

命令の類似度を用いた命令コードの自動割り当て*

5 Z B - 0 3

中田 晃一 塩見 彰睦†
静岡大学大学院情報学研究所‡

1 はじめに

今日の VLSI の集積度の向上に伴い、大規模な ASIC の実装が容易になってきた。しかし、プロセッサの規模の増加と共に、アーキテクチャは多様化してきている。そのため、より良いアーキテクチャを探索するためには、膨大な工数が必要となる。そこで、より良いアーキテクチャを少ない工数で探索でき、短期間で設計するための手法が強く望まれている。

我々の研究グループでは、設計品質の向上と設計工数削減を図るために、プロセッサ設計支援システム PEAS-III を提案している [1]。PEAS-III システムを利用したアーキテクチャ設計において、命令形式の定義及び命令コードの割り当ての設計工程が設計全体に要する工数の約 3 割を占めている。その理由として、命令形式の定義は、データ形式、アドレッシング法など計算機システムの性能に関わるために重要であり、より良いアーキテクチャを探索するために、命令形式の変更及び命令の追加・削除が頻繁に起きることが考えられる。また、命令コードを従来のように設計者が決定する方法では、命令形式が変更した際に設計者の負担が大きくなる。

本研究では、より設計工数の削減を行なうために、設計者が指定した命令コードフィールドに命令コードをシステムで割り当てる方法を提案する。また命令コードの割り当て方によって命令デコーダの回路面積が変化することから、デコードの容易化を意識した命令コードを割り当てる。

2 命令コードの割り当て方針

デコードの容易化を意識した命令コードの割り当て手法は、文献 [2] で提案されている。文献 [2] の手法では、命令機能や処理対象別など、命令を特徴づける関係をもとに命令のグループ分けを行ない命令コードを割り当て、デコーダの面積縮小を実現している。しかしこの方法では、分けるグループ数がはじめから決まっているため、あるグループに命令が偏った場合、設計者が指定した命令コードフィールドに命令コードを割り当てることが難しい。本研究では設計者が指定した命令コードフィールドに、命令コードが収まるよう命令の動作からグループ分けを行ない命令コードを割り当てる。

本研究では、PEAS-III システムと同じ固定長命令で、命令形式毎に異なる命令コードフィールドを持つパイプ

ラインアーキテクチャを対象とする。そのため、命令コードフィールドは、全命令共通の共通命令コードフィールドと、命令形式固有の非共通命令コードフィールドから構成される。

以上のことから、各非共通命令コードフィールドで区別する命令、共通命令コードフィールドのみで区別する命令に分け、命令のグループ化を行ない、グループ毎に命令コードを割り当てる。共通命令コードフィールドのグループ化では、非共通命令コードフィールドを使用する命令形式を区別するために、非共通命令コードフィールドのグループ化の結果をもとにグループ化を行なう。

共通及び非共通命令コードフィールドの情報は、PEAS-III システムの命令形式の情報から求める。また各命令のパイプラインステージ毎の動作を記述する PEAS-III システムのマイクロ動作記述設計工程の情報から、各命令間の動作の類似度を算出し、クラスタ分析を行なうことで、命令のグループ化を実現する。

3 命令の類似度の算出方法

各命令間の類似度を求めるために、PEAS-III システムのマイクロ動作記述設計工程の情報を利用する。マイクロ動作記述から、各命令のパイプラインステージで利用するリソースや命令動作の情報を得ることができる。マイクロ動作記述から、命令間の類似度を算出する流れを以下に示す。

1. マイクロ動作記述から、命令が利用するリソースとファンクションの組合せをパイプラインステージ毎に求める。
2. 各命令の求めたパイプラインステージ毎のリソースとファンクションの組合せを総当たりで比較し、類似度を求める。

命令 x と命令 y の類似度を求める場合、まず各パイプラインステージ毎に各命令間の類似度を求める。 i 番目のパイプラインステージにおいて、命令 x, y のリソースとファンクションの組合せを総当たりで比較し、リソース及びファンクション共に同じである数 $Nr f_i^{x,y}$ 、リソースのみ同じである数 $Nr_i^{x,y}$ を求める。そして、命令 x の i 番目のパイプラインステージのリソースとファンクションの組合せの数 N_i^x 、命令 y の i 番目のパイプラインステージのリソースとファンクションの組合せの数 N_i^y とするとき、命令 x と命令 y の i 番目のパイプラインステージの類似度 $C_i^{x,y}$ は式 (1) によって求める。

$$C_i^{x,y} = \frac{Nr f_i^{x,y} + 0.5 \times Nr_i^{x,y}}{N_i^x + N_i^y - Nr f_i^{x,y} - Nr_i^{x,y}} \times 100 \quad (1)$$

* An automatic assignment of instruction codes using closeness of instruction

† Koichi Nakata, Akichika Shiomi

‡ Shizuoka University Graduate School of Information

そして、命令 x と命令 y の類似度 $C_{x,y}$ は式 (2) で求める。

$$C_{x,y} = \frac{1}{n} \left(\sum_{i=1}^n C_i^{x,y} \right) \quad (2)$$

ここで n はパイプラインステージ数を表している。

4 命令コードの割り当てアルゴリズム

各命令間の類似度を利用し、階層的クラスタ分析によって命令のグループ化を行ない命令コードを割り当てる。クラスタ分析を行なう際の距離の再定義には最短距離法 [3] を利用する。また、命令形式から各非共通命令コードフィールドで区別する命令、共通命令コードフィールドのみで区別する命令に分け、命令コード割り当てアルゴリズムを実行する。共通命令コードフィールドの命令コード割り当てを行なう場合は、非共通命令コードフィールドを持つ命令形式を区別するために、非共通命令コードフィールドの数だけクラスタを用意する。そして、非共通命令コードフィールドのクラスタ併合後の距離を利用してアルゴリズムを実行する。

命令コードのビット幅を W_{inst} 、グループを識別するビット幅を W_g 、グループを識別するビット幅以外のビット幅を $W_e = W_{inst} - W_g$ とする。このときグループ数 N_c は 2^{W_g} とする。命令コードの割り当てアルゴリズムを以下に示す。

- (1) グループを識別するビット幅を $W_g = 1$ とする。
- (2) 1つの命令を1つのクラスタと考え、類似度からクラスタ数 N_c になるまでクラスタ併合を行なう。
- (3) クラスタ併合の結果を命令コードに反映させることが可能かを調べる。
 - (3-1) 併合したクラスタに所属するクラスタ数が 2^{W_e} を越えている場合は、越えたクラスタを所属するクラスタ数が 2^{W_e} 以下になるまで分割する。分割後のクラスタの中で所属するクラスタ数が一番多いクラスタを1つのグループとして決定する。残りのクラスタでクラスタ数 $N_c - 1$ とし (2) を行なう。
 - (3-2) グループを識別するビット幅 W_g では明らかにクラスタの併合が行なえない場合は、 $W_g = W_g + 1$ とし (2) を行なう。
- (4) 各クラスタを区別できるよう命令コードにビット幅 W_g 分命令コードを割り当てる。
- (5) 残りのビット領域 W_e が0になるまで、各クラスタに所属する命令毎に $W_{inst} = W_e, W_g = 1$ とし (2) を行なう。

5 実験と考察

実験として MIPSR3000 互換のアーキテクチャをもとに、本手法を用いて命令コードを割り当てた場合、MIPSR3000 の命令コードの場合、そしてランダムに命令コードを割り当てた場合の制御部の VHDL 記述を用意し、論理合成を行ない面積を比較する。合成ツ

ルは Synopsys 社の Design Compiler を利用し、vsc754d ライブラリを使用して論理合成を行なう。また、本手法を利用して命令コード割り当てた場合に要した時間は、Pentium III 866MHz で 0.13 秒であった。合成結果は表 1 に示す。

表 1: 制御回路の論理合成結果 (gate)

	本手法	オリジナル	ランダム
Combinational area	714.75	808.75	908.50
Nocombinational area	483.25	483.25	483.25
Net Interconnect	571.17	637.76	703.37
Total area	1769.17	1929.76	2095.12

表 1 よりオリジナル及びランダムの結果と比べ、本手法を用いることで制御回路を構成する組合せ回路及び配線が小さくなるのが分かる。この理由として、命令動作の近い命令毎にグループ化を行ない命令コードを割り当てることにより、各コンポーネントへの制御信号を少ないゲート数で実現できた結果だと考えられる。

6 おわりに

今回提案した手法を用いることで、制御回路を構成する組合せ回路及び配線が小さくなることを示した。またこの命令コード自動割り当てシステムを利用することで、設計者が命令コードを割り当てる必要がなく、PEAS-III システムのアーキテクチャ設計において設計工数を削減することができる。

現在グループ数を変化させ、探索を行なうことで命令のグループ化を実現しているため、最適なグループ分けが行なえているとは言えない。今後はよりデコードの容易化を行なうために、グループ数の決定手法を確立する必要があると思われる。また、命令コードフィールドを単純に共通及び非共通命令コードフィールドに分けているため複雑な命令コードフィールドに対応できない。そのため、複雑な命令コードフィールドに対応することも今後の課題である。

参考文献

- [1] 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, 情報処理学会, 1995.
- [2] 高尻寛之, 吉田たけお, “命令デコーダを縮小する命令コード割り当て方法” 1999 年度電子情報通信学会情報・システムソサイエティ大会講演論文集 D-6-1, p30, 1999.
- [3] 続多変量解析入門, 日科技連出版社, 奥野 忠一, 芳賀 敏郎, 矢島 敬二, 奥野千恵子, 橋本 茂司, 古河陽子, 1991