

仮想 CPU 作成ツールに対応したシミュレータソフトの作成

原美智子 ^{†1} 佐々木純一 ^{†2} 秋和潤 ^{†3} 西田誠幸 ^{†4} 原田紀夫 ^{†5}
 拓殖大学工学部情報工学科 ^{††}

1. はじめに

計算機アーキテクチャの導入教育支援の1つとして、CPUのシミュレータを教材として用いる方法がある。これまでに、いくつかのCPUのシミュレータソフトが研究、開発されている [1][2]。これらのシミュレータは対象となるCPUがあらかじめ定められたものであり、メモリの内容を変更し、実行状況をGUIにより表示することでその動作が確認できるソフトウェアである。

ここで、シミュレータが対象とするCPUの仕様を使用者が定めることができれば、学生は講義などで得た計算機アーキテクチャの知識を手軽に確認できるようになる。また教師は、講義において自身が望むCPUの仕様に沿ったシミュレータを教材として用いることができる。

そこで本稿では、ファイルに格納されたCPUの仕様記述を読み込み、計算機上で仮想実行するシミュレータソフトを開発する。

本シミュレータを用いることで、使用者はCPUの仕様を自由に記述し、GUIを通じてCPUの実行の様子を確認することが可能となる。

また我々は、本稿と並行して、シミュレータに対応した仮想CPU作成ツールの開発を進めている。これは、機能回路を単位としてCPUを設計することができるソフトウェアである。

2. 本ソフトの構成

使用者は本ソフトを用いて、CPUの仕様記述の作成、制御信号の作成、メモリの内容の書き込み、CPUの動作確認ができる。

仕様記述の作成には2通りの方法を用意する。1つは、CPUの仕様をエディタで直接記述する

方法である。もう1つは、仮想CPU作成ツールにより作成する方法である。

制御信号の入力方法は、仕様記述ファイルに記述する方法と、シミュレータ上でGUIを用いて作成、変更する方法を用意する。

2.1 対象となるCPUの構成

本ソフトが対象とするCPUは、アキュムレータアーキテクチャか汎用レジスタアーキテクチャのどちらかとし、使用者によって決定され、その演算命令当たりの明示的オペランド数は1または2とする。また、オペランドはメモリまたはレジスタを直接指定し、命令セットでの演算操作の種類や各々のオペランドのタイプ、サイズは使用者に設定させることとし、メモリ参照には直接アドレッシング方式のみを可能とする。

機能回路として使用可能なものは、プログラムカウンタ、インストラクションレジスタ、汎用レジスタ、メモリ、ALU、マルチプロセッサの6種類である。このうち汎用レジスタとマルチプロセッサの使用数は無制限とする。

2.2 CPUの仕様データ

本ソフトでは、設計されたCPUの仕様記述を読み込むことでCPUの動作をシミュレートすることができる。シミュレートするために必要なデータは、命令についての情報と、CPUを設計する際に選択されたレジスタ等の回路とその位置情報、各回路の名前と機能、各回路間の接続情報、制御部の情報である。命令についての情報とは、命令の名前とその働き、命令部、アドレス部の長さ、アドレス方式についての情報である。これらのデータを用いて本ソフト側で実行、確認を行えるようにする。

2.3 表示方法

実行画面上には、設計されたCPUの構成図の他に、タイムチャートウィンドウや、メモリ書き込みウィンドウを表示させるツールボタンを用意する。タイムチャートウィンドウは、制御信号

Development of a CPU Simulator Based on Dynamically Specified CPU Description

^{†1}Michiko HARA (mhara@cs.takushoku-u.ac.jp)

^{†2}Junichi SASAKI (j-sasaki@cs.takushoku-u.ac.jp)

^{†3}Jun AKIWA (jakiwa@cs.takushoku-u.ac.jp)

^{†4}Seikoh NISHITA (snishita@cs.takushoku-u.ac.jp)

^{†5}Norio HARADA (nharada@cs.takushoku-u.ac.jp)

^{††}Dept. of Computer Science, Faculty of Engineering, Takushoku University

の確認と、その入力順序の指定に用いられる。また、CPUの仕様記述内に制御信号が記述されていない場合には、CPUの処理を実行する直前に、制御信号の入力を促す。

2.4 CPUの実行、制御方法

本ソフトを用いて使用者は、1) 制御部の設計、2) CPUの実行、3) メモリの書き込み、書き換えを行うことができる。このうち制御部の設計は、タイムチャートウィンドウを用いて制御信号のタイムチャートを作成して行う。さらに、CPUの実行中に設計の誤りを発見した場合などには、その場で制御信号を変更し、再び実行し直すことが可能である。

また、CPUの実行方法は3種類あり、それぞれ、プログラム全体、1命令の処理、1クロック分の処理が実行される。

2.5 誤りを含むCPU仕様に対する処理

本ソフトの特徴の1つは、CPUの仕様記述に誤りがあった場合の処理である。誤りとして、次の5つを想定した。

1. 機能回路やデータパスの配置、配線が不適切なため、一部の命令を実行できない。
2. 制御信号の入力順序の誤りにより、一部の命令を実行できない。
3. データパスの入力ビット幅と出力ビット幅に不整合がある。
4. IRの命令部に格納されたデータがデコードできない。
5. 機能回路の入出力がデータパスに接続されていない。

このうち、1~3の場合はそのまま動作させ、使用者にその誤りの内容について考えさせる。但し、2でレーシングが発生する場合と、4では、CPUのシミュレートが不可能となるため、その旨のエラーメッセージを表示し、実行を停止する。5の場合は、CPUについての記述ファイルを読み込んだ段階でエラーメッセージを表示し、実行を禁止する。

3. 仮想CPU作成ツールについて

仮想CPU作成ツールでは、GUIで回路や命令を指定しながらCPUを設計することができる。使用者は、設計するCPUの仕様に合わせて回路

を選択し、設計画面上に自由に配置する。また、それらをデータパスで接続することでCPUの構成を決定する。同時に命令の仕様も定めることが可能である。制御部については、作成ツール側、シミュレータソフト側、双方で設計可能にする。

ここで設計されたCPUの仕様は、本ソフトの入力として使用できる。

4. 本ソフトの実装

本シミュレータソフトは、Java1.4を用いて作成した。実行画面を図1に示す。

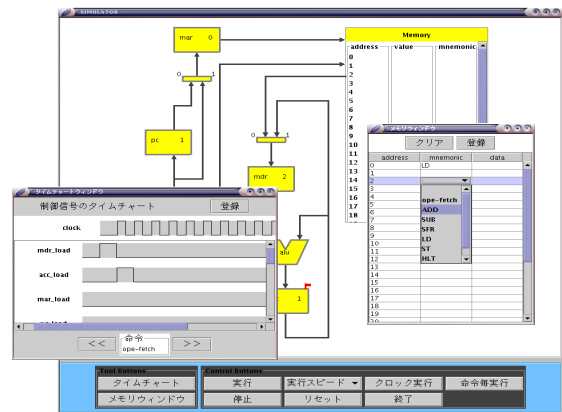


図 1: CPUシミュレータの実行の様子

5. おわりに

本稿では、CPUの仕様記述を入力とするCPUのシミュレータソフトの開発を行った。今後の課題として、シミュレータの評価、さらにCPUの仕様記述からの仕様書の作成がある。

参考文献

- [1] 渡辺博芳: “WCASL-2(COMET2/CASL2シミュレータ)”, 帝京大学.
- [2] 山村周史, 新實治男: “Java言語を用いたKUE-CHIP2シミュレータの開発”, 京都工芸繊維大学公開文書, [1997].
- [3] 佐々木純一, 西田誠幸, 原田紀夫: “CPUの構成と動作を学習するための実験教材の作成”, 情報処理学会第65回全国大会講演論文集, [2003].
- [4] John L.Hennessy, David A.Patterson: “Computer Architecture: A Quantitative Approach”, 日経BP社, [2003].