九州大学学術情報リポジトリ
Kyushu University Institutional Repository

低消費エネルギー化システムLSI設計手法 : システム設計からのアプローチ

安浦, 寛人
九州大学システムLSI研究センター

https://hdl.handle.net/2324/4037

出版情報：電子情報通信学会技術研究報告. ICD2002 (179), pp.31-36, 2002-12. 電子情報通信学会ICD研究会
バージョン：
権利関係：

[特別招待論文] 低消費エネルギー化システム LSI 設計手法
－システム設計からのアプローチ－

安浦 寛人†

†九州大学システム LSI 研究センター 〒816-8580 春日市春日公園６―１
E-mail: †yasuura@slrc.Kyushu-u.ac.jp

あらまし 低消費エネルギー化はシステム LSI 設計における重要な技術課題である．発熱に起因する回路の性能や信頼性の低下を防ぎ，微細加工技術のもたらす集積度を向上の恩恵を受けるために，低消費エネルギー化は必須の技術である．さらに，携帯情報機器等のバッテリー駆動機器へ搭載されるシステム LSI においては，バッテリーの容量（機器の重量や大きさに大きな影響を与える）の削減や充電頻度の削減による利便性の向上と関係し，低消費エネルギー化の成否が機器の商品価値を決定することも少なくない．地球規模での環境問題が重要視される中，低消費エネルギー化技術は，半導体や情報機器の市場規模そのものを規定する要因ともなりうる．本稿では，システム設計レベルでの低消費エネルギー化技術の基本方針を議論し，具体的な手法をいくつか紹介する．

キーワード システム LSI 設計，消費エネルギー，消費電力，動的電圧制御，データパス幅最適化

Design Methods for Low-Energy-Consumption System LSIs
－Approaches from System Design－

Hiroto YASUURA†

†System LSI Research Center, Kyushu University 6-1 Kasuga-Koen, Kasuga, Fukuoka, 816-8580 Japan
E-mail: †yasuura@slrc.Kyushu-u.ac.jp

Abstract Reduction of energy consumption is one of the most important technical issues in System-on-a-Chip (SoC or System LSI) design. For preventing the decrease of system performance and reliability caused by heating and for enjoying the increase of the number of transistors on a chip by the progress of fabrication technology, the reduction of energy consumption is very important. In mobile information terminal devices, which are driven by batteries, the energy consumption of SoCs on the devices strongly influences the frequency of battery charge and the weight and the size of the batteries, which are directly involved with the commercial value of the devices. Since the global environment problem is argued as a large political problem in the 21st century, the low-energy design technology for SoC may be a significant factor to determine the size of the market of semiconductors and information technology. In this article, basic concepts of the energy reduction technology for SoC in the system design level are discussed, and several examples of energy reduction techniques are presented.

Keyword SoC Design, Energy Consumption, Power Consumption, Dynamic Voltage Scaling, Datapath width Optimization

1. はじめに

消費エネルギー（電力）は，集積回路の技術の方向性を決定してきた大きな性能尺度である．回路の集積度の 向上に伴ない，バイポーラトランジスタからMOS トランジスタへ，n-MOS から CMOS へと，より消費エネルギーの小さなデバイスや回路構造が採用されてきた．すでに，数十ワットの電力を消費する LSI もあり，その単位面積辺りの発熱量は暖房器具やホットプレートのような調理器をはるかに越えている．

システム LSI 設計における消費エネルギーの問題は，

１）回路の発熱によるシステム構成・性能・コストへの影響．

２）消費エネルギーの増加によるシステム構成・性能・コストへの影響．

３）地球規模の環境問題への影響．

の３つの視点から考える必要がある．

回路の発熱は，LSI のパッケージを規定すると共に，その LSI を搭載する機器の放熱設計や LSI 上に集積可能な素子数の制約などに大きな影響を与える．発熱と放熱のバランスが壊れると回路の誤動作や故障につながるため，平均的なエネルギー消費としての電力の低減が重要な設計目標となる．発熱が大きい場合，高価なセラミックパッケージへの搭載や空冷・水冷機構の導入などを必要とし，システム全体のコストを引き上げる大きな要因となる．

一方，２）の問題は，電源装置や電源ラインの電流供給能力，電源の重量や大きさなどに影響する．特に昨今は，電池で駆動される携帯型の機器が多くなり，低消費エネルギー化が携帯情報機器の価値を左右する大きな要因として注目されている．電池の持続時間は，対象とするの処理を完了するのに必要なエネルギー量を最小化することで最適化できる．このようなエネルギー消費の低減は，機器の運用コストの削減や利用しやすさの向上にもつながる．一方，瞬時的な電流消費が大きいと，電源の性能をそれに対して向上させる必要があり，信頼性とコストのトレードオフが生じる．

地球規模での環境問題が世界的な問題としてクローズアップされている今日、社会全体のエネルギー消費を増加させるような新しい製品やシステムは禁止的になる．ユビキタスコンピューティングなど多くの半導体製品を社会の中に埋め込むシステムの構想が提案されているが，社会全体としてエネルギー消費を増大させるような提案では最終的に社会に受け入れられない．システム LSI の市場を拡大する意味からも，個々の LSI の消費エネルギーの削減は重要な技術課題である．

LSI の低消費エネルギー(電力)化設計に関しては，近年多くの論文や書籍が発行されている．最近の研究成果をまとめた著書として[1]，関連するサーベイ論文および資料として[2]-[6]をあげておく．また，多くの設計技術の国際会議がこの問題を取り上げているが，システム設計レベルから回路・デバイス設計レベルまで，幅広い分野の研究者が参加する中心的な会議としては，ISLPED (International Symposium on Low Power Electronics and Design)[7]がある．

2. 電力消費のモデル

単位時間の発熱量（消費電力）P，電流 I，電圧 V，電荷 Q，抵抗 R，消費エネルギー U の間には下記のような関係が成り立つ．

$$P = dU/dt = V\,dQ/dt = VI = V^2/R = I^2 R$$

消費電力 P は消費エネルギー U の時間微分あるいは単位時間辺りの平均値であり，消費エネルギーは消費電力を時間方向に積分したものである．発熱や電源線の容量等の議論には消費電力が重要であり，バッテリーの寿命や機器としてのエネルギー消費を議論する場合には，対象とする一連の処理を完了する間に消費されるエネルギーを議論しなければならない．

デジタル CMOS 回路の主な電力消費は，各ゲートのスイッチング時の負荷容量への充放電による動的電力消費，ゲートのスイッチング時の貫通電流による電力消費，漏れ電流による電力消費などからなる．

$$P = C_L \cdot V_{DD}^2 \cdot f_{0\to1} + t_{SC} \cdot V_{DD} \cdot I_{peak} \cdot f_{0\to1} + V_{DD} \cdot I_{leakage}$$

C_L:負荷容量，　V_{DD}：電源電圧，

t_{SC} ：スイッチング時間，I_{peak} ：貫通電流，

$I_{leakage}$ ：漏れ電流，　$P_{0\to1}$ ：遷移確率，

$f_{0\to1}$ ：単位時間の遷移頻度 $f_{0\to1} = f \cdot P_{0\to1}$

従来の CMOS 回路では，第一項の動的電力消費が支配的であったが，回路の微細化および電源電圧の低下に伴うしきい値電圧の低下により，今後は第三項の漏れ電流に起因する静的な電力消費も無視できなくなってくる．電源電圧 V_{DD} は，すべての項に影響を与える重要なパラメータである．過去の低消費電力化は，電源電圧の削減を中心に行われてきた．しかし，電源電圧の低減は，しきい値電圧の低減も伴い，トランジスタを OFF の状態で用いるときの漏れ電流（オフ電流）が大きくなって第三項の増大を招くようになっている．100nm 以下のプロセスでは，電源電圧の削減は逆に消費電力を増加させる効果を持つ可能性が高い．

一般に，電源電圧を下げると論理素子の遅延は大きくなり，システムの処理速度は低下する．動的電力消

費を下げるために電源電圧を低く設定した場合，一つのまとまった処理を完了するまでの時間は長くなる．エネルギー消費は，電力消費の時間積分であるから，処理時間の増大は，その処理を完了するまでに使われる漏れ電流によるエネルギー消費の増大につながる．単位時間辺りのエネルギー消費である電力消費の削減とひとつのまとまった処理を完了するまでに使われるエネルギ消費の関係は，単純な比例関係ではない．

3. 動的エネルギー消費の削減手法

3.1. 動的消費電力によるエネルギー消費

従来の CMOS 論理回路においては,その主たるエネルギー消費は,動的消費電力に起因するものであった．あるまとまったタスクを処理するのに消費される動的消費電力に起因するエネルギー消費は下記のようにモデル化できる．

$$E_{task} = Cycle_{task} \cdot p$$
$$= Cycle_{task} \cdot \Sigma CL_k \cdot Swit_k \cdot V_{DD}^2$$

p: 1 クロック周期あたりのエネルギー消費
$Cycle_{task}$: タスク処理に要するクロックサイクル数
CL_k: ゲート k の負荷容量
$Swit_k$: ゲート k の 1 クロックサイクルの平均スイッチング回数
V_{DD}: 電源電圧

この式から，エネルギーの削減には，クロックサイクル数の削減，負荷容量の削減，スイッチング回数の削減，電源電圧の低減が可能であることがわかる．

3.2. 電源電圧の低減

電源電圧の低減は，２乗の効果があるので最も基本的なエネルギー削減手法である．しかし，電源電圧の低減は，素子遅延を引き起こすので，性能を一定に保つためには，並列処理等と併用して処理速度の低下に対応しなければならない．

粗い近似として，電源電圧と素子遅延が反比例すると仮定すれば，電源電圧を半分にして処理回路を２つ用意することで，完全に並列化ができたとすれば（この仮定は情報工学では馬鹿げた仮定であることは明らかではあるが),処理速度を一定に保って消費エネルギーを 4 分の 1 に削減できる．しかし，一般にはこのように完全に並列化できることは少なく，アムダールの法則からも明らかなように，並列化できない部分が性能を強く規定する．

現実のシステムにおいては，処理の時間的あるいは空間的偏りが存在する．回路のある部分は高い処理効率を要求されるが他の部分はそれほどでも無い．この場合，高い処理速度に合わせてシステム全体を高い電源電圧で動作させるより，高い処理効率を必要とする部分だけを高い電源電圧に設定し，他の部分は低い電源電圧でゆっくりと動かす（空間的な電源電圧の最適化）．これにより，システム全体の効率を犠牲にせずに効率を必要としない部分のエネルギー消費を削減することが可能となる．

処理の偏りは時間的にも発生する．携帯電話では，待機時と通話時の処理量は大幅に異なる．このような場合，最高性能を要求する場合に合わせて電源電圧を固定するのでは無く，高性能が要求されるときにのみ高い電源電圧を利用して，その他は低い電源電圧で運用する手法が有効である．動的電圧制御(Dynamic Voltage Scaling あるいは V_{DD} Hopping)と言われる技術である．

特に,マイクロプロセッサにおける処理においては，高い性能のプロセッサを低い電源電圧で動作させることにより，性能を必要としない部分のエネルギー消費を大幅に削減できることが報告されている．電源電圧の制御は，

- プログラムのコンパイル時に最適化を行う．
- 実行中にオペレーティングシステムが制御する．
- 性能要求を観測する特別の制御回路を用意し，その回路からの情報で制御する．

などの方法で行われる．エネルギー消費を最小化する動的電圧制御に関する基本的な理論，コンパイラによる静的な最適化手法，オペレーティングシステムや制御回路を用いた動的な最適化手法などに関する多くの手法が提案されている[2][5][6].

3.3. スイッチング回数の削減

スイッチング回数の削減も広く用いられるエネルギー削減手法である．基本的には，

計算に不要なスイッチングを除去する．

符号化やプロトコルを工夫する．

スイッチングが少ないアルゴリズムを選ぶ．

不要なスイッチングを削減する手法としては，ゲーティッドクロックやパイプラインレジスタの挿入による方法などが広く知られている．また，計算をしなくても結果が簡単に分かるような入力の組み合わせに対して，計算をせずに直接結果を作るプレ計算の手法なども有効である．例えば，共通鍵暗号系の AES の暗号化回路において，パイプラインレジスタの挿入で消費電力 6 分の 1 にまで削減できる．

データとして保持すべき情報を記憶・演算・転送するレジスタ，メモリ，演算器，バスなどの幅も重要な最適化パラメータである．応用によって異なるデータ

のビット数を意識したシステム設計法については次節で詳説する．

この他，データを表現する符号化やバス上の通信プロトコルなどを工夫してスイッチング回数を削減する手法や動作の非対称性を利用して頻度の高い入力に対するスイッチング回数を小さくする回路構成法等が知られている．

3.4. 負荷容量の削減

負荷容量を削減する方法は，回路を小さくして容量を下げることが基本である．特に，スイッチング回数が大きい部分の負荷容量を下げる方法が有効である．例えば，頻繁に実行される命令だけを小さなメモリに常駐させる方法[8]や頻繁にアクセスされる変数で語長が短いものをビット幅の小さいメモリに格納するなどの方法がある[9]．

3.5. クロックサイクル数の削減

プロセッサベースの設計において，最も無駄なサイクルはキャッシュミスやパイプラインストールに起因する計算とは直接関係しないサイクルである．命令やデータのメモリ上での配置を工夫し，キャッシュミスを減らす手法が種々提案されている．この他，キャッシュメモリのラインサイズを動的に制御してキャッシュミスを削減する方法，ストールの多いプログラムに対して命令パイプラインの段数を動的に削減する方法なども検討されている．

アルゴリズムの改良は，クロックサイクル数の削減の王道ではあるが，アルゴリズムの変更にともなう動作検証や周辺回路への影響等附随する問題も発生する．さらに，画像処理や音声処理においては，最終的に必要とされる画像や音声の品質に応じて省略できる演算を削除してクロックサイクル数を削減するような方法も考えられる．情報の品質と消費エネルギーのトレードオフを考えた Quality Driven Design の手法も提案されている[10][11]．

4. データパス幅をパラメータとする消費エネルギー最小化設計手法

システム内のバスや演算器のビット幅であるデータパス幅は，システムの規模や消費電力（エネルギー）に大きな影響を与える．ハードウェア設計は，従来の設計においても設計者はデータパス幅に注意を払って設計を行ってきている．システム内で実行される処理で必要とされるデータパス幅を注意深く調べて，設計者はレジスタや演算器のビット幅をチップ面積や消費電力を最小化するように決定している．プロセッサを利用するシステム設計では，各プログラムに対してデータパス幅をプログラマが制御することは難しい．システムのデータパス幅は，システム設計者がプロセッサを選択する時点で固定されてしまう．一方，各応用は入力や出力あるいはアルゴリズムによってそれぞれ異なった計算精度を必要とし，必要となるデータパス幅は必ずしも選択されたプロセッサのデータパス幅と一致するという保証は無い．

例として MPEG-2 ビデオ復号器の C プログラムの中に現れる変数をそれらが必要とするビット幅ごとに数え上げた結果を示す．プログラムは C 言語で記述された約 6000 行のプログラムで，384 個の変数が int 型として宣言されている．このうち，50 個の変数はフラグとして用いられており，1 ビットの情報しか保持しない．384 個の 32 ビットで宣言された変数のうち，わずか 35%のビットだけが実際の計算で用いられる．短いビット幅の情報しか入らない変数に長いビット幅のレジスタを割り当てることは，計算の中で無駄な電力消費を引き起こす原因を作ることになる．

システムのデータパス幅は，データパス上の回路やメモリの面積に大きな影響を与える[12][13]．面積が大きくなると回路内の配線の容量が大きくなり，動的な消費電力に影響を与える．プロセッサベースのシステム設計においては，データパス幅はシステム設計者が決定することになる．プロセッサの面積は，データパス幅にほぼ線形に比例する．しかし，データやプログラムを格納するメモリの面積は必ずしもデータパス幅に対して単調な関係を持つわけでは無い．このため，メモリの消費電力もデータパス幅に対して単調な関係にはならない[12]．

一般に，計算の精度を一定に保ちながらデータパス幅を削減すると，1 つの命令を多倍長演算命令として実現する必要があるため，プロセッサのサイクル数が増加し，プログラム中の命令の数も増加する．32 ビットの精度を必要とするデータに対する演算は，32 ビット幅のデータパスを持つプロセッサでは 1 命令ですむが，16 ビット幅のデータパスを持つプロセッサでは 2 命令を要する．

データパス幅が短くなると，多倍長演算命令が増加するのでプログラム実行時の命令サイクル数は増加する．すなわち，クロック周波数が同じであれば，性能は悪くなる．例えば，次の n ビット精度の演算を考える．

```
int x, y, z; /* n bits */
z = x + y
```

もし，データパス幅 m が $n/2 \leq m < n$ であれば，この演算は次のように行われなければならない．

```
int x, y, z; /* n bits */
z_low = x_low + y_low /* m bits */
```

z_high = x_high + y_high + carry / m bits */*

このように，計算精度を保つためには，データパス幅が減少した時には命令を増加させる必要がある．2命令を使うと，データパスは2回利用されるから，消費されるエネルギーは増加する可能性がある．すなわち，1サイクル当たりのエネルギー消費（消費電力すなわち単位時間当たりのエネルギー消費)が減っても，処理全体をとおして消費されるエネルギーの総量が減少する保証は無い．携帯機器などのバッテリー寿命を議論する場合には，１命令サイクルあたりの消費電力を議論するより，処理全体のエネルギー消費を議論することが重要である．

プロセッサベースシステムに対するデータパス最適化によるエネルギー最小化問題は次のように定義される．

エネルギー最小化問題

与えられたプログラムと入力データ集合に対し，与えられたプログラムが与えられた入力データを処理するのに必要な消費エネルギーを最小化するデータパス幅を求めよ．

１）動的なデータパス幅最適化

プロセッサベースシステムは多様なビット幅のデータを取り扱う．これに対し，システムのデータパス幅を固定して，消費エネルギーの最小化を図るのは，一般的にいって無理がある．できれば，動的にデータパス幅を制御できることが望ましい．使わないと分かっているデータパスの部分を電気的に停止した状態にすることで，無駄なスイッチングによる電力消費を削減することが可能である[3].例えばプロセッサの命令セットに32ビット命令と8ビット命令を用意し，8ビット以下のデータに対しては，8ビット命令を用いて，上位 24 ビットはスイッチングしないようにする手法が考えられる．以下のような３つのデータが順次転送される場合にを考える．

A 1100 1101 0010 1010 0000 0101 0011 0001
B 0000 0000 0000 0000 0000 0000 1011 0101
C 0101 1001 0000 1110 0110 0111 1010 1110

この場合，通常のデータ転送路では，スイッチングが28回発生する．しかし，上記のように2種類の命令セットを用意する手法を使えば，Bの上位24ビットを変化させないことができる．

A 1100 1101 0010 1010 0000 0101 0011 0001
B <u>1100 1101 0010 1010 0000 0101</u> 1011 0101
C 0101 1001 0000 1110 0110 0111 1010 1110

これによって，全体のスイッチング回数を 28 回から14回へ削減することができる．

8 ビットと 32 ビットの命令セットを持つ簡単なRISCプロセッサによる実験では,UnixのSPLITとSORTのプログラムに対し，それぞれ 16% と 28% のエネルギー削減効果があることが報告されている．命令セットの拡張に伴う面積の増加は10%以下であった．

動的なデータパス幅最適化には，この他にもいろいろな方法が考えられる．今後の研究が期待される．

２）有効ビット幅解析

データパス幅の最適化において，プログラムに含まれる各変数がそれぞれどれだけのビット幅の計算を要求するかを知ることは重要である．一般に使われているプログラミング言語は，各変数が必要とするビット幅を細かく指定する仕組みを持っておらず，プログラマもビット幅に関しては，データ型の選択以外は特に注意を払わないことが多い．プログラマに，プログラムを作成する時からビット幅の情報を管理してプログラムの付加情報とすることを要求することは現実的でない．そこで，既存のプログラムを解析して各変数が必要とするビット幅を抽出する技術(ビット幅解析)が必要となる．ビット幅解析は，与えられたプログラムに対し，プログラムへの入力の定義域とプログラムの出力として要求される計算結果の精度を手がかりに，プログラム中の各変数が必要とするビット幅を計算する技術である．ハードウェア設計およびコンパイラの分野でいくつかのビット幅解析の技術が開発されている．

ビット幅解析技術には大きくわけて２つのアプローチがある．一つは，シミュレーションの技術を用いて実際にプログラムを実行し，各変数の利用される状況をモニタするもので，動的解析と呼ばれる．もう一つは，形式的検証やシンボリックシミュレーションの手法を利用して各変数の値域の理論的な上下限を求める方法で，静的解析と呼ばれる．動的解析は，比較的高速に実行できるが，対象とする入力データに対してしか結果が保証されないため，完全では無い．一方，静的解析は完全性は保証されるが，計算時間がかかったり，極めて悲観的な値しか得られないことが多い．

静的解析は，下記のように行われる．与えられた入力の定義域から，プログラムの各命令ごとにその代入先の変数の値域を割り出していく．ループ回数が限定されないようなループがあると，その解析精度は一般に悪くなる．また，出力が要求する値域の方から，変数のビット幅を後ろからさかのぼるように解析する方法もあり，両者を組み合わせることで，より精度の良い解析結果を得ることができる．例えば，$z = x + y$ という加算命令に対し，x と y の定義域がそれぞれ[0, 2000]と[30, 500]とした場合，出力 z の値域は[30,

2500]と判断することができる．

動的解析は，画像処理のような場合に標準的な入力データや，定義域の中で最も極端な値をとるコーナーデータを入力として与えてシミュレーションを行い，各変数のビット幅を解析する．動的解析と静的解析をうまく組み合わせて，信頼性の高い精度良いビット幅解析を行う技術が重要になる．

３）ソフトコアプロセッサ上のデータパス幅最適化

ソフトコアプロセッサは，データパス幅もパラメータ化できる RTL レベルのプロセッサコアであり，プロセッサベースシステムのデータパス幅最適化に利用できる．ここでは，ソフトコアプロセッサのプロトタイプである Bung DLX を用いた設計環境と実験結果を報告する．Bung-DLX は DLX RISC アーキテクチャを基本としたパイプライン機構を持たないプロセッサである．命令数は 72 命令で，命令長は 32 ビットである．基本記述は 32 ビットのデータパス幅を持ち，32 本の汎用レジスタを持っている．レジスタ数やデータパス幅は，設計パラメータとしてユーザーに解放されており，カスタマイズ時に変更が可能である．プロセッサは，約 7000 行の VHDL 記述で与えられている．基本記述の論理合成結果は 23,282 ゲートである[14]．

ソフトコアプロセッサ上で動作するプログラムは，ビット幅解析を終えたプログラムで，各変数のビット幅が既知となっている．これらのビット幅情報を陽にプログラム内に記述するために，C 言語を拡張した Valen-C (Variable Length C) を開発した．さらに，Valen-C で記述されたプログラムを計算精度を落とすこと無くデータパス幅の異なるプロセッサのオブジェクトコードへ変換するリターゲッタブルコンパイラも開発した．

このような設計環境の下で，Lempel-Ziv データ圧縮／伸長器に関するデータパス幅を変化させた場合の設計例を図１に示す．

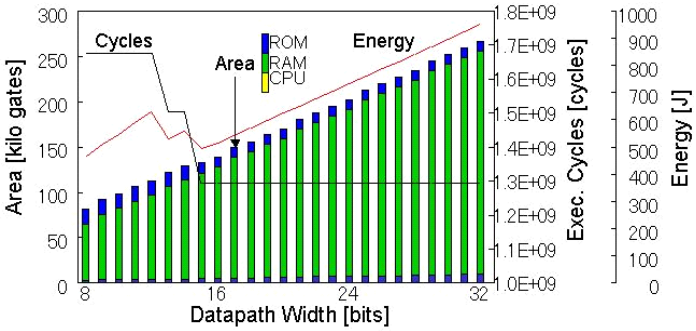

図１．Lempel-Ziv データ圧縮／伸長器に関するデータパス幅最適化例

5. おわりに

発熱やパッケージあるいは携帯情報端末としての商品価値等低消費エネルギー化がシステム LSI のコストや価値に大きな影響を与えている．従来の性能やチップ面積を重視したシステム設計から電力やエネルギーを重視した設計への移行が要求される．ソフトウェアやアーキテクチャも含めた統合的な取り組みが必要となる．

文　献

[1] M. Pedram and J. Rabaey, ed., : Power Aware Design Methodologies, Kluwer Academic Publishers, Boston, 2002.

[2] H. Yasuura and T. Ishihara, "System LSI Design Methods for Low Power LSIs," IEICE Trans. Electronics, vol. E83-C, no. 2, pp. 143-152, Feb. 2000.

[3] T. Kuroda, "Low-Power CMOS Design", the Journal of the IEICE, vol.81, no.11, pp.1144-1149, Nov. 1998 , (in Japanese).

[4] L. Benini, A. Bogliolo, and G. de Micheli, "A Survey of design techniques for System-Level Dynamic Power Management", IEEE Trans. on VLSI Systems, vol.8, no.3, pp299-316, June 2000.

[5] T. Okuma, H. Yasuura, and T. Ishihara, "Software energy reduction techniques for variable-voltage processors," IEEE Design & Test of Computers vol.18, no.2 , pp.31-41, March/April 2001.

[6] http://www.cse.psu./~mji

[7] http://ee.ucla.edu/~islped

[8] T. Ishihara and H. Yasuura, ``Programmable Power Management Architecture for Power Reduction,'' IEICE Trans. on Electronics, vol. E81-C no. 9, pp.1473-1480, September 1998.

[9] Y. Cao and H. Yasuura ``A System-level Energy Minimization Using Datapath Optimization'', International Symposium on Low Power Electronics and Design, August 2001.

[10] C.N. Taylor, S. Dey, and D. Panigrahi, ``Energy/Latency/Image Quality Tradeoffs in Enabling Mobile Multimedia Communication'', Proc. of Software Radio: Technologies and Services, Enrico Del Re, Springer Verlag Ltd., January 2001.

[11] Y. Cao and H. Yasuura, ``Video Quality Modeling for Quality-driven Design'', the 10th Workshop on System and System Integration of Mixed Technologies (SASIMI 2001), Oct. 2001.

[12] B. Shackleford, et al, ``Memory-CPU Size Optimization for Embedded system Designs,'' Proc. of 34th Design Automation Conference (34th DAC), June 1997.

[13] S. Mahlke, R. Ravindran, M. Schlansker, R. Schreiber, and T. Sherwood, ``Bitwidth Cognizant Architecture Synthesis of Custom Hardware Accelerators,'' IEEE Trans. CAD, vol. 20, no. 11, pp. 1355--1371, Nov. 2001.

[14] F. N. Eko, et.al., ``Soft-Core Processor Architecture for Embedded System Design,'' IEICE Trans. Electronics, vol. E81-C, no. 9, 1416-1423, Sep. 1998.