

2021/4/19

コンピュータアーキテクチャ論_Ex02

s1260027

Shunsuke Onuki

課題：MIPS プログラミング（乗算）

MIPS プログラミングとして符号なし乗算を行うプログラムを作る。乗数 A、被乗数 B、積 C は 32 ビット。乗数と被乗数は最大 16 ビットまでを仮定するし、求める乗算結果は 64 ビットではなく、下位の 32 ビットだけで十分である。

課題：乗算アルゴリズムの実装

MIPS プログラミングとして符号なし乗算を行うプログラムを作る。乗数 A、被乗数 B、積 C は 32 ビット。乗数と被乗数は最大 16 ビットまでを仮定するし、求める乗算結果は 64 ビットではなく、下位の 32 ビットだけで十分である。

(考え方)

被乗数を 1 ビットずつ 0/1 判定して、1 ならば積の値に乗数を加算する。

左シフトを使い加算していくことで乗算を実装する。

5(101)と 3(11)の場合、3(11)の 1 ビット目が 1 なので、積は 5(101)になる。

次に、乗数と積を左シフトで 2 ビット目に注目する。

被乗数の 2 ビット目も 1 なので、左シフトした積に左シフトした乗数を加算すると 15(1111)になる。

(プログラムとその説明)

```
.data
A:      .word 26      #被乗数
B:      .word 27      #乗数
C:      .word 0       #積

D:      .word 16      #計算する分のビット数
        .text

main:
    lw $a0, A          #被乗数の値
    lw $a1, B          #乗数の値
    lw $v0, C          #積の値

    lw $t0, D          #計算する分のビット数
    or $t1, $0, $0     #i=0
    or $t2, $0, 1      #各ビットで1/0をチェックする(チェックビット)

loop:
    slt $t3, $t1, $t0   #1<N ならば $t3=1
    beq $t3, $0 store   #t3=0 ならば store

    ###各ビットが1/0を判断
    and $t4, $a1, $t2   #$t4=$a1+$t2
    beq $t4, $0 loopend #$t4=0 ならば loopend

    add $v0, $v0, $a0   #積をたす
    j loopend

loopend:
    add $a0, $a0, $a0   #チェックビットを左シフト
    add $t2, $t2, $t2   #乗数を左シフト
    addi $t1, $t1, 1    #i++
    j loop

store:
    la $t5, C           #積のアドレス
    sw $v0, 0($t5)      #結果をCに保存

exit:
    j exit
```

(結果)

Data Segments				
	DATA			
	[0x00005000]	0x0000001a	0x0000001b	0x0000002be 0x00000010
	[0x00005010]... [0x00025000]	0x00000000		
	STACK			
	[0x7ffffeffc]	0x00000000		
	[0x7ffff000]... [0x80000000]	0x00000000		