

マイクロプロセッサ設計のための命令ダイアグラムツールの初期開発 Early Development of Instruction Diagram Tool for Microprocessor design

岩本 稜平[†]
Ryohei Iwamoto

森本 智之[‡]
Tomoyuki Morimoto

堤 利幸[†]
Toshiyuki Tsutsumi

1. 概要

我々は、命令形式の構成に注目し、マイクロプロセッサの各命令の動作とハードウェア部品の接続を表すブロック図を生成するツールを開発した。一つの命令について、意味を中心とした表現と機能を中心とした表現に分けることによって、各命令が持つ意味や動作を考慮することができる。さらに、処理ステージの順番の変更によって、命令の実行に与える影響を調べることができる。

2. 背景

マイクロプロセッサを設計するためには、論理設計、電子回路、と言った基本的な知識や、コンピュータアーキテクチャ、ハードウェア記述言語、CAD ツールの使い方を習得している必要がある。これらを学習するツールとして、様々な設計教育システムが開発されているが、これまでの設計システムは論理設計を中心としたものでしかなかった[1][2]。一方、アーキテクチャレベルで仕様を探索するツールや、動作合成等の高位の設計ツールも出てきている。しかしながら、技術者自身が仕様を深く考えなくても設計できるので、マイクロプロセッサの仕様を設計する力が身につかないという問題があった。

3. 研究目的

マイクロプロセッサを自力で設計することのできるツールの開発を目指している[3][4]。開発している命令ダイアグラムは、ユーザが実現したい命令をどのように設計すれば良いのかを可視化することができるツールである。

4. 命令ダイアグラムツール

4.1 意味分割命令と機能分割命令

我々のシステムでは命令を設計する際に、意味を中心とした側面と機能を中心とした側面の両面を考慮することによって、各命令が持つ意味と動作について理解する事ができる。前者を意味分割命令、後者を機能分割命令と呼ぶ。意味分割命令は、処理内容を表すオペコードと、処理対象を表すオペランドから成る。機能分割命令は、機能毎に分割された処理ステージで構成される。命令ダイアグラムツールで、意味分割における演算の種類や対象データの操作方法と、機能分割における演算や対象データの操作方法とハードウェア部品との対応を明確にすることができる。

4.2 ツールの概観

図 1 が命令ダイアグラムツールである。図 1 の上部(図 2)から命令を機能分割から設計することができ、図 1 の左部(図 3)から命令を意味分割から設計することができる。意味分割、機能分割のどちらかから命令を設計すると、図 1 のように各命令の動作とハードウェア部品の接続を表すブロック図が生成される。これによって、より直感的にどのようにしたら命令を設計することができるのかを把握することができる。

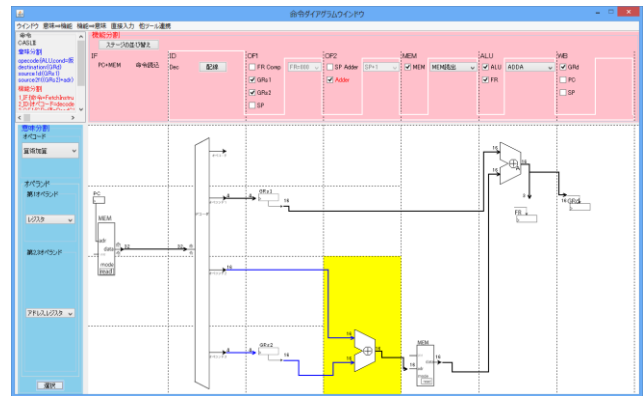


図 1 命令ダイアグラムツール

4.3 機能分割からの命令設計

機能分割から命令を設計するには、ステージの順番を決定し、図 2 の機能分割から使用するハードウェア部品を選択し、部品の接続を行なうことによって、命令を設計することができる。

機能分割は 7 つの処理ステージの機能の分割されており、それぞれのステージは、命令を読み込む IF(Instruction Fetch)ステージ、命令を解読する ID(Instruction Decode)ステージ、オペランドの読み込みを行う OF(Operand Fetch)1 ステージ、OF2 ステージ、メモリアクセスを行う MEM(MEMory access)ステージ、各種演算を行う ALU ステージ、結果を書き戻す WB(Write Back)ステージである。

OF1 ステージから WB ステージでそれぞれ使用するハードウェア部品を選択するようになっており、OF1 ステージには、比較器、汎用レジスタ GRs1,GRs2 の読み出し部、スタックポインタ(SP)の読み出し部がある。OF2 ステージには、SP を 1 加減算する加減算器、レジスタのデータとオペランドに書かれたアドレスを足す加算器(Adder)がある。MEM ステージには、メモリからのデータの読み込み部、及びメモリへのデータの書き込み部がある。ALU ステージには演算を行なうための ALU とフラグレジスタ(FR)への書き込み部がある。WB ステージでは結果を書き込むための汎用レジスタ GRd、SP、プログラムカウンタ(PC)への書き込み部がある。この用意されているハードウェア部品を選択し、ID ステージで部品の接続を決定する。

[†] 明治大学大学院理工学研究科基礎理工学専攻情報科学系

[‡] 株式会社ヴィアックス

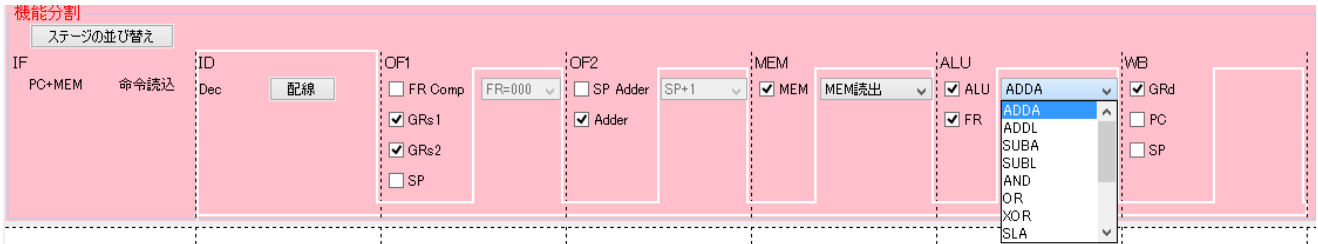


図2 機能分割からの命令設計

4.4 意味分割からの命令設計

意味分割から命令を設計するには、図3の意味分割からオペコード、オペランドを選択することによって命令設計を行える。オペコードには、「算術加算、論理加算、算術減算、論理減算、ロード、ロード(フラグセット)、ストア、正分岐、負分岐、零分岐、非零分岐、オーバーフロー分岐、無条件分岐、論理積、論理和、排他的論理和、算術左シフト、算術右シフト、論理左シフト、論理右シフト、算術比較、論理比較、PUSH、POP、CALL、RET、NOP、HALT」といった処理内容を表すオペコードが用意されている。オペランド選択は第1オペランドと第2,3オペランドの選択の2つに分かれている。この第1オペランドは、「レジスタ、無し」の2通りの選択ができる。また第2,3オペランドには、「アドレス・レジスタ、アドレス、レジスタ、無し」の4通りの選択ができるようになっている。

意味分割から命令設計を行なうと、情報処理技術者試験にて用いられているCOMET2プロセッサの動作を可視化することができ、どのように設計すれば良いのか理解できるようになっている。

4.5 意味と機能との対応

意味と機能の対応は、その対応が分かりやすいブロック図の位置にハードウェア部品を配置していることで実現している。図1ではブロック図の中央下に位置するAdderにマウスを合わせた状態である。意味分割命令のオペランド2,3と、機能分割命令のOF2ステージのAdderとの対応が明瞭に視覚化されている。

4.6 処理ステージの順番の変更

この命令ダイアグラムツールでは、処理ステージの順番を変更することができる。ステージの順番の変更によって設計した命令が正常動作できなくなる場合、誤っている箇所のハードウェア部品の接続が赤く表示される。この機能によってパイプライン動作する命令を設計する際に、どのような時に命令の実現ができなくなるか分かるようになっており、命令の仕様と設計技術について深く理解することができる。

5. 結論

この命令ダイアグラムツールでは、意味分割と機能分割からの命令設計という手法を用いて、命令のブロック図を描画することによって、命令の設計技術を自力で理解できるツールとなっている。意味と機能との対応を表現することによって、命令が持つ意味や動作を考えることができ、さらに処理ステージの順番指定によって、処理するハードウェアの順番が命令に与える影響も調べられる。また、現

在は意味分割命令についてはCASL2を主に扱っているが、より普及しているプロセッサの命令についても扱えるようにする予定である。

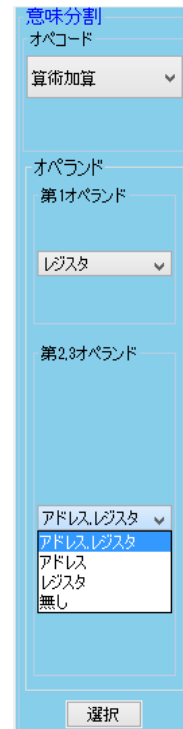


図3 意味分割からの命令設計

参考文献

- [1] 難波 翔一郎, 志水 建太, 山崎 勝弘, 小柳 滋, “プロセッサ設計支援ツールの設計・実装とハード/ソフト協調学習システムの評価”, FIT2007, LC-002, (2007).
- [2] 高橋 隆一, 児島 彰, 上土井 陽子, 吉田 典可, “マイクロコンピュータ設計教育環境 City-1”, 情報処理学会研究報告, 設計自動化研究会報告, IPSJ SIG Notes, Vol.97, No.17(19970214) pp 41-48, (1997).
- [3] 森本 智之, 堤 利幸, “パイプラインプロセッサ設計教育システム MEIMES-DESIGN の初期開発”, FIT2005, N-025, (2005).
- [4] Tomoyuki Morimoto, Toshiyuki Tsutsumi, “Development of Instruction Analysis Tool for Microprocessor Design Education”, The 17th International Conference on Computers in Education ICCE 2009, pp 489-491, (2009).