

## 原始プログラムに必要な命令セットについて

※この説明は、東京工業大学での大学院授業課題の回答レポートを元に作成した。元となったレポートを作成した加藤大智氏(2013年度計算工学修士)に感謝する。

### 1. 命令セットの定義

原始プログラムに必要な(真に最低限ではないが)と思われる命令列を以下に定義する。なお、レジスタ番号の指定が必要な命令にはレジスタ番号が#1となるもののみ記述する。#2~#4については同様に使用できる。(定数レジスタは参照のみとする。)

命令に続く部分(例: get#1 [1]の[1])を目的語と呼ぶ。目的語の値は、目的語が即値ならばその値、番地ならその番地のメモリまたはレジスタの値、番地をレジスタ番号で表している(例: [#1])ならそのレジスタに保存された番地のメモリの値である。定義の前に\*を付した命令は目的語をとらない。

命令	定義
get#1	目的語の値をレジスタ#1に入れる
put#1	目的語の値を、レジスタ#1の最下位1ビットにする
add#1	レジスタ#1の値に目的語の値を加える
sub#1	レジスタ#1の値から目的語の値を引く (注: #1 < 目的語の値 のときは0とする)
mult2#1	*レジスタ#1の値を2倍にする
div2#1	*レジスタ#1の値を2で割る
and#1	レジスタ#1と目的語の値のビットごとのANDを#1に求める
or#1	レジスタ#1と目的語の値のビットごとのORを#1に求める
xor#1	レジスタ#1と目的語の値のビットごとの排他論理和を#1に求める
not#1	*レジスタ#1のビットを反転させる
0eq?#1	*レジスタ#1の値が0であるときは次の行へ、そうでないときは、その1つ先の行へ飛ぶ
true?#1	*レジスタ#1の最下位ビットが1であるときは次の行へ、そうでないときは、その1つ先の行へ飛ぶ
goto	目的語の値の番号の行へ飛ぶ
halt	停止命令。目的語の値を出力して正常終了する

## 2. MIPS の命令セットとの比較

命令セットがこれで十分であることを確認するため、一般のコンピュータで使われている機械語の命令セット<sup>1</sup>である MIPS と比較する。比較した結果が下の表である。MIPS で使われる命令の多くが原始計算機でも実行可能なことがわかる。ただし、MIPS と異なり次のような制約がある。

- MIPS アーキテクチャではメモリから値をワード単位で読みだすことが可能だが、原始計算機では1ビットずつしか読みだすことができない。しかし、メモリの値を1ビットずつレジスタに読み出し、その値を演算に用いることで MIPS と同等の計算を行うことができる。
- MIPS 命令セットでは目的語が即値かアドレスかで命令が異なるが、原始計算機では同一なので下表では一つにまとめた。
- 原始計算機では符号付き整数および小数（浮動小数点表記）を直接は利用できない。したがって、それらを用いる MIPS の命令は用いることができない。（ただし、小数に関する演算を行うプログラムを作成することで、これらの命令を模倣することはできる。）
- jal, jr, jalr は対応する命令が存在しない。これは、これらの命令が原始計算機では不可能なプログラムカウンタの直接制御を行っているためである。これらの命令は関数呼び出しを実現するために存在しているので、関数呼び出しが不可能な原始計算機ではそもそも使う必要がない。

MIPS	原始計算機	MIPS	原始計算機
addu, addiu	add	beq	xor->0eq?
subu	sub	bne	xor->not->0eq?
and, andi	and	j	goto
nor	or の後 not	jal	なし
or, ori	or	jr	なし
xor, xori	xor	jalr	なし
sll, sllv	mult2 を繰り返し	lb, lbu, ...	get
srl, srlv	div2 を繰り返し	sb, sh, sw	put
sltu, sltiu	sub->0eq?		

<sup>1</sup> <http://www.cs.uregina.ca/Links/class-info/201/SPIM-IO/MIPSrefCard.pdf>