mips设计文档

李昊宸 计算机科学与技术 2017K8009929044

int main()

{

int i = 1;

int sum = 0;

do

{

sum = sum + i;

i = i + 1;

} while (i <= 100);

printf(“sum = %d\n”,sum);

return 0;

}

将以上代码修改为MIPS汇编实现

.section .data

output: .ascii "sum = " #输出结果的文字段

.section .text

.globl \_\_start

\_\_start:

addiu $v0, $0, 4004 #写文件调用

addiu $t0, $0, 1

addiu $t1, $0, 1 #t1为每次循环中增量的部分

addiu $t2, $0, 0 #t2存放部分和

addiu $t3, $0, 101 #t3寄存器存放循环次数

addiu $t4, $0, 10 #t4存放立即数10，用于将

部分和转换成ascii码的形式

addiu $t6, $0, 4 #L2循环次数

L1:

add $t2, $t2, $t1 #sum = sum + i

addiu $t1, $t1, 1 #i = i + 1

bne $t1, $t3, L1 #i < 101？

nop

la $a1, output #sum =占5个字节

addiu $a1, $a1, 9 #数字5050占4个字节

L2:

divu $t2, $t4 #mod运算，逐位将sum变为

#ascii形式

mfhi $t5 #[HI] = GPR[t2] mod GPR[t4]

mflo $t2 #[LO] = GPR[t2]/GPR[t4]

addiu $t5, $t5, 48 #ascii化

sb $t5, ($a1) #存到output后面

addiu $a1, $a1, -1 #地址向前移动

addiu $t6, $t6, -1

bne $t6, $0, L2

nop

addiu $a1, $a1, -5 #地址回归

addiu $a2, $0, 9 #输出字节数

syscall

addiu $v0, $0, 4001 #退出

addiu $a0, $0, 0

syscall

运行结果

