

# Homework 5 Solution

一、

(1) 程序执行的 CPI=没有分支的基本 CPI(1)+分支带来的额外开销

分支带来的额外开销是指在分支指令中, 缓冲命中但预测错误带来的开销与缓冲没有命中带来的开销之和

分支带来的额外开销=15%\* (90%命中\*10%预测错误\*4+10%没命中\*3) =0.099

CPI = 1+0.099 = 1.099

(2)采用固定的 2 个时钟周期延迟的分支处理 CPI=1+15%\*2=1.3

由(1)(2)可知分支目标缓冲方法执行速度快

二、

(1) 每 6 个 FLOP, 读 4 个浮点数, 写 2 个浮点数, 共访问 24 个字节。

运算密度  $6/(6 * 4) = 0.25$

(2) 将长度为 300 的向量分解为长度为 44 和 4 个长度 64 的向量。第一次计算长度为 44 的向量, 之后长度均为 64。

```

        li        $V1,44          # perform the first 44 ops
        li        $r1,0           # initialize index
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       # check if first iteration
        addi      $r1,$r1,#44      # first iteration,
                                   # increment by 44
        j loop      # guaranteed next iteration
else:    addi      $r1,$r1,#256     # not first iteration,
                                   # increment by 256
skip:    blt      $r1,1200,loop    # next iteration?
```

(3)

```
1.    mulvv.s    lv      # a_re * b_re (assume already
                        # loaded), load a_im
2.    lv         mulvv.s # load b_im, a_im*b_im
3.    subvv.s    sv      # subtract and store c_re
4.    mulvv.s    lv      # a_re*b_im, load next a_re vector
5.    mulvv.s    lv      # a_im*b_re, load next b_re vector
6.    addvv.s    sv      # add and store c_im
```

6 chimes

本题中的  $T_a$  计算较为复杂，不算分。套用公式即可

$$\left\lceil \frac{n}{MVL} \right\rceil (T_{loop} + T_{start}) + n * T_{chime}$$

(4)

```
1. mulvv.s          # a_re*b_re
2. mulvv.s          # a_im*b_im
3. subvv.s  sv      # subtract and store c_re
4. mulvv.s          # a_re*b_im
5. mulvv.s  lv      # a_im*b_re, load next a_re
6. addvv.s  sv  lv  lv  lv # add, store c_im, load next b_re,a_im,b_im
```

三、

(1)  $1.5 * 16 * 16 = 384$  GFLOPS/s

(2) 维持吞吐量需要  $12 \text{ bytes/FLOP} \times 384 \text{ GFLOPs/s} = 4.6 \text{ TB/s}$  带宽，给定的存储器带宽不满足，所以吞吐量不能持续

四、补充习题

假设程序中无条件转移指令的比例为 5%，没有无条件转移指令的基础程序 CPI=1，未设置分支目标缓冲时 CPI=1.1。现增设分支目标缓冲，命中率为 90%，假设分支目标缓冲中包含分支目标指令，允许无条件转移指令进入分支目标缓冲，则 CPI 的值为多少？（只考虑程序包含无条件转移指令，不需要考虑其他分支指令）

设每条无条件转移指令的延迟为  $x$ ，则有

$$1 + 5\%x = 1.1$$

$$x = 2$$

当分支目标缓冲命中时，无条件转移指令的延迟为 0

$$\text{所以程序的 CPI} = 1 + 2 * 5\% * (1 - 90\%) = 1.01$$