

## データ流量解析に基づくアーキテクチャレベルでの 設計品質見積もり手法

米岡 昇<sup>†</sup> 上田 恭子<sup>†</sup> 坂主 圭史<sup>†</sup> 武内 良典<sup>†</sup> 今井 正治<sup>†</sup>

<sup>†</sup> 大阪大学 大学院情報科学研究科 情報システム工学専攻  
〒565-0871 大阪府吹田市山田丘 1-5

E-mail: {yoneoka@aoba.ics.es.osaka-u.ac.jp, {k-ueda,sakanusi,takeuchi,imai}@ist.osaka-u.ac.jp}

あらまし SoC などの大規模なシステムにおいては、開発期間の短縮のため、設計資産の再利用を行う IP ベース設計が着目されている。IP ベース設計においては、処理時間や面積、消費電力などの厳しい設計制約の中で、いかに最適なアーキテクチャを探索するかということが非常に重要な問題となっている。そこで本稿では、SystemC を用いたシステムレベルにおけるプロセス間のデータ流量解析に基づく消費電力量と面積の見積もり手法を提案し、設計早期段階において、多数のアーキテクチャの設計品質見積もりを行う手法を提案する。評価実験においては、音声・動画処理用 VLSI を想定し、提案手法を用いて多数のアーキテクチャの性能、面積、消費電力量のトレードオフを比較することが可能であることが確認された。

キーワード IP ベース設計, SystemC, 設計品質見積もり, バスアーキテクチャ

## Architecture Level Design Quality Estimation Method based on Data Flow Analysis

Noboru YONEOKA<sup>†</sup>, Kyoko UEDA<sup>†</sup>, Keishi SAKANUSHI<sup>†</sup>, Yoshinori TAKEUCHI<sup>†</sup>, and  
Masaharu IMAI<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University  
1-5 Yamada-Oka, Suita-shi, Osaka, 565-0871 Japan

E-mail: {yoneoka@aoba.ics.es.osaka-u.ac.jp, {k-ueda,sakanusi,takeuchi,imai}@ist.osaka-u.ac.jp}

**Abstract** In this paper, an architecture-level area and power estimation method is proposed. This method is based on system level data-flow analysis using SystemC. In the experiment, the proposed method is applied to an audio/video encoding application system. The experimental results show that proposed method can estimate design qualities of different 32 architectures in short time. The proposed method is very effective for architecture level design exploration in embedded system design.

**Key words** IP-based design, SystemC, Design quality estimation, Bus architecture

### 1. はじめに

近年、半導体技術の進歩により、従来複数個の LSI によって構成されていたシステムを一つの LSI に集積することが可能になった。このような大規模システムにおいては、開発期間の短縮のため、抽象度の高い言語を用いた C ベース設計や、設計資産の再利用を行う IP (Intellectual Property) ベース設計が着目されている。アーキテクチャ設計においては、処理時間や面積、消費電力といった厳しい設計制約を満たすため、多数のアーキテクチャの評価を行なう必要があり、設計早期段階での性能、

面積、消費電力の見積もりを行うことが重要となっている。

そこで本稿では、コンポーネント間のデータ流量に着目したアーキテクチャ設計品質見積もり手法を提案する。組み込みシステム設計の設計早期段階における処理時間の見積もりに関しては、システムをトランザクションレベルモデル (TLM) で記述し、アーキテクチャレベルでのシミュレーションにより評価を行う手法 [1]~[4] が提案されている。しかし、一つのアーキテクチャに対してそれぞれアーキテクチャレベルの記述、シミュレーションが必要となるため、多数のアーキテクチャを比較する場合には、記述の修正やシミュレーションに多くの時間が必

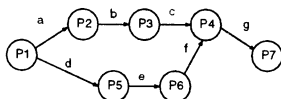


図1 システムレベルモデルの例

要となる。そこで、SystemC [5] 記述のシステムレベルのプロファイリング結果を用いた性能見積り手法が提案された [6]。この手法は、アーキテクチャに依存しないシステムレベルでプロファイリングを行い、そのプロファイリング結果とアーキテクチャ情報から性能見積りを行うため、多数のアーキテクチャに対して高速に性能を見積もることができる。

処理時間、面積、消費電力を考慮した設計手法としては、プロセッサのキャッシュ容量や専用ハードウェアの有無による複数のアーキテクチャに対して、処理時間、面積、消費電力を見積もる手法 [7], [8] が提案されているが、機能ブロックの構成やバス構成がある程度固定されており、対象アーキテクチャの範囲が限定されている。

本稿では、システムレベルのプロファイリング結果を用いたアーキテクチャレベルでの性能見積り手法 [6] を利用した、面積および消費電力量見積り手法を提案する。提案手法により、多数のアーキテクチャの処理時間、面積、消費電力量を比較することが容易になる。

本稿は次に示す構成となっている。まず、第2節で見積り対象のモデルについて述べる。そして、第3節で面積および消費電力量の見積り手法について述べ、第4節ではバス電力見積り精度に関する実験と、音声・動画処理用 VLSI に本手法を適用した結果について述べ、最後に第5節で本報告のまとめを述べる。

## 2. 見積り対象のモデル

本節では、提案する見積り手法が対象とするシステムのモデルについて述べる。

### 2.1 システムレベルモデル

システムレベルモデルは、システムの動作を表すプロセスと、プロセス間のデータ転送を表すチャンネルによって表現したモデルである。システムレベルモデルにおいて、プロセスは処理に必要とする全てのデータを受信後、データの処理を開始し、データの処理が完了した後、データを送信する。

図1にシステムレベルモデルの例を示す。ノード P1, P2, P3, P4, P5, P6, P7 がプロセスを、有向枝 a, b, c, d, e, f, g がチャンネルとそのデータ転送方向を表す。プロセス P1 で処理されたデータは、チャンネル a, d を経由し、プロセス P2 と P5 へ送信される。

### 2.2 アーキテクチャレベルモデル

アーキテクチャレベルモデルは、機能ブロック群構成とバスアーキテクチャから構成される。各機能ブロックには1個以上のプロセスがマッピングされ、各バスには1個以上のチャンネルがマッピングされる。マッピングされた各プロセスの実行時間や消費電力は、あらかじめ IP データベースに登録されている。

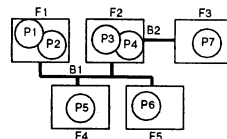


図2 アーキテクチャレベルモデルの例

図2にアーキテクチャレベルモデルの例を示す。矩形 F1, F2, F3, F4, F5 は機能ブロックを、直線 B1, B2 はバスを表す。機能ブロック F1 は、バス B1 に接続されており、プロセス P1 と P2 の処理を行う。

## 3. 面積、消費電力量見積り手法

システムの面積は、機能ブロックの面積  $A_{FB}$  とバスの面積  $A_{Bus}$  に大きく分かれる。同様に、システムの消費電力量も、機能ブロックの消費電力量  $W_{FB}$ 、バスの消費電力量  $W_{Bus}$  に分けることができる。次節以降、機能ブロックの面積と消費電力、そして、バスの面積と消費電力の見積り手法について、順に説明する。

### 3.1 機能ブロックの面積、消費電力見積り手法

#### 3.1.1 機能ブロックの面積

機能ブロックの総面積は、対象アーキテクチャにおいて使用する IP の総面積となる。システムの機能ブロックの総面積  $A_{FB}$  は、式 (1) で見積もる。ここで、 $F$  は対象アーキテクチャに含まれる機能ブロックの集合、 $a(f_i)$  は機能ブロック  $f_i$  の面積を表す。

$$A_{FB} = \sum_{f_i \in F} a(f_i) \quad (1)$$

#### 3.1.2 機能ブロックの消費電力

機能ブロックの消費電力は、各機能ブロックの動作状態別の消費電力と、各状態の動作時間から求める。

システムレベルプロファイリングに基づく性能見積り手法 [6] を用いることで、各機能ブロックの状態とその時間を取得することができる。

本稿では、機能ブロックの状態として、以下の状態を想定する。

- プロセス実行状態 ( $exec(p_i)$ )

プロセス実行状態とは、機能ブロックにおいてプロセスのデータ処理が実行されている状態である。

- アイドル状態 (idle)

アイドル状態とは、機能ブロックでいつれのプロセスも実行されていない状態である。

- 省電力状態 (sleep)

省電力状態とは、消費電力を削減するため、使用していない機能ブロックへのクロックの停止などによる省電力モードである状態である。

システムの機能ブロックの総消費電力量  $W_{FB}$  は、式 (2) で見積もる。ここで、 $S_i$  は機能ブロック  $f_i$  の状態の集合、 $p(f_i, s)$ 、 $t(f_i, s)$  はそれぞれ、機能ブロック  $f_i$  の状態  $s \in S_i$  の消費電

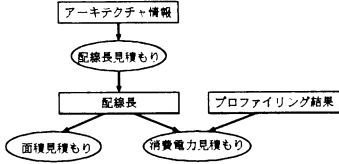


図3 面積および消費電力量の見積り手順

力、動作時間を表す。

$$W_{FB} = \sum_{f_i \in F, s \in S_i} \{p(f_i, s) \times t(f_i, s)\} \quad (2)$$

### 3.2 バスの面積、消費電力量見積もり手法

本節では、バスの面積、消費電力量の見積もり手法について述べる。提案手法は、配線長予測モデルを用いて見積もったバスの配線長を利用して、バスの種類別にバスの面積を見積もる。

図3に提案手法での見積り手順を示す。提案手法では、アーキテクチャの情報からまずバスの配線長を見積もる。そして、配線長から面積を見積り、配線長とプロファイリング結果から消費電力量を見積もる。

#### 3.2.1 バス配線長の見積もり

提案手法では、Rent's Rule [9] を応用し、配線のファンアウト数と接続された機能ブロックの面積から配線長を予測する手法 [10] を利用して、配線長を見積もる。この手法では、MRST (Minimum Rectilinear Steiner Tree) モデルを用いて、複数の機能ブロック間を接続する配線長を見積もる。

MRST モデルを図4に示す。Block Bounding Area は、対象バスに接続される機能ブロックを全て囲む矩形のうち、面積が最小のものを表す。また、接続するピンは Block Bounding Area に均等に配置されるものと仮定する。以上より、接続するピンを全て囲む矩形のうち面積が最小の Net Bounding Area の幅  $w_{nb}$  と高さ  $h_{nb}$  は式 (3)、式 (4) によって算出する。ここで、 $w_{bb}$ 、 $h_{bb}$  はそれぞれ、Block Bounding Area の幅と高さを表し、 $n_{pin}$  は接続するピンの本数を表す。

$$w_{nb} = w_{bb} \times \frac{n_{pin} - 1}{n_{pin} + 1} \quad (3)$$

$$h_{nb} = h_{bb} \times \frac{n_{pin} - 1}{n_{pin} + 1} \quad (4)$$

MRST の長さすなわち配線長  $l$  は、算出された Net Bounding Area の幅と高さ、ピン数を用いて、式 (5) によって算出する。配線長は、Net Bounding Area の半周長と、内部の接続線の長さの和となる。

$$l = (h_{nb} + w_{nb}) + 1.1(\sqrt{n_{pin}} - 2.0) \times \frac{h_{nb} \times w_{nb}}{h_{nb} + w_{nb}} \quad (5)$$

提案手法では、Block Bounding Area の幅と高さを式 (6) で見積もる。ここで、 $F_b$  をバス  $b$  に接続されている機能ブロックの集合とする。

$$w_{bb} = h_{bb} = \sqrt{\sum_{f \in F_b} A_f} \quad (6)$$

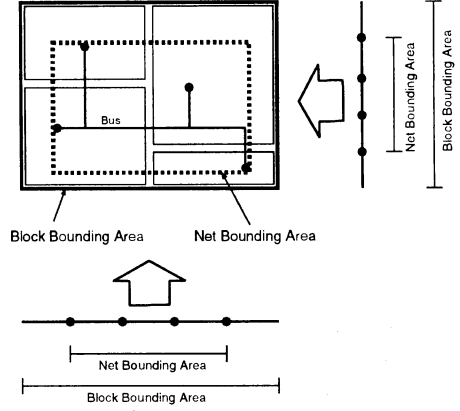


図4 MRSTにおける配線モデル

また、バス配線は、データ、アドレス、制御線など複数のネットから構成されるが、提案手法では全て同一配線長として扱う。以上より、提案手法ではバス  $b_i \in B$  の配線長  $l_{b_i}$  を、式 (7) を用いて見積もる。ここで、 $n_{pin_i}$  はバス  $b_i$  のピン数を表し、バスにマッピングされたチャンネルの入出力から、マスタ数およびスレーブ数を割り出し、バスのネットのピン数を算出する。

$$l_{b_i} = (0.9 + 0.55\sqrt{n_{pin_i}}) \sqrt{\sum_{f_j \in F_{b_i}} A_{f_j}} \quad (7)$$

#### 3.2.2 バス面積の見積もり

バスの面積  $A_{Bus}$  は、各バス  $b_i$  の面積  $A_{b_i}$  の和として見積もる。各バスの面積は、主にバス配線における面積と、調停回路やマルチプレクサ等のロジックによる面積とに分けられる。バス  $b_i \in B$  の配線面積  $A_{wire_i}$  は、図5に示すモデルを用いて、式 (8) によって見積もる。ここで、 $w_{pitch}$ 、 $n_{data_i}$ 、 $n_{addr_i}$  はそれぞれ、配線幅と配線間隔の和、バス  $b_i$  のデータ配線本数とアドレス配線本数を表す。

$$A_{wire_i} = l_{b_i} \times w_{pitch} \times (n_{addr_i} + n_{data_i}) \quad (8)$$

バス配線は、機能ブロックの上を通る場合（オーバー・ザ・セル）と、機能ブロックの外側を通る場合の二通りが考えられる。機能ブロックの上を通る場合には、バス配線はチップ面積には影響しない。そこでオーバー・ザ・セルの割合を考慮したバス面積の見積もりを考え、バス  $b_i \in B$  の配線面積  $A_{b_i}$  を式 (9) によって見積もる。ここで、 $A_{logic_i}$  はロジックの面積を表す。調停回路、マルチプレクサ等のロジックは、あらかじめ論理合成を行った値を利用する。また、パラメタ  $r_{wire}$  はバス配線が機能ブロックの外側を通る割合（オーバー・ザ・セルでない割合）を表し、設計者が与えるものとする。

$$A_{b_i} = r_{wire} \times A_{wire_i} + A_{logic_i} \quad (9)$$

ロジックの種類や数、バスの配線本数は、バスの種類によって異なる。バスの種類としては、AMBA AHB [11] などの単一バス、AMBA AXI [12] などのマルチレイヤバス、クロス

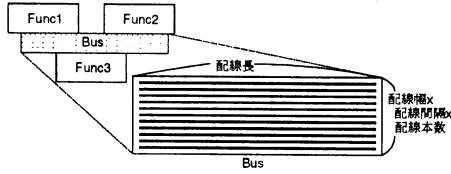


図5 配線面積モデル

バースイッチ [13], RF/Wireless [14] バス, NoC(Network-on-chip) [15], [16] などが存在する。以降に、オンチップバスとして代表的な単一バス, マルチレイヤバス, クロスバースイッチの3種類を見積もり対象とし、それぞれのバスのロジックの面積, バスの配線本数について述べる。

a) 単一バス

単一バスは、一本のバスを複数の機能ブロックが共有し、機能ブロック間の転送は調停回路によってスケジューリングされる。単一バスでは、調停回路がマスタからの要求を調停し、マルチプレクサによってスレーブのアドレス, マスタおよびスレーブに渡されるデータが選択される。したがって、バス  $b_i$  のマスタ数, スレーブ数, アドレスビット幅, データビット幅がそれぞれ,  $n_{mstr_i}$ ,  $n_{slv_i}$ ,  $w_{addr_i}$ ,  $w_{data_i}$  の場合, 下記の配線およびロジックが必要となる。

• 配線

- スレーブ用アドレス線 ( $w_{addr_i}$  ビット)
- マスタ書き込み用データ線 ( $w_{data_i}$  ビット)
- マスタ読み込み用データ線 ( $w_{data_i}$  ビット)

• ロジック

- 調停回路 ( $n_{mstr_i}$  入力)
- スレーブアドレス用マルチプレクサ ( $n_{mstr_i}$  入力  $w_{addr_i}$  ビット)
- マスタ書き込みデータ用マルチプレクサ ( $n_{mstr_i}$  入力  $w_{data_i}$  ビット)
- マスタ読み込みデータ用マルチプレクサ ( $n_{slv_i}$  入力  $w_{data_i}$  ビット)

したがって、バス  $b_i \in B$  のアドレス配線本数  $n_{addr_i}$  は  $w_{addr_i} \times n_{mstr_i}$ , データ配線本数は  $n_{data_i}$  は  $2 \times w_{data_i} \times n_{mstr_i}$  となる。また、ロジックの面積  $A_{logic_i}$  は式 (10) で見積もる。ここで、 $A_{arb}(m)$ ,  $A_{mux}(n, k)$  はそれぞれ、 $m$  入力調停回路の面積,  $n$  入力  $k$  ビットマルチプレクサの面積を表す。

$$A_{logic_i} = A_{arb}(n_{mstr_i}) + A_{mux}(n_{mstr_i}, w_{addr_i}) + A_{mux}(n_{mstr_i}, w_{data_i}) + A_{mux}(n_{slv_i}, w_{data_i}) \quad (10)$$

b) マルチレイヤバス

マルチレイヤバスは複数のデータバスとアドレスバスを持ち、マルチプレクサによってバスを切替えることにより、複数のマスタが同時に複数のスレーブに転送を行うことができる。マルチレイヤバスでは下記の配線およびロジックが必要となる。

• 配線

- スレーブ用アドレス線 ( $w_{addr_i}$  ビット,  $n_{mstr_i}$  本)

- マスタ書き込み用データ線 ( $w_{data_i}$  ビット,  $n_{mstr_i}$  本)
  - マスタ読み込み用データ線 ( $w_{data_i}$  ビット,  $n_{slv_i}$  本)
  - ロジック
    - スレーブアドレス用マルチプレクサ ( $n_{mstr_i}$  入力  $w_{addr_i}$  ビット,  $n_{slv_i}$  個)
    - マスタ書き込みデータ用マルチプレクサ ( $n_{mstr_i}$  入力  $w_{data_i}$  ビット,  $n_{slv_i}$  個)
    - マスタ読み込みデータ用マルチプレクサ ( $n_{slv_i}$  入力  $w_{data_i}$  ビット,  $n_{mstr_i}$  個)
- したがって、バス  $b_i \in B$  のアドレス配線本数  $n_{addr_i}$  は  $w_{addr_i} \times n_{mstr_i}$ , データ配線本数  $n_{data_i}$  は  $w_{data_i} \times (n_{mstr_i} + n_{slv_i})$  となる。また、ロジックの面積  $A_{logic_i}$  は式 (11) で見積もる。単一バスに比べ、マスタ数とスレーブ数が増大するにつれ、飛躍的に配線本数が増大する。

$$A_{logic_i} = A_{mux}(n_{mstr_i}, w_{addr_i}) \times n_{slv_i} + A_{mux}(n_{mstr_i}, w_{data_i}) \times n_{slv_i} + A_{mux}(n_{slv_i}, w_{data_i}) \times n_{mstr_i} \quad (11)$$

c) クロスバースイッチ

クロスバースイッチは、スイッチによってデータを制御し、同時に複数のデータ転送を行うことができる。クロスバースイッチでは下記の配線およびロジックが必要となる。

• 配線

- スレーブ用アドレス線 ( $w_{addr_i}$  ビット,  $n_{mstr_i}$  本)
- マスタ書き込み用データ線 ( $w_{data_i}$  ビット,  $n_{mstr_i}$  本)
- マスタ読み込み用データ線 ( $w_{data_i}$  ビット,  $n_{slv_i}$  本)

• ロジック

- スレーブアドレス用スイッチ ( $w_{addr_i} \times n_{mstr_i} \times n_{slv_i}$  個)
- マスタ書き込みデータ用スイッチ ( $w_{data_i} \times n_{mstr_i} \times n_{slv_i}$  個)
- マスタ読み込みデータ用スイッチ ( $w_{data_i} \times n_{mstr_i} \times n_{slv_i}$  個)

したがって、バス  $b_i \in B$  のアドレス配線本数  $n_{addr_i}$  は  $w_{addr_i} \times n_{mstr_i}$ , データ配線本数  $n_{data_i}$  は  $w_{data_i} \times (n_{mstr_i} + n_{slv_i})$  となる。また、ロジックの面積  $A_{logic_i}$  は式 (12) で見積もる。ここで、 $A_{switch}$  は、スイッチの面積を表す。

$$A_{logic_i} = n_{mstr} \times n_{slv} \times (w_{addr} + 2 \times w_{data}) \times A_{switch} \quad (12)$$

3.2.3 バス消費電力量の見積もり

バスの消費電力量  $W_{Bus}$  は、各バス  $b_i$  の消費電力量  $W_{b_i}$  の和として見積もる。本稿では、ダイナミック消費電力量を対象とする。各バスの消費電力量は、主に配線における消費電力量  $W_{wire_i}$  と、調停回路やマルチプレクサ等のロジックにおける消費電力量  $W_{logic_i}$  とに分けられる。配線における消費電力量は、図6に示すモデルを用いて見積もる。配線の消費電力は、配線容量, 電圧の二乗とスイッチング回数の積で求められる。そこで提案手法では、電圧  $V$ , 配線キャパシタンス  $C_w$ , その先

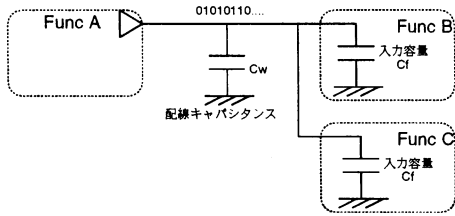


図 6 配線消費電力モデル

に接続される機能ブロックの入力容量  $C_f$ 、データ送信量  $M_{b_i}$ 、データのスイッチング確率  $R_{data}$  とアドレスのスイッチング確率  $R_{addr}$  から消費電力量を計算する。バス  $b_i \in B$  の配線における消費電力量  $W_{wire_i}$  は、式 (13) で見積もる。

$$W_{wire_i} = (C_w + C_f) \times V^2 \times M_{b_i} \times \frac{w_{addr_i}}{w_{data_i}} \times R_{addr} \times n_{addr_i} + (C_w + C_f) \times V^2 \times M_{b_i} \times R_{data} \times n_{data_i} \quad (13)$$

バス上で流れるデータ流量は、システムレベルにおけるプロファイリングと、アーキテクチャ情報によって取得する。

マルチプレクサや調停回路等のロジックにおいて消費される電力は、RTL の論理合成結果とスイッチング回数から計算する。提案手法においては、論理合成は Synopsys Design Compiler [17] を用いて行い、消費電力の測定は Synopsys Power Compiler [18] を使用した。各ロジックの電力測定値は、Power Compiler でクロックを 100MHz、遷移確率を 0.25、動作時間が 1 秒間である場合の消費電力測定値を基準としている。ロジックの消費電力量は、Power Compiler で測定された消費電力の基準値、見積もり時の対象アーキテクチャの周波数、およびデータ転送量とスイッチング確率の積で見積もる。

ロジックにおける消費電力量  $W_{logic_i}$  は、3.2.2 節で述べた単一バス、マルチレイバス、クロスバススイッチなど、バスの種類による配線本数、ロジックの種類と数を用いて、バスの種類に応じて消費電力の見積もり式を変更する。

## 4. 評価実験

### 4.1 バス消費電力見積もり精度の評価

マルチプロセッサシステムに対して、提案するバス消費電力見積もり手法の値と実際の値がどの程度ばらつくか調査を行った。実験では、まず 4 つのプロセッサコアとメモリ、それらを接続するデータバスを RTL で設計した。バスの構成として、単一バスが 1 本の場合と 2 本の場合をデータを用意した。そして、4 つのプロセッサそれぞれが、ロードストア命令を行うプログラムを実行し、データバスにアクセスする場合の消費電力を Synopsys Power Compiler を用いて測定し、見積もり式の結果と比較を行った。結果を表 1 に示す。プログラム  $S(x)L(y)$  は、ストア命令を  $x$  回、ロード命令を  $y$  回実行するプログラムを表す。また、1bus が単一バス 1 本、2bus がバス 2 本の構成を表す。実験結果より、バス電力の見積もり値と測定値の誤差は最大でも約 16% であった。Power Compiler を用いて消費電

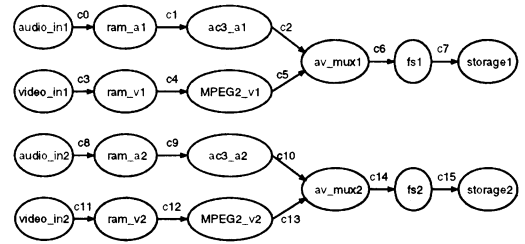


図 7 実験対象のシステムレベルモデル

力を計測する場合、対象のバス構成ごとに RTL 記述を用意して測定する必要がある。提案する見積もり手法を用いることで、RTL 設計前に誤差が 2 割以下で消費電力の見積もりを行うことが可能となることがわかった。

### 4.2 アーキテクチャ設計品質見積もり

次に、提案手法の有効性を示すため、音声・動画処理用 VLSI システムを想定し、機能ブロック構成やバス構成が異なる 32 種類のアーキテクチャに対して性能、面積、消費電力の見積もりを行った。実験対象のシステムレベルモデルを図 7 に、アーキテクチャレベルモデルを図 8 に示す。実験対象システムは、音声入力と動画の入力をそれぞれエンコードし、マルチプレクスしてストレージデバイスに記録するという処理を同時に行う。

図 9 に、見積もり結果から 32 種類のアーキテクチャの性能、面積、消費電力のトレードオフを表したグラフを示す。実験により、提案手法はさまざまなアーキテクチャにおける処理時間、面積、消費電力を見積もることが可能であることがわかった。したがって、提案手法を用いることにより、さまざまなアーキテクチャの設計品質の比較、最適なアーキテクチャの探索が容易となることがわかった。

## 5. 終わりに

本稿では、アーキテクチャレベルで面積、消費電力を短時間に見積もる手法を提案した。文献 [6] で提案された性能見積もりにおけるプロファイリング結果を利用した見積もり手法を提案した。評価実験では、音声・動画処理用 VLSI システムに提案手法を適用し、さまざまなアーキテクチャに対して性能、面積、消費電力を見積もり、値の比較を行った。実験結果より、設計早期段階において、アーキテクチャレベルでの設計品質見積もりを短時間に行うことにより、最適なアーキテクチャの探索が容

表 1 見積もり式と Power Compiler との比較

プログラム	アーキテクチャ	見積もり (mJ)	Power Compiler (mJ)	誤差 (%)
S(20)L(20)	1bus	$2.7131 \times 10^{-6}$	$2.42473 \times 10^{-6}$	11.9
	2bus	$1.2303 \times 10^{-6}$	$1.44372 \times 10^{-6}$	-14.8
S(40)L(40)	1bus	$5.4277 \times 10^{-6}$	$4.96766 \times 10^{-6}$	9.3
	2bus	$2.4606 \times 10^{-6}$	$2.80981 \times 10^{-6}$	-12.4
S(80)L(80)	1bus	$10.8555 \times 10^{-6}$	$9.34493 \times 10^{-6}$	16.2
	2bus	$4.9211 \times 10^{-6}$	$5.54058 \times 10^{-6}$	-11.2

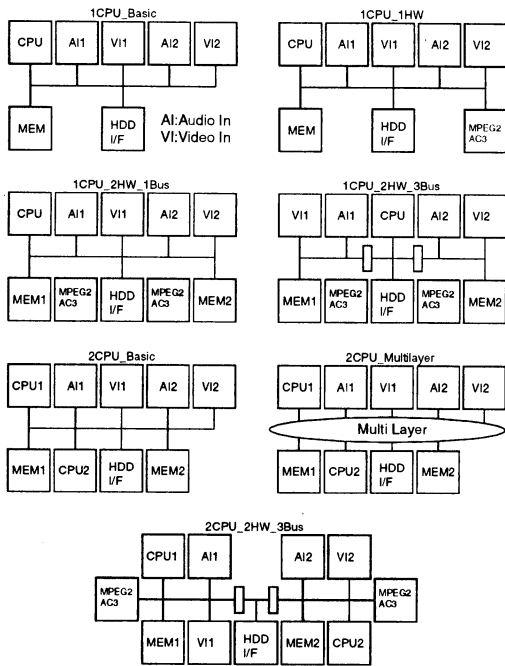


図 8 実験対象のアーキテクチャレベルモデル

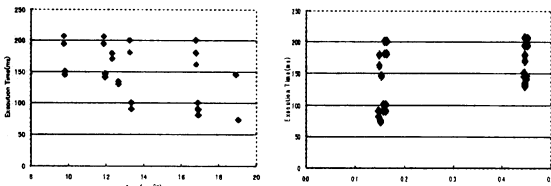


図 9 設計品質の比較

易となることを確認した。今後、様々なシステムに本手法を適用し、提案する手法の有効性についてさらに評価を行う。また、見積もり式のさらなる評価、アーキテクチャ最適化手法について研究を進める。

#### 謝辞

本研究を進めるにあたり貴重なコメントをいただいた、株式会社半導体理工学研究センターの宮本俊氏、株式会社富士通研究所の藤田隆司氏、ソニー株式会社の柿本勝氏、シャープ株式会社の大西充久氏、三洋電機株式会社の大山達史氏、鶴岡工業高等専門学校の佐藤淳助教授、大阪大学今井研究室の諸氏に感謝する。本研究の一部は、株式会社半導体理工学研究センターとの共同研究（研究番号 202）による。

#### 文 献

- [1] C.Norris Ip and Stuart Swan, "Using Transaction-Based Verification in SystemC," Cadence Design Systems, June 2002
- [2] Jon Connell and Bruce Johnson, "Early Hardware/Software Integration Using SystemC 2.0," Class 552, ESC San Francisco 2002
- [3] Sudeep Pasricha, Nikil Dutt and Mohamed Ben-Romdhane,

- "Extending The Transaction Level Modeling Approach for Fast Communication Architecture Exploration," Proceedings of DAC 2004, pp.113-118, 2004.
- [4] Sudeep Pasricha, Nikil Dutt and Mohamed Ben-Romdhane, "Fast Exploration of Bus-based On-chip Communication Architectures," Proceedings of CODES+ISSS2004, pp242-247.
- [5] "Functional Specification for SystemC 2.0," <http://www.systemc.org/>
- [6] Kyoko Ueda, Keishi Sakanushi, Yoshinori Takeuchi and Masaharu Imai, "Architecture-level Performance Estimation for IP-based Embedded Systems," Proceedings of DATE2004, pp.1002-1007, 2004.
- [7] Tony Givargis, Frank Vahid, and Jorg Henkel, "System-Level Exploration for Pareto-Optimal Configurations in Parameterized System-on-a-Chip," IEEE Trans. VLSI Systems, Vol. 10, No. 4, pp.416 - 422, August 2002.
- [8] Hiroshi Mizuno, Hiroyuki Kobayashi, Takao Onoye and Isao Shirakawa, "An Architecture Level Power Estimation Method for Embedded Systems," Proceedings of SASIMI 2001, pp.78-85, 2001.
- [9] B. S. Landman and R. L. Russo, "On a pin versus block relationship for partitions of logic graphs," IEEE Trans. Comput., Vol. C-20, pp.1469-1479, December 1971.
- [10] Payman Zarkesh-Ha, Jeffrey A. Davis, and James D. Meindl, "Prediction of Net-Length Distribution for Global Interconnects in a Heterogeneous System-on-a-Chip," IEEE Trans. VLSI Systems, Vol. 8, No. 6, December 2000.
- [11] "AMBA 仕様書 (Rev 2.0)," <http://www.arm.com>
- [12] "AMBA AXI プロトコル v1.0 仕様書," <http://www.arm.com>
- [13] J. Chang, S. Ravi, and A. Raghunathan, "FLEXBAR: a crossbar switching fabric with improved performance and utilization," IEEE CICC, pp.405-408, 2002.
- [14] Mau-Chung Frank Chang, Vwani P. Roychowdhury, Liyang Zhang, Hyunchol Shin, and Yongxi Qian, "RF/Wireless Interconnect for Inter- and Intra-Chip Communications," proc. IEEE Vol. 89, No. 4, pp.456-466, April 2001.
- [15] William J. Dally and Brian Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," Proceedings of DAC2002, pp. 524-529, June 2003.
- [16] J. Hu and R. Marculescu, "Exploiting the routing flexibility for energy/performance aware mapping of regular NoC Architectures," Proceedings of DATE2003, pp.688-693, March 2003.
- [17] "Design Compiler リファレンスマニュアル: Optimization and Timing Analysis 2000.05 日本語版," Synopsys Inc, 2000.
- [18] "Power Compiler リファレンスマニュアル 2000.05 日本語版," Synopsys Inc, 2000.