C-016

汎用命令を用いた命令シミュレーションツールの初期開発
Early Development of Instruction Simulation Tool with Meta-instruction

小野木 俊介† 鈴木祐一郎†,‡ 荻田 忠弘† 川股 凌太† 森本 智之†,§
Syunsuke Onogi Yuichiro Suzuki Tadahiro Ogita Ryota Kawamata Tomoyuki Morimoto
堤 利幸†
Toshiyuki Tsutsumi

1. 概要

我々は、マイクロプロセッサの任意の命令を実装するためのアーキテクチャツールの研究開発を行っている。我々は設計した命令を実行するため、汎用的な命令表現を用い、命令を動かす仕組みを提案する。命令を、意味を中心とした意味表記および、使用するハードウエアとその接続を中心とした機能表記で表現する。提案する仕組みは機能表記を用いてシミュレーションを行う。このため、ユーザは、意味表記でプログラムを記述すると、意味表記と機能表記を結びつけることできるので、設計した命令を実際に動かすことができる。今回、教育用マイクロプロセッサである COMET-II の命令を提案した手法で動作できることを確認できたので報告する。

2. 背景

我々は、アーキテクチャレベルでのマイクロプロセッサの設計を目指し、アーキテクチャ設計ツール MEIMAT(Meiji University Microprocessor Architecture design Tools)の開発を行っている。アーキテクチャレベルで命令を設計するためには、命令の意味と機能の対応を考えることが必要であり[1]、命令を意味表記と機能表記の 2 つの表現で表すことにより、命令の意味とプロセッサの動作の対応を表現する方法を提案した。また、この手法を用いて教育用マイクロプロセッサである COMET-II の命令を図示するツールを開発した[2][3]。さらに任意の命令を設計できるようにするために、意味表記と機能表記を改良して汎用命令を提案した[4]。しかし、これまでの命令ダイアグラムでは、図示された命令は実行できないという欠点があった。そこで、その命令を実行するためのシミュレータを開発する必要性がある。

3. 研究目的

我々のツールでは、意味表記で記述されたプログラムの各命令を機能表記に変換することができる。機能表記で命令を動作させることができれば、ユーザが設計した命令を動かすことができると考えている。今回、COMET-II の命令を意味表記に変換したプログラムをシミュレーションできたので報告する。

4. 命令シミュレーションツール

4.1 意味表記と機能表記

図 1 は、COMET-II の命令”ADDA GR2 GR3”を我々が提案している汎用命令の意味表記と機能表記で表したものである。意味表記では命令の意味を記述し GR2 値と GR3 値を加算し、GR2 に格納するという意味である。機能表記では命令を実行するために必要なステージおよびハードウェアモジュールとそのモジュール間の接続情報を記述する。本論文で我々が提案するシミュレーションは、機能表記の記述を用いて、シミュレーションを行う。

```
意味表記
(GRn1)<=(GRn2)@(GRn3),@ALU=S16_ADD,SRset;GRn1=GRn[2],GRn2=GRn[2],GRn3=GRn[3]
機能表記
 1_IF{pc(pc),PC_ADDER(PC_ADDER_16),terminal(TERMNAL),i_cache(i_cache)};
 2_ID{decoder(decoder)};
 3_OF1{GRn2(GRn2),GRn3(GRn3)};
 4_OF2{};
 5_MEM{};
 6_ALU{ALU(S_ALU_16),ALUMODE(ALUMODE),SR(SR)};
 7_WB{GRn1(GRn1)};
 Wire_{(IF.PC.out)->(IF.PC_ADDER.pc_in),
 (IF.PC_ADDER.out)->(IF.PC.in),
 (IF.TERMINAL.one_word)->(IF.PC_ADDER.val_in),
 (IF.PC.out)->(IF.I_CACHE.adr),
 (IF.I_CACHE.out)->(ID.DECODER.inst_in),
 (ID.DECODER.equation)->(OF1.GRn2.num),
 (ID.DECODER.equation)->(OF1.GRn3.num),
 (ID.DECODER.equation)->(WB.GRn1.num),
 (OF1.GRn2.out)->(ALU.ALU.in1),
 (ALU.ALUMODE.s16_adda)->(ALU.ALU.mode),
 (OF1.GRn3.out)->(ALU.ALU.in2),
 (ALU.ALU.status)->(ALU.SR.in),
 (ALU.ALU.out)->(WB.GRn1.in)};
```

図 1. COMET-II 命令”ADDA GR2 GR3”の意味表記と機能表記

4.2 機能表記を用いたシミュレーション

提案するシミュレーション手法では、機能表記で記述された各ハードウェアモジュールと配線を評価する。例えば、図 1 の(OF1.GRn2.out)->(ALU.ALU.in1)という接続情報から、OF1 ステージのモジュール GRn2 を動作させて出力端子から出力されるレジスタ値を ALU ステージのモジュール ALU の入力端子に渡す。また、(ALU.ALU.out)->(WB.GRn1.in)という接続情報を用いて、ALU ステージのモジュール ALU を動作させて、その出力端子から出力される演算結果値を WB ステージのモジュール GRn1 の入力端子に受け渡す。すべての接続情報を評価することにより、その命令を最後まで実行することができ、命令をシミュレートすることができるようになる。

4.3 図示ツールの外観

図 2 は、”ADDA GR2 GR3”を MEM ステージまで実行させた際の命令ダイアグラムツールである。実行する命令の意味表記とその命令に必要な回路のブロック図がツール上に図示されている。図 2 のブロック図では命令が実行されている MEM ステージまで背景が青色になっている。ユーザは、プログラム上の任意の命令、ステージを指定し、シミュレーションをそのステージまで実行することができる。ツール上のブロック図では実行されたステージまで背景が

† 明治大学 Meiji University
‡ 株式会社アルトナー ARTNER Co., Ltd
§ 株式会社ヴィアックス VIAX Co., Ltd

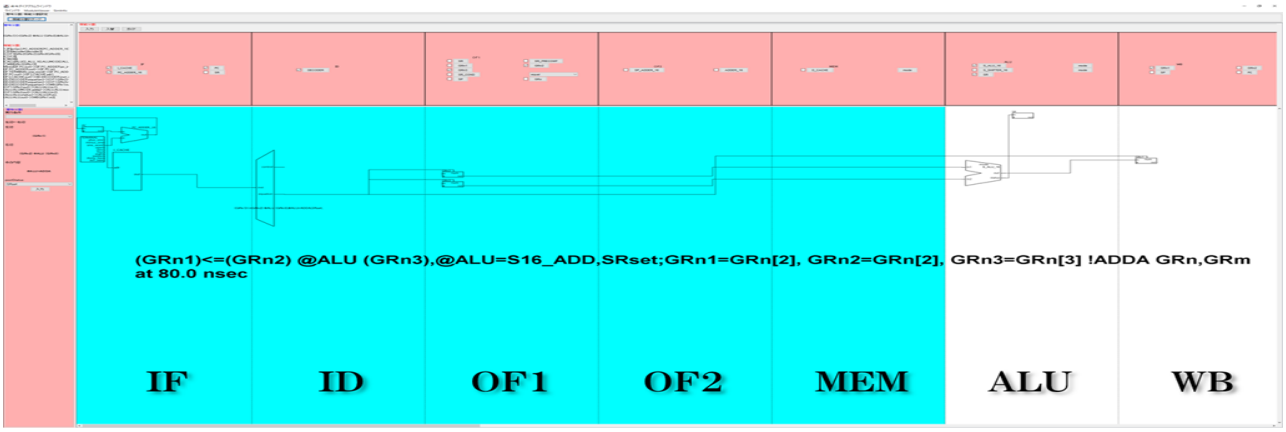

図 2. COMET-II 命令”ADDA GR2 GR3”の MEM ステージまで実行した命令ダイアグラムウィンドウ

青色になり、シミュレータが実行したステージがわかるようになっている。

4.4 命令による汎用レジスタの値の変化

我々はシミュレーション結果を GUI 上で確認できるようにモジュールビューワを開発している。モジュールビューワは、CPU の構成に使用されている各ハードウェアモジュールの中身の情報を表示する。表示する情報は、モジュールの名前、シミュレーションの経過時間、格納されている値などである。

図 3 は、”ADDA GR2 GR3”を MEM ステージまで実行した時の汎用レジスタのモジュールビューワである。シミュレーションを開始してから 80.0nsec 経過したことがわかる。また、その時の汎用レジスタの中身が表示されている。”GRn2”には 0x0004、”GRn3”には 0x0006、それ以外のレジスタには 0x0000 が格納されていることがわかる。この時点では加算処理や結果の書き込みをしていないため、汎用レジスタには命令実行前のデータが格納されている。

次に図 4 は、”ADDA GR2 GR3”の実行をさらに進め、WB ステージまで実行した時の汎用レジスタのモジュールビューワである。WB ステージまで実行すると加算結果が格納される。シミュレーションを開始してから 84.0nsec 経過したことがわかる。また、この時、”GRn2”には 0x000A、”GRn3”には 0x0006、それ以外には 0x0000 が格納されている。WB ステージで、”GRn2”に、図 3 の時の”GRn2”と”GRn3”の値を加算した値を代入したため、図 4 の”GRn2”の中身が 0x000A になっている。このようにして COMET-II 命令の実行が正しくシミュレーションされていることを確かめることができた。

5.結論

我々は、以前提案した 1 つの命令を意味表記と機能表記に分けて表現する方法を発展させ、機能表記を用いて COMET-II の命令を実際に動かすことができる簡単なシミュレータを開発した。汎用レジスタの内容を確認するモジュールビューワを設計し、レジスタの中身を確認することによってシミュレータの動作を確認した。

しかし，汎用レジスタのモジュールビューワのみではシミュレータとして不十分である。少なくとも、メモリや ALU の値などのモジュールビューワも必要であり、現在開発中である。また、パイプライン処理に対応したシミュレーションの開発も予定している。

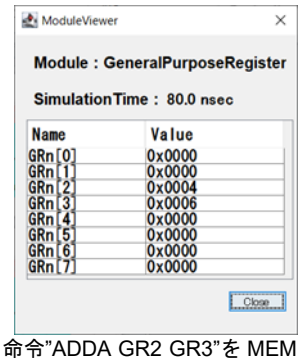

図 3. COMET-II 命令”ADDA GR2 GR3”を MEM ステージまで実行した時の汎用レジスタのモジュールビューワ

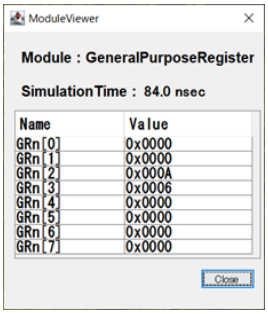

図 4. COMET-II 命令”ADDA GR2 GR3”を WB ステージまで実行した時の汎用レジスタのモジュールビューワ

参考文献

[1] 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).

[2] 岩本 稜平,森本 智之,堤 利幸 "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",FIT2014,C-024(2014)

[3] T. Morimoto, R. Iwamoto and T. Tsutsumi, "Early Development of Visualization Tool for Instruction Implement of Microprocessor," 2019 19th International Symposium on Communications and Information Technologies (ISCIT), pp. 5-9, (2019)

[4] 榎本 崇,花井 北斗, 鈴木 祐一郎, 佐野 友輝,森本 智之,堤 利幸,”マイクロプロセッサアーキテクチャツール MEIMAT の開発-汎用命令の提案-“,FIT2018,C-016(2018)