ダウンロード・データあります

ハードの理解で差がつく時代

シミュレータの自作からはじめるCPU動作メカニズム

第2回 シンプルRISC-Vシミュレータを自作する

ご購入はこちら

中森 章

実は意外と手軽に作れる命令セット・シミュレータを，自作しながらCPUの動作メカニズムを理解しようというのが本連載の趣旨です．今回からRISC-Vシミュレータを自作していきます．本誌ウェブ・サイト（http://interface.cqpub.co.jp）からダウンロードできるようにしておきますので，自由に参照してください．（編集部）

今回自作に挑戦する命令セット・シミュレータ

● ターゲット…由緒正しき教育向きRISC-V

RISC-Vはカリフォルニア大学バークレー校（UCB）の研究から生まれたRISCプロセッサのISAで，オープンソースであることが大きな特徴です．

また，RISC-VのISAは，UCBで開発されたRISC-I（RISC-VはRISC-Iの5代目を意味する）に似ているという話もあります(1)．個人的には，MIPSのISAに非常に似ていると思っています注1．そのココロは「由緒正しきRISC（縮小命令セット・コンピュータ）」ということです．つまり，教育用には最適なISAです．コンピュータの教科書で有名な『ヘネパタ本』も，最初はMIPSに類似したISAを使って説明されていましたが，最新版ではRISC-Vに一新されています．

● サポート範囲…32ビット長基本コードの実行

基本的なISAは32ビット固定長の命令形式を採用し，特定分野のための命令拡張が用意されています．

RISC-Vは，プログラマ見え（＝プログラマから見える領域）には，32本の汎用レジスタと1本のプログラム・カウンタ（PC）を持っています．それらのレジスタのビット長（*XLEN*）は，32ビット，64ビット，128ビットのいずれかを採ります（**図1**）．このビット長がCPUのビット長を示します．

本連載では，RISC-Vの32ビットの基本的な実装であるRV32Iに乗除算を加えたRV32IMをサポートするISS（命令セットを実行するソフトウェア・シミュレータ）を作成します．

● 特権や割り込みは未対応

ただし，その目的は，コンパイラの吐き出すコードが実行できる程度なので，特権命令や割り込み／例外はサポートしません．

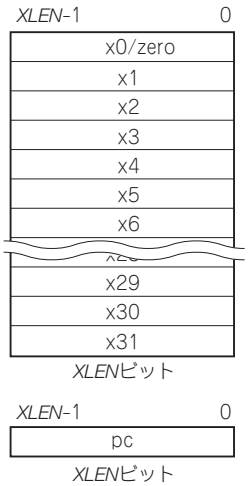

図1　由緒正しき教育用に最適なRISC-Vのレジスタは最長で128ビットだけどシンプルな32ビットのシミュレータを自作することにする

レジスタのビット長（*XLEN*）は32ビット，64ビット，128ビットの3種類があるが，本稿では32ビットを使用．x0は常に値が0であるレジスタで，定数値「0」の生成や，演算結果の書き捨て場所に利用する

命令フェッチ部を自作する

まずは命令フェッチ（＝命令の読み出し）です．命令フェッチは実行する命令をメモリから取り出します．

命令フェッチの操作は**リスト1**のようになります．PC（プログラム・カウンタ）が指すアドレスのメモリから4バイト（1ワード）を取ってきて命令レジスタに格納します．

命令レジスタとは，命令コードを一時的に保持しておくレジスタです．CPUは命令レジスタの内容をデ

注1：RISC-VのISAは先進的でMIPSのISAは古臭いという風潮がありますが，これは必ずしも正しくありません．MIPSのISAもRelease 6になって洗練され，RISC-Vと遜色のないものになっています．