

# オープンソースCPU 「RISC-V」の研究

[ご購入はこちら](#)

第3回 ハード&ソフト開発環境あれこれ

@msyksphinz

## RISC-Vの実装

### ● まずは本家バークレイから

本連載では、オープンソースCPUであるRISC-Vアーキテクチャを紹介しています。既に多くの半導体メーカーやIPベンダが、RISC-Vの実装や、対応したソフトウェアを提供しています。それらの全てを紹介しようとすると煩雑になってしまうので、ここではRISC-Vの開発元であるカリフォルニア大学バークレイ校(UCB)が開発、運用しているプロジェクトおよび、そこから派生したSiFive社の提供しているチップ/ボードを中心に紹介します<sup>編集注</sup>。SiFive社は早くから個人でも入手できるRISC-Vマイコン・ボードなどを提供しているので、興味がある人は知っておいてよいと思います。

バークレイの実装の利点は、全てのプロジェクトの成果物がオープンソースとしてGitHubに公開されており、簡単に中身を参照できることです。UCB以外のコミットによっても開発は行われています。特定の半導体ベンダやグループが開発するのではなく、オープンな場所で議論を行いながら、発展していくアーキテクチャというのは興味深く、その議論に気軽に参加できるというのは非常に魅力的です。

### ● その1：元祖Rocket Core

Rocket Coreは、UCBが開発しているRISC-V実装です(<https://github.com/freechipsproject/Rocket-Chip>)。Rocket CoreはCPUコアの部分を示しており、L1キャッシュやバスなどを含めてSoC環境を構成しています(図1)。

Rocket Coreは5ステージのシンプルなパイプラインのRISC-V実装ですが、マシン・モード、スーパーバイザ・モード、ユーザ・モードの実行モードをサポート

トし、デフォルトで64ビット・アドレッシングを備えています(図2)。また浮動小数点数演算まで実行できるほぼフルスペックのRISC-V実装となっており、手軽にRISC-Vプロセッサの動きを見たいときにはRocket Chipを見てみるというのがお勧めです。

また、Rocket Chipの特徴として、Verilog HDL(以下Verilog)などのハードウェア記述言語のフルクラッチで実装してあるわけではなく、Chiselという

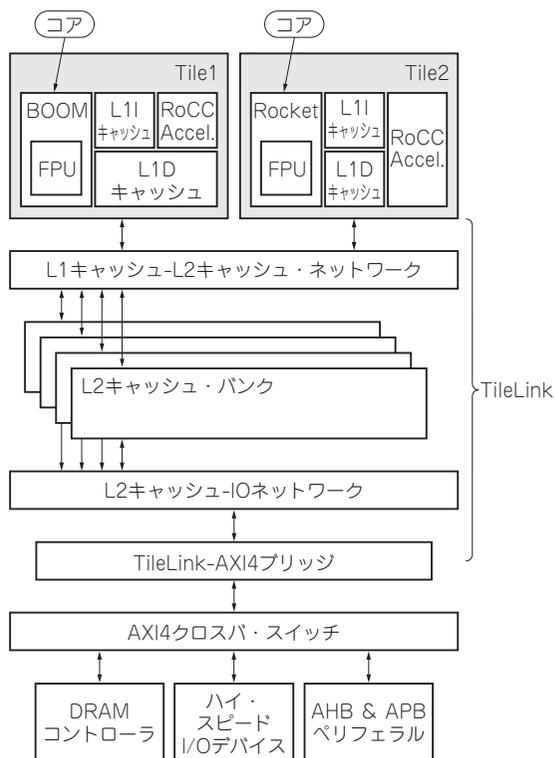


図1 (17) 元祖Rocket CoreとL1キャッシュなどを含めてSoCを構成している

注：本稿の内容は執筆時点のもので、随時更新されていく可能性があります。

編集注：本誌2018年2月号特集2「新時代プロセッサシステム作り」ではバークレイ以外のさまざまなコアも紹介しています。