



计算机组成原理实验报告

姓名： 卫焱滨

学号： 11710823

一、实验内容

1. Exchange the highest 8 bits with the lowest 8 bits in a word.
2. Calculate the bit inversion (0->1,1->0) of the odd digits in a word.
3. For an integer x, calculate the result of 10x. DO NOT use mult/mul/multu in your code.
4. Calculate the absolute value of a word by basic operations other than abs.

二、实验步骤（阐述代码思路或操作步骤）

1.

题意分析：

本题要求对于用户输入的一个 word，交换其最高八位和最低八位

操作思路和步骤：

1. 以十进制形式读入用户输入，并存入内存中，用寄存器（t5）保存这块内存的地址
2. 为了方便与处理后的结果比较，先将其以 16 进制展示打印出来
3. 实现交换：
通过基线寻址（3（\$t5）表示最高字节在内存中地址，（\$t5）表示最低字节的地址），
将内存中数据的最高 byte 和最低 byte 分别 lb 到寄存器内，再重新 sb 到交换后的地址
4. 将交换后的结果同样以 16 进制的形式打印输出

代码：

```
.include "macro_print_str.asm"

.data
    num:.word 0
.text
    main:
        #t5 is address of wrd
        la $t5 num

        #read the integer
        print_string("Please input an Integer :")
        li $v0 5
        syscall

        #print the hexadecimal form of the original integer
        sw $v0 ($t5)
        print_string("Hexadecimal form before exchange :")
        li $v0 34
        lw $a0 ($t5)
```

```
syscall

#exchage the lowest byte and the highest byte
lb $t0 ($t5)
lb $t1 3($t5)
sb $t0 3($t5)
sb $t1 ($t5)

#print the result after exchange
print_string("\nHexadecimal form after exchange :")
li $v0 34
lw $a0 ($t5)
syscall

end
```

3.

题意分析：

本题要求将 word 的奇数位取反

操作步骤与思路：

1. 以十进制形式读入用户输入，并存入内存中，用寄存器（t5）保存这块内存的地址
2. 为了方便与处理后的结果比较，先将其以 2 进制展示打印出来
3. 实现奇数位取反：对于某一位来说，异或 1 该位取反，异或 0 该位不变，因此可以构造一个 word 变量，其奇数位为 1，偶数位为 0，即 0xAAAAAAAA。将读入的数据异或此变量即可实现奇数位取反
4. 将奇位取反后的结果以二进制形式打印输出

```
.include "macro_print_str.asm"
.data
    num: .word 0
.text
    main:
        #t5 is address of wrd
        la $t5 num

        #read the integer
        print_string("Please input an Integer : ")
        li $v0 5
        syscall

        #print the Binary form of the original integer
        sw $v0 ($t5)
```



```
print_string("Binary form before odd Inverse : ")
li $v0 35
lw $a0 ($t5)
syscall

#do the odd bits inverse
li $t0 0xAAAAAAAA
lw $t1 ($t5)
xor $t2 $t0 $t1

print_string("\nBinary form after odd Inverse : ")
li $v0 35
move $a0 $t2
syscall

end
```

3.

题意分析:

本题要求对于输入 x , 计算 $10x$

实现思路:

1. 读入数据
2. $10x = 8x + 2x = x \ll 3 + x \ll 1$, 因此求取 x 左移 3 位后的结果, 加上 x 左移一位后的和即为 $10x$
3. 输出所求 $10x$ 的值

代码:

```
.include "macro_print_str.asm"
.text
main:
#read the integer x
print_string("Please input Integer x: ")
li $v0 5
syscall

#compute  $10x = 8x + 2x = x \ll 3 + x \ll 1$ 
move $t0 $v0 #t0 is x
sll $t1 $t0 3
sll $t2 $t0 1
add $t3 $t2 $t1

#print the result 10x
print_string("The result of 10x : ")
```



```
li $v0 1
move $a0 $t3
syscall

end
```

4.

题意分析:

本题要求对于输入 x , 求其绝对值

实现思路:

1. 读入输入 x
2. 通过取反+1 获得 x 的 2 的补码, 即 $-x$ 的值。取反操作可以通过异或 0xFFFFFFFF 实现
3. 使用 `slt` 比较 x 和 0 的相对大小确定其正负, 若为正, $t1$ 为 0。否则为 1.
4. 通过 `beq` 和 `bnc` 以及设置指令标签实现分支输出功能
当 $t1$ 为 1 ($x < 0$), 跳转到语句 L1 输出 $-x$ 。
否则顺序执行语句, 输出 x 并在 L1 之前退出程序
5. 通过以上步骤。即可实现输出 x 的绝对值

代码 :

```
.include "macro_print_str.asm"
.text
main:
#read the integer x
print_string("Please input Integer x: ")
li $v0 5
syscall

#compute -x
move $t0 $v0
slt $t1 $t0 $zero
li $t2 0xFFFFFFFF
xor $t2 $t2 $t0
addi $t2 $t2 1

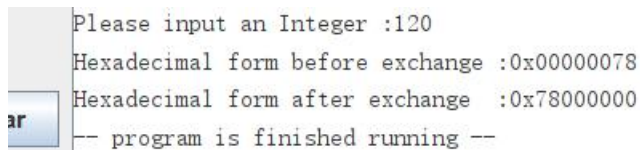
#print the abs,use beq to choose -x or x
print_string("The abs of x :")
li $v0 1
beq $t1 0 L1
move $a0 $t2
syscall
end
L1:
```

```
move $a0 $t0
syscall
end
```

三、实验结果（截图并配以适当的文字说明）

1.

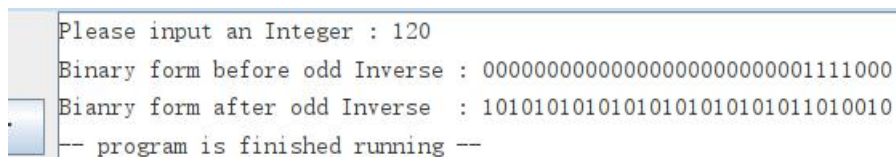
运行结果如下，若输入 120，输出为交换前后的 16 进制形式
输出包含题目要求的答案和所设定好的提示信息



```
Please input an Integer :120
Hexadecimal form before exchange :0x00000078
Hexadecimal form after exchange :0x78000000
-- program is finished running --
```

2

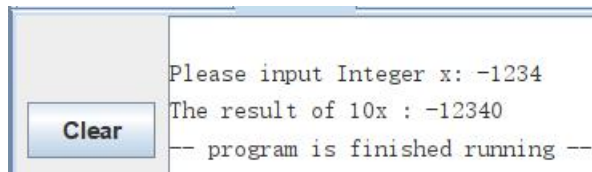
运行结果如下，若输入 120，输出为奇数位取反前后的二进制形式
输出包含题目要求的答案和提示信息



```
Please input an Integer : 120
Binary form before odd Inverse : 0000000000000000000000001111000
Binary form after odd Inverse : 1010101010101010101010101010010
-- program is finished running --
```

3

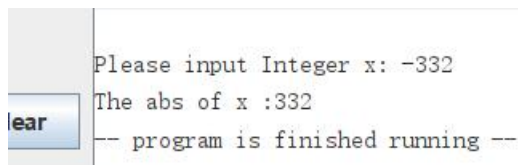
运行结果如下，输入 x，输出 10x
如输入-1234，输出为-12340



```
Please input Integer x: -1234
The result of 10x : -12340
-- program is finished running --
```

4.

运行结果如下，输入 x，输出 x 的绝对值
如输入-332，输出为 332



```
Please input Integer x: -332
The abs of x :332
-- program is finished running --
```

四、实验分析（遇到的问题以及解决方案）

问题 1：输出并非预测的值而是地址

解决：检查发现调用宏 print_string 时改变了 v0 和 a0 的值，重设后解决

五、实验小结与体会

熟悉了二进制位运算操作（左右移、异或、和等）

熟练使用了 slt、beq、bnc 实现分支操作的过程

加深了对码制（反补码运算）和 MIPS 数据类型，以及寄存器与内存的了解