体系结构 第五次作业

PB18000221 袁一玮

1

输出相关：S1-S3 A[i]

反相关：S1-S3 A[i], S1-S2 B[i], S2-S3 A[i], S3-S4 C[i]

真相关：S1-S2 A[i], S3-S4 A[i]

展开二级循环，并不会引入相关

S	Instr
S1	A[i] = A[i] * B[i]
S2	B[i] = A[i] + c
S3	A[i] = C[i] * c
S4	C[i] = D[i] * A[i]
S1'	A[i+1] = A[i+1] * B[i+1]
S2'	B[i+1] = A[i+1] + c
S3'	A[i+1] = C[i+1] * c
S4'	C[i+1] = D[i+1] * A[i+1]

修改变量名如下

S	Instr
S1	A1[i] = A[i] * B[i]
S2	B1[i] = A1[i] + c
S3	A2[i] = C[i] * c
S4	C[i] = D[i] * A2[i]

2

a

执行 6 次浮点运算，读 4 个浮点数，写 2 个浮点数，访问 $(4+2) \times 4 = 24$ 个字节

内核运算密度为 $\frac{6}{(4+2)\times 4} = \frac{1}{4}$

b

```
        li          $VL, 44
        li          $r1, 0
loop:   lv          $v1, a_re + $r1 ; load a_re
        lv          $v3, b_re + $r1 ; load b_re
        mulvv.s     $v5, $v1, $v3   ; a_re * b_re
        lv          $v2, a_im + $r1 ; load a_im
        lv          $v4, b_im + $r1 ; load b_im
        mulvv.s     $v6, $v2, $v4   ; a_im * b_im
        subvv.s     $v5, $v5, $v6   ; a_re * b_re - a_im * b_im
        sv          $v5, c_re + $r1 ; store c_re
        mulvv.s     $v5, $v1, $v4   ; a_re * b_im
        mulvv.s     $v6, $v2, $v3   ; a_im * b_re
        addvv.s     $v5, $v5, $v6   ; a_re * b_im + a_im * b_re
        sv          $v5, c_im + $r1 ; store c_im
        bne         $r1, 0, else
        addi        $r1, $r1, #44   ; loop increment by 44
        j           loop
else:   addi        $r1, $r1, #256  ; loop increment by 256
skip:   blt         $r1, 1200, loop
```

c & d

```
mulvv.s     lv
lv          mulvv.s
subvv.s     sv
mulvv.s     lv       ; load next vector
mulvv.s     lv       ; load next vector
addvv.s     sv
```

6 次，每次需要周期为

$$\frac{64 \times 6 + 15 \times 6 + (8 \times 4 + 5 \times 2)}{2 \times 64} = \frac{129}{32} = 4.03\text{clock}$$

3

a

$$1.5\text{GHz} \times 80\% \times 85\% \times 70\% \times 10 \times 8 = 57.12\text{GFLOP/s}$$

b

$$\text{S}_1 = \frac{1.5\text{GHz} \times 80\% \times 85\% \times 70\% \times 10 \times 16}{57.12\text{GFLOP/s}} = \frac{114.24\text{GFLOP/s}}{57.12\text{GFLOP/s}} = 2$$

$$\mathrm{S}_2 = \frac{1.5\mathrm{GHz} \times 80\% \times 85\% \times 70\% \times 15 \times 8}{57.12\mathrm{GFLOP/s}} = \frac{85.68\mathrm{GFLOP/s}}{57.12\mathrm{GFLOP/s}} = 1.5$$

$$\mathrm{S}_3 = \frac{1.5\mathrm{GHz} \times 80\% \times 95\% \times 70\% \times 10 \times 8}{57.12\mathrm{GFLOP/s}} = \frac{63.84\mathrm{GFLOP/s}}{57.12\mathrm{GFLOP/s}} = \frac{19}{17} = 1.11$$

4

$$1.5\mathrm{GHz} \times 16 \times 16 = 384\mathrm{GFLOP/s}$$

每个单精度运算需要读 2 个操作数，写 1 个操作数，访问 $(2+1) \times 4 = 12$ 个字节，需要 $12\mathrm{Byte} \times 384\mathrm{GFLOP/s} = 4608\mathrm{GB/s}$ 比存储器的带宽大，因此吞吐量不可持续。