1 实验目的与内容

1.1 实验目的

通过本次实验熟悉 RISC 指令集 (RV32I 指令集) 的部分指令和编码格式

1.2 实验内容

1.2.1 斐波那契数列

编写汇编程序，计算斐波那契数列的第 N 项（$0 \leq N \leq 30$ ）。初始时，N 的值保存在 R_2 中。程序执行完成后，数列的第 N 项保存在 R_3 中。

1.2.2 大整数处理

编写汇编程序，计算斐波那契数列的第 N 项（$0 \leq N \leq 80$ ）。初始时，N 的值保存在中。程序执行完成后，数列的第 N 项保存在 R_3 和 R_4 中，其中 R_3 存储结果的高 32 位，R_4 存储结果的低 32 位。

1.2.3 导出 COE 文件

完成汇编程序的编写之后，导出指令段的 COE 文件，以供后续实验使用。

2 逻辑设计

2.1 斐波那契数列

斐波那契数列计算公式为

$$F_N = \begin{cases} 1 & 1 \leq N \leq 2 \\ F_{N-1} + F_{N-2} & N > 3 \end{cases}$$

其中，N 为整数。

```
.data
    #设置N的值
    N:
      .word 2
.text
MAIN:
    #x2用于存储N的值(即R2), x1存储结果(即R3)
    lw x2,N
    #先判断N是否大于2, 若否, 则直接将结果设置为1并跳转到代码结尾
    li t1,3
    bge x2,t1,N3
    addi x1,x1,1
    bge x0,x0,END
N3:
    addi x2,x2,-2
    li t1,1
    li t2,1
    #计算斐波那契数列, 直到x2等于0时退出循环, 结果在x1中
FOR:
    add x1,t1,t2
    addi t1,t2,0
    addi t2,x1,0
    addi x2,x2,-1
    blt x0,x2,FOR
END:
```

Listing 1: 斐波那契数列

2.2 大整数处理

代码整体设计思路同上，但是由于斐波那契数列第 80 项已经大于 2^{32}，故需要两个寄存器来进行计算，分别存储结果的低 32 位和高位。

在进行低 32 位的加法计算时需要判断是否进位，判断方法为将加法结果与任意一个加数比较，若结果小于加数，则产生了进位。在进行高位加法计算时需要加上进位。

代码如下：

```
.data
    N:
      .word 69
.text
MAIN:
    #x2用于存储N的值(即R2), x3存储结果高位(即R3), x4存储结果低位
        (即R4)
    lw x2,N
    li t1,3
    bge x2,t1,N3
    addi x4,x4,1
    bge x0,x0,END
N3:
    addi x2,x2,-2
    #t1为高位, t2为低位
    li t1,0
    li t2,1
    #t3为高位, t4为低位
    li t3,0
    li t4,1
FOR:
    #低位相加
    add x4,t2,t4
    #t5存储进位
    sltu t5,x4,t2
    #高位相加
    add x3,t1,t3
    #高位加进位
    add x3,x3,t5
    addi t1,t3,0
    addi t2,t4,0
    addi t3,x3,0
    addi t4,x4,0
    addi x2,x2,-1
    blt x0,x2,FOR
END:
```

Listing 2: 大整数处理

2.3 导出 COE 文件

将代码段与数据段分别导出，并在文件开头加上如下代码：

memory_initialization_radix = 16;

memory_initialization_vector =

结果如下：

```
memory_initialization_radix  = 16;
memory_initialization_vector =
00000006
00000000
```

Listing 3: 斐波那契数列数据段

```
memory_initialization_radix  = 16;
memory_initialization_vector =
0fc10117
00012103
00300313
00615663
00108093
02005263
ffe10113
00100313
00100393
007300b3
00038313
00008393
fff10113
fe2048e3
```

Listing 4: 斐波那契数列代码段

```
memory_initialization_radix  = 16;
memory_initialization_vector =
```

```
00000050
00000000
```

Listing 5: 大整数处理数据段

```
memory_initialization_radix  = 16;
memory_initialization_vector =
0fc10117
00012103
00300313
00615663
00108093
04005063
ffe10113
00000313
00100393
00000e13
00100e93
01d38233
00723f33
01c301b3
01e181b3
000e0313
000e8393
00018e13
00020e93
fff10113
fc204ee3
```

Listing 6: 大整数处理代码段

3 总结

通过本次实验初步掌握了 RV32I 指令集部分命令的使用以及编译和运行的方法。