アーキテクチャ拡張仕様書 (Group 18)

1029-28-9483 勝田 峻太朗　　1029-28-1547 住江 祐哉

2018 年 5 月 31 日

目 次

拡張機能の仕様 **2**

拡張機能の使用例 **2**

拡張機能の仕様

拡張機能として, 即値演算命令を 3 つ実装した.

Table 1: 拡張命令

ニモニック	タイプ	op1	rs	rd	d	操作
ADDI	拡張命令	10	001	input	input	`r[rd] = r[rd] + sign_ext(d)`
SUBI	拡張命令	10	010	input	input	`r[rd] = r[rd] + sign_ext(d)`
CMPI	拡張命令	10	011	input	input	`r[rd] - sign_ext(d)`

これは, 命令解釈部で,ALU にレジスタの値ではなく即値を渡すように改良することで実行した.

拡張機能の使用例

lst. 1 のコードは, メモリ中の値をバブルソートによって昇順ソートするアセンブリコードである.

ここで,

```
SUBI R[1] 00000001
ADDI R[2] 00000001
```

の操作は, 拡張命令がない場合, 以下のようになってしまう.

```
#load 1 from memory
LD R[4] R[0] xxxxxxxx
SUBI R[1] 00000001
ADDI R[2] 00000001
```

これが 1 回なら大して違いはないが, レジスタの数は少ないので, この操作を何度も実行しなければなる可能性もある. これは, 命令数を増加させるし, プログラマの負担にもつながる.

即値命令によって, このような負担をなくすことができており, この命令は有用である.

Listing 1 バブルソート

```
#===PREPARATION===#
#R[7]:START
LD R[7] R[0] 00000000
#R[1]:LOOP_MAX
LD R[1] R[0] 00000001
#R[2]:INDEX
MOV R[7] R[2]
#===PREPARATION===#
SUBI R[1] 00000001
ADDI R[2] 00000001
#===SWAP===#
LD R[5] R[2] 00000000
LD R[6] R[2] 00000001
#IF_R[5]_<_R[6],_SKIP_STORE
ST R[5] R[2] 00000000
ST R[6] R[2] 00000001
CMP R[6] R[5]
BLT 00000010
ST R[5] R[2] 00000001
ST R[6] R[2] 00000000
#===SWAP===#
#IF_INDEX_<_MAX,_GO_BACK_TO_ADDI_(-11)
CMP R[1] R[2]
BLT 11110101
MOV R[7] R[2]
#IF_MAX_!=_0_GO_BACK_TO_SUBI_(-15)
CMP R[1] R[7]
BNE 11110001
HLT
```