

```

1 void inner5(vec_ptr u, vec_ptr v, data_t *dest)
2 {
3     long int i;
4     int length = vec_length(u);
5     int limit = length-3;
6     data_t *udata = get_vec_start(u);
7     data_t *vdata = get_vec_start(v);
8     data_t sum = (data_t) 0;
9
10    /* Do four elements at a time */
11    for (i = 0; i < limit; i+=4) {
12        sum = sum
13        + udata[i] * vdata[i]
14        + udata[i+1] * vdata[i+1]
15        + udata[i+2] * vdata[i+2]
16        + udata[i+3] * vdata[i+3];
17    }
18
19    /* Finish off any remaining elements */
20    for (; i < length; i++) {
21        sum += udata[i] * vdata[i];
22    }
23    *dest = sum;
24 }

```

- A. We must perform two loads per element to read values for udata and vdata. There is only one unit to perform these loads, and it requires one cycle.
- B. The performance for floating point is still limited by the 3 cycle latency of the floating-point adder.