

PA5

2013551 雷贺奥

PA5

实验目的

实验内容

实验过程

浮点数支持

f2F函数

F_mul_F 函数

F_div_F 函数

Fabs

FLOAT 和 int 之间的相互转换

实现shrd、shld

Bug

结果 (结局)

实验目的

我们在PA3中把仙剑奇侠传运行起来了,但却不能战斗,这是因为还有一些浮点数相关的工作需要处理要在NEMU中实现浮点指令也不是不可能的事情.但实现浮点指令需要涉及x87架构的很多细节,根据KISS法则,我们选择了一种更简单的方式:我们通过整数来模拟实数的运算,这样的方法叫 binary scaling。

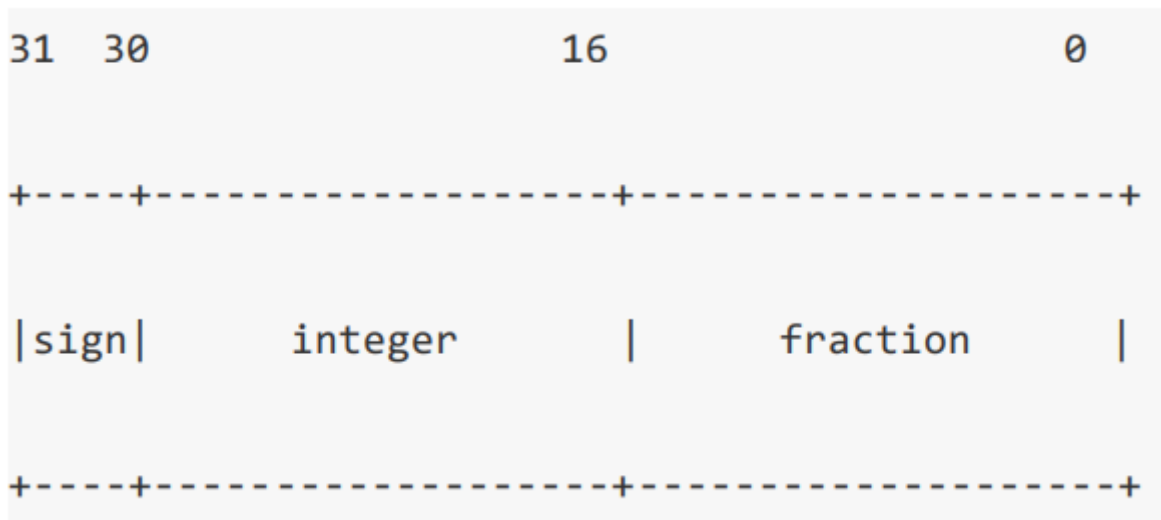
实验内容

1. 实现浮点数的支持
2. 通过整数来模拟实数的运算
3. 理解 binary scaling 的支持

实验过程

浮点数支持

用 **binary scaling** 方法表示的实数的类型为 **FLOAT**. 我们约定最高位为符号位, 接下来的 15 位表示整数部分, 低 16 位表示小数部分, 即约定小数点在第 15 和第 16 位之间(从第 0 位开始)



用float IEEE 754方法表示虽然在nemu中同样为32位，但却大不相同。如下图所示，s为标志位，exp为指数位，frac为尾数位。



因此，作者先实现了两种float表示方法的结构体，分别如下：

- float IEEE 754

```
typedef union
{
    float value;
    struct
    {
        unsigned m:23; // 尾数
        unsigned e:8;  // 指数
        unsigned s:1;  // 符号位
    };
} ieee_float;
```

- binary scaling

```
typedef union
{
    FLOAT value;
    struct
    {
        unsigned f:16;
        unsigned i:15;
        unsigned s:1;
    };
} Float;
```

f2F函数

- (1) 判断指数e是否为0，若为0，则直接返回0，表示浮点数为0。
- (2) 判断指数e是否为0xFF，若为0xFF则表示此时的浮点数为特殊数值即（NAN或Infinity），分别返回32位可以表达的最大正数0x7fffffff和可以表示的最小负数0x80000000。
- (3) 计算浮点数的偏移量，用e-127+16可得，若计算后小于0，返回0，表示无法转换为合法的整数。若大于0，计算出float的base值，用于计算浮点数有效数部分，pow需要减去23，若pow 小于 0，则将 base 右移-pow 位；否则将 base 左移 pow 位。
- (4) 代码根据符号位 值来确定最终结果的正负

```
FLOAT f2F(float a) {
    ieee_float value = {a};
    if(value.e==0)
    {
        return 0;
    }
}
```

```

}
if(value.e==0xff)
{
    //特殊值 (NaN 或 Infinity)
    return (value.s ? 0x80000000 : 0x7fffffff);
}
int pow=value.e-127+16;
if(pow<0)
{
    return 0;
}
int base=value.m|(1<<23);
pow-=23;
return (value.s ? -1:1)*(pow<0 ? (base>>-pow):(base<<pow));
}

```

F_mul_F 函数

(1) 由于指导书上说不用考虑溢出，只要将结果除以 2^{16} ，就能得出正确的结果了。

```

FLOAT F_mul_F(FLOAT _a, FLOAT _b) {
    return ((uint64_t)_a*_b)>>16;
}

```

F_div_F 函数

(1) 确保除数 b 不为零

(2) 分别计算被除数的绝对值 x 和除数的绝对值 y。

(3) 之后进入 for 循环，用于执行浮点数的小数部分的除法计算。在每次循环中，将 x 左移一位，将 ret 左移一位，然后检查 x 是否大于等于 y。若 $x > y$ ，ret 即可 +1

```

FLOAT F_div_F(FLOAT a, FLOAT b) {
    assert(b != 0);
    FLOAT x = Fabs(a);
    FLOAT y = Fabs(b);
    FLOAT ret = x / y;
    x = x % y;
    //处理低16位
    for (int i = 0; i < 16; i++) {
        x <<= 1;
        ret <<= 1;
        if (x >= y) {
            x -= y;
            ret++;
        }
    }
    if (((a ^ b) & 0x80000000) == 0x80000000) {
        ret = -ret;
    }
    return ret;
}

```

Fabs

```
FLOAT Fabs(FLOAT a) {  
    return a<0 ? -a:a;  
}
```

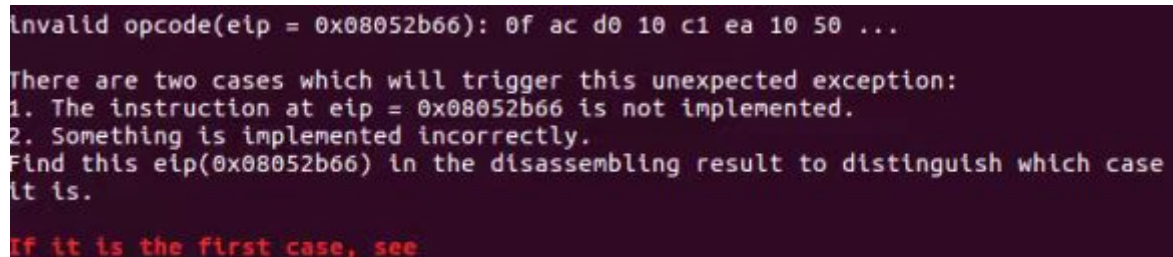
FLOAT 和 int 之间的相互转换

都可以直接操作，这里不再赘述

```
static inline int F2int(FLOAT a) {  
    //assert(0);  
    //FLOAT = a*2^16  
    return a>>16;  
}  
  
static inline FLOAT int2F(int a) {  
    //assert(0);  
    return a<<16;  
}  
  
static inline FLOAT F_mul_int(FLOAT a, int b) {  
    //assert(0);  
    //可以直接相乘  
    return a*b;  
}  
  
static inline FLOAT F_div_int(FLOAT a, int b) {  
    //assert(0);  
    //可以直接相除  
    return a/b;  
}
```

实现shrd、shld

make run 报错，需要实现shrd、shld指令。



```
Invalid opcode(eip = 0x08052b66): 0f ac d0 10 c1 ea 10 50 ...  
  
There are two cases which will trigger this unexpected exception:  
1. The instruction at eip = 0x08052b66 is not implemented.  
2. Something is implemented incorrectly.  
Find this eip(0x08052b66) in the disassembling result to distinguish which case  
it is.  
  
If it is the first case, see
```

- shrd

```
make_EHelper(shrd) {  
    rtl_shr(&t0, &id_dest->val, &id_src->val);  
    if (decoding.is_operand_size_16) {  
        rtl_addi(&t1, &tzero, 16);  
    } else {  
        rtl_addi(&t1, &tzero, 32);  
    }  
    rtl_sub(&t1, &t1, &id_src->val);  
    rtl_shl(&t2, &id_src2->val, &t1);
```

```

rtl_or(&t0, &t0, &t2);
operand_write(id_dest, &t0);
rtl_update_ZFSF(&t0, id_dest->width);
print_asm_template2(shrd);
}

```

- shld

```

make_EHelper(shld) {
    rtl_shl(&t0, &id_dest->val, &id_src->val);
    if (decoding.is_operand_size_16) {
        rtl_addi(&t1, &tzero, 16);
    } else {
        rtl_addi(&t1, &tzero, 32);
    }
    rtl_sub(&t1, &t1, &id_src->val);
    rtl_shr(&t2, &id_src2->val, &t1);

    rtl_or(&t0, &t0, &t2);
    operand_write(id_dest, &t0);

    rtl_update_ZFSF(&t0, id_dest->width);

    print_asm_template2(shld);
}

```

Bug

实现完float，发现报错，提示shld指令没有实现，群里也有同学有相似的问题，在实现之后，pa就可以正常运行了

结果（结局）

