

Verilog-HDLによるRISC-V 5段パイプラインプロセッサの設計

菊池智也[†] 大津金光[†] 大川猛[†] 横田隆史[†]
[†]宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

近年、オープンな命令セットアーキテクチャとしてRISC-V[1]が注目を集めており、様々な実装のRISC-Vプロセッサが公開されている。このオープンであるという特徴から、今後より一層、RISC-Vプロセッサを利用したシステムの研究や開発が行われていくと予想される。

我々は、FPGA上で高性能データ並列処理を実現するソフトコアプロセッサによって、FPGAを利用したシステム開発の短期間化を目指した研究を行っているが、そのためのプロセッサコアとして、前述したRISC-Vを用いることを検討している。本研究において、単純な5ステージパイプラインを持ち、Verilog-HDLで記述されたプロセッサコアを求めているが、我々の知る範囲ではそのようなオープンソース実装は存在しなかった。そこで、上記の要求を満たすプロセッサを我々は開発する。

よって本稿では、命令の追加拡張を行いやすく、Verilog-HDLを用いて記述されたRISC-Vソフトコアプロセッサを開発する。本プロセッサコアは、単純な5段パイプライン構成を採用し、RISC-Vの最小命令セットであるRV32I[2]を採用する。また、論理シミュレーションによる動作検証について示す。

2 プロセッサコアの仕様と実装

設計するプロセッサコアの内部構成を図1に示す。データバス幅とアドレスバス幅はともに32bitとして、プロセッサコアのアーキテクチャは性能とシンプルさを考慮して、5段のシングルパイプライン構成を採用する[3]。

パイプラインは命令フェッチ(IF)ステージ、命令デコード(ID)ステージ、演算(EX)ステージ、メモリアクセス(MA)ステージ、レジスタ書き込み(WB)ステージの5ステージで構成する。制御ハザードによるペナルティコストを減らすために、分岐命令の処理はIDステージで行うことに加えて、分岐予測を利用する。分岐予測機構には、実装が容易である静的分岐予測(常に不成立と予測する)を採用する。

パイプライン構造であるために発生する命令間のデータ依存(フロー依存)が原因のストールを減らすために、データフォワーディング機構を備える。フォワーディングは、分岐命令の処理を行うIDステージと、そ

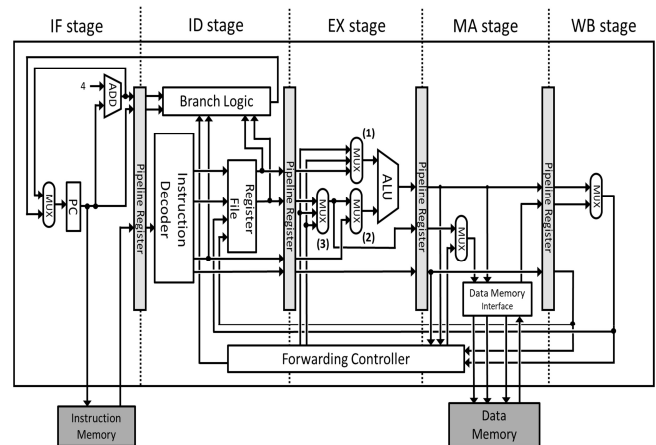


図1: プロセッサコアの内部構成

の他の演算命令の処理を行うEXステージ、ロード/ストア命令の処理を行うMAステージに対して行う。

さらに、レジスタファイルが原因となる構造ハザードを解消するために、レジスタファイルには入力アドレスバスを書き込み用と読み出し用それぞれ別に用意し、レジスタへの書き込みと読み出しを同時に行うことを可能にする。ただし、同じアドレスに対して書き込みと読み出しが同時に発生した際には、書き込みによって更新された新しい値が読み出されるようにする。

また、実装する命令が多いとプロセッサ回路の規模が増加したり、回路が複雑化する。このように設計、実装が複雑化してしまうことを避けるために、本プロセッサではRISC-Vの最小の命令セットであるRISC-V 32bit基本命令セット(RV32I)に限って実装をする。なお、当面は例外処理および割込み処理機能等を使わないので、それらの処理に関する機能の実装は行わない。

次にプロセッサの実装について考えていく。RV32I命令は最大で2つのソースレジスタ(rs1, rs2)を指定可能であるが、レジスタの読み書きを同時に行う場合、レジスタファイルには入力アドレスバスが3つ必要になる。FPGAチップ内のブロックRAMを利用することを考えると、ブロックRAMは最大2入力であるため、1つのブロックRAMではレジスタファイルを構成することができない。そのため、IDステージに含まれるレジスタファイルはブロックRAMを2つ用いて構成する。

EXステージにおいては、ALUに入力されるデータは2つあり、1つ目のデータはrs1とプログラムカウンタ、フォワーディングデータから選択し、2つ目のデータはrs2と即値、フォワーディングデータから選

Design of RISC-V 5-stage pipeline processor with Verilog-HDL

[†]Tomoya Kikuchi, [†]Kanemitsu Ootsu, [†]Takeshi Ohkawa, [†]Takashi Yokota

Department of Information Systems Science, Graduate School of Engineering, Utsunomiya University ([†])

択する。ALUの各入力データの選択にはマルチプレクサを2つ(図中(1), (2))用いるが、ALUの2つ目の入力データであるrs2とフォワーディングデータは、どちらかのデータが選択されてMAステージにおけるデータメモリに対するメモリアクセスに用いられるため、ALU入力用とは異なるマルチプレクサ(図中(3))を使用する。

また、パイプラインレジスタとレジスタファイルの動作クロックは同じものを使用する。この際に両者がともにクロックの立ち上がり(または立ち下がり)に同期しており、IDステージで処理している命令とWBステージで処理している命令にデータ依存が存在する場合、1サイクルのストールが発生する。これを避けるためには、WBステージからIDステージへのフォワーディングパスが必要になる。

しかし、パイプラインレジスタはクロックの立ち上がりに同期させ、レジスタファイルはクロックの立ち下がりに同期させることで、IDステージとWBステージにデータ依存が存在する場合でもストールを避けることができる。これにより、WBステージからIDステージへのフォワーディングパスを削減することができる。

3 動作検証

今回実装したプロセッサコアの動作検証は、Xilinx社FPGAの開発環境であるVivado(2017.2)を用いた論理シミュレーションを行い、テストプログラムを実行してその結果を確認する。テストプログラムには、RISC-VのGitHubリポジトリ(<http://github.com/riscv/riscv-tests>)にて公開されている、RV32I用のテストプログラムを用いる。また今回の検証では、プロセッサコア部分の動作確認を行うことが目的であるため、プロセッサ外部のメモリ要素である命令メモリやデータメモリは、FPGAチップ内のブロックRAMを用いて構成する。テストプログラムをブロックRAMに初期値として設定しておくことで、プロセッサの動作開始と同時にテストプログラムを実行し、動作検証を行う。

シミュレーションにはVivado Simulatorを使用する。Vivado Simulatorを用いて論理シミュレーションによる検証を行ったところ、実装したすべての命令に対して、テストプログラムの結果がパスしたことを確認した。

また表1に、Vivadoを使用して論理合成をした際に出力された、FPGAリソースの使用量を示す。ここで示しているリソース使用量は、ターゲットデバイスのFPGAボードとしてNexysVideo(Xilinx XC7A200T-1SBG484C)[4]を選択した場合のものである。なお、表中で示されている数値は、命令メモリとデータメモリを除いたプロセッサコア部分のみのリソース使用量となっている。

表1で示されている各リソースの使用量は、Xilinx社製7シリーズFPGAの内、最もリソース量が少な

表 1: リソース使用量

FPGA リソース	使用量
LUT	964
FF	426
BRAM	1
BUFG	1

いFPGA(XC7S6)でも実装することが可能なリソース量である。そのため、よりリソースの少ない低価格なFPGAを用いた場合でも、本プロセッサを実装することは可能である。

なお、表1で示しているブロックRAMの使用量は1となっているが、これは36KbサイズのブロックRAMが1つ使用されていることを示している。

4 おわりに

本稿では、命令の追加によるプロセッサの拡張が容易な、Verilog-HDLを用いて記述されたシンプルなRISC-V 5段パイプラインプロセッサの設計を行った。設計したプロセッサコアの動作検証を、Vivado Simulatorを用いた論理シミュレーションによって行うことで、設計したプロセッサが正常に動作することを確認した。

今後の課題としては、今回行った論理シミュレーションでなく、FPGA上にプロセッサを実装して動作検証を行うこと、さらに、実際にこのプロセッサを利用して追加拡張を行ったプロセッサを設計して実装、評価することが挙げられる。

謝辞

本研究は、一部JSPS科研費16K00068, 17K00072の助成による。

参考文献

- [1] デイビッド・パターソン, アンドリュー・ウォーターマン: “RISC-V 原典 オープンアーキテクチャのスヌメ”, 日経BP社, 2018.
- [2] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovic: “The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.0”, Technical Report UCB/EECS-2014-54, EECS Department, University of California, Berkeley, May 2014.
- [3] David A. Patterson, John L. Hennessy: “Computer Organization and Design RISC-V Edition: The Hardware Software Interface”, Morgan Kaufmann, 2017.
- [4] “Nexys Video FPGA Board Reference Manual”, https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-video/nexysvideo_rm.pdf, 2019年1月アクセス.