

プロセッサアーキテクチャ学習のための スーパースカラシミュレータの開発

志水 建太[†] 山崎 勝弘[†]

立命館大学大学院 理工学研究科[†]

1. はじめに

本研究室では、ハードウェアとソフトウェア両面の知識を持つ人材を育成するために、命令セットとマイクロアーキテクチャを設計することで、両面の境界を学習できるハード/ソフト協調学習システム (Hardware / Software Co-learning System:HSCS) の研究を進めている [1]-[3]。現在のシステムは、シングルサイクル・マルチサイクル・パイプラインアーキテクチャが学習できる。

本論文では、さらにスーパースカラプロセッサにおけるハザードとその対策について学習できるスーパースカラシミュレータの開発について述べる。

2. ハード/ソフト協調学習システム

ハード/ソフト協調学習システムとは、学習者がプロセッサ設計を通してソフトウェアとハードウェアの知識を習得し、境界を理解していく教育システムである。HSCSは、MIPSのサブセット MONIを用いたアセンブリプログラミングとプロセッサ設計を行うプロセッサ学習システム、及び独自の命令セットを定義し、そのプロセッサ設計と、FPGAボード上で検証できるプロセッサ設計支援ツールから構成される。図1にHSCSの学習フローを示す。まず、学習者はMONI命令セットを用いてアセンブリプログラミングを行い、MONIシミュレータで実行してアセンブリプログラミング技術とプロセッサアーキテクチャの知識を習得する。次に、MONIプロセッサをHDLで設計して、プロセッサデバッグ・モニタを用いて実機検証し、プロセッサの基本的な設計能力を習得する。さらに、命令セット定義ツールを用いて独自の命令セットを定義して命令セットシミュレータで実行し、命令セットを検証する。命令セットアセンブラで検証プログラムを作成し、HDLシミュレータや実機で設計したプロセッサのデバッグに入る。本研究室では、2004年から4回生の卒業研究の導入実習を行い、HSCSの評価を行っている[2]。

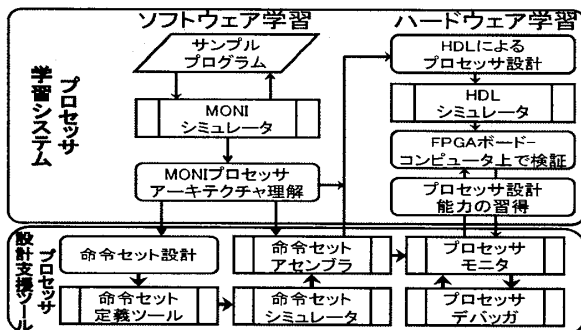


図1: ハード/ソフト協調学習システムの学習フロー
Development of a Superscalar Simulator for Processor Architecture Study, Kenta Shimizu and Katsuhiro Yamazaki, Graduate School of Science and Engineering, Ritsumeikan University.

3. スーパースカラシミュレータの開発

3.1 設計方針

本シミュレータで学習できるアーキテクチャは、2命令を同時実行するシングルサイクルとパイプラインの2つである。設計方針を以下に述べる。

(1) 2命令同時実行シングルサイクルプロセッサ

まず、シングルサイクルでは、2命令を同時に処理する際に発生するデータハザードと制御ハザードについて理解する。ハザードが発生した場合のデータの流れを学習していく。また、ハザードを発生させないようなアセンブリプログラミングの学習も行う。

(2) 2命令同時実行パイプラインプロセッサ

次に、本来のスーパースカラのアーキテクチャを学習する。ここでは、複数の命令をパイプライン処理していく際に発生するデータハザードと制御ハザードについて理解し、それぞれのハザードに対するフォワーディング、ストールや遅延分岐を行うアーキテクチャについて学習を進めていく。さらにアセンブリプログラミングの能力も高めていく。

(3) ハザードの理解とその対策

2つのアーキテクチャでシミュレーションを行うことで、2命令間に発生するデータハザードや制御ハザード、パイプライン処理による複数の命令間に発生するデータハザードや制御ハザードについて理解していく。シングルサイクルでは、ハザードがどのような場合に発生するかを知る。パイプラインでは、複数の命令間に発生するデータハザードはフォワーディングによって解決し、メモリ読み出し命令はストールすることを理解する。同時実行中の2命令間のデータハザードや制御ハザードでは、ストールや遅延分岐を行うことを理解する。

プログラムをエディタに書き、命令メモリに書き込む際に、プログラムやデータのチェックを行う。エラーがなければ、データメモリにデータを書き込み、実行を行う。実行には、複数のモードが用意されており、デバッグが容易に行える。表示には、実行ごとにデータが更新され、様々なデータを確認することができる。シミュレーション後は、各命令の実行回数などを表示し、プログラムや命令セットの評価を行う。データパスも実行ごとにデータの流れを強調表示するため、アーキテクチャについて理解を行いながら、ハザードを発生させない最適なアセンブリプログラミングの学習ができる。

3.2 シミュレータの機能と構成

図2にシミュレータの機能と構成を示す。機能は、表示、リセット、メモリ書き込み、実行の4つに大別される。表示では、実行した命令、データ・制御ハザード、データパスなどプログラム作成とアーキテクチャの学習に必要な情報を表示する。リセットでは、データメモリなどを初期化することで繰り返しテストが行え、またエディタに記述した内容も初期化する。メモリ書き込みで

は、プログラムなどの書き込みや記述が正しいかチェックする。また、パイプラインでは遅延分岐用に入れ替えて書き込みを行う。実行では、4つの実行モードがありデータハザード、制御ハザードの検出も行う。

コマンドボタンの種類によって、リセット、メモリ書き込みや実行を行い、結果のデータを表示する。

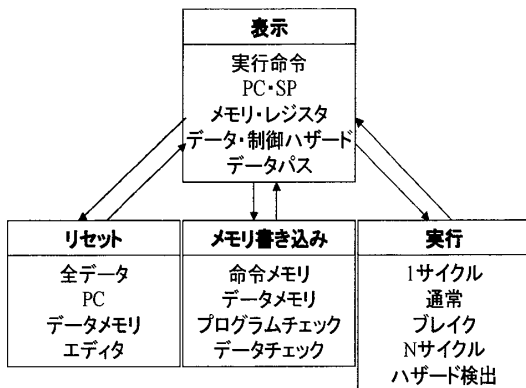


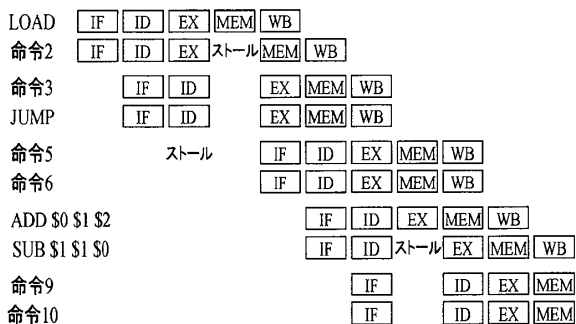
図 2: スーパーカラシミュレータの構成

3.3 シングルサイクルとパイプラインの動作

シングルサイクルは 2 命令を同時実行し、ハザード発生時は 1 命令だけが実行される。パイプラインは 2 命令を命令読み出し、命令解読、実行、メモリアクセス、レジスタ書き込みの 5 サイクルでインオーダー実行する。それぞれの動作を図 3 に示す。シングルサイクルでは、1 命令目の書き込み先を 2 命令目が参照する際に発生するデータハザードと、1 命令目が分岐命令の際に発生する制御ハザードがあり、いずれの場合も 2 命令目を実行しない。1 命令目が条件分岐の場合は 2 命令目も実行し、条件が成り立てば 2 命令目の結果を破棄する。パイプラインでは、データハザードが発生した場合、実行とメモリアクセスのデータを利用するフォワードリングで解消する。ロード命令の結果を次の命令が参照する場合は、メモリアクセスでストールする。制御ハザードが発生した場合は、JUMP 命令では命令解読でストールする。1 命令目の結果を 2 命令目が使用する場合は、実行でストールする。条件分岐命令の場合は、遅延分岐を行う。



(a) シングルサイクル



(b) パイプライン

図 3: シングルサイクルとパイプラインの動作

4. スーパーカラシミュレータのテスト

図 4 にシングルサイクルのデータバスを示す。2 命令同時実行のシングルサイクルなので、ALU モジュール、レジスタとメモリの入出力のポートなどを 2 個ずつ用意している。テストプログラムは 9 個 (N までの和, 最大値検出, 乗算, 除算, 素数判定, 根の判別, 三角形判定, 一次方程式, 最大公約数) を用いた。これらのテストで、全ての命令の動作や複雑なプログラムの動作も正しく動作することを確認できた。今回のテストでは、MONI シミュレータで作成したプログラムであったため、ハザードが何度も発生した。本シミュレータを用いることにより、ハザードを発生させないような深く考えられたプログラミングができると思われる。

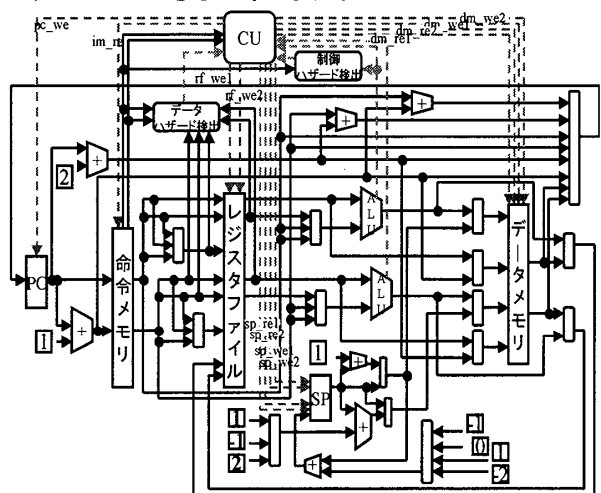


図 4: シングルサイクルのデータバス

5. おわりに

本論文では、2 命令を同時実行するシングルサイクルとパイプラインスーパーカラシミュレータの概要、機能と構成、動作、シングルサイクルのテストについて述べた。テストでは 9 個のプログラムを用い、全て正しい動作をすることを確認した。テストプログラムは 1 命令実行用のプログラムであり、ハザードが何度も発生した。本ツールを使用することで、ハザード発生回数を減らしたプログラミング能力が身につく、並列処理を考慮した独自プロセッサの設計ができると思われる。

今後の課題として、2 命令同時実行のパイプラインの設計と実装を行うことが挙げられる。パイプラインでは、順番に命令を実行しないアウトオブオーダーの処理も学習させたい。

参考文献

- [1] 難波, 他: 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] 井手, 他: 学生によるプロセッサ設計実験の基づいたハード/ソフト協調学習システムの評価, FIT2008, C-009, 2008.
- [3] 志水, 他: プロセッサ設計教育における命令セット定義ツールと命令セットシミュレータの試作, 情報処理学会関西支部, 支部大会講演論文集, A-05, 2008.