

LSI のデータ通信消費電力を削減するリソースバインディング手法

世渡 秀和[†] 伊藤 和人[‡]

[†] 埼玉大学 理工学研究科

[‡] 埼玉大学 情報メディア基盤センター

〒338-8570 埼玉県さいたま市桜区下大久保 255

E-mail: [†] setohide@elc.ees.saitama-u.ac.jp, [‡] kazuhito@ees.saitama-u.ac.jp

あらまし LSI チップ上のモジュール間データ通信に伴う電力消費は、モジュールの配置だけでなくモジュール間のデータ通信状況にも依存する。モジュール間データ通信状況はモジュールへの演算やデータのバインディングによって決まり、さらにバインディングの可否は演算スケジュールによって決まる。データ通信消費電力最小化をバインディング最適化とフロアプラン最適化を分けて行う場合、データ通信消費電力を最小化するフロアプランが得られるようなバインディングを得ることが重要となる。本研究ではデータ通信消費電力削減可能なバインディングを選択するための評価関数を提案し、その有効性を比較検討する。

キーワード 低消費電力、スケジューリング、バインディング、フロアプラン、データ通信消費電力

A Resource Binding Method for Reducing Power Consumption of LSI Data Communications

Hidekazu SETO[†] and Kazuhito ITO[‡]

[†] Graduate School of Science & Engineering

[‡] Information Technology Center

Saitama University

255 Shimookubo, Sakura-ku, Saitama, Saitama 338-8570 Japan

E-mail: [†] setohide@elc.ees.saitama-u.ac.jp, [‡] kazuhito@ees.saitama-u.ac.jp

Abstract The power consumption by data communications on a LSI chip depends on the layout of modules as well as how data are communicated among modules. The data communications are determined by the resource binding of operations to functional units and data to registers. The binding in turn depends on the operation schedule. To separate the binding optimization and floorplan optimization to minimize the power consumption by data communications, it is important to derive a binding which introduces the floorplan with minimized power consumption. In this paper, several evaluation functions of the resource binding are proposed and the effectiveness of each evaluation function is compared.

Keyword Low Power, Scheduling, Binding, Floorplan, Data Communication

1. はじめに

大規模集積回路(LSI)が消費する電力のうち、演算器やレジスタ等のモジュール間データ通信によるものが大きな割合を占めている[1]。配線に起因する消費エネルギーは、通常の配線長では、配線長にほぼ比例することが報告されており[2]、データ通信は通信路を構成する配線を用いて行うので、データ通信の消費電力はその通信路配線長に比例する。そのため、一般にデータの通信元と通信先を物理的に近接して配置することによって配線を最短化し、データ通信の消費エネルギーを最小化することが可能となる。

LSI は、実現したい処理内容を入力として、スケジューリング、バインディング、フロアプランの各工程により設計される。スケジューリングでは演算の実行

開始時刻が決定され、バインディングでは各演算をどの演算器で実行するか、各データをどのレジスタに記憶するかが決定される。そしてフロアプランでは各モジュールの位置と向きが決定される。

データ通信消費電力は、モジュール間通信路長と各通信路を利用して行う通信回数の両方に依存する。モジュール間データ通信路長はフロアプランにより決定される。しかし、バインディングはモジュール数を決定するためフロアプランに影響するので、通信路長は実際にはフロアプランとバインディングに依存する。モジュール間データ通信回数はバインディングに直接依存する。そしてバインディングはスケジューリング結果に依存する。すなわち、データ通信消費電力最小化のためにはスケジューリング、バインディング、フロアプランの3つの組み合わせの最適化を行う必要が

ある[3]。しかしその組み合わせ数は膨大であり、最適解を見つけるには多大な時間を要する。

データ通信消費電力を最小化する設計を短時間で得るため、最適化問題をスケジューリングとバインディング、およびフロアプランの2つの部分問題に分割し、前者の最適化解に対して後者の最適化問題を適用する手法が提案されている[4]。そこでは、データ通信消費電力を最小化する適切なバインディングを得るためのスケジューリングとバインディングの一手法が提案されているが、その妥当性について十分な検討がなされていない。

そこで本論文では、データ通信消費電力を最小化するフロアプランを導きうる適切なバインディングを得るためのバインディング手法を複数種類考案し、その比較検討を行う。

2. データ通信消費電力最小化

文献[4]で示されたデータ通信消費電力(エネルギー)を最小化するためのスケジューリング・バインディングとフロアプランを分離した最適化手法を簡単に確認する。

ある処理ALGが与えられ、そのLSI設計を考える。LSI設計は一般に複数種類(加算、乗算など)かつ複数個の演算器と複数個のレジスタからなり、ALG中の演算実行や演算データの記憶を行う。演算器およびレジスタ(以下モジュールと呼ぶ)の集合を M とする。バインディングの結果、ALGを1回実行するときにモジュール i, j 間に必要なデータ通信回数 C_{ij} が決定する。また、フロアプランの結果、各モジュールの配置およびモジュール間配線経路が決定する。セグメント化バス(分割バス[5])またはマルチプレクサを用いたpoint-to-point接続によりモジュール i, j 間のデータ通信経路を最短化し、その経路長を WL_{ij} とする。データ通信消費エネルギーは経路長に比例することから、ALGを1回実行するときの全データ通信消費エネルギー Ec は以下の式で与えられる。

$$Ec = \sum_{i, j \in M} (K_{ij} C_{ij} WL_{ij}) \quad (1)$$

ここで K_{ij} はモジュール i, j 間のデータ通信消費エネルギー係数とする。 K_{ij} はLSIプロセスに依存する値であり、全てのモジュール間通信で共通と仮定すると、データ通信消費エネルギーは以下のパラメータ Pc で測ることができる。

$$Pc = \sum_{i, j \in M} (C_{ij} WL_{ij}) \quad (2)$$

式(2)より、 Pc を低減するには C_{ij} および WL_{ij} を削減すればよいことが分かる。ところが、 C_{ij} の総和は与えられた処理ALGに固有の値となり自由に減少できず、 WL_{ij} についても相互に影響させずに削減することは

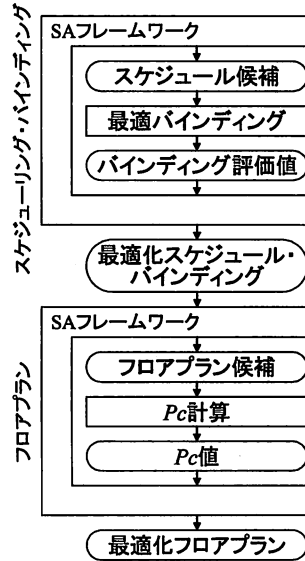


図1 スケジューリングとバインディング、およびフロアプランの関係

一般的に不可能である。そのため、 C_{ij} の値が均衡していると Pc を大きく削減することは難しく、むしろ積極的に C_{ij} の値に偏りをつくり、 C_{ij} の大きい通信路を優先して配線長 WL_{ij} を短くすることで Pc の低減を図ることができる。

モジュール間データ通信回数 C_{ij} は、スケジューリングとバインディングに依存し、フロアプランとは無関係である。そこで、まず C_{ij} の値が大きく偏るようにスケジューリングとバインディングを行い、その後に C_{ij} の値に基づいて、 C_{ij} の大きいモジュール i, j が近接して配置するようにフロアプランを行うことで Pc を低減したLSI設計を得ることが出来る。

モジュール間データ通信回数 C_{ij} は直接的にはバインディングの結果として決定されるが、バインディングはスケジュールに強く依存するため、 C_{ij} を最適化するにはスケジューリングとバインディングを同時に考慮すべきである。そこで、スケジュール候補に対して決定的な最適化バインディングを行い、評価値が最大(または最小)のバインディングが得られるスケジュールを選択することで C_{ij} を最適化するスケジューリングとバインディングの組み合わせを求める。

スケジューリングとバインディング、およびフロアプランの関係を図1に示す。

3. データ通信電力消費最小化

3.1. スケジュール探索

スケジューリングはDFG中の全ての演算間先行制約を満足し、与えられた目的関数を最適化するように

演算実行開始時刻を決定する問題であり、NP 困難な問題である。本研究では、焼き鈍し法(SA)により最適なスケジュールを探索する手法[4]を用いる。そこでは演算間に 0 以上の Strut(待ち時間)を挿入して、ASAP スケジュールを求める事で演算実行時刻を決定するが、Strut の組み合わせを SA により探索することで目的関数を最適化するスケジュールを見つける(図 2)。データ通信消費エネルギー最小化においては、 P_c を最小化するフロアプランを導きうるバインディングが得られるスケジュールを探索することを目的とし、3.2 節で示すバインディング評価値を目的関数とする。

3.2. 電力消費指向バインディング

与えられたスケジュール(演算実行時刻)に基づきバインディングを行う。得られたバインディングに、後に示す評価関数を適用して与えられる評価値を SA によるスケジュール探索の際の目的関数値 $Cost$ とする。

入力として与えられたスケジュールについて、繰り返し周期が T_r のとき、時刻 t は $tc = t \bmod(T_r)$ で与えられる時刻クラスに分類される。すなわち各演算 $k \in N$ の実行時刻 t_k は時刻クラス $t_k \bmod(T_r)$ に属し、同じ時刻クラスに属する演算は同時に実行されることを意味する。したがって、同じ時刻クラスに属する演算は、それぞれ異なる演算器に割り当てなければならない。

同様にデータについても、保持期間中の時刻に対して時刻クラスが定まり、同じ時刻クラスにおいて保持すべきデータはそれぞれ異なるレジスタに割り当てる。

バインディングは以下のように行う。未処理の時刻クラス tc を選び、 tc で実行する演算 $I(tc)$ と対応する演算器の全ての可能なバインディングを列挙する。一般に演算数を n 、演算器数 m とすれば、可能なバインディングは $m \cdot P_n$ 個存在する。既にレジスタに割り当て済みの演算入出力データと演算間データ通信状況に基づき、各バインディング候補を評価し、評価値 S_α が最大(評価方式によっては最小)のものをを選び、 $I(tc)$ のバインディングを確定すると共に tc を処理済みとする。

$I(tc)$ が出力するデータ集合 $R(tc)$ について、利用可能なレジスタとの可能なバインディングを列挙し、演算器とレジスタ間のデータ通信状況に基づき、最適なものを選んで $R(tc)$ のバインディングを確定する。データ保持状況により、最初に見積もったレジスタ数では割り当てできない場合は、必要数だけレジスタを増やす。

以上述べた時刻クラスに関する部分バインディングを未処理時刻クラスが無くなるまで繰り返し行うことで全演算、全データのバインディングを得る(図 3)。

なお、モジュール数は直接データ通信消費電力に影響せず、モジュール間データ通信回数の最適化がデータ通信消費電力に効果があるため、モジュール数最小化は考慮しない。

- 2 初期解として適当なスケジュール SC を得る。 SC に目的関数を適用し、評価値 $Cost$ を得る。
- 3 温度 T を T_{start} に初期化する。
- 4 SC の近傍解 SC' を得る
- 5 SC' について目的関数を適用し、評価値 $Cost'$ を得る。(図 2 を参照)
- 6 $Cost > Cost'$ の場合、 $SC = SC'$ とする。または、 $Cost < Cost'$ の場合は以下の確率 p で $SC = SC'$ とする。

$$p = \exp\left(-\frac{Cost - Cost'}{T}\right)$$

- 7 所定回数 M だけ 3 から 6 を繰り返す。
- 8 温度 T に係数 $\alpha < 1$ を乗じることで T を低下させる。あらかじめ決められた終了温度 $T_{end} > T$ であれば終了する。さもなければ 3 に戻る。
- 9 最小の評価値 $Cost$ を与える SC を解として出力する。

図 2 SA を用いたスケジュールの探索

1. データの保持期間に時刻クラス tc を含むデータの保持期間の二乗和を求め、最大数を与える時刻クラスを選び、データを個別のレジスタに割り当てる。
2. 演算種類ごとに同時に実行する演算数の最大値を調べ演算器の必要最小数を求める。また同時に保持が必要なデータ数の最大値を調べレジスタの必要最小数を求める。
3. $I(tc)$ の入力データ、出力データについて未割り当てのものが最少の時刻クラス tc を選ぶ。
4. 評価 S_α が最大(最小)となるように $I(tc)$ の部分バインディングを行う。
5. 評価 S_α が最大(最小)となるように $R(tc)$ のバインディングを行う。現在のレジスタ数では $R(tc)$ の割り当てが出来ない場合は必要最小なレジスタを増やす。
6. 時刻クラス tc を処理済みとする。未処理の時刻クラスが存在すれば 3.へ戻る。
7. 得られた全体バインディングに対し、評価 S を求める。

図 3 バインディングアルゴリズム

2 節で述べたように、モジュール間通信回数 C_{ij} の値の偏りが生じるバインディングを得ることを目的とする。そのため、3.2.1 節以降に示すバインディング評価関数を検討する。 S_v は方式 v による全体バインディング評価関数であり、 S_{av} は方式 v による評価値 S_v を最小化する全体バインディングを得ることを目的として行う

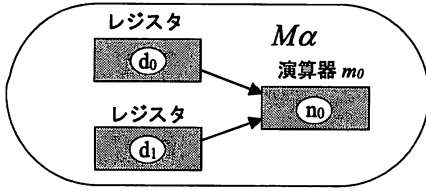


図 4 Ma の概念図

部分バインディングの評価関数である。各変数を以下のように定義する。

M	モジュールの集合
$C_{ij} \ (i, j \in M)$	モジュール (i, j) 間の通信回数
$F_{ij} = \begin{cases} 1 & (C_{ij} \geq 1) \\ 0 & (C_{ij} = 0) \end{cases}$	モジュール (i, j) 間通信の有無
$\overline{C}_i = \frac{\sum_{j \in M} C_{ij}}{\sum_{j \in M} F_{ij}}$	モジュール i における各通信先への通信回数の平均値
Ma	ノード n_0 をモジュール m_0 にバインドする場合、 n_0 が使用するデータを保持するモジュール全てと、 m_0 の集合 (図 4)

3.2.1. 評価 A : ファンアウト

各モジュールの出力先モジュール数 (ファンアウト) の 2 乗の総和を以て評価値とし、値が小さいほど望ましいものとする。ファンアウトが小さければデータ通信が少数の配線に集中する事が期待できる。この評価は [4] で提案されたものである。

$$S_{FO} = \sum_{i \in M} \left(\sum_{j \in M} F_{ij} \right)^2 \quad (3)$$

$$Sa_{FO} = \sum_{i \in Ma} \left(\sum_{j \in M} F_{ij} \right)^2 \quad (4)$$

3.2.2. 評価 B : モジュール間通信回数

モジュール間の通信回数の二乗の総和を評価値とし、値が大きいほど望ましいものとする。通信回数の多い通信路が多いほど、通信が少数の通信路に集中する事が期待できる。

$$S_{CC} = \sum_{i, j \in M} C_{ij}^2 \quad (5)$$

$$Sa_{FO} = \sum_{i \in Ma} \left(\sum_{j \in M} C_{ij} \right)^2 \quad (6)$$

3.2.3. 評価 C : モジュール毎分散

各モジュール毎に算出した出力通信回数の分散の

全モジュールに対する総和を評価値とし、値が大きいほど望ましいものとする。単一のモジュールに通信が集中するのを防ぐ効果が期待できる。

$$S_{TV} = \sum_{i \in M} \left(\frac{\sum_{j \in M} (C_{ij} - \overline{C}_i)^2}{\sum_{j \in M} F_{ij}} \right) \quad (7)$$

$$Sa_{TV} = \sum_{i \in Ma} \left(\frac{\sum_{j \in M} (C_{ij} - \overline{C}_i)^2}{\sum_{j \in M} F_{ij}} \right) \quad (8)$$

3.2.4. 評価 D : 全体分散

全通信路における通信回数の分散を以て評価値とし、値が大きいほど望ましいものとする。全通信の通信回数をばらつかせることができ、同時に通信路数を減らす効果も期待できる。

$$S_{AV} = \frac{\sum_{i, j \in M} (C_{ij} - \overline{C})^2}{\sum_{i, j \in M} F_{ij}} \quad (9)$$

$$Sa_{AV} = \frac{\sum_{i \in Ma} \sum_{j \in M} (C_{ij} - \overline{Ca}_i)^2}{\sum_{i \in Ma} \sum_{j \in M} F_{ij}} \quad (10)$$

ただし、 $\overline{Ca}_i$ はモジュール i ($i \in Ma$) における各通信先への通信回数の平均値を表す。

$$\overline{Ca}_i = \frac{\sum_{j \in M} C_{ij}}{\sum_{j \in M} F_{ij}} \quad (11)$$

3.2.5. 評価 E : 通信路数

総通信路数を以て評価値とし、値が小さいほど望ましいものとする。通信路数を減少させることにより、総配線長を短くする事が期待できる。通信路数が少ないことは、他にも配置配線の複雑性、可検査性等の点において優れており、しばしば回路規模最小化バインディングの目標として用いられる [6]。

$$S_{CL} = \sum_{i, j \in M} F_{ij} \quad (12)$$

$$Sa_{CL} = \sum_{i \in Ma} \sum_{j \in M} F_{ij} \quad (13)$$

3.3. 電力消費指向フロアプラン

バインディング結果に基づいて、Sequence-Pair [7] を

用いた SA によって最適なフロアプランを探索する。各フロアプラン候補についてデータ通信の転送元と転送先の最短経路を求めて配線長とし、式(2)の P_c を最小化するフロアプランを求める。

4. 実験結果

提案するスケジューリングとバインディング手法の有効性を確認するため実験を行った。実験の概要を図5に示す。与えられた処理について、指定した繰り返し周期 Tr で実行可能な多数のスケジュール候補を生成し、各スケジュール候補 SC について3.2節のバインディングを行って最適化(最大化または最小化)したバインディングおよびその評価値 S を求める。得られたバインディング結果に対して、データ通信消費電力パラメータ P_c を最小化するフロアプランを求める。多数のスケジュール候補により多数の (S, P_c) の対が得られ、 S と P_c の相関の様子から、スケジュールとバインディングの最適化(S の最適化)がデータ通信消費電力を最小化するフロアプラン(P_c の最小化)を導きうるか否か、および複数種類のバインディング評価関数のうち、データ通信消費電力最小化に最も効果があるのはどれかを知ることができる。

提案手法を C++言語を用いて実装し、PC(Core 2 Duo 2.66GHz, 2GB メモリ)上で実行した。

実験に用いた DFG は 5 次ウェーブ楕円フィルタ(WEF)[8]とその3次アンフォールディング(WEF3)、及び8点1次元DCT[9]の3種類を用いた。ハードウェア仕様は、文献[4]と同じものを用いた。各 DFG についてそれぞれ3通りの繰り返し周期 Tr を指定し、それぞれ4.2節で示した各評価関数を用いて設計を行った。同一のバインディング結果に対しフロアプラン結果は一意に定まらないため、それぞれ $k=2$ 回ずつフロアプランを行った。

図6にWEF3、 $Tr=53$ の時の各評価関数と消費電力との関係を示す。グラフの縦軸は式(2)で与えられる P_c である。横軸は各評価関数によるバインディング評価値を示しており、サンプル中の最大値により正規化している。比較を容易にするため、評価B、評価C、評価Dは横軸を反転している。なお、図中の $RMSE$ は式(14)で定義される測定値のばらつき具合を示す指数で、平均二乗誤差の平方根で与えられる。本実験では同一の

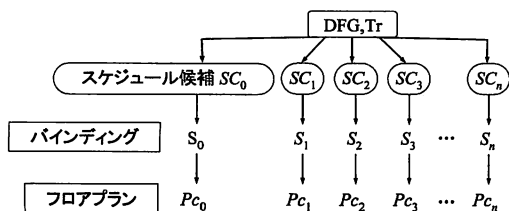


図5 ある DFG, Tr での設計における多数の (S, P_c) 取得法

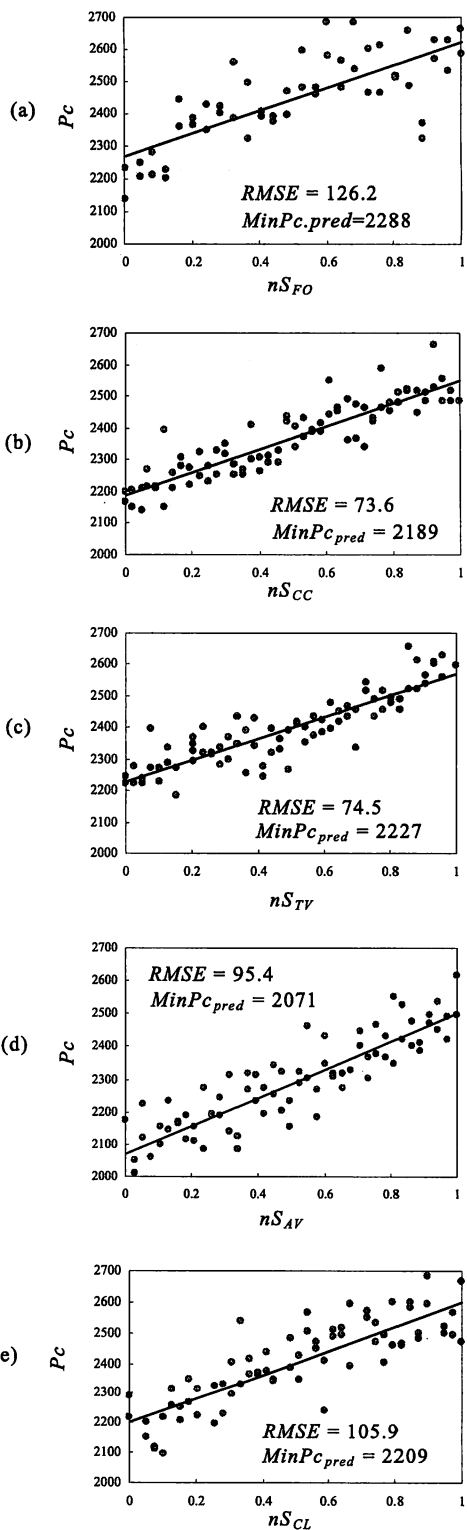


図6 WEF3, $Tr=53$ における各評価関数と P_c との関係
(a)評価 A, (b)評価 B, (c)評価 D, (d)評価 C, (e)評価 E

表 1 各 DFG, 周期における各評価関数毎の $MinPc_{pred}$ と $RMSE$

DFG	周期	$MinPc_{pred}$					$RMSE$				
		評価A [4]	評価B	評価C	評価D	評価E	評価A [4]	評価B	評価C	評価D	評価E
WEF	16	677	686	698	669	692	54.0	30.6	37.0	42.4	38.9
	17	667	644	692	623	687	49.9	41.2	34.0	47.7	43.9
	18	660	638	669	629	674	43.1	34.5	38.5	45.1	42.8
DCT	14	810	714	818	724	836	69.8	55.2	65.6	58.9	51.8
	15	831	767	834	766	801	68.6	70.0	57.6	53.1	57.0
	16	806	740	847	789	786	67.8	48.7	48.3	55.1	59.1
WEF3	48	2283	2200	2232	2071	2281	110.0	94.1	93.8	118.6	121.0
	50	2337	2195	2228	2124	2281	126.7	99.7	90.8	104.9	102.7
	53	2288	2189	2227	2071	2209	126.2	73.6	74.5	95.4	105.9

バインディングに対して k 回フロアプランを行っているので、その平均値に対する $RMSE$ を計算する。式(14)では、 x_i をバインディングとし、その集合を X とする。また、 x_i に対する k 回目のフロアプラン結果の Pc を y_{ik} とし、その集合を Y_i 、平均値を $\bar{y}_i$ とする。

$$RMSE = \frac{1}{|X|} \sum_i \left(\frac{1}{|Y_i|} \sum_k (y_{ik} - \bar{y}_i)^2 \right) \quad (14)$$

図6中の $MinPc_{pred}$ は各グラフにおける Pc の最小推定値(近似直線の y 切片)を示している。消費電力最小化を目的としていることから、比較に用いるサンプルは Pc の推定値が各グラフの $MinPc_{pred}$ の最小値の125%以下であるものに限定した。回帰直線はそのサンプルついて最小二乗法で求めた近似直線である。

図6では評価B(b)が最も $RMSE$ 値が低く、測定値が近似直線付近にまとまっている一方、最も $RMSE$ 値の高い評価A(a)は測定値がばらついている。 $RMSE$ 値が大きいと、バインディングの最適化を行っても、それがデータ通信消費電力最小化に結びつきにくいことを意味する。

一方 $MinPc_{pred}$ は消費電力の最小推定値を示しており、それが低いほど、より大きな消費電力の削減が期待できる事を意味する。図6では評価D(d)がもっとも $MinPc_{pred}$ が低い、 $RMSE$ が比較的大きい。しかし、ばらつきの範囲を考慮しても他の評価法の $MinPc_{pred}$ よりも小さく、最も大きなデータ通信消費電力削減が期待できると考えられる。

表1に各DFG、繰り返し周期における $RMSE$ と $MinPc_{pred}$ を示す。太字で示されている値は、その設定における最小値を示している。 $MinPc_{pred}$ については、ほとんどの場合で全体分散(評価D)が最小値である。全体分散の $RMSE$ は必ずしも最小ではないが、他の評価法と比べて遜色ない値である。以上より、今回比較した5つの評価の中では、全体分散(評価D)がデータ通信消費電力の削減に最も効果が高いと言える。

また、一般的な方法である通信路数最小化(評価E)は、 $MinPc_{pred}$ 、 $RMSE$ 共に比較的高く、データ通信消費エネルギーの観点からは、必ずしも有効なバインディング評価ではないことが分かる。

5. まとめ

本研究ではVLSI設計をスケジューリング・バインディングとフロアプランの二段階に分けて最適化を行う手法について、スケジュールの探索により最適なバインディングを得るために用いる評価関数をいくつか提案し、それらの比較検討を行った。

実験により、既存の手法に対し今回提案したモジュール間データ通信回数の全体分散を最大化する手法がデータ通信消費エネルギー最小化という観点でより優れたものであることを示した。

文 献

- [1] S. Komatsu, M. Ikeda, and K. Asada, "Low Power Microprocessors for Comparative Study on Bus Architecture and Multiplexer Architecture," Proc. ASP-DAC '98, pp. 323-324, 1998.
- [2] N. Tylor, S. Dey, and Y. Zhao, "Modeling and Minimization of Interconnect Energy Dissipation in Nanometer Technologies," Proc. DAC 2003, pp. 754-757, 2003.
- [3] P. Prabhakaran, P. Banerjee, J. Crenshaw, and M. Sarrafzadeh, "Simultaneous Scheduling, Binding and Floorplanning for Interconnect Power Optimization," VLSI Design 1999, pp. 423-427, 1999.
- [4] 伊藤, "スケジューリング探索によるデータ通信消費電力削減," 信学技報, CPSY2005-81, pp. 25-30, 2006.
- [5] R. Lu and C.-K. Koh, "A High Performance Bus Communication Architecture through Bus Splitting," Proc. ASP-DAC 2004, pp. 751-755, 2004.
- [6] T. Yorozyua, K. Ohashi, M. Kaneko, "Assignment-Driven Loop Pipeline Scheduling and Its Application to Data-Path Synthesis," IEICE Trans. Fundamentals, Vol. E85-A, No. 4, pp. 819-825, April 2002.
- [7] M. Murata, S. Natakake, K. Fujiyoshi, and Y. Kajitani, "VLSI Module Placement based on Rectangle Packing by Sequence-Pair," IEEE Trans. Comput. Aided Des. Integrated Circuits & Syst., vol. 15, pp. 1518-1524, 1996.
- [8] S. M. Heemstra de Groot, S. H. Gerez, and O. E. Herrmann, "Range-Chart-Guided Iterative Data-Flow Graph Scheduling," IEEE Trans. Circuits Syst.-I: Fund. Theory & Appl., vol. 39, pp. 351-364, 1992.
- [9] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical Fast-1D DCT Algorithms with 11 Multiplications," Proc. IEEE ICASSP '89, pp. 988-991, 1989.