

ご購入はこちら

オープンソースCPU 「RISC-V」の研究

第10回 CPU自作の醍醐味…Myカスタム命令を追加する

@msyksphinz

inst[4:2]	000	001	010	011	100	101	110	111
inst[6:5]								(>32b)
00	LOAD	LOAD_FP	custom0	MISC_MEM	OP_IMM	AUIPC	OP_IMM_32	48b
01	STORE	STORE_FP	custom1	AMO	OP	LUI	OP_32	64b
10	MADD	MSUB	NMSUB	NMADD	OP_FP	reserved	custom2/rv128	48b
11	BRANCH	JALR	reserved	JAL	SYSTEM	reserved	custom3/rv128	>80b

図1 RISC-Vの命令フィールドにはカスタム命令として利用できるフィールドが幾つか定義されている。custom0/custom1は将来的な利用が独自拡張を想定している。custom2/custom3はRV128命令と共用

● 今回やること

RISC-VのRocket Coreを拡張する方法にはいろいろあります。1つは命令を拡張してALUにいろいろ手を加える方法です。もう1つは、アクセラレータを外部に接続してそれに対するアクセスを実行するためのRoCC (Rocket Custom Coprocessor) というインターフェースを使う方法です。今回はRoCCインターフェースを使って、Rocket Coreのそばにアクセラレータを接続する方法について解説します注1。そしてソフトウェアのみでプログラムを構築した場合と比べて、どれくらい高速化できるか見てみましょう。

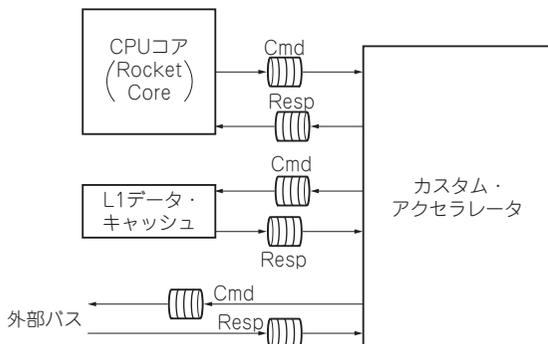


図2 RISC-VのRocket Chip実装でカスタム命令を実行するアクセラレータを追加可能な仕組みRoCC

Rocket Coreにカスタム命令を追加できる仕組み「RoCC」

● カスタム命令フィールド

RoCCはRocket Coreに対してカスタムのアクセラレータを接続するインターフェースです。RISC-Vの命令フィールドを見てみると、4種類のカスタム命令を追加することのできるフィールドが残されています(図1)。4種類のうち、custom2, custom3はRV128命令セットとかぶっていますが、custom0, custom1命令は将来の拡張か、カスタム命令用に定義されているため、自由に使用できます。

● CPUコアと外部アクセラレータとのインターフェース

Rocket Chipは、カスタム・アクセラレータを追加できる3種類のインターフェースを備えています。custom0, custom1, custom2命令を実行することで、このインターフェースを介してカスタム・アクセラレータと通信できます。

Rocket Chipのインターフェースの概要を図2に示

注1: 最新版のfpga-zynqリポジトリは2018年7月現在Rocket-Chipが正常にブートできなくなっています。筆者はfpga-zynqのリビジョンf03982e, 内包しているRocket Chipリポジトリはリビジョンf3299aeで実験を行っています。

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。