

7.使用逻辑运算实现32位乘法

1.实验原理

先看一个实例：1011*1010，分为两个运算
1011*0010，相当于左移一位得到10110
1011 1000相当于左移三位则为1011000
故两者乘积为10110与1011000之和1101110。
因而乘法可以通过一系列移位和加法完成

2.代码部分

```
.data
#定义输出时候所用的字符
#本程序采用5A286752H和6E12BA98相乘进行测试#
$VAR0: .ascii  "5A286752H*6E12BA98=\000"
$VAR1: .ascii  "%x\n\000"

.text
.globl main
.ent main
main:
#入栈保护
addiu   $sp,$sp,-32
sw      $ra,28($sp)
sw      $fp,24($sp)
move    $fp,$sp
#输出 "5A286752H*6E12BA98="
la      $a0,$VAR0
jal     printf

#装入测试的数据
li      $a0,0x5a286752
li      $a1,0x6e12ba98

#运算初始化开始
#t2用来记录当前位的序号初始为第0位
#t3是用来辅助取出当前位到t1
#t6是循环条件，次数为32
move    $t0,$a0
move    $t1,$a1
li      $t2,0
li      $t3,31
li      $t6,32

L2: move    $t1,$a1
subu    $t4,$t3,$t2
#取出a1当前的t2位。先左移，后右移。最低位为1保留
sll     $t1,$t1,$t4
sr1     $t1,$t1,$t3
```

```

#注意都要右移因为最后只保留高32位
srl $t5,$t5,1
beq  $t1,$0,L3
#只要取出的结果不为0, 就相加
addu  $t5,$t0,$t5

```

L3:

```

#一共有32位要做, 只要t2计数器未到32就继续L2。
addiu $t2,$t2,1
bltu  $t2,$t6,L2
#由于截取高32位, 遇到0, 相当于在结果t5前加个0
srl $t5,$t5,1
#结果放入v0
move  $v0,$t5

#输出结果
move  $a1,$v0
la  $a0,$VAR1
jal  printf

#出栈恢复
move  $sp,$fp
lw  $ra,28($sp)
lw  $fp,24($sp)
addiu $sp,$sp,40
j  $ra

.end  main

```

2.测试结果

测试过程采用5A286752H和6E12BA98相乘进行测试

```

zheng@ubuntu:~$ mips-linux-gnu-gcc -static -g mul.s -o mulfinal1
zheng@ubuntu:~$ mips-linux-gnu-qemu mulfinal1
5A286752H*6E12BA98=26c3f4f3

```

这与计算器验证结果是一致的!