社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2005－ＳＬＤＭ－121（31）
2005／10／21

ハードウエア動作モデルからの電力・性能推定に関する一検討

井上 典之† 押川 克寛† 泉 知論‡ 福井 正博‡

† 立命館大学大学院理工学研究科情報システム学専攻

‡ 立命館大学理工学部電子情報デザイン学科

〒525-0058 滋賀県草津市野路東 1-1-1

E-mail: {re001010, re003001, t-izumi, mfukui}@se.ritsumei.ac.jp

あらまし システムの大規模化により，設計の早い段階で電力を推定して，設計戦略を立てることが重要となっている．そのためには，より抽象度の高いレベルでの電力解析の方法が求められている．本稿では，ハードウエア動作モデルからの電力推定を考える．その際，スピードとのトレードオフで考える必要があるが，著者らは，いくつかの条件に対して，トレードオフ解析のための推定データを求める方法について検討し，簡単な実験を行ったので報告する．

キーワード 高位合成，高位推定，電力推定，トレードオフ解析

A study for power and speed tradeoff estimation from behavioral hardware model

Noriyuki Inoue,† Katsuhiro Oshikawa,† Tomonori Izumi,‡ and Masahiro Fukui‡

† Department of Information Science and Systems Engineering, Ritsumeikan University

‡ Department of VLSI System Design, Ritsumeikan University

Noji-higashi 1-1-1, Kusatsu, Shiga 525-0058, Japan

E-mail: {re001010, re003001, t-izumi, mfukui}@se.ritsumei.ac.jp

Abstract Due to rapid growth of the scale of systems, it becomes a very important task to plan the design strategy based on the power estimation in the early design stage. Power analysis in higher abstraction level is required. Based on a control data flow graph, the authors have developed a structured tradeoff model of power and speed of a given function and a method to obtain the model. This paper gives a fundamental evaluation through a prototype system and discusses its usefulness and future works.

Keyword high-level synthesis, high-level estimation, power estimation, tradeoff analysis

1. はじめに

近年，半導体微細化技術の進歩により，大規模で複雑なシステムを１つのLSI上で実現することが可能になった．しかし，設計生産性の伸びが集積回路の集積度の伸びよりも低いために「設計生産性危機」と呼ばれる深刻な問題が起きつつある．大規模な回路の設計を短期間で行う方法として，設計記述の抽象度を高めることが挙げられる．ハード・ソフトの区別がないシステムレベル設計（システムの仕様段階での設計）は，設計期間の短縮化のために今後，重要となっていく[1]．

また，携帯通信・モバイル情報機器が急速に発展してきた今日では製品の小型化に伴って，電池駆動での長時間動作および低発熱が要求される．一方では，製品の機能の多様化・高度化・高性能化を実現するためにLSIの規模やLSIに実装される機能が増える傾向にあり，消費電力の増える要因が増加している．したがって，低消費電力化を達成するための設計技術の重要性はますます大きくなっている．

設計期間の短縮と低消費電力化を実現するために，システムレベル設計での消費電力推定が必要になる[9][10]．しかし低消費電力設計技術の現状は，

レイアウト・回路・プロセスレベルで行う手法が中心であり，システムレベル設計において適応可能な消費電力技術はまだあまり確立されていない．現在の消費電力推定のアプローチとして，バスやメモリなどに注目し推定する方法や，仮動作合成を行い推定する方法，ボトムアップ的に演算器ごとに電力モデルを作成しスイッチング率とともに推定する方法，C 言語などのコードからプロファイリングを作成し推定する方法などがある[5].

我々は，仮動作合成に注目し，消費電力の最適な高位合成の実現を目指している．高位合成のスケジューリング手法には ASAP(As Soon As Possible)，ALAP(As Late As Possible)，リストスケジューリング[6]などがあるが，消費電力の最適解を求められるものではなく，その場合人手による介入が必要とされている．

我々の提案する手法ではデータ・フロー・グラフ（DFG）を構成可能なハードウェアモデルを電力・スピードのトレードオフカーブを各ノードで定義し，DFG 全体のトレードオフカーブを各ノードのスケジューリングにより探索し求める．

さらに，DFG での最適解を求める方法を，制御を追加したコントロール・データ・フロー・グラフ（CDFG）に応用した．これによって，CDFG で表現される大規模回路の低消費電力化を図ることが可能となる．

本稿では，トレードオフカーブを求める手法と，それを用いた最適化手法について述べる．

2. 理論

2.1. 高位合成手法について

高位合成とは C 言語などのアルゴリズムから RTL の動作記述を生成するものであり，その手順は図 2.1 に示すフローチャートで表せられる．まず始めに入力されたアルゴリズムから CDFG を作成する．CDFG は DFG に分岐とループを持ち合わせたものである．次に DFG のリソースへの割り当てをする．リソースの割当てとはどのような演算器をいくつ使用するか決定することである．例えば，加算をする演算は加算器，加減算器，ALU など様々ある中で対象とする DFG に最も適したリソースを割り当てる．次にスケジューリングを行う．スケジューリングとは各クロックサイクルでどの演算を行うかを決めることである．スケジューリングのアルゴリズムについては次節で説明する．スケジューリングの結果に基づいて次にレジスタ割当てを行う．レジスタ割当てとは，データ依存が複数のクロックサイクルにまたがる時，レジスタを挿入することである．「変数の生存区間の解析」を行うと，挿入するレジスタの個数を削減することができる．レジスタの割当ての次にはバインディングをする．バインディングは DFG の頂点にハードウエアの演算器を割当てることである．最後にデーターパス，状態遷移機械（FSM）の生成をする．これはセレクタを挿入し，制御を加えた回路を生成することである．FSM を生成するまで行うと，RTL での動作記述と同等の記述レベルになる．

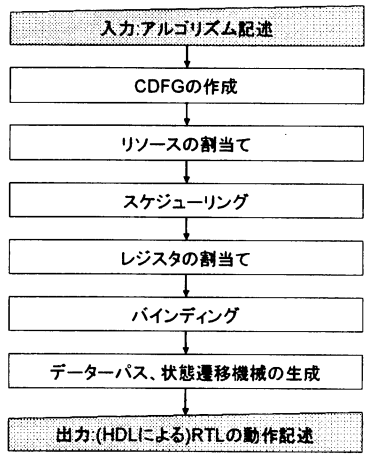

図 2.1　高位合成の流れ

2.2. スケジューリングアルゴリズムの紹介

高位合成において重要視されているのがスケジューリングのアルゴリズムである．代表的なのが ASAP，ALAP がある[6]．この二つのアルゴリズムを応用したものであるリストスケジューリングについて説明する．

リストスケジューリングには 2 種類あり，使うリソースの数を決めた上でスケジューリングを行う「面積制約スケジューリング」といつまでに処理を終えるかを決めた上でスケジューリングを行う「時間制約スケジューリング」がある．

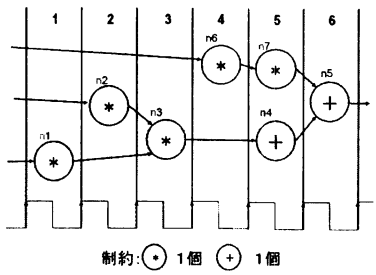

図 2.2　面積制約スケジューリング

「面積制約スケジューリング」の目標はいかに並列性を抽出してステップ数を最小にするかである．図2.2に示すようにこの場合，1サイクルに乗算器1個，加算器1個しか使用できないので最小ステップで6サイクルかかる結果になる．

「時間制約スケジューリング」の目標は演算資源の共用することで相互排他性を抽出してハードウエア量（面積）を最小にすることである．図2.3では4サイクルで処理を終えるために1サイクルに乗算器を2個，加算器1個を最低使用しなければいけない結果になった．

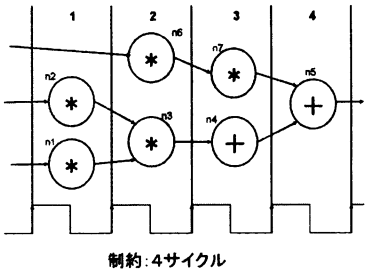

図 2.3　時間制約スケジューリング

このように，リストスケジューリングを行えば目的に合った最適な解が見つけることも可能だが，例外もありこのアルゴリズムを用いてスケジューリングをした結果が必ずしも最適解になるわけではない．よって，現在は人手が加えられて最適解を求められている．

3. トレードオフモデル

基本アルゴリズムを述べる前に，3.1，3.2，3.3でノードに割り当てられた演算のトレードオフモデルについて3.4でDFGから出力されるトレードオフモデルについて述べる．

3.1. ノード

演算器ごとに各ノードは，平均消費電力と時間のトレードオフモデルを複数持つ．それらを時間最大のモデルから時間最小のモデルまですべてのモデルを入れ替える．それ以外にマルチサイクルを考慮し，ある一定値より時間のかかる処理は複数サイクルにわける．図3.1にノードの演算に割り当てるトレードオフモデルを示す．

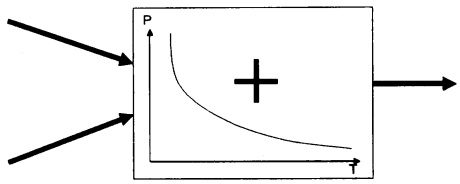

図 3.1　演算に割り当てるトレードオフモデル

3.2. 電力モデル

平均消費電力はノードの演算に割り当てられるトレードオフモデルの縦軸にあたり，これを横軸の時間と掛け合わせることによってノードの演算でのトータルな消費電力を求めることが出来る．これは電池の寿命に直接関係してくる．

ピーク電力はノードの平均消費電力から見積もる．信号が入ってくる立上り時からピーク時にいたるまで，ピーク時から立下り時にいたるまでを一定の形と仮定し，平均消費電力からピーク電力を計算する．ピーク電力は熱問題に関係する．よってピーク電力を用いたトレードオフカーブを作成することにより，消費電力は同じだが熱を分散して発生させるより最適な設計が可能になる．

3.3. 性能の測り方

1サイクルの定義をする．まず，それぞれのノードの演算に処理時間がある．それに付け加えてクロック遅延とセットアップ・ホールドタイムを考慮する．すべてのノードの演算の中でこの合計が一番大きいものをクリティカルスロット（ＣＳ）と呼びこれを1サイクルに設定する．図3.2には一般的な1サイクルのスロットを示す．

マルチサイクルの場合にはクリティカルスロットの大きさに合わせて区切られる(図3.3)．その中で，クロック遅延は最初のノードだけに反映され，セットアップ・ホールドタイムは最後のノードにだけ反映される．

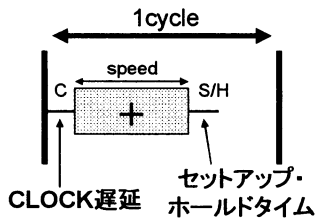

図 3.2　1サイクルの定義

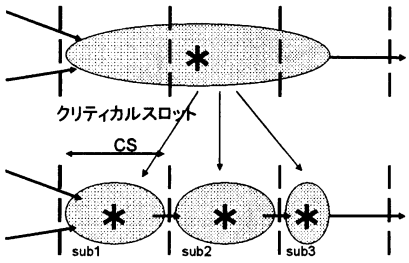

図 3.3　マルチサイクル

3.4. DFG のトレードオフカーブモデル

各ノードの演算からクリティカルスロットが求まりそれをサイクルの数だけ掛け合わせたものが DFG 全体の時間となる．また，すべての平均消費電力を足し合わせて時間を掛けたものが DFG のトータルな消費電力になる．それぞれのノードの演算のピーク電力を同サイクル内で足し合わせることで DFG でのピーク電力が求まる．このピーク電力を縦軸，時間を横軸にすることにより，DFG のトレードオフカーブを作成する点をプロット出来る．

4. 基本アルゴリズム

本アルゴリズムでは，ＤＦＧの処理のためのクロック数を決め，その条件のもとでの複数個の（電力，時間）の頂点をプロットする．その後，クロック数を１づつ減らし，各々のクロック数のもとでの複数個の（電力，時間）の頂点を追加プロットし，最終的に，全てのクロック数での実現可能な全ての（電力，時間）頂点がプロットできることになる．これを求めるトレードオフカーブと定義する．4.1.にそのクロック数探索手順について述べる．4.2.に得られたトレードオフカーブの例を示す．

4.1. クロック数の探索手順

ここでは，図 4.1 に示すような，縦にノードの数(n 個)，横に処理時間が最大になるケースのサイクル数(t 個)の n×t マスを持つスケジューリングテーブルを使用する．このテーブルの横幅がクロック数に相当し，縦長がピーク電力の総和に相当する．

まず，スケジューリングテーブルに DFG の各ノードを頂点Ｓからの最小距離の順番で左側から順に配置する．このとき，スケジューリングテーブルの同じ行と同じ列に複数個のノード（演算）は配置されない．この状態では，演算器の共有が無い場合を意味し，サイクル数 t は最大である．このクロック数が最大から最小になるまでの状態を以下の手順により探索する．

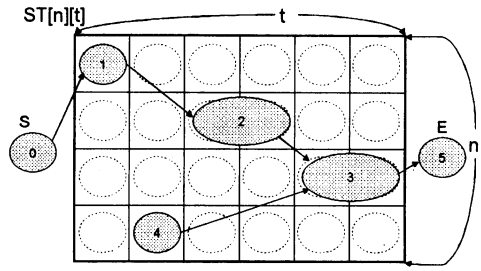

図 4.1　スケジューリングテーブル

本処理において，移動度を「ＤＦＧの順序関係と他のノードの配置を変更することなく移動可能なタイムスロットの個数，と定義する．たとえば，図 4.2 の例では，ノード４の移動度は３である．ノードの移動度を１つ減らすということは，最も左のタイムスロットを配置領域から除外することを意味する．

（クロック数を減らす処理）

移動の自由がない(移動度が１の)ノードはそのままで，移動度が２以上のノードを１個選択し，その移動度を１つ減らす操作を行う．その時に，縦方向に見て，ノードが全く存在しないタイムスロットがあれば，それを削除する．無ければ，別のノードについて同様の処理を行う．

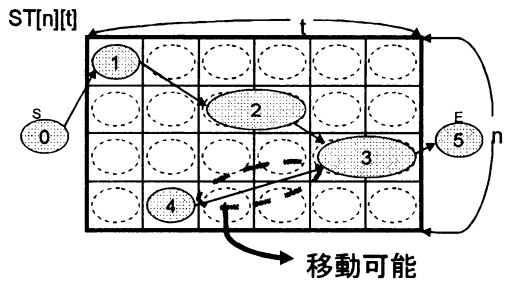

図 4.2　移動度の定義

4.2. 結果

実験に与えたデータは仮想のものであるが，加算器，減算器，乗算器のモデルをそれぞれ３個ずつ与えた．結果，35 個のスケジューリング結果が出てきた．この結果をグラフにすると，図 4.3 のようになりこの点の集合体の内側をとれば最適な設計点といえる．また，この点の一つ一つからスケジューリング結果とそのときのノードに割り当てられたトレードオフモデルを導き出すことが出来る．

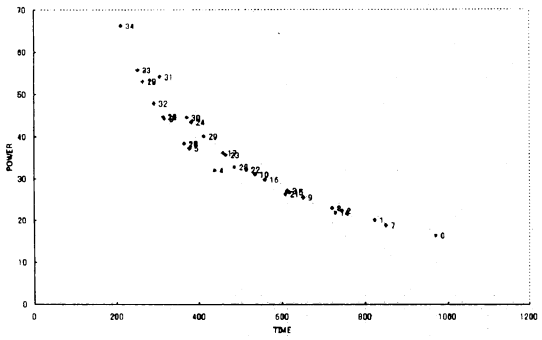

図 4.3　実験結果

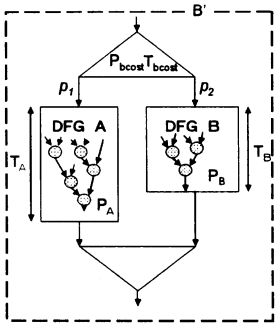

図 5.1　分岐

5. 応用アルゴリズム

基本アルゴリズムで各 DFG についてのトレードオフカーブを導き出すことが可能になった．これをより一般的な回路を想定して使用するためにはCDFGの形からトレードオフを導き出す必要がある．つまり，制御を考慮したトレードオフを出力しなければならない．この章で紹介する分岐とループは制御で主に使用されるものである．4章で紹介した基本アルゴリズムを応用することによって求める手法を述べる．

5.1. 分岐

CDFG の制御に値するものの一つである条件で状態を分岐させる時に使用する．分岐に使用するコンパレータやセレクタを分岐コストとし消費電力と時間である一定値を与えるものとする．また，分岐する確率は確率分布テーブルを参照し決定する．図 5.1 は分岐を表す．DFG A と DFG B のどちらかを通るルートによって消費電力は決まる．また，時間については大きい方に依存する．よってこれらを DFG のトレードオフカーブから解析的に求める．

DFG A はピーク電力を P_A，処理時間を T_A とし，n 個の集合体である．DFG B も同様に P_B，T_B を m 個の集合体である．また，DFG A と DFG B から出力した結果を P_T，T_T とする．制御に使用するコンパレータ，セレクタを考慮した分岐コストの値を P_{bcost}，T_{bcost} とする．確率分布テーブルより DFG A を選択する確率を p_1，DFG B を選択する確率を p_2 とする．以下にアルゴリズムを記述する．

計算手順　分岐

$G = \Phi$
(DFG A,Bのトレードオフグラフの各点i,jについて)
{　$P_T = p_1 \cdot P_{A[i]} + p_2 \cdot P_{B[j]} + P_{bcost}$
　(T_AがT_Bよりも大きいならば)
　{　$T_T = T_{A[i]} + T_{bcost}$
　}
　(T_BがT_Aよりも大きいならば)
　{　$T_T = T_{B[j]} + T_{bcost}$
　}
　Gに(P_T, T_T)を加える
}
B'のトレードオフグラフをGとする

5.2. ループ

ループは前判定，後判定によって状態が変わる．よってこれらを分けて考える．前判定の場合は DFG に一度も到達しない場合があるが，後判定の場合は必ず一回は DFG に入力が入る．分岐同様にこちらも導き出されたトレードオフカーブから解析的に求める手法をとる．よって，繰り返す回数を各最適な点の値に掛け合わせることから求める．図 5.2 は前判定，図 5.3 は後判定の状態を示す．

前判定，後判定共に DFG A はピーク電力を P_A，処理時間を T_A とし，n 個の集合体である．また，結果を P_T，T_T とする．制御に使用するコンパレータ，セレクタを考慮したループコストの値を P_{lcost}，T_{lcost} とする．ループ回数を N とする．以下にアルゴリズムを記述する．

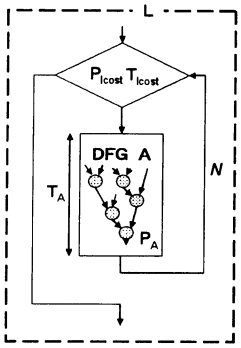

図 5.2　前判定ループ

計算手順　前判定ループ

$G'=\Phi'$

(DFG Aのトレードオフグラフの各点iについて)

{　$P_T = N(P_{A[i]}+P_{lcost})$

$T_T = N(T_{A[i]}+T_{lcost})$

G'に(P_T,T_T)を加える

}

LのトレードオフグラフをG'とする

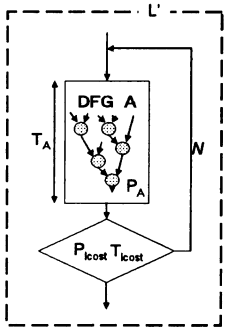

図 5.3　後判定ループ

計算手順　後判定ループ

$G''=\Phi''$

(DFG Aのトレードオフグラフの各点iについて)

{　$P_T = P_{A[i]}+(N-1)(P_{A[i]}+P_{lcost})$

$T_T = T_{A[i]}+(N-1)(T_{A[i]}+T_{lcost})$

G''に(P_T,T_T)を加える

}

L'のトレードオフグラフをG''とする

6. 結論

基本アルゴリズムにおいて DFG からトレードオフカーブを生成することを示した．また，それを部分的に用いて，大規模回路で通常用いられる分岐，ループを含む動作表現 CDFG のトレードオフカーブを生成する手法を述べた．

本稿では消費電力と時間という２次元で DFG を評価した．今後，パイプラインを想定したハードウエア最適化を行う場合の有効なトレードオフデータの抽出とその活用方法を検討する．この場合，消費電力と時間以外に，面積を指標として取り入れる必要性があるので，３次元的なトレードオフ表現を含め，有効な手段を検討する．また，配線とコントロール部のオーバーヘッドの考慮についても検討する．

謝辞

本研究において，立命館大学の山内寛紀教授には有益なご助言に対して感謝致します．また，福井研究室の各位には，日ごろのシステム構築に関する協力と議論に感謝致します．

本研究の一部は科研費基盤研究(C)(2) 16560316(平成 16 年～17 年)「大規模電子システムの仕様設計段階における高精度消費電力解析手法の研究，都市エリア産学官連携促進事業「診断治療のためのマイクロ体内ロボットの開発(びわこ南部エリア)」による．

文　献

[1] 今井正治，“システムレベルデザインに向けて，”情報処理，Vol.45, No.5, P451-P455, May 2004.

[2] 黒坂均，竹村和祥，橘昌良，“システムレベル設計フローと設計言語，”情報処理，Vol.45, No.5, P456-P463, May 2004.

[3] 大塚正人，荒木大，吉田紀彦，“システムのモデル化と計算モデル，”情報処理，Vol.45, No.5, P464-P470, May 2004.

[4] 塚本泰隆，温兆祺，“動作合成技術の動向，”情報処理，Vol.45 No.5, P471-P476, May 2004.

[5] 荒木大，竹村和祥，齊藤博文，“低消費電力設計と消費電力見積もり，”情報処理，Vol.45 No.5, P492-P499, May 2004.

[6] Daniel D. Gajski, “Principles of Digital Design,” PRENTICE HALL, 1997.

[7] John P. Elliott, “Understanding Behavioral Synthesis,” Kluwer Academic Publishers, 1999.

[8] 押川克寛，福井正博，“SystemC を用いた動作レベル電力解析の一手法，”DA シンポジウム 2004 論文集, Vol.2004, No.8, pp.307-312, July. 2004.

[9] 福井正博，“[招待講演]システム設計における性能推定技術，”情報処理学会関西支部 VLSI システム研究会, July. 2004.

[10] 福井正博，“[チュートリアル講演]システム設計の概要 -システム設計で見えるもの，見えないもの-，“情報処理学会 研究報告, 2004-SLDM-115, pp.19-21 May 2004.