パイプライン・プロセッサのための アーキテクチャレベル面積見積り手法

塩見彰睦[†] 久須裕之[†,☆]
伊藤真紀子[††,☆☆] 今井正治[††]

本論文では，パイプライン・プロセッサの設計初期段階におけるアーキテクチャ探索のための面積見積り手法を提案する．見積りには，アーキテクチャ・レベルの設計情報であるパイプライン段数，命令数，リソース数，データ幅に基づく近似式を用いる．本手法を評価した結果，重み係数の推定に用いた32個の設計サンプルに対して，論理合成の面積値と比較して最大9.14%の誤差および94.75%の忠実度で見積りが行われた．さらに，MIPS R-3000をベースとした新規設計のサンプルプロセッサに対して，最大2.87%の誤差および100%の忠実度で見積りが行われた．

An Architecture Level Area Estimation Method for Pipelined Processors

Akichika Shiomi,[†] Hiroyuki Kusu,[†,☆] Makiko Itoh[††,☆☆] and Masaharu Imai[††]

This paper proposes an architecture level area estimation method for pipelined processor, that can be used for design space exploration in an early stage of processor design. The proposed method utilizes the number of pipeline stages, number of user defined resources, number of instructions, and data bit width. The experimental results using some derivative processors based on MIPS R3000 instruction set demonstrate the effectiveness of the proposed method. Area estimation error was up to 9.14% and fidelity of estimated value was at least 94.75% for 32 learning samples; and estimation error was up to 2.87% and fidelity was at least 100% respectively for 5 evaluation samples.

1. はじめに

近年における技術の進歩によりVLSIの集積度が向上し，大規模なASIP（Application Specific Instruction set Processor，特定用途向きプロセッサ）の実現が容易になってきた．しかし，プロセッサの規模の増加とともに，実現可能なアーキテクチャも多様化したため，より良いアーキテクチャを決定しようとすると，アーキテクチャを探索するために膨大な工数が必要となる．

適切なアーキテクチャを短期間で決定するには，プロセッサの設計工程の初期段階でアーキテクチャの設計品質を評価し，さまざまなアーキテクチャを比較する必要がある．アーキテクチャ候補を絞り込むためには，少なくとも10%程度の誤差の範囲で設計品質の見積り値が必要となる．このような精度の高い見積り値を得るためには，実装方法や機能のある程度詳細な設計を行う必要がある．しかし，設計の変更やアーキテクチャの比較には多くの工数を要する．

設計の上流段階におけるハードウェアコストの面積見積りの手法としては，協調設計の分野における機能分割時のコストの見積り手法[1)~5)]や既知のプロセッサに演算器を追加する際の見積り手法[6),7)]等が提案されている．協調設計の分野における機能分割時の見積り値は，機能分割の効率や性能向上のために用いられ，設計対象となるアーキテクチャの見積り値として十分

† 静岡大学情報学部情報科学科
Department of Computer Sciences, Faculty of Information, Shizuoka University
†† 大阪大学大学院基礎工学研究科情報数理系専攻
Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University
☆ 現在，株式会社日立情報ネットワーク
Presently with Hitachi Information Network, Ltd.
☆☆ 現在，株式会社半導体理工学研究センター
Presently with Semiconductor Technology Academic Research Center

な精度が得られないという問題点がある．既知のプロセッサに演算器を追加する際の見積り手法は，ベースとなる最小構成のプロセッサに演算器を追加するアプローチであるため，ベースとなるプロセッサに対する変更に対して柔軟に対応できないという問題点がある．

提案手法は，プロセッサの設計初期段階における設計項目をもとに精度，忠実度の良い面積見積りを目指す．設計初期段階の具体的な設計項目として，命令を実現する演算器やレジスタ等の主要なリソース，命令の個数，パイプライン段数とデータ幅を想定する．高位合成や機能分割等の設計や最小構成のプロセッサへの追加命令等設計手法における見積り手法に比べて本手法は見積りに要する設計情報の抽象度が高いといえる．

設計初期段階の抽象度の高い設計パラメータだけを用いることで，パイプライン段数の変更や命令の実現方法や追加・削除に対して迅速かつ柔軟に対応でき，精度と忠実度を向上させることで，設計の初期段階でのアーキテクチャの探索や候補の絞り込みに対して，短期間に有用な見積り値を設計者に提供できる．

本論文の構成は次のとおりである．2 章では，関連研究とその問題点について述べ，3 章では，本手法が想定するプロセッサ・モデルと見積りの近似式について説明する．4 章では近似式に用いた重み係数の推定方法について解説し，5 章で提案手法の評価実験の方法と結果および考察について述べる．

2. これまでの研究

設計の上流段階におけるハードウェアコストの面積見積りの手法としては，これまで協調設計の分野における機能分割時のコストの見積り手法や既知のプロセッサに演算器を追加する際の見積り手法等が提案されている．

機能分割では，制約条件とデータフローグラフ（DFG: Data Flow Graph）を入力とし，制約条件のもとで DFG のノードのスケジューリングが行われる．スケジューリングの段階で用いられる見積り手法は，実装に用いる機能ユニット（FU: Functional Unit），レジスタおよびマルチプレクサ等の単体の面積とそれらの数量の下界や上界を求め，機能分割の効率や性能を向上する用途に用いられる[1)~3)]．これらの手法の面積見積りはいずれも FU，レジスタ，マルチプレクサ単体の面積を基準にそれぞれの個数を乗じて面積の見積り値とするものである．Sharma ら[4)]は，DFG をバスを用いて実現するため，FU，レジスタ，マルチプレクサのほかにバスの本数と 3 ステート・ドライバの個数も考慮しているが，他の手法と同様の単体の面積と個数の積で全体の面積を見積もるアプローチである．これらの見積り値は探索木の枝刈りを行うために利用されるため，実装される設計対象は下界の値よりも大きくなることが多く，見積り精度の面では設計対象となるアーキテクチャの見積り値とはいいがたい．

Vahid ら[5)]のアプローチは，制御部とデータパス部からなるモデルから，それぞれを構成する FU，レジスタ，マルチプレクサの種類と個数と入出力や制御信号線の本数を決定し，面積を見積もる手法である．機能分割の際にデータパスが変化する場合には，変化前の設計とハードウェア・ユニットの面積の差分を用いて，変化後の設計の見積りを繰り返すことで，見積りの精度と速度を向上させている．この手法も，初回の見積りは，DFG より得られる FU，レジスタ，マルチプレクサ単体の面積を基準に，それぞれの個数より見積りを得ることになる．さらに，初回の見積り値に対して，繰返しにより誤差が蓄積されることから，見積り精度が悪くなると考えられる．

Binh らは，ASIP 開発システム PEAS-I での面積見積り手法[6)]を提案した．この手法も最小構成のプロセッサを用意し命令を追加するとともにプロセッサの周辺に演算器を追加する．この設計手法に用いる見積り手法は，最小構成のプロセッサの面積，演算器（FU）の種類，FU が実現できる命令，プロセッサ側のインタフェースや制御部を含めた FU の面積をモジュール・データベースに用意し，プロセッサ全体の面積は，最小構成のプロセッサと追加演算器の面積の和で求めている．Miyaoka らも，最小構成のプロセッサの面積と追加演算器の面積の和でプロセッサ全体の面積を見積もるアプローチを採用している[7)]．

最小構成のプロセッサと追加演算器の面積の和で見積もる手法は，精度の良い見積りが可能であるが，ベースとなる最小構成のプロセッサを用意しなければならない．したがって，最小構成のプロセッサのパイプライン段数を変更した場合は，設計を変更し，面積を計測し直す必要がある．さらに追加演算器の方も変更されたプロセッサに応じたインタフェースや制御部の面積の調整が必要となる．さまざまなアーキテクチャを比較するという目的を考慮すると，設計や計測のための工数が大きいという問題がある．

本論文は，プロセッサの設計初期段階における設計項目をもとに精度，忠実度の良い面積見積り手法を提案する．

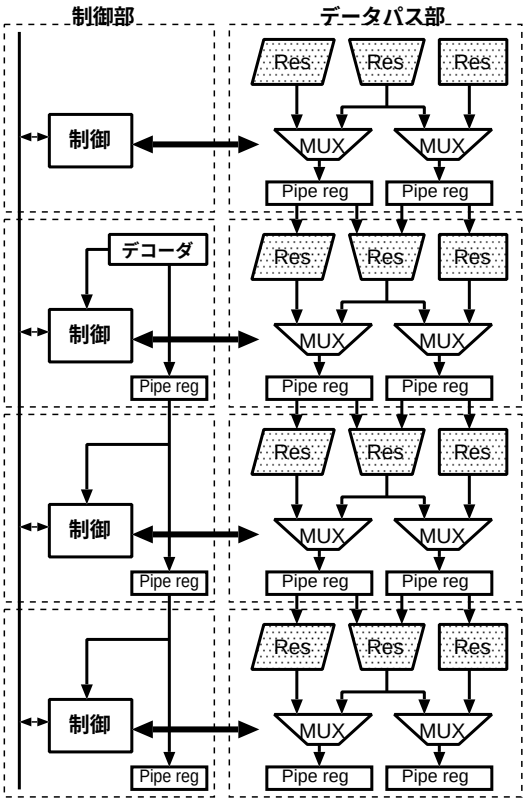

図 1 想定するプロセッサ・モデル
Fig. 1 The processor model to assume.

3. プロセッサ・モデルと見積り近似式

3.1 プロセッサ・モデル

提案手法で想定するパイプライン・モデルを**図 1** に示す．パイプライン・プロセッサは，ステージごとの制御部とデータパス部で構成される．制御部は，デコーダでデコードされた命令の種類に応じて，各ステージでリソースを制御する制御信号を出力する．データパス部は，演算器やレジスタ等のリソース “Res” とパイプライン・レジスタ “Pipe reg” がセレクタ “MUX” を介して接続される．このモデルは，ほとんどのパイプライン・プロセッサの設計に対して適用可能なモデルであると考えられる．

筆者らは，設計品質の向上と設計工数の削減を目的としたプロセッサ設計支援システム PEAS-III を提案してきた[8]．PEAS-III のプロセッサ生成も図 1 に基づいて生成されている[9]．

3.2 設計パラメータ

設計の初期では，設計者はプロセッサの大枠と命令を実現する主要なリソースを決定すると考えられる．プロセッサの大枠を決定するために，設計者はパイプライン段数，データビット幅，命令ビット幅，遅延分岐スロット数，ハードウェア・インタロックの有無等に関する設計情報を決定すると考えられる．

命令を実現する主要なリソースの決定では，演算器（ALU，加算器，乗除算器等）やレジスタ等を選択する．図 1 の網掛けで示される “Res” がこれに該当し，これらの設計者により選択されたリソースを,「明示されたリソース」と呼ぶ．通常，各リソースは既設計のコンポーネントを再利用することが行われる．また，新たに設計する場合でも概略の面積は決定されていると考えられる．したがって，本提案手法では明示されたリソースの部分は得られるものとする．

そこで，本手法では，設計情報の中で面積に大きく影響する設計情報に着目し，設計対象のプロセッサの面積見積りを行う．図 1 の網掛け以外の部分の詳細を決定するためには，さらに時間をかけて詳細な設計を行わなければならない．面積に大きく影響する設計情報は，以下の 4 種類と考えられる．パイプライン段数，宣言されたリソース数，データビット幅は，パイプライン・レジスタ数やセレクタ数等のデータパス部の規模と制御の複雑さに関係し，命令数は制御部やデコード部の規模に大きく関係していると考えられる．

本論文で使用する設計パラメータは以下の表記を用いる．

- パイプライン段数：N_{pipe}
- 宣言されたリソース数：N_{res}
- データのビット幅：W_{data}
- 命令数：N_{inst}

以降の節では，設計パラメータであるパイプライン段数，リソース数，データビット幅，命令数を用いた面積見積り近似式について説明する．なお，以降現れる α_i は設計パラメータの面積に対する影響の度合を表す重み定数である．

3.3 プロセッサ面積 $AREA$ の近似式

図 1 のプロセッサ・モデルより，プロセッサ全体の面積 $AREA$ は式 (1) に示すとおり，セレクタの面積 A_{sel}，パイプライン・レジスタの面積 A_{pipe}，制御部の面積 A_{cnt}，明示されたリソースの面積 A_{res} の和で求めることができる．

$$AREA = A_{sel} + A_{pipe} + A_{cnt} + A_{res} \quad (1)$$

ここで，明示された個々のリソースに関しては，その種類と面積は既知とする．したがって，明示されたリソースの和 A_{res} も既知である．それに対し，制御部の複雑さやパイプライン・レジスタおよびセレクタのビット幅等の諸元はアーキテクチャレベル見積りを行う段階では明確でない．本手法では，パイプライン段

数，明示されたリソース数，データ幅，命令数を用いて推定し，各部分項 A_{sel}，A_{pipe}，A_{cnt} の見積りを行う．

パイプラインの段数 N_{pipe} を固定して段数ごとに見積り式をたてて，それぞれの重み係数を推定する手法も考えられたが，本論文では，パイプライン段数 N_{pipe} も設計パラメータとして見積りに有効な重み係数を推定する方法を採用する．本手法の重み係数を推定する回帰分析では，一般に推定に用いるサンプル数が推定する係数の個数よりかなり大きいときでないと，有効な重み係数を求めることが難しい．パイプライン段数ごとに有効な重み係数を推定するためには，推定する段数に応じた多くの設計サンプルが必要になるため，本手法では，少ない設計サンプルから有効な重み係数を推定できるように，パイプライン段数も設計パラメータの 1 つとする．

本手法は，制御部，パイプライン・レジスタ，セレクタの面積を，各設計パラメータに関して線形に近似する．いくつかの設計データを解析した結果，各設計パラメータの値が二乗や累乗のオーダで大きく影響するのは，明示されたリソースの部分であり，制御部，パイプライン・レジスタ，セレクタの面積に対しては，二乗や累乗のオーダで大きく影響することはないことが分かった．また，デコード部等設計パラメータ n に対し $n\log n$ のオーダで影響する部分があるが，設計パラメータ n が 2 桁程度の場合，$n\log n$ の値は n で近似できると考えられる．以下，式 (1) の各項の面積に関する近似式について述べる．

3.4 セレクタの面積 A_{sel} の近似式

セレクタの面積 A_{sel} の見積り近似式を式 (2) に示す．

$$A_{sel} = W_{data} \times (\alpha_1 \times N_{inst} + \alpha_2 \times N_{res} + \alpha_3 \times N_{pipe} + \alpha_4) \quad (2)$$

セレクタは，信号衝突の解消のために生成系によって自動的に挿入される．セレクタの面積の合計は，データ幅，命令数，リソースの個数に比例すると考えられる．セレクタの面積 A_{sel} を見積もるうえで支配的になるセレクタは，主にデータパス中に挿入されるため，セレクタのデータ幅はプロセッサのデータ幅と等しいと考えられる．リソース数の増加や命令数の増加によってもデータパスが複雑化し，共有されるリソースが増えるため，セレクタ数が増加する．さらに，パイプライン段数の増加によっても，セレクタ数が増加すると考えられる．

3.5 パイプライン・レジスタの面積 A_{pipe} の近似式

今回見積りの対象とするプロセッサはパイプライン・アーキテクチャである．複数ステージにわたってデータを転送する場合，パイプライン・レジスタを介する必要がある．生成系によって挿入されるパイプライン・レジスタの面積 A_{pipe} を式 (3) に示す．

$$A_{pipe} = W_{data} \times (N_{pipe} - 1) \times (\alpha_5 \times N_{inst} + \alpha_6 \times N_{res} + \alpha_7) \quad (3)$$

式 (3) は，データパス中に挿入されるパイプライン・レジスタの面積が，プロセッサのデータ幅に比例し，パイプライン・レジスタ数は，パイプライン段数とデータパスの複雑さに依存することを表している．データパスの複雑さはセレクタと同様，N_{inst}，N_{res} に比例する．

3.6 制御部の面積 A_{cnt} の近似式

制御部は生成系により，各命令のマイクロ動作記述から自動生成される部分である．制御部の面積 A_{cnt} を式 (4) に示す．

$$A_{cnt} = \alpha_8 \times N_{inst} + \alpha_9 \times N_{res} + \alpha_{10} \times N_{pipe} + \alpha_{11} \times N_{inst} \times N_{pipe} + \alpha_{12} \times N_{res} \times N_{pipe} + \alpha_{13} \quad (4)$$

制御部は，それぞれのパイプライン・ステージにおいて，個々の命令に対してリソースを制御すると考えられる．したがって，式 (4) では，命令数 N_{inst}，リソース数 N_{res} に比例するデータパスの複雑さとパイプライン段数との両方に依存することを表している．

3.7 明示されたリソースの面積 A_{res} の近似式

明示されたリソース $Res_1, Res_2, \cdots$ の集合を Res とすると Res は次式で表される．

$$Res = \{Res_1, Res_2, \cdots, Res_i, \cdots, Res_n\} \quad (5)$$

また，$A(Res_i)$ はリソース Res_i の面積とする．明示されたリソースの面積 A_{res} は，各リソースに対する面積見積り値の総和によって求められる．

$$A_{res} = \sum_{Res_i \in Res} A(Res_i) \quad (6)$$

設計情報の値（たとえば，W_{data}）に対して，二乗や累乗のオーダで大きく影響するのは，明示されたリソースの面積となる．この面積を正確に見積もることで見積り精度の向上をはかることができる．

ここで筆者らは Res_i を正確に見積もるために，筆者らが提案した設計資産の再利用手法[10]のモデルである FHM（Flexible Hardware Model）を用いる．FHM はデータベース（以下，FHM-DB と呼ぶ）に登録されている．設計者は，FHM-DB から使用するモデルを選択し，ビット幅，ハードウェア・アルゴリズム等のパラメータの指定を行う．指定されたパラメータに応じた機能の記述や面積，遅延時間，消費電力の見積り値，VHDL 記述等を FHM-DB より得ることができる．FHM を用いることにより，正確なリソースの

表 1 重み係数の推定に用いたプロセッサの諸元
Table 1 Specification of processors used for estimation of parameters.

	設計パラメータ				実測値				
設計名	W_{pipe}	N_{inst}	N_{res}	W_{data}	A_{res}	A_{cnt}	A_{pipe}	A_{sel}	$AREA$
BASE3a	3	53	17	32	40961.24	1253.45	6252.09	2345.73	53323.10
BASE3c	3	53	18	32	41118.13	1262.05	6252.09	2345.73	53500.98
BASE4a	4	53	17	32	41287.83	1619.28	10743.79	2347.30	54823.76
BASE4b	4	53	17	32	41287.83	1729.25	11624.84	2347.30	56080.71
BASE4c	4	53	18	32	41444.71	1657.26	10743.79	2347.30	54941.80
BASE5a	5	53	17	32	41488.21	1762.37	17341.07	2369.45	56873.70
BASE5b	5	53	17	32	41488.21	1858.05	18222.13	2369.45	57728.75
BASE5c	5	53	18	32	41645.10	1747.01	16941.16	2369.45	56726.58
MA3a	3	53	17	32	54612.10	1363.41	6478.38	2252.41	71699.80
MA3c	3	53	18	32	54768.99	1408.66	6479.23	2252.41	72436.92
MA4a	4	53	17	32	54938.69	1837.05	11064.19	2381.09	74378.75
MA4b	4	53	17	32	54938.69	1828.29	11930.28	2381.09	75758.97
MA4c	4	53	18	32	55095.57	1796.57	11064.19	2381.09	74565.05
MA5a	5	53	18	32	55565.49	1959.04	16448.98	2403.24	77657.14
MA5b	5	53	18	32	55565.49	1964.35	17315.07	2403.24	78621.05
MA5c	5	53	19	32	55722.38	1866.68	16448.98	2403.24	77674.17
ML3a	3	52	15	32	47246.55	1603.88	5971.95	1803.45	62617.19
ML3c	3	52	16	32	47403.44	1401.73	5986.05	1803.45	63048.88
ML4a	4	52	15	32	47573.14	1818.69	10037.22	1932.13	64546.11
ML4b	4	52	15	32	47573.14	1849.16	11330.59	1932.13	65677.05
ML4c	4	52	16	32	47730.03	1743.98	10037.22	1932.13	64553.89
ML5a	5	52	15	32	47773.53	1776.81	15355.25	1954.27	65383.66
ML5b	5	52	15	32	47773.53	1903.28	16648.61	1954.27	66743.16
ML5c	5	52	16	32	56492.23	1797.67	15355.25	1954.27	65509.77
MAL3a	3	55	18	32	55038.52	1658.66	6904.80	2534.34	74132.08
MAL3c	3	55	19	32	55195.41	1683.75	6918.91	2534.34	73866.75
MAL4a	4	55	18	32	55365.11	1852.12	11490.61	2663.02	77081.62
MAL4b	4	55	18	32	55365.11	2030.44	12356.70	2663.02	77873.43
MAL4c	4	55	19	32	55521.99	1884.30	11490.61	2663.02	77086.38
MAL5a	5	55	19	32	55991.91	2012.36	16875.40	2685.17	80482.55
MAL5b	5	55	19	32	55991.91	2104.79	17741.49	2685.17	80575.08
MAL5c	5	55	20	32	56148.80	1936.61	16875.40	2685.17	80514.84

面積 $A(Res_i)$ を得ることができる．

4. 重み係数 α_i の推定

3 章では見積りの近似を行ったが，各見積り式中に定数として置いたすべての α_i の値を求める必要がある．この章では重み係数 α_i の推定方法と，推定に用いたプロセッサ設計および α_i の推定結果について述べる．

4.1 推 定 方 法

重み係数の推定方法は，設計パラメータに対する実設計サンプルの各部の論理合成結果の値を回帰分析（最小二乗近似）することで行う．

まず，32 種類のサンプルプロセッサの詳細設計を行って得られた RT レベルのプロセッサ VHDL 記述を個々のリソース，制御部，セレクタ部，パイプライン・レジスタに分割する．次に，それぞれを面積最小オプションで論理合成し，その結果を実測値とする．最後に，各部の実測値と設計情報の組に対して回帰分析により重み係数 α_i を推定した．本論文で扱う「実測値」は，VLSI テクノロジー社の VSC753d（CMOS 0.5 μm）ライブラリと，Synopsys 社の Design Compiler を用いて，面積最小オプションを付けて論理合成して得られた値である．

4.2 推定に用いたプロセッサ設計

回帰分析には，MIPS-R3000 のアーキテクチャを基本とした，32 種類の設計を用いた．これらの設計記述は，PEAS-III を用いて設計し，生成された RT レベルの VHDL 記述を用いた．その諸元を**表 1** に示す．

表 1 で示される 32 個の設計は，大きく 4 つのアーキテクチャに分類される．各アーキテクチャに対して，パイプライン段数，および一部命令の実現方法の違いによって，8 種類のバリエーションがある．

4 つのアーキテクチャは，基本アーキテクチャ（base）と拡張命令を含むアーキテクチャ（ma，ml，mal）に分類される．

base：MIPS-R3000 準拠の命令セットを持つプロ

セッサ

ma：base の設計に対し，乗除算命令を削除し，積和命令，モジュロ・アドレッシングを支援する命令を追加したプロセッサ

ml：base の設計に対し，乗除算命令を削除し，積和命令，ループ命令を追加したプロセッサ

mal：ma と ml の両方の命令を持つプロセッサ

パイプライン段数は，3 段，4 段，5 段となっており，設計名中の数字で表される．3 段の場合は，命令フェッチ，命令デコード，実行の 3 ステージで構成される．4 段の場合は，3 段の構成に対して，メモリ・アクセス，レジスタ書き込みを第 4 ステージ目に移動させた構成となっている．5 段の場合は，4 段の構成に対して，レジスタ書き込みを第 5 ステージ目に移動させた構成になっている．

各アーキテクチャ，パイプライン段数に対して，分岐命令の実現方法が 3 種類用意されている．命令の実現方法の違いは，設計名中の最後に付加されているアルファベット（a，b，c）で示される．実現方法 a は，ALU を用いて分岐条件を判定し，同ステージで分岐を行う．実現方法 b では，a の構成に対して，分岐ステージを次ステージに送ることで，動作周波数を向上させている．パイプライン段数が 3 段の設計では，分岐ステージを変更することができないため，b 系列のバリエーションは存在しない．さらに，実現方法 c は，実現方法 a に対して，比較用の演算器を ALU から専用の比較器に変更することで，動作周波数を向上させている．

4.3 推定結果

見積り式と表 1 の諸元を基に回帰分析により α_i を推定した結果，**表 2** に示す自由度調整済重相関係数 R，標準誤差 Se および重み係数 α_i が得られた．自由度調整済重相関係数 R は，1.0 に近いほど回帰式のあてはまりが良いことを示しており，見積り近似式が非常によくあてはまっていることを示している．標準誤差 Se は，見積り近似式による見積り値の標準偏差を示している．標準誤差は，それぞれの実測値と比較して非常に小さい．これらの結果より，実測値と見積り近似式の回帰直線のずれは，非常に小さいといえる．

表 2 では，α_4，α_9，α_{11}，α_{13} が負の値となっている．α_4 と α_{13} は見積り近似式の切片（オフセット分）となっているものと考えられる．α_9 に関しては，リソース数が増加する場合には，命令とリソースの対応が簡略化され制御部の面積が減少する方向にあると考えられる．また，α_{11} に関しては，命令数とパイプライン段数の積が増加する場合には，1 段・1 命令あたりの制御の複雑さが軽減され，制御部の面積に対して減少の方向に働くと考えられる．

表 2 重相関係数 R，標準偏差 Se，重み係数 α_i の値
Table 2 Value of adjusted multiple correlation R, standard error Se and parameter α_i.

A_{sel}		A_{pipe}		A_{cnt}	
R	0.93	R	0.94	R	0.86
Se	83.83	Se	1207.64	Se	99.54
α_1	4.36	α_5	0.69	α_8	332.95
α_2	2.58	α_6	2.55	α_9	−312.62
α_3	1.55	α_7	42.77	α_{10}	1769.70
α_4	−211.00			α_{11}	−47.98
				α_{12}	57.03
				α_{13}	−11400.26

5. 評価実験と考察

本手法を評価するために評価実験を行った．評価は，見積りの精度（相対誤差）と忠実度[11]について検証する．見積りの忠実度は，任意の異なるアーキテクチャ間で見積り値の大小関係と実測値の大小関係が保存されている割合で定義される．忠実度が高いほど，見積り手法は異なるアーキテクチャ間の差異を適切に反映しているといえる．アーキテクチャの探索を行ううえで大小関係が保存されていることは有効であり，適切なアーキテクチャの決定が可能になる．以下に見積りの忠実度を求める方法を示す．

忠実度 F は，アーキテクチャ D_i とアーキテクチャ D_j の間での大小関係が保存されているか否かを表す．ここで，$D = D_1, D_2, \ldots, D_n$ を異なる設計の組とし，$1 \leq i,\ j \leq n$ かつ $i \neq j$ を満たす i，j に対して，μ_{ij} は以下のように定義される．

$$\mu_{ij} = \begin{cases} 1 & \text{if } E(D_i) > E(D_j) \text{ and} \\ & \quad M(D_i) > M(D_j), \text{or} \\ & E(D_i) = E(D_j) \text{ and} \\ & \quad M(D_i) = M(D_j), \text{or} \\ & E(D_i) < E(D_j) \text{ and} \\ & \quad M(D_i) < M(D_j) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

上式において $E(D_n)$ は見積り値，$M(D_n)$ は実装後の測定値を示す．見積りの忠実度 F は式 (7) で定義された μ_{ij} を用いて以下の式で求められる．

$$F = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mu_{ij} \times 100 \ \ [\%] \quad (8)$$

5.1 評価実験 1

評価実験 1 は，重み係数を推定するために用いた

表 3 サンプル設計に対する面積見積り値と実測値の比較
Table 3 Comparison of estimated area and measured value.

設計名	実測値	見積り値	誤差 [%]
BASE3a	53323.10	52514.78	−1.52
BASE3c	53500.98	52775.57	−1.36
BASE4a	54823.76	57006.83	3.98
BASE4b	56080.71	57006.83	1.65
BASE4c	54941.80	57406.11	4.49
BASE5a	56873.70	61372.67	7.91
BASE5b	57728.75	61372.67	6.31
BASE5c	56726.58	61910.44	9.14
MA3a	71699.80	66165.64	−7.72
MA3c	72436.92	66426.43	−8.30
MA4a	74378.75	70657.69	−5.00
MA4b	75758.97	70657.69	−6.73
MA4c	74565.05	71056.97	−4.70
MA5a	77657.14	75830.83	−2.35
MA5b	78621.05	75830.83	−3.55
MA5c	77674.17	76368.60	−1.68
ML3a	62617.19	58219.75	−7.02
ML3c	63048.88	58480.54	−7.25
ML4a	64546.11	62460.80	−3.23
ML4b	65677.05	62460.80	−4.90
ML4c	64553.89	62860.08	−2.62
ML5a	65383.66	66575.65	1.82
ML5b	66743.16	66575.65	−0.25
ML5c	65509.77	67113.42	2.14
MAL3a	74132.08	67441.03	−9.03
MAL3c	73866.75	67701.82	−8.35
MAL4a	77081.62	72019.61	−6.57
MAL4b	77873.43	72019.61	−7.52
MAL4c	77086.38	72418.89	−6.05
MAL5a	80482.55	77279.29	−3.98
MAL5b	80575.08	77279.29	−4.09
MAL5c	80514.84	77817.05	−3.35

プロセッサ 32 種類に対する見積り値と実測値の比較，評価実験 2 では，新たに MIPS-R3000 ベースのプロセッサ 5 種類を設計し，見積り値と実測値を比較した．

それぞれ設計に対する見積り値と実測値の比較を**表 3** と**図 2** に示す．横軸の設計は左から，表 3 に掲載したアーキテクチャの順に並べている．

評価実験の結果，見積りの平均相対誤差は 4.83%，最大の相対誤差は 9.14%であった．また，見積りの忠実度 F は 94.75%であり，相対的なアーキテクチャの比較に十分に利用できる忠実度であると考えられる．

5.2 評価実験 2

評価実験 2 は重み係数 α_i を求めた設計以外のアーキテクチャに関する見積り値を評価することが目的である．評価実験 2 では，PEAS-III を用いて，5 種類のプロセッサを設計し，見積り値と実測値を比較した．ここで用いた設計記述も，評価実験 1 と同様に，PEAS-III を用いて設計し，生成された RT レベルの VHDL 記述を用いている．設計したプロセッサの諸元を以下に示す．

設計 1：MIPS-R3000 に準拠の命令セットを持つ 5 段パイプライン・プロセッサからフォワーディング機能と特殊レジスタアクセス（`MFCO`）命令を削除したプロセッサ

設計 2：設計 1 に対し，積和命令を追加した 4 段パイプライン・プロセッサ

設計 3：設計 1 に対し，乗除算命令を削除し，積和命令を追加した 3 段パイプライン・プロセッサ

設計 4：設計 1 に対し，ループ命令を追加した 5 段パイプライン・プロセッサ

設計 5：設計 1 に対し，積和命令，ループ命令を追加した 4 段パイプライン・プロセッサ

評価実験に用いたプロセッサでは，リソースの個数 N_{res} に関して，すべての設計で推定に用いたパラメータよりも小さく設定し，命令の個数 N_{inst} に関しては，設計 3 では 49 個，設計 5 では 59 個とパラメータの範囲外に設定してある．

5 種類の設計に対する見積りの値と論理合成で得られた実測値の比較を**表 4** と**図 3** に示す．評価実験の結果，5 種類の設計に対する見積り値の平均相対誤差は 1.74%，最大の相対誤差は 2.87%であった．

一般に，回帰式による近似式は，推定に用いたパラメータ値範囲内で有効な推定値が得られるものである．本見積り式では，設計パラメータの範囲外でも設計パラメータの近傍であれば，精度の良い見積りが得られることが分かった．

また，面積見積りの忠実度 F は 100%であった．これは，全組合せ数 10 通りすべての大小関係が保存された見積りになっていることを表す．評価実験 2 における各設計は，パイプライン段数や規模の大きなリソース（積和演算器）の有無という違いがある．このようなアーキテクチャの違いは，プロセッサの面積に対して支配的に働くと考えられる．提案手法では，設計パラメータを通してアーキテクチャの違いを見積り値に正確に反映させることができた．

5.3 考　　察

本節では提案手法によって得られた見積り値の忠実度について考察し，アーキテクチャ探索の指標として利用可能なことを示す．そのうえで提案手法の限界と今後の課題について明らかにする．

まず，評価実験 1 について，見積り値と実測値の大小関係が保存されない原因を明らかにする．大小関係が保存されていないのは設計 a と b の間である．設計 a と b の違いは分岐命令の動作である．本提案手法で用いているアーキテクチャ・レベルの設計情報が

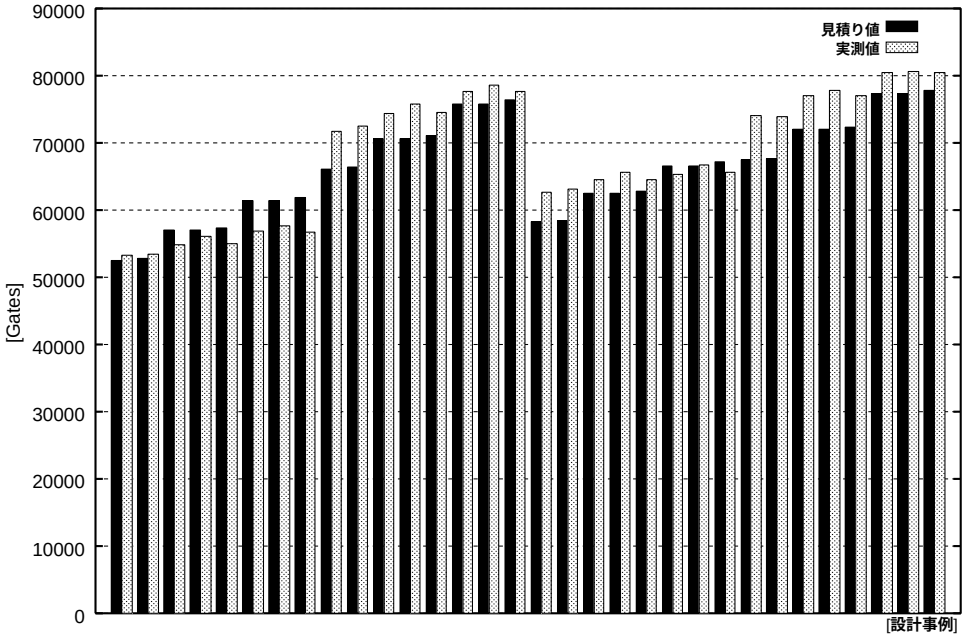

図 2 サンプル設計に対する面積見積り値と実測値の比較
Fig. 2 Comparison of estimated area and measured value.

表 4 評価サンプルに対する面積見積り値と実測値の比較
Table 4 Comparison of estimated area and measured value of evaluation samples.

	設計パラメータ					結果		
モデル	N_{pipe}	N_{inst}	N_{res}	W_{data}	A_{res}	実測値	見積り値	誤差 [%]
設計 1	5	52	11	32	39342.69	55744.05	56621.30	1.57
設計 2	4	57	12	32	54495.14	72470.61	70388.39	−2.87
設計 3	3	49	10	32	48054.42	58034.46	57390.48	−1.11
設計 4	5	54	11	32	39342.69	57939.35	57262.45	−1.17
設計 5	4	59	12	32	54495.14	72524.87	71081.50	−1.99

まったく等しいため，得られる見積り結果も等しくなる．しかし，実際には命令の実現方法により面積は異なるため，大小関係が保存されない．提案手法は見積りの粒度が粗いために，このような実現方法といった詳細設計段階における微妙な違いは反映できないといえる．

しかし，今回の評価実験の結果を見ると見積り値の大小関係が大きく逆転しているわけではない．見積り値の差の小さいアーキテクチャであれば，論理合成やレイアウト等の実現方法によって大小関係が逆転する可能性があるアーキテクチャといえる．したがって，本手法はアーキテクチャの評価としては，十分に利用可能な見積り方法と考えられる．

次に，本手法の限界について言及する．本見積り手法の誤差が大きくなる要因は以下の 3 種類の要因が考えられる．

(1) 見積り式の近似誤差

見積り式は回帰式による近似式であるために，どうしても誤差は避けられない．推定した設計パラメータの値の範囲の近傍の設計であれば良い見積り値が得られるものの，推定した設計パラメータの値の範囲から大きくはずれる設計に対しては誤差は大きくなると考えられる．見積り精度を追求するためには，設計パラメータの範囲をカバーする設計サンプルを用いて重み係数 α_i の値を推定しなおさなければならない．

しかし，広範囲な設計パラメータに対して，本手法で採用した $n \log n$ のような曲線を n で近似した回帰式を用いると，設計パラメータの範囲内でも誤差が大きくなる可能性がある．高い精度と広範囲なパラメータに対応するには，設計パラメータの範囲を複数の区間に分け，区間ごとに近似式を求めることで解決できると考えられる．高い精度と広範囲なパラメータに対応する近似式については今後の課題である．

(2) 論理合成の最適化の効果

本見積り手法では，論理合成ツール Synopsys 社の Design Compiler の面積最小オプションでの見積りは

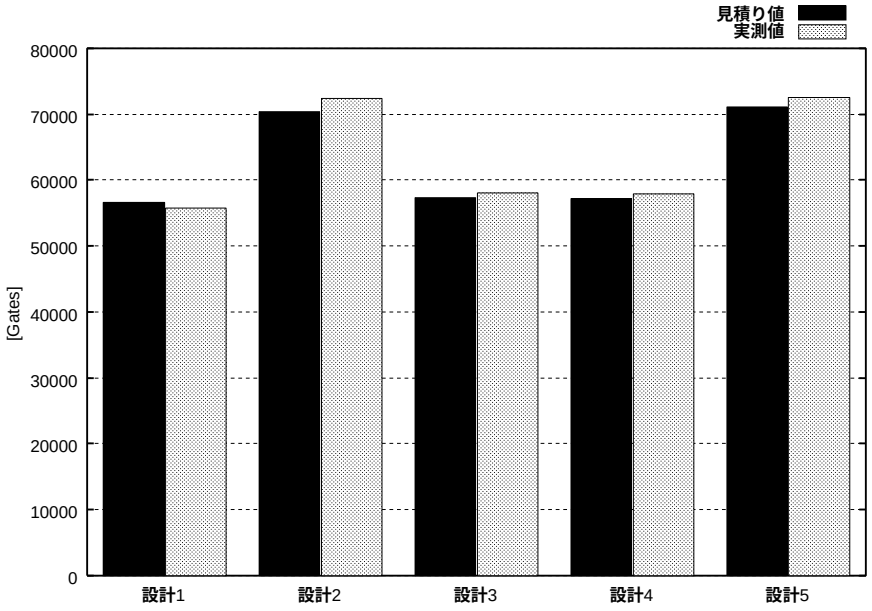

図 3 評価サンプルに対する面積見積り値と実測値の比較
Fig. 3 Comparison of estimated area and measured value of evaluation samples.

比較的良い結果が得られている．しかし，合成時のオプションを遅延時間最小のオプションに変更された場合に，本見積り式をそのまま適用すると小さく見積もることになる．

遅延時間最小オプションで合成される設計の場合には，A_{sel}，A_{pipe}，A_{cnt}，A_{res} それぞれの見積り値に対して最適化の効果を考慮した重み係数を導入し，本手法と同じ手法で推定できると思われる．

また，論理合成ツールが代わる場合にも，誤差は大きくなる．論理合成ツールが代わった場合は，論理合成ツールの特性等により，最適化の効果が変化するために，合成オプションの変更のように重み係数を導入するだけでは精度の向上は難しいと思われる．

論理合成時の制約条件や論理合成ツールの性能の影響を切り分ける見積り近似式は今後の課題である．

(3) ライブラリや設計ルールの変更

今回の評価実験で使用した VSC753d（CMOS 0.5 μm）ライブラリとほぼ比例関係にあるライブラリや設計ルールの変更に関しては，見積り値 $AREA$ に比例定数を乗じることで解決できると考えられる．しかし，精度を追求するためには，重み係数 α_i の値を設計サンプルを用いて推定しなおさなければならない．ライブラリや設計ルール変更に対する校正方法は今後の課題である．

6. おわりに

本論文では，抽象度の高いレベルでのプロセッサ・アーキテクチャの設計情報を用いた面積見積り手法を提案し，MIPS-R3000 を基本としたサンプルに対して精度と忠実度を評価した．重み係数を推定したアーキテクチャに関して，忠実度が 94.75%，平均相対誤差が 4.83%，最大相対誤差は 9.14%という結果が得られた．

さらに，重み係数を推定したアーキテクチャと異なる設計 5 種類に関しては，忠実度が 100%，平均相対誤差が 1.74%，最大相対誤差は 2.87%となり，実用的な見積り値が得られたといえる．本手法の見積り値は設計初期段階におけるアーキテクチャの探索に十分利用できると思われる．

多種多様なアーキテクチャによる見積り結果の評価，論理合成の最適化による差を補正する方法，ライブラリや設計ルール変更に対する校正方法は今後の課題である．

謝辞 本研究を行うにあたり討論していただいた，大阪大学武内良典助教授，豊田工業高等専門学校木村勉助手ならびに大阪大学今井研究室，静岡大学塩見研究室の諸氏に感謝いたします．なお，本研究の一部は（株）半導体理工学研究センターとの共同研究による．

参 考 文 献

1) Ohm, S.Y., Kurdahi, F. and Dutt, N.: Comprehensive Lower Bound Estimation from Behavioral Descriptions, *IEEE/ACM Proc. International Conference on CAD* (*ICCAD*) 1994, pp.182–186 (1994).
2) Jain, R., Parker, A.C. and Park, N.: Predicting System-Level Area and Delay for Pipelined and Nonpipelined Design, *IEEE Trans. CAD*, Vol.11, No.8, pp.955–965 (1992).
3) Rabaey, J.M. and Potkonjak, M.: Estimating Implementation Bounds for Real Time DSP Application Specific Circuits, *IEEE Trans. CAD*, Vol.13, No.6, pp.669–683 (1994).
4) Sharma, A. and Jain, R.: Estimating Architectural Resources and Performance for High Level Synthesis Applications, *IEEE Trans. VLSI Systems*, Vol.1, No.2, pp.175–190 (1993).
5) Vahid, F. and Gajski, D.D.: Incremental Hardware Estimation During Hardware/Software Functional Partitioning, *IEEE Trans. VLSI Systems*, Vol.3, No.3, pp.459–464 (1995).
6) Binh, N.N., Imai, M., Shiomi, A. and Hikichi, N.: An Instruction Set Optimization Algorithm for Pipelined ASIPs, 電子情報通信学会論文誌 (英文誌 (A) VLSI と CAD アルゴリズム小特集), Vol.E78-A, No.12, pp.1707–1714 (1995).
7) Miyaoka, M., Kataoka, Y., Togawa, N., Yanagisawa, M. and Ohtsuki, T.: Area / Delay Estimation for Digital Signal Processor Cores, *ASP-DAC 2001*, pp.156–161 (2001).
8) 塩見彰睦，今井正治，片岡健二，青山義弘，佐藤 淳，引地信之：ASIP 設計用コデザインワークベンチ PEAS-III の提案，情報処理学会設計自動化，pp.73–80 (1995).
9) 伊藤真紀子，檜垣茂明，塩見彰睦，佐藤 淳，武内良典，今井正治：パイプライン・ハザードを考慮したプロセッサ生成手法の提案，情報処理学会論文誌，Vol.41, No.4, pp.851–862 (2000).
10) 森藤孝文，繁原英一郎，武内良典，今井正治：FHM データベースと FHM を用いた設計例，第 2 回システム LSI 琵琶湖ワークショップポスター資料集，pp.329–330 (1998).
11) Gajski, D.D., Vahid, F., Narayan, S. and Gong, J.: *Specification and Design of Embedded Systems*, PTR Prentice Hall, Inc. (1994).

(平成 13 年 9 月 14 日受付)
(平成 14 年 3 月 14 日採録)

塩見 彰睦（正会員）

昭和 62 年豊橋技術科学大学情報工学課程卒業．平成 4 年同大学院博士課程修了．博士（工学）．平成 8 年より静岡大学情報学部情報科学科講師．平成 12 年同助教授．設計自動化，ハードウェア/ソフトウェア協調設計手法，最適化設計支援に関する研究に従事．

久須 裕之

平成 11 年静岡大学工学部知能情報工学科卒業．平成 13 年同大学院修士課程計算機工学専攻修了．同年株式会社日立情報ネットワーク勤務．現在ヘルプデスク，財務分析等，企業向け業務支援システムの開発に従事．

伊藤真紀子（正会員）

平成 13 年大阪大学大学院基礎工学研究科博士後期課程修了．博士（工学）．平成 13 年より株式会社半導体理工学研究センターで，特定用途向きプロセッサのためのコンパイラ生成に関する研究に従事．電子情報通信学会会員．

今井 正治（正会員）

昭和 49 年名古屋大学工学部電気工学科卒業．昭和 54 年同大学院博士後期課程修了（工学博士）．同年豊橋技術科学大学奉職．平成 6 年同教授．平成 8 年大阪大学大学院基礎工学研究科情報数理系専攻教授．その間，昭和 59 年から昭和 60 年にかけて米国サウスカロライナ大学工学部電気計算機工学科客員助教授（文部省在外研究員）．これまで組合せ最適化アルゴリズム，ハードウェア/ソフトウェア協調設計等の研究に従事．平成 3 年より日本電子機械工業会および IEEE/DASC において EDA 標準化作業に従事．IEEE，ACM，電子情報通信学会，人工知能学会各会員．