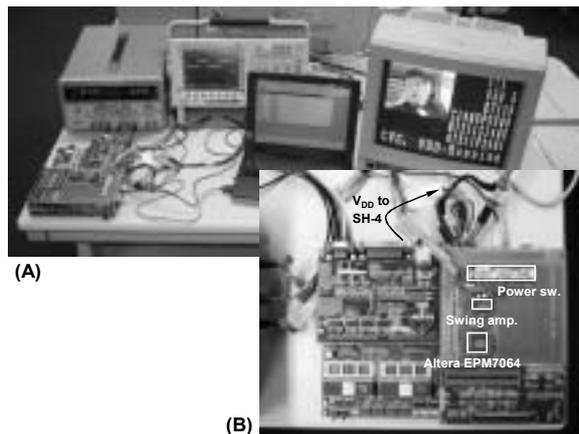


図 7. (A) システム全景. (B) SH-4 組込システム基板

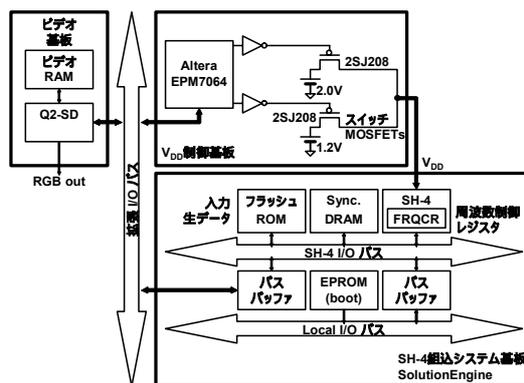


図 8. ブロック図.

前述どおり最適な f と V_{DD} に関する情報がシステムコールと FV ルックアップテーブルによって得られるが、 V_{DD} 情報はバスバッファ経由で拡張 I/O バスに送られ、FPGA (Altera EPM7064) によって実装された V_{DD} 制御基板を制御する。

また SH-4 は FRQCR と呼ばれる周波数制御レジスタを内蔵しており、即時に内部クロック周波数を変更できる。 f の制御はこれを利用する。この内部クロック周波数は外部周波数 33MHz と同期している。このシステムでは 33MHz の整数倍である 200MHz と 100MHz を f として使っており、外部システムとのインターフェイスにおける同期の問題は発生しない。 V_{DD} については $f = 200\text{MHz}$ のとき $V_{DD} = 2.0\text{V}$ 、 $f = 100\text{MHz}$ のとき $V_{DD} = 1.2\text{V}$ となるように 2SJ208 (スイッチ MOSFETs) で変更される。これらの V_{DD} の値はプロセッサの測定により得られたものである。

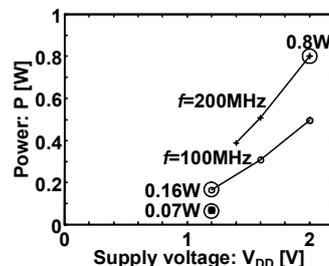


図 9. SH-4 の消費電力特性.

図 9 に測定された SH-4 の消費電力特性を示す。 $f = 200\text{MHz}$ において 0.8W、 $f = 100\text{MHz}$ において 0.16W、スリープにおいて 0.07W である。 $f = 200\text{MHz}$ を利用する時間は 10.5%、 $f = 100\text{MHz}$ は 54.0%、スリープは 35.5% であったので平均電力は 0.20W ($10.5\% \times 0.8 + 54.0\% \times 0.16 + 35.5\% \times 0.07$) である。なお SH-4 の I/O バッファは定格以下の電源電

圧で最適化されていない。もし注意深く設計されていたなら $f = 100\text{MHz}$ の場合、 $V_{DD} = 0.9\text{V}$ で動作するはずである。その場合は 100MHz 時の消費電力はさらに半分程度まで低減できる。

4. 結果

図 10 はプロセッサの V_{DD} とスリープ信号の測定波形を示す。平均負荷は 37.5% ($10.5\% \times 1 + 54\% \times 0.5 + 35.5\% \times 0$) である。 $\mu\text{ITRON-LP}$ とオリジナルとの消費電力比較を図 11 に示す。オリジナルではタスクによってプロセッサが占有されていないときは NOP を実行する。 $\mu\text{ITRON-LP}$ は 0.20W の電力を消費した。もし I/O バッファを低電源電圧において最適化し、 0.9V でも動作可能である場合、さらに 0.16W にまで低減できる。すなわち $\mu\text{ITRON-LP}$ はオリジナルの $1/4$ 以下の電力にまで節減可能である。

$\mu\text{ITRON-LP}$ の電力節減は実際にはタスク周期の組み合わせに依存する。すなわち締切までの時間をどれだけ利用できるかで効果が異なる。さらにはタスクの実行時間のばらつきにも依存している。しかしながらタスク間の時間余裕とマルチメディアアプリケーションを持つ実行時間のばらつきを最大限に利用しており、 $\mu\text{ITRON-LP}$ は複数のタスクに対して十分な電力節減を期待できる。

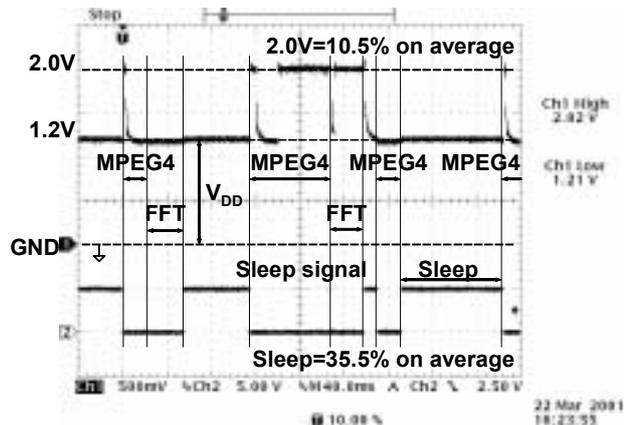


図10. プロセッサの V_{DD} とスリープ信号の測定波形。MPEG4 の周期は 114ms でその WCET は 79ms である。FFT のそれらはおのおの 171ms と 35ms である。MPEG4 と FFT はそれぞれ 22 と 2 スライスに分割されている。

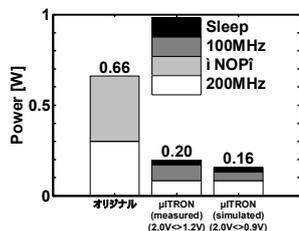


図11. オリジナルと $\mu\text{ITRON-LP}$ との電力比較。

5. 結論

この論文では $\mu\text{ITRON-LP}$ の実装とその有効性について述べた。 $\mu\text{ITRON-LP}$ の本質はタスク間の時間余裕とタスク内部における実行時間の WCET からのばらつきを利用することである。測定結果から $\mu\text{ITRON-LP}$ は MPEG4 と FFT のマルチタスク環境において電力をオリジナル μITRON の $1/4$ 以下にまで低減できることを示した。

謝辞

日立製作所中央研究所の相坂一夫氏、十山圭介氏、石橋孝一郎氏、日立米沢電子の八巻一志氏に感謝いたします。

なおこの研究は日立製作所と日本学術振興会から受託されたものである。

参考文献

- [1] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, iData Driven Signal Processing: An Approach for Energy Efficient Computing, i Proceedings of International Symposium on Low Power Electronics and Design, pp. 347-352, 1996.
- [2] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda, T. Sakurai, and T. Furuyama, iVariable Supply-Voltage Scheme for Low-Power High-Speed CMOS Digital Design, i IEEE Journal of Solid-State Circuits, vol. 33, no. 3, pp. 454-462, 1998.
- [3] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, iPower Optimization of Variable Voltage Core-Based Systems, i Proceedings of Proceedings of Design Automation Conference, pp. 176-181, 1998.
- [4] T. Pering, T. Burd, and R. Brodersen, iThe Simulation and Evaluation of Dynamic Voltage Scaling Algorithms, i Proceedings of International Symposium on Low Power Electronics and Design, pp. 76-81, 1998.
- [5] T. Ishihara, and H. Yasuura, iVoltage Scheduling Problem for Dynamically Variable Voltage Processors, i Proceedings of International Symposium on Low Power Electronics and Design, pp. 197-202, 1998.
- [6] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, iA Dynamic Voltage Scaled Microprocessor System, i IEEE International Solid-State Circuits Conference Digest of Technical Papers, pp. 294-295, 2000.
- [7] T. Okuma, H. Yasuura and T. Ishihara, iSoftware Energy Reduction Techniques for Variable-Voltage Processors, i IEEE Design & Test of Computers, March-April, 2001.
- [8] Transmeta Web Site, <http://www.transmeta.com/>.
- [9] David R. Ditzel, iTransmeta's Crusoe: A Low-Power x86-Compatible Microprocessor Built with Software, i Proc. International Symposium on Low-Power and High-Speed Chips (Cool Chips), pp. 1-30, 2000.
- [10] Y. Shin, and K. Choi, iPower Conscious Fixed Priority Scheduling for Hard Real-Time Systems, i Proceedings of Design Automation Conference, pp. 134-139, 1999.
- [11] M. Weiser, B. Welch, A. Demers, and S. Shenker, iScheduling for Reduced CPU Energy, i Proceedings of USENIX Symposium on Operating Systems Design and Implementation, pp. 13-23, 1994.
- [12] F. Yao, A. Demers, and S. Shenker, iA Scheduling Model for Reduced CPU Energy, i Proceedings of IEEE Annual Foundations of Computer Science, pp. 374-382, 1995.
- [13] C. Hwang and A. Wu, iA Predictive System Shutdown Method for Energy Saving of Event-Driven Computation, i Proceedings of IEEE/ACM International Conference on Computer Aided Design, pp. 28-32, 1997.
- [14] Y. Lee and C. Krishna, iVoltage-Clock Scaling for Low Energy Consumption in Real-time Embedded Systems, i Proceedings of International Workshop on Real-Time Computing Systems and Applications, 1999.
- [15] S. Lee, and T. Sakurai, iRun-time Power Control Scheme Using Software Feedback Loop for Low-Power Real-time Applications, i Proceedings of Asia and South Pacific Design Automation Conference, pp. 381-386, 2000.
- [16] S. Lee, and T. Sakurai, iRun-time Voltage Hopping for Low-power Real-time Systems, i Proceedings of Design Automation Conference, pp. 806-809, June 2000.
- [17] H. Kawaguchi, G. Zhang, S. Lee, and T. Sakurai, iAn LSI for V_{DD} -Hopping and MPEG4 System Based on the Chip, i Proceedings of IEEE International Symposium on Circuits and Systems, pp. 17-23, May 2001.
- [18] 川口 博, 張 綱, 李 誠洙, 桜井 貴康 「電圧ホッピングとアプリケーションスライシングによるリアルタイムアプリケーション向け低電力プロセッサシステム」, 第 3 回 LSI IP デザイン・アワード IP 優秀賞, 平成 13 年 5 月. <http://ne.nikkeibp.co.jp/award/papers/pdf2.pdf>.
- [19] S. Lim, Y. Bae, G. Jang, B. Rhee, S. Min, C. Park, H. Shin, K. Park, and C. Kim, iAn Accurate Worst Case Timing Analysis for RISC Processors, i Proceedings of IEEE Real-Time Systems Symposium, pp. 97-108, 1994.
- [20] Tron Project Official Home Page, <http://www.tron.org/>.
- [21] Hitachi Semiconductor and Integrated Circuits Web Site: <http://www.hitachi.co.jp/Sicd/English/Products/micome.htm>.
- [22] Y. Shin, H. Kawaguchi, and T. Sakurai, iCooperative Voltage Scaling (CVS) between OS and Applications for Low-Power Real-Time Systems, i Proceedings of IEEE Custom Integrated Circuits Conference, pp. 553-556, May 2001.