第四章

4.47

(1)

```
void bubble_p(long *data,long count){
    long *i,*last;
    for(last = data+count -1;last > data;last--){
        for(i = data;i < last;i++){
            if(*(i+1) < *i){
                long t = *(t+1);
                *(t+1) = *i;
                *i = t;
            }
        }
    }
}
```

(2)

```
.pos 0
    irmovq stack,%rsp
    call main
    halt

.align 8
data:
    .quad 0x0000000000000004
    .quad 0x0000000000000003
    .quad 0x0000000000000002

data_end:
    .quad 0x0000000000000001

main:
    irmovq data,%rdi
    call ysBubbleP
    ret

ysBubbleP:
    jmp L2
L4:
    mrmovq 8(%rax),%r9
    mrmovq (%rax),%r10
    rrmovq %r9,%r8
```

```
    subq %r10,%r8
    jge L3
    rmmovq %r10, 8(%rax)
    rmmovq %r9,(%rax)
L3:
    irrmovq %8,%r8
    addq %r8,%rax
    jmp L5
L6:
    rrmovq %rdi,%rax
L5:
    rrmovq %rsi,%r8
    subq %rax,%r8
    jg L4
    irmovq $8,%r8
    subq %r8,%rsi
L2:
    rrmovq %rsi,%r8
    subq %rdi, %r8
    jg L6
        ret

.pos 0x200
stack:
```

4.51

取指	icode:ifun = M1[PC]; rA:rB = M1[PC+1]; valC = M8[PC+2]; valP = PC + 10;
译码	valB = R[rB]
执行	ValE = valB + valC; set CC
访存	/
写回	R[rb] = valE
更新 PC	PC = valP

4.55

1. 预测错误情况：

将`(E_icode == IJXX && !e_Cnd)`

改为`(E_icode == IJXX && E_ifun != UNCOND && e_Cnd)`

2. 修改跳转，对所有的 jxx 操作

将 `M_icode == IJXX && !M_Cnd : M_valA;`

改为 `M_icode == IJXX && M_ifun != UNCOND && M_Cnd : M_valA;`

4.59

4.47 的性能更好.

50%的 jge 指令会跳转成功，对于跳转成功的循环会执行 9 条指令，失败的会执行 5 条指令，平均每个循环执行 7 条指令。
4.48 无 jge，每个循环执行 9 条，
4．49 无 jge，每个循环执行 11 条。