2020/5/18

コンピュータアーキテクチャ論_Ex2

s1260027　　Shunsuke Onuki

課題：乗算アルゴリズムの実装

乗算をおこなうプログラムを作る。 乗数 A、被乗数 B、積 C は 32 ビット。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。 つまり、乗数と被乗数は最大 16 ビットまでを仮定する。

課題：乗算アルゴリズムの実装

乗算をおこなうプログラムを作る。 乗数 A、被乗数 B、積 C は 32 ビット。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。 つまり、乗数と被乗数は最大 16 ビットまでを仮定する。

（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 13    #乗数
B:      .word 37    #被乗数
C:      .word 0     #積

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

main:
        lw $t1, A           #乗数の値
        lw $t2, B           #被乗数の値
        lw $t3, C           #積の値
        lw $t4, D           #計算する分のビット数
        or $t5, $0, $0        #i=0
        addi $t6, $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 $t7, $t5, $t4   #1<N ならば $t7=1
        beq $t7, $0 store #$t7=0 ならば loopend

        ###各ビットが1/0を判断
        and $t8, $t2, $t6   #$t8=$t2+$t6
        beq $t8, $0 loopend #$t8=0 ならば loopend

        add $t3, $t3, $t1   #積をたす
        j loopend

loopend:
        addu $t6, $t6, $t6  #チェックビットを左シフト
        addu $t1, $t1, $t1  #乗数を左シフト
        addi $t5, $t5, 1    #i++
        j loop

store:
        la $t9, C           #積のアドレス
        sw $t3, 0($t9)      #結果をCに保存

exit:
        j exit
```

（結果）

Data Segments

```
        DATA
[0x00005000]                    0x0000000d  0x00000025  0x000001e1  0x00000010
[0x00005010]...[0x00025000]     0x00000000

        STACK
[0x7fffeffc]                    0x00000000
[0x7ffff000]...[0x80000000]     0x00000000
```