

# 計算機科学実験及演習3 ハードウェア 実験レポート ユーザーズマニュアル

22 班

小林雅季 学籍番号：1029284333

山田瑛平 学籍番号：1029282731

平成 30 年 6 月 14 日

## 概要

当プロセッサは altera 社の FPGA ボード CycloneIV EP4CE30F23I7N 上で動作するソフトウェアプロセッサです。SIMPLE (SIxteen-bit MicroProcessor for Laboratory Experiment) と呼ばれるプロセッサをベースに、改良を行ったものです。

## 特徴・性能

- メモリやレジスタに格納されるデータサイズの最小単位は 16 ビットです。
- 命令の種類は 23 種類です。
- 命令セットは 2 オペランド方式を採用しています。
- 演算命令で扱える数値は 16 ビット符号付き整数です。
- ハーバードアーキテクチャを採用しています。すなわち、命令を格納しているメモリと、データそのものを格納しているメモリがそれぞれ別に用意されています。
- 最大動作周波数は 51MHz です。
- 並列化により、1 命令につきおよそ 4 クロックで動作します。

## アーキテクチャ

### メインメモリ

前述の通り、当プロセッサは命令データ格納用のメモリとデータ格納用のメモリの 2 つを搭載しています。いずれも 16 ビット長のデータを 4096 語格納することが可能です。データはプロジェクトのルートに存在する *mem1.mif* および *mem2.mif* に記述することでメモリに格納されます。*mem1.mif* は命令メモリ、*mem2.mif* はデータメモリに対応します。当プロセッサは命令メモリの 1 番地の命令から実行を開始します。命令メモリの 0 番地に格納された命令はジャンプ命令などで 0 番地に飛ばない限り実行されないのをご注意ください。特別な事情のない限り、命令メモリの 0 番地には halt 命令 (C0F0) や値に一切変更を加えない命令 (C070) を格納しておくことを推奨します。

### 汎用レジスタ

当プロセッサの搭載する汎用レジスタの容量は 8 語です。この 8 語に関してはいずれも自由にロード・ストア・算術演算などによる値の更新が可能です。

命令コード					概要	詳細
15-14	13-11	10-8	7-4	3-0		
11	Rs	Rd	0000	d	$r[Rd] = r[Rd] + r[Rs]$	加算
11	Rs	Rd	0001	d	$r[Rd] = r[Rd] - r[Rs]$	減算
11	Rs	Rd	0010	d	$r[Rd] = r[Rd] \& r[Rs]$	論理積
11	Rs	Rd	0011	d	$r[Rd] = r[Rd] \parallel r[Rs]$	論理和
11	Rs	Rd	0100	d	$r[Rd] = r[Rd] \wedge r[Rs]$	排他的論理和
11	Rs	Rd	0101	d	$r[Rd] = r[Rd] - r[Rs]$	比較演算
11	Rs	Rd	0110	d	$r[Rd] = r[Rs]$	移動演算
11	Rs	Rd	0111	d	$r[Rd] = r[Rd] + \text{sign\_ext}(d)$	即値加算
11	Rs	Rd	1000	d	$r[Rd] = \text{sl}(r[Rd], d)$	左論理シフト
11	Rs	Rd	1001	d	$r[Rd] = \text{slr}(r[Rd], d)$	左循環シフト
11	Rs	Rd	1010	d	$r[Rd] = \text{srl}(r[Rd], d)$	右論理シフト
11	Rs	Rd	1011	d	$r[Rd] = \text{sra}(r[Rd], d)$	右算術シフト
11	Rs	Rd	1100	d	$r[Rd] = \text{input}$	入力
11	Rs	Rd	1101	d	$\text{output} = r[Rs]$	出力
11	Rs	Rd	1111	d	$r[Rd] = \text{halt}()$	プロセッサの停止
00	Ra	Rb	d.1	d.2	$r[Ra] = *(r[Rb] + \text{sign\_ext}(d.1, d.2))$	ロード
01	Ra	Rb	d.1	d.2	$*(r[Rb] + \text{sign\_ext}(d.1, d.2)) = r[Ra]$	ストア
10	000	Rb	d.1	d.2	$r[Rb] = \text{sign\_ext}(d.1, d.2)$	即値ストア
11	100	Rb	d.1	d.2	$\text{PC 値} = \text{PC 値} + 1 + \text{sign\_ext}(d.1, d.2)$	PC 相対アドレッシング
11	111	000	d.1	d.2	if Z==1 PC 値 = PC 値 + 1 + sign_ext(d.1, d.2) else PC 値 = PC 値 + 1	条件分岐
11	111	001	d.1	d.2	if S^V ==1 PC 値 = PC 値 + 1 + sign_ext(d.1, d.2) else PC 値 = PC 値 + 1	条件分岐
11	111	010	d.1	d.2	if Z==1    S^V==1 PC 値 = PC 値 + 1 + sign_ext(d.1, d.2) else PC 値 = PC 値 + 1	条件分岐
11	111	111	d.1	d.2	if Z==0 PC 値 = PC 値 + 1 + sign_ext(d.1, d.2) else PC 値 = PC 値 + 1	条件分岐

表 1.1: 命令一覧

## 命令セット・アーキテクチャ

前述の通り、当プロセッサは2オペランド方式を採用しています。以下に、プロセッサが搭載する命令の一覧を掲載します。

$r[Rd]$  および  $r[Rs]$  は、Rd ないし Rs をレジスタのアドレス番地として解釈した際のレジスタに格納されている (あるいは格納される) 値を指します。また、演算子  $*(x)$  はデータメモリの  $x$  番地を指します。  $\text{sign\_ext}(x)$  は、 $x$  を 16 ビット符号付き整数に符号拡張した値です。また、条件分岐命令にて登場する条件コード  $S, Z, V$  は、直前の算術命令の演算結果に依存するフラグです。以下に各条件コードの意味の説明を行います。

**S** 直前の命令の演算結果が負ならば 1、そうでなければ 0

**Z** 直前の命令の演算結果がゼロならば 1、そうでなければ 0

**V** 演算結果が符号付き 16 ビットで表せる範囲を超えた場合 1、そうでなければ 0

以下、各命令についての詳細な説明を加えます。

略称	命令名	詳細
ADD	加算	2つのレジスタの値を加算し、結果をレジスタに格納します。
SUB	減算	レジスタ値同士での減算を行い、結果をレジスタに格納します。
AND	論理積	2つのレジスタの値を2進法表示した上での論理積を取ります。
OR	論理和	2つのレジスタの値を2進法表示した上での論理和を取ります。
XOR	排他的論理和	2つのレジスタの値を2進法表示した上での排他的論理和を取ります。
CMP	比較	レジスタ値同士の減算を行います。結果をレジスタに格納しません。 主に条件分岐命令のために使用されます。
MOV	移動	レジスタの値を別のレジスタに移動させます。
ADDI	即値加算	レジスタの値に命令中で指定した4ビット符号付き整数を加算し、別のレジスタに格納します。
SLL	左論理シフト	命令中に指定した値だけ左シフトし、空いた部分に0を格納します。
SLR	左循環シフト	命令中に指定した値だけ左シフトし、空いた部分にシフトアウトされたビット列を格納します。
SRL	右論理シフト	命令中に指定した値だけ右シフトし、空いた部分に0を格納します。
SRA	右算術シフト	命令中に指定した値だけ右シフトし、空いた部分に符号ビットを格納します。
IN	入力	16個のDIPスイッチに入力されている値を16ビット整数値と解釈し、指定されたレジスタに格納します。
OUT	出力	指定されたレジスタの値を16進法4桁の値に変換し、7SEG LEDに出力します。
HALT	停止	プロセッサを停止させます。
LD	ロード	レジスタにデータメモリの値をロードします。
ST	ストア	ロードするメモリの番地は、レジスタ値と即値の加算により指定します。 データメモリにレジスタの値を格納します。
LI	即値ロード	命令中に指定した即値をレジスタに格納します。
B	無条件分岐	命令中に指定した即値だけプログラムカウンタ値を進捗させます。
BE	条件分岐	直前の演算命令の演算結果が0ならば、指定された即値の分だけジャンプします。
BLT	条件分岐	直前の演算命令の演算結果が負ならば、指定された即値の分だけジャンプします。
BLE	条件分岐	直前の演算命令の演算結果が正でないならば、指定された即値の分だけジャンプします。
BNE	条件分岐	直前の演算命令の演算結果が0でないならば、指定された即値の分だけジャンプします。

表 1.2: 命令セットの詳細一覧

## 演算・入出力命令

演算・入出力命令は、以下のような形式で表されます。

ビット数	16 15	14 13 12	11 10 9	8 7 6 5	4 3 2 1
命令値	11	Rs	Rd	op3	d

**算術演算** ADD, SUB はそれぞれ、レジスタ Rd とレジスタ Rs の演算結果をレジスタ Rd に格納します。

**論理演算** AND, OR, XOR はそれぞれ、1ビットごとにレジスタ Rd とレジスタ Rs の論理積、論理和、排他的論理和を計算し、結果をレジスタ Rd に格納します。条件コード C はいかなる場合においても 0 となります。

**比較演算** CMP はレジスタ Rd からレジスタ Rs を減算しますが、演算結果はどこにも格納せず、条件コードのみを出力します。主に条件分岐命令のために用いられます。

**移動演算** MOV はレジスタ Rd にレジスタ Rs の値を格納します。条件コード C はいかなる場合においても 0 に設定されます。

**即値加算** ADDI はレジスタ Rs の値に即値 d を符号拡張した値を加算し、レジスタ Rd に格納します。

シフト演算 SLL, SLR, SRL, SRA はレジスタ Rs の値をシフトし、レジスタ Rd に格納します。シフトの詳細を以下に述べます。

- SLL : 左シフトし、空いた部分に 0 を入れます。
- SLR : 左シフトし、空いた部分にシフトアウトされた部分列を入れます。
- SRL : 右シフトし、空いた部分に 0 を入れます。
- SRA : 右シフトし、空いた部分に符号ビットの値を入れます。

入力 IN は FPGA ボード上の DIP スイッチを 16 ビット値と解釈し、レジスタ Rd に格納します。DIP スイッチは 8 個 × 2 存在し、上の DIP スイッチ 8 個が 16 桁目～9 桁目に、下の DIP スイッチ 8 個が 8 桁目～1 桁目に対応します。

出力 OUT はレジスタ Rs の値を、7SEG LED に出力します。出力する場所は左端の最上部 4 つで、2 進法 16 ビット値を 16 進法 4 桁の値に変換して出力します。

停止 HALT は実行されるとその時点でプロセッサを停止させます。

#### ロード・ストア命令

ロード・ストア命令は、以下のような形式で表されます。

ビット数	16 15	14 13 12	11 10 9	8 7 6 5 4 3 2 1
命令値	op1	Ra	Rb	d

ロード LD は、レジスタ Rb の値と即値 d を符号拡張した値を計算し、計算結果の値をアドレスと解釈した上でデータメモリにアクセスし、その場所に格納されていた値をレジスタ Ra に格納します。op1 は 00 です。

ストア ST は、レジスタ Rb の値と即値 d を符号拡張した値を計算し、計算結果の値をアドレスと解釈した上でデータメモリにアクセスし、その場所にレジスタ Ra の内容を格納します。op1 は 01 です。

#### 即値ロード・無条件分岐命令

即値ロード・無条件分岐命令は、以下のような形式で表されます。

ビット数	16 15	14 13 12	11 10 9	8 7 6 5 4 3 2 1
命令値	10	op2	Rb	d

即値ロード 即値 d を符号拡張した値を、レジスタ Rb に格納します。op2 は 000 です。

無条件分岐 即値 d を符号拡張した値を、プログラムカウンタ値に加算します。op2 は 100 です。

ビット数	16 15	14 13 12	11 10 9	8 7 6 5 4 3 2 1
命令値	10	111	cond	d

## 条件分岐命令

条件分岐命令は、上記のような形式で表されます。

**BE** BE は直前の命令の演算結果の条件コード Z が 1 ならば、即値 d を符号拡張した値をプログラムカウンタ値に加算します。cond は 000 です。この条件分岐は、直前の命令の演算結果が 0 ならば発生します。

**BLT** BLT は直前の命令の演算結果の条件コード S と V の排他的論理和が 1 ならば、即値 d を符号拡張した値をプログラムカウンタ値に加算します。cond は 001 です。この条件分岐は、直前の命令の演算結果が負ならば発生します。

**BLE** BLE は直前の命令の演算結果の条件コード Z が 1 または S と V の排他的論理和が 1 ならば、即値 d を符号拡張した値をプログラムカウンタ値に加算します。cond は 010 です。この条件分岐は、直前の命令の演算結果が正でないならば発生します。

**BNE** BNE は直前の命令の演算結果の条件コード Z が 0 ならば、即値 d を符号拡張した値をプログラムカウンタ値に加算します。cond は 011 です。この条件分岐は、直前の命令の演算結果が 0 でないならば発生します。

## 構造と動作

### 命令の動作

当プロセッサの命令は完了に 5 ステージを要します。各ステージの動作は以下の通りです。

1. 命令フェッチフェーズ：命令メモリからプログラムカウンタ値の指す番地の命令をフェッチします。
2. デコード：命令を解釈し、必要に応じてレジスタからの値のロードを行います。
3. 演算：算術演算・シフト演算および、分岐命令においてはプログラムカウンタ値への加算を行います。
4. メモリアクセス：必要であればデータメモリにアクセスし、データのロードおよびストアを行います。
5. 汎用レジスタ書き込み：汎用レジスタにデータを書き込みます。

### 命令の並列実行

当プロセッサは先述の 5 フェーズのうち、フェーズ 1 とフェーズ 5 の同時実行を行います。すなわち、1 命令あたりおよそ 4 クロックで実行することが出来ます。フェーズ 1 とフェーズ 5 の同時実行においてパイプラインハザードは生じないため、当プロセッサはいかなる命令も逐次的に実行します。

## 実際の動作

FPGA ボード上にプロセッサをダウンロードすると、命令メモリの 1 番地から順番に命令の実行が開始されます。halt 命令によりプロセッサ全体を停止させることが出来るので、命令メモリの適切な場所に halt 命令を格納することを推奨します。