```
void inner6(vec_ptr u, vec_ptr v, data_t *dest)
{
long int i;
int length = vec_length(u);
int limit = length-3;
data_t *udata = get_vec_start(u);
data_t *vdata = get_vec_start(v);
data_t sum0 = (data_t) 0;
data_t sum1 = (data_t) 0;
data_t sum2 = (data_t) 0;
data_t sum3 = (data_t) 0;

/* Do four elements at a time */
for (i = 0; i < limit; i+=4) {
sum0 += udata[i] * vdata[i];
sum1 += udata[i+1] * vdata[i+1];
sum2 += udata[i+2] * vdata[i+2];
sum3 += udata[i+3] * vdata[i+3];
}

/* Finish off any remaining elements */
for (; i < length; i++) {
sum0 = sum0 + udata[i] * vdata[i];
}
*dest = sum0 + sum1 + sum2 + sum3;
}
```

A. For each element, we must perform two loads with a unit that can only load one value per clock cycle.

B. With IA32, the limited set of registers causes some of the accumulator values to be spilled into memory.