第五章

5.15

```c
void inner4(ver_ptr u, vec_ptr v, data_t *dest)
{
    long i;
    long length = vec_length(u);
    data_t *udata = get_vec_start(u);
    data_t *vdata = get_vec_start(v);
    data_t s1 = (data_t)0;
    data_t s2 = (data_t)0;
    data_t s3 = (data_t)0;
    data_t s4 = (data_t)0;
    data_t s5 = (data_t)0;

    for (i = 0; i < length; i += 6)
    {
        s1 = s1 + udata[i] * vdata[i];
        s2 = s2 + udata[i + 1] * vdata[i + 1];
        s3 = s3 + udata[i + 2] * vdata[i + 2];
        s4 = s4 + udata[i + 3] * vdata[i + 3];
        s5 = s5 + udata[i + 4] * vdata[i + 4];
        s6 = s6 + udata[i + 5] * vdata[i + 5];
    }
    for (; i < length; i++)
    {
        s1 = s1 + udata[i] * vdata[i];
    }

    *dest = s1 + s2 + s3 + s4 + s5 + s6;
}
```

可能原因：
浮点数加法的单元的 issue time 为一个周期，而 Capacity 也为 1，所以每个时钟周期最多能完成 1 个假发操作，此时 CPE 下界为 1.

5.19

瓶颈在于 val = val+a[i]，这句中存在加法数据依赖，又有 K>= L*C，浮点数的加法中 L = 3，C= 1，故选择 3*1a

```c
void faster_psum1a(float a[], float p[], long n)
{
    long i;
    float val = 0;
    for (i = 0; (i+2) < n; i += 3)
    {
        float tmp1 = a[i];
        float tmp2 = tmp1 + a[i+1];
        float tmp3 = tmp2 + a[i+2];

        p[i] = var + tmp1;
        p[i+1] = var + tmp2;
        p[i+2] = var = var + tmp3;
    }
    for (; i < n; ++i)
    {
        var += a[i];
        p[i] = var;
    }
}
```