社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
ICD2001-65 (2001-08)

Bus Shuffling

－低消費電力向けの新しいバス技術－

山田　大裕　　辛　英　珠　　川口　博　　桜井　貴康
東京大学国際・産学協同センター
〒153-8505　東京都目黒区駒場 4-6-1
電話番号　03-5452-6253
E-mail　{dyamada, ysshin, kawapy, tsakurai}@iis.u-tokyo.ac.jp

あらまし　現在の通信・マルチメディア用途等の組込システムにおいては、メモリへのアクセスにおける電力消費が全体の電力消費の大部分を占める。従って、バスは十分な速度でかつ低消費電力で動作するように設計する必要がある。この論文では、特定アプリケーション用途での低消費電力システムにおけるバスシャッフリングを取り扱う。配線の側面間容量による消費電力の実効値が減るようにバスをシャッフルすることで、バスでの消費電力を低減させるヒューリスティックアルゴリズムを提案する。様々なデータ例を使って実験を行った所、消費電力を 30%から 46.7%程度、オーバーヘッドなしに低減させることができた。

キーワード　低消費電力，特定アプリケーション用途システム，ヒューリスティック，バスシャッフリング

Coupling-driven bus design for low-power application-specific systems

Daisuke Yamada,　Youngsoo Shin,　Hiroshi Kawaguchi,　and　Takayasu Sakurai
Center for Collaborative Research, University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan
Tel: 03-5452-6253
E-mail　{dyamada, ysshin, kawapy, tsakurai}@iis.u-tokyo.ac.jp

Abstract　In modern embedded systems including communication and multimedia applications, large fraction of power is consumed during memory access. Thus, buses should be designed and optimized to consume reasonable power while delivering sufficient performance. In this paper, we address a bus ordering problem for low-power application-specific systems. A heuristic algorithm is proposed to determine the order in a way that effective lateral component of capacitance is reduced, thereby reducing the power consumed by buses. Experimental results for various examples indicate that the average power saving from 30% to 46.7% depending on capacitance components can be obtained without any circuit overhead.

key words　Low-power, application-specific system, heuristic algorithm, Bus shuffling

1. はじめに

プロセス技術が微細化し、デザイン規模が大きくなるに従って、配線が回路面積、遅延及び消費電力に大きな影響を与えるようになっている[1]。特にシュリンクによって、配線側面間の容量が全配線容量の大部分を占めるようになった。高密度化のため配線ピッチが小さくなったことと、配線による抵抗を補償するために配線の縦横比が増大したことがその原因である。典型的な 0.35 ミクロン 3 層メタル CMOS プロセスで基板を接地している場合、配線側面間の容量はその対基板容量の 2 倍以上に達する。

組込システムではプロセサコアを基本ユニットとして利用することが増えてきている。特に、通信やマルチメディア組込システムに応用する場合、メモリへのアクセス時に電力が大量に消費されている。従って、各サブシステム間のデータ転送を行うシステムバスの設計は重要といえる。またその設計にあたっては、十分なパフォーマンスを出し、かつ消費電力を最小にする必要がある。コーディング技術を用いてチップ外部のバスの消費電力を低減させる方法を述べた論文は多数あるものの[2]-[7]、これらの技術をオンチップのバスに応用する場合、遅延・面積・消費電力の点から、コーディングによるオーバーヘッドは許容できないことが多い[8]。あるいは、低消費電力化のために行ったコーディングがそのハードウェアオーバーヘッドによってかえって消費電力の増大を招くことも多い。また上述のとおり、オンチップのバスをディープサブミクロン領域で設計する場合、配線側面間容量を考慮する必要がある。

本論文では、側面間容量とともに対基板容量も考慮した、組込アプリケーションシステム用の低消費電力オンチップバスの設計手法を提案する。本論文の手法は、プロセサ上で動く組込アプリケーションにおいて有効である。信号処理や画像処理といったアプリケーションのアルゴリズムと扱うデータの性質がわかっている場合、アドレスバスのパターンは特定の性質を持つと考えられる。ここで提案するバス・シャフリングではアドレスバスのパターンの性質に基づいて、バスにおける消費電力が最小になるようにバスの配置を最適化するものである。もちろん、本手法が有効なのはアドレスバスのみならず、データバスなどでも、一定の性質を有するバスの場合には常に有効である。本論文で提案する設計手法は、似たような信号を流す可能性が高いバスラインを隣接させれば側面間容量による消費電力が減るという原理を利用している。一切のオーバーヘッドとなる回路や面積増加などはなく、プロセサのレイアウトを改良するのみで実現できる。

この論文では、まず配線容量を考慮した回路モデルを提案し、消費電力を最小化するバス線のシャッフリング問題を定式化する。次に、バス線の配置を決定するヒューリスティックアルゴリズムを提案する。バス幅が狭く、アドレスのパターンが短いときに限り配置の最適解が求まるが、通常パターンは長い。そのために、全探索アルゴリズムには限界があり、ヒューリスティックアルゴリズムが重要となる。提案したヒューリスティックアルゴリズムを、シミュレーティドアニーリングを使った場合と比較する。

本論文の構成は以下のとおりである。2 章でバスのモデルを解説し、シャッフリング問題の定義を行い、その後にヒューリスティックアルゴリズムを提示する。3 章に実験結果を提示する。最後に、4 章において結論を示す。

2. 協調動作バスデザイン

2.1. 電力消費モデルの概要

n をバス幅、b_j を j 番目のバス線とした場合、バスを $\boldsymbol{B}=(b_0, b_1, \cdots b_{n-1})$ とする。時刻 i において信号 b^i_j をバス線 b_j に流れる信号とすると、$\boldsymbol{B}_i=(b^i_0, b^i_1, \cdots b^i_{n-1})$ で表現される信号群が与えられる。このバス $\boldsymbol{B}$ において消費電力が最小になるように各バス線を並び替える。仮に配線容量が対基板容量しかない場合、バス線をシャッフルしたところで、消費電力の低減効果は全くない。しかし実際には側面間容量があるため、

消費電力は隣接するバス線の信号に依存する。

図1に配線の寄生容量を示す。C_cは隣接する配線との側面間容量である。C_aと C_fはそれぞれ、配線と基板間に生じる垂直方向とフリンジ部の容量を表す。

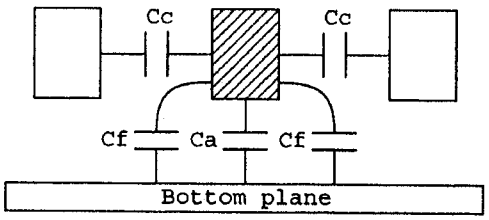

図1　隣接する配線と寄生容量

C_aと $2C_f$の和を対基板容量 C_lとする。C_cとC_lとの比を式(1)で定義する。

$$\eta = \frac{C_c}{C_l} \tag{1}$$

ミラー効果のため、側面間容量による消費電力は隣接するバス線の信号に依存する[9]。そのためこの論文ではミラー効果を考慮した実効遷移確率を扱う。隣接するバス線に逆相の信号が流れた場合、実効容量による消費電力は倍となる。逆に同相の信号が流れた場合は零となる。

CMOSデジタル回路では、消費電力は容量と信号の遷移確率に比例する。そこでまずj番目のバスでの信号の遷移を式(2)のように符号化する。

$$s_j^i = \begin{cases} 1 & \text{if } b_j^{i-1} = 0 \text{ and } b_j^i = 1 \\ -1 & \text{if } b_j^{i-1} = 1 \text{ and } b_j^i = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

さらに隣接するバス線j及びkの時刻iにおける信号の同相性を$\xi_i(j,k)$で表す。$\xi_i(j,k)$はバス線jから見たC_cの効果を意味する。従って$\xi_i(j,k)$と$\xi_i(k,j)$は異なる。仮に時刻i-1において逆相だった隣接する2線が時刻iにおいて互いに逆相となるような遷移をした場合$\xi_i(j,k)=2$である。従って、$\xi_i(j,k)$は式(3)のように表記できる。

$$\xi_i(j,k) = \left|s_j^i\right|\left(2 - \left|s_j^i + s_k^i\right|\right) \tag{3}$$

C_l及びC_cの影響をともに考慮する。時刻iにおけるバスb_jの信号の遷移をα^i_jとし、式(4)のように表す。このα^i_jはC_lのみを考慮した信号の遷移に対して正規化されでいる。

$$\alpha_j^i = \begin{cases} \left|s_j^i\right|(1+\eta\xi_i(j,j+1)) & \text{if } j=0 \\ \left|s_j^i\right|(1+\eta\xi_i(j,j-1)) & \text{if } j=n-1 \\ \left|s_j^i\right|(1+\eta(\xi_i(j,j+1)+\xi_i(j,j-1))) & \text{otherwise} \end{cases} \tag{4}$$

これに基づいてバスシャッフリング問題を図2のように定義する。

Compute taransition probability(p_j) of each line;
Compute switching corelation (ρ_{jk}) of each pair of lines;
Find a set of shielding lines $S \leftarrow \{b_j \mid p_j < \xi\}$;
$R \leftarrow \{b_0, b_1, \cdots b_{n-1}\} - S$;
$\{b_j\} \leftarrow$ *arrange-clusters*(Ψ,S);

図2　バスシャッフリング問題におけるヒューリスティックアルゴリズム

このヒューリスティックアルゴリズムより、α^i_jのi及びjについての総和を最小にするバス線の組み合わせ$\tilde{B}$を求める。

2.2. 配置アルゴリズム

バス幅をnとするとシャッフリングには$n!$通りの組み合わせが存在するため、真の$\tilde{B}$を見つけるには指数関数的な数の組み合わせを探索することになる。そのため、バス幅が小さく、また信号のパターンが短い時に限り真の最適解が求めることができる。しかし組込アプリケーション用のプロセサを扱う場合、このようなことはほとんど当てはまらない。ここでは信号の遷移の相関性と遷移確率に基づくヒューリスティックアルゴリズムを提示する。バス線j,

k における信号の遷移の相関を式(5)のように定義する。

$$\rho_{jk} = \frac{K_{jk}}{\sigma_j \sigma_k} \tag{5}$$

ここでの σ_j は s_j(s_jの時刻 iについての和)の標準偏差である。K_{jk} は s_jと s_kの共相関係数であり、(6)式に表わす。

$$K_{jk} = E\{s_j s_k\} - m_j m_k \tag{6}$$

$E\{x\}$は xの期待値であり、m_jは s_jの平均値である。ここで仮に2線の信号の遷移の相関が1に近い場合、この2線を通る信号は同相になるような遷移をする確率が高いことを意味する。しかし、信号の遷移の相関を求めただけではこの問題を解くのに不十分である。なぜなら、もしこの2線を通る信号がほとんど遷移しない場合、いくら相関が1に近くても、この2線を隣接させたところで消費電力は低減しないからである。従って、相関性とともに各バス線の信号の遷移確率を考慮に入れる必要がある。

図2に示した通り、最初にある閾値ξ[1]よりも遷移確率の低いバス線をふるい分ける。これらの線はクラスタ同士を分けるシールド線として用いられる。残ったバス線を一定の規則に従って分類する。このバス線の集合体をクラスタと呼ぶ。クラスタとシールド線を並べてこの問題に対する解とする。各クラスタにおけるバス線の配置は、遷移確率が変動しない限り、クラスタが形成された時点で固定される。

バス線をクラスタ化するヒューリスティックアルゴリズムを図3に示す。今まで使われていないバス線の中で、最も信号の遷移確率が高いものを選び出し、新しいクラスタの組み立てを始める。内側にあるwhileループ内では、クラスタの左端及び右端にあるバス線との相関が最大になるようなバス線 b_kを選んでいる。このループは左端もしくは右端にあるバス線と正の相関性があるバス線がなくなるまで続く。

```
build-clusters(R , pj , ρjk )
  while R not empty do
  Select bj∈R  s.t. pj is maximum,
  and R ← R − {bj};
  Form a new cluster Ψi ← {bj};
  while true do
    Find bk∈R  maximizing ρkl>0, where bl is the
    first or the last element of Ψi;
    if bk is not found then exit loop; end if
    if bl is the first element of Ψi
     then Ψi ← {bk}∪Ψi;
     else Ψi ←Ψi∪{bk};  end if
  end do
  Ψi ←Ψ∪Ψi;
end do
return Ψ;
```

図3　クラスタリングアルゴリズム

このようにして、信号の遷移確率が高く、また遷移の相関性も高いバス線同士が高確率で集まるようにクラスタが形成される。従って、側面間容量の実効値を減少させることができる。さらに、各クラスタの左端及び右端に配置されたバス線は比較的低確率で信号が遷移するため、側面間容量の実効値をより減少させることができる。このことを図4に明示する。

---

[1] 閾値はいろいろな方法で決定される。信号の遷移確率の分布に切れ目がある場合、閾値を切れ目にすればよい。線引きが難しい場合は閾値を色々変化させて、図2に示した処理を繰り返して、その中で最もよい結果を選ぶ。

```
arrange-clusters(Ψ,S).
  Select Ψl∈Ψ s.t. ρ(Ψl) is maximum,
  and Ψ ←Ψ−Ψl ;
  Select Ψr∈Ψ s.t. ρ(Ψr) is maximum,
  and Ψ ←Ψ−Ψr ;
  if the first element of Ψl has ρ(Ψl)
  then F ←Ψl ;
  else F ← reverse(Ψl) ; end if
  foreach Ψl∈Ψ do
    Ψ ←Ψ−Ψl , and F ←F∪Ψl ;
    Select bi∈S, S ←S−{bi},
    and F ←F∪{bi} ;
  end do
if S is not empty then F ←F∪S ;end if
if the last element of Ψr has ρ(Ψr)
then F ←F∪Ψr ;
else F ← reverse(Ψr) ; end if
return F;
```

図 4　最終的な解を得るためのヒューリスティクアルゴリズム。$\rho(\Psi_i)$は、Ψ_iを構成するバスの両端に配置されている線における信号の遷移確率の大きい方を意味する。$reverse(\Psi_i)$はΨ_iを左右転置する関数である。

図4に示したヒューリスティックアルゴリズムを用いて、クラスタとシールド線を組み合わせ、最終的な解を得る。まず、解の両端に配置する2つのクラスタ(Ψ_l, , Ψ_r)を遷移確率が高い順に選ぶ。両端に配置されたバス線による側面間容量は片面分しか影響しないため、高い遷移確率を持つバス線が両端に配置されることが望ましいからである。その後、シールド線がある限り、クラスタを順にシールド線とシールド線の間に配置していく。図3にある通り、各クラスタは両端に配置されているバス線が、他のクラスタの両端にあるバス線と互いに負の相関を持つように組み立てられているため、クラスタをシールド線の間に挟み込むことで側面間容量の実効値を減少させる効果がある。以上のアルゴリズムを単純な例を挙げて説明する。

＜例＞

以下のような8ビットのパターン流列を考える。ここでは左端を b_7とし、右端を b_0とする。

00011100
01110011
00110010
01001101
10000101

ここでは b_4と b_7がシールド線として選ばれたとする。最初に、残りのバス線の中で最も遷移確率の高い b_6を選ぶ。次に、b_6と最も正の相関が高い b_0(相関 0.9)を選択する。さらに b_3を選択し、b_6の隣に配置する。ρ_{36}が0.3であるのに対しρ_{30}は0.1だからである。これで、クラスタ内部でバス線は{b_0, b_6, b_3}の順か、左右が逆の順に配置されている。同様に b_2を選択し b_3の隣に配置する。ρ_{23}が0.9であるのに対しρ_{20}は0.0だからである。クラスタは{b_0, b_6, b_3, b_2}となる。

この時点で、このクラスタの両端に配置されているバス線 b_0, b_2に対し正の遷移相関を持つバス線はなくなるので、1番目のクラスタは以上か、またはその逆の順で構成される。同様にして、2番目のクラスタは{b_1, b_5}で構成される。

最初のクラスタにおいて、b_0は b_2よりも大きな遷移確率を持つ。従って、最終的な配置においてこのクラスタが左端に配置されるのなら{b_0, b_6, b_3, b_2}の順になる。それに対し、クラスタが右端に配置された場合この順とは逆になる。もし1番目のクラスタが左端に配置された場合、2番目のクラスタは右端に配置されることになる。従って、このヒューリスティックア

ルゴリズムから最終的に求まる配置は{b_0, b_6, b_3, b_2, b_4, b_7, b_1, b_5}となる。

3. 実験結果

提案したアルゴリズムの性能を評価するため、以下のサンプルパターンを使って実験を行った。

- wavelet, linear, laplace, compress, lowpass

ベンチマーク用データに使用したアドレスバスのパターンは、典型的な画像及び信号処理アルゴリズムから集めた[10]。全てのプログラムに関し 16 ビットのアドレスバスが使われている。データパターンは Shade を利用して手に入れた[11]。

- fft

図 5 に示されているオーディオデコーダー用 fft プロセサとメモリ間の 7 ビットデータアドレスのパターン[12]。VHDL シミュレーションによりパターンを抽出した。

- ac3

フレームメモリに記憶されている入力データを読み出すオーディオデコーダー用の解析プロセサとメモリ間との 16 ビットアドレスのパターン。VHDL シミュレーションによりパターンを抽出した。

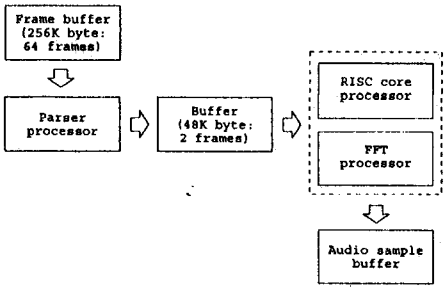

図 5 オーディオレコーダーのブロックダイアグラム

シールド線の閾値(ξ)は 0.01 に設定した。C_c と C_l は全てのバス線で一定と仮定して、η =1,2,3,4,5,∞で実験を行った。ヒューリスティックアルゴリズムによる消費電力の低減率を図 6 に示す。シミュレーティドアニーリング(SA)[2][13]で配線組換えを行った場合の結果は図 7 に示す。図 6,7 より、ヒューリスティックアルゴリズムを使うことによって良好な結果が得られていることがわかる。しかし、バスの幅や長さが適当なものであれば、SA をヒューリスティックアルゴリズムのかわりに用いることができる[3]。

ヒューリスティックアルゴリズムを使うことで得られる消費電力の低減率と SA を用いた際に得られる結果を比較したのが図 8 である。η=1 のときヒューリスティックアルゴリズムを用いることで平均 30%の低消費電力化がはかれ、η=∞で最大 46.7%の低減率を得ることができる。ヒューリスティックアルゴリズムと SA の結果は 1.9%～4.4.%違う。

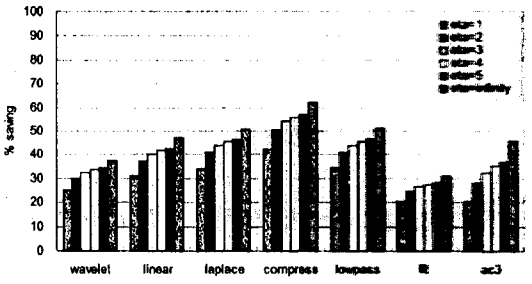

図 6 ヒューリスティックアルゴリズムを用いた場合の消費電力低減率

---

[2] このアルゴリズムでは、2 線をランダムに選び、取り替える。また 2 つの任意のバス線の集合もランダムに選び、取り替える。どちらを実行するかもランダムである。

[3] SA では実行時間の都合上サンプルデータの一部のみを用いた。linear では 485503 パターンあるサンプルデータの最初の 2000 パターンのみ用いた。それでも Sun Ultra1 で SA を実行させた場合約 14 分かかる。それに対し、この論文で提案したヒューリスティックアルゴリズムを用いた場合 1 秒以内に解が得られる。

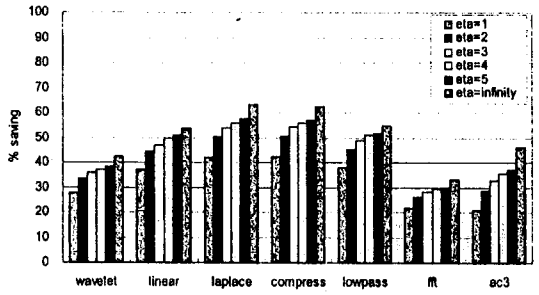

図 7　シミュレーティドアニーリングを用いた場合の消費電力低減率

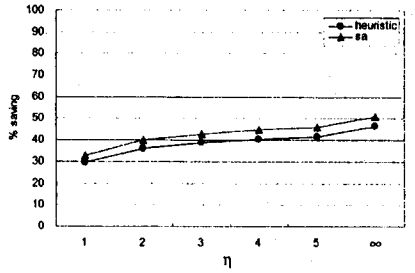

図 8　ヒューリスティックアルゴリズムと SA を用いた場合の消費電力低減率の比較

4. 結論

本論文では、特定アプリケーション向けの低消費電力システムを対象に、オンチップでのバスの設計技術を述べた。提案した手法では、配線の側面間容量と対基板容量を考慮に入れて、実効遷移確率が最小になるようバスのシャッフリングを行うことで、オンチップのバスの消費電力を最小化した。また同時に、バス線のシャッフルを行うヒューリスティックアルゴリズムを示した。提案した手法は、メモリ集約型アプリケーションに特化したシステムのアドレスバスを設計するのに特に向いている。ベンチマーク用のデータやオーディオデコーダーにおけるアドレスパターンといった大きなデータを扱う場合においても、消費電力を確実に減らせることが実験結果により示された。またシミュレーティドアニーリングを用いる場合と比較して、ヒューリスティックアルゴリズムを用いた場合、実行時間が大幅に削減されることが示された。更に、ヒューリスティックアルゴリズムにより得られる消費電力の削減率が、シミュレーティドアニーリングにより得られる削減率と比べ 1.9%～4.4.%程度であることも示された。

参考文献

[1] M. T. Bohr, "Interconnect scaling – the real limiter to high performance ULSI," in Proc. IEEE Int'l Electron Devices Meeting, pp. 241-244, Dec. 1995.

[2] M. R. Stan and W. P. Burleson, "Bus- invert coding for low power I/O," IEEE Trans. on VLSI Systems, vol.3, pp.49-58. Mar.1995.

[3] L. Benini, G. D. Micheli, E. Macii, D.Sciuto, and C. Silivano, "Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems," in Proc. Great Lakes Symposium on VLSI, pp.77-82, Mar. 1997.

[4] L. Benini, G. D. Micheli, E. Macii, M.Poncino, and S. Quer, "System-level power optimization of special purpose applications: The Beach Solution," in Proc. Int'l Symposium on Low Power Electronics and Design, pp. 202-207, Aug. 1997.

[5] E. Musoll, T. Lang, and J. Cortadella, "Exploiting the locality of memory references to reduce the address bus energy," in Proc. Int'l Symposium on Low Power Electronics and Design, pp. 202-207, Aug. 1997.

[6] S. Ramprasad, N. R. Shanbhag and I. N. Hajj, "A coding framework for low-power address and data busses," IEEE Trans. on VLSI Systems, vol.7, pp.212-221, June 1999.

[7] Y. Shin and K. Choi, "Narrow bus encoding for low power systems," in Proc. Asia South Pacific Design Automat. Conf., pp.217-220, Jan. 2000.

[8] P. Sotiriadis and A. Chandrakasan, "Low power bus coding techniques considering inter-wire capacitances," in Proc. IEEE Custom Integrated Circuits Conf., pp. 507-510,May 2000.

[9] H. B. Bakoglu, Circuits, Interconnections and Packaging for VLSI. Addison-Wesley, 1990.

[10] P. Panda and N. Dutt, "1995 high level synthesis

design repository," in Proc. Int'l Symposium on System Synthesis, 1995.

[11] R. Cmelik and D. Keppel, "Shade: A fast instruction-set simulator for execution profiling," Tech. Rep. TR-93-12, Sun Microsystems Laboratories, 1993.

[12] S. Lee and W. Sung, "A parser processor for MPEG-2 audio and AC-3 decoding," in Proc. Int'l Symposium on Circuits and Systems, pp. 2621-2624, June 1997.

[13] S. Kirkpatrick, J. C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, pp. 671-680, May 1983.