

第 5 章家庭作业

——201808010515 计科 1805 黄茂荣

****5.15** 假设我们想编写一个计算两个向量 X 和 Y 内积的过程。这个函数的一个抽象版本对整数、单精度和双精度数据，在 x86-64 上 CPE 等于 16~17，在 IA32 上 CPE 等于 26~29。通过进行与我们将抽象程序 combine1 变换为更有效的 combine4 相同类型的变换，得到如下代码：

```
1  /* Accumulate in temporary */
2  void inner4(vec_ptr x, vec_ptr y, data_t *dest)
3  {
4      long int i;
5      int length = vec_length(x);
6      data_t *xdata = get_vec_start(x);
7      data_t *ydata = get_vec_start(y);
8      data_t sum = (data_t) 0;
9
10     for (i = 0; i < length; i++) {
11         sum = sum + xdata[i] * ydata[i];
12     }
13     *dest = sum;
14 }
```

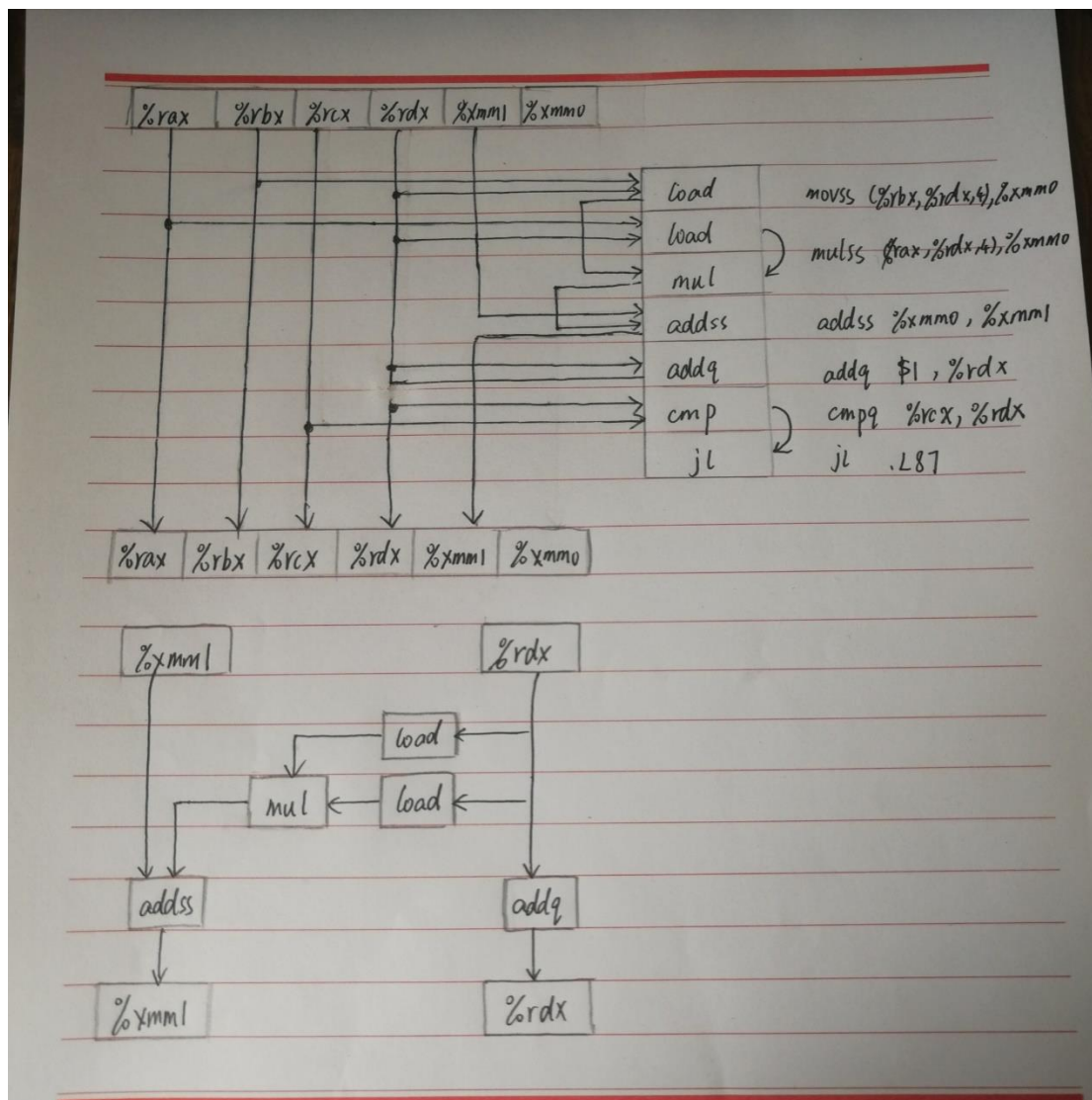
测试显示对于整数和浮点数据，这个函数的 CPE 等于 3.00。对于数据类型 float，内循环的 x86-64 汇编代码如下所示：

```
inner4: data_t = float
xdata in %rbx, ydata in %rax, limit in %rcx.
i in %rdx, sum in %xmm1

1  .L87:                                loop:
2      movss    (%rbx,%rdx,4), %xmm0    Get xdata[i]
3      mulss    (%rax,%rdx,4), %xmm0    Multiply by ydata[i]
4      addss    %xmm0, %xmm1            Add to sum
5      addq     $1, %rdx                Increment i
6      cmpq     %rcx, %rdx              Compare i:limit
7      jl      .L87                    If <, goto loop
```

假设功能单元特性如图 5.12 所示。

A. 按照图 5-13 和 5.14 的风格，画出这个指令序列会如何被译码成操作，并给出它们之间的数据相关如何形成一条操作的关键路径。



由第 2 张图片可以看到程序主要有两条数据相关链，左边的数据链限制了程序的运行时间，关键路径是 %xmm1 更新路径上的加法。

B. 对于数据类型 float，这条关键路径决定 CPE 的下界是什么
浮点数加法的延迟

C. 假设对于整数代码也有类似的指令序列，对于整数数据的关键路径决定 CPE 的下界是什么？
两个 load 单元的吞吐量界限。

D. 请解释两个浮点版本的 CPE 怎么会都是 3.00，即使乘法操作需要 4 或 5 个时钟周期。

因为乘法不在关键路径上，其也是流水线执行的，整个程序的限制因素为最后的浮点数加法的延迟，从表 5-12 可以知道 float 和 double 的延迟都是 3.00，所以两个版本的 CPE 一样。

***5.17** 编写习题 5.15 中描述的内积过程的一个版本，使用 3 次循环展开和 3 个并行累积变量。对于 x86-64，我们对这个函数的测试得到对所有类型的数据 CPE 都等于 2.00。

A. 什么因素制约了性能达到 CPE 等于 2.00？

B. 请解释为什么这个版本对于整数在 IA32 上 CPE 等于 2.67，比只做 4 路循环展开时的 CPE 等于 2.33 还要差。

实现函数如下：

```
void inner5 (vec_ptr x, vec_ptr y; data_t *dest)
{
    long int i;
    int length = vec_length(x);
    data_t *xdata = get_vec_start(x);
    data_t *ydata = get_vec_start(y);
    data_t sum0 = (data_t)0;
    data_t sum1 = (data_t)0;
    data_t sum2 = (data_t)0;
    long int limit = length-2;
    for(i=0; i<limit; i+=3)
    {
        sum0 = sum0 + xdata[i]*ydata[i];
        sum1 = sum1 + xdata[i+1]*ydata[i+1];
        sum2 = sum2 + xdata[i+2]*ydata[i+2];
    }
    for(; i<length ; i++)
    {
        sum0 = sum0 + xdata[i]*ydata[i];
    }
    *dest = sum0+sum1+sum2;
}
```

A. 什么因素制约了性能达到了 CPE 等于 2.00

Load 执行单元的吞吐量限制，即存储器的加载性能。

B. 请解释为什么这个版本对于整数在 IA32 上 CPE 等于 2.67，比只做 4 路循环展开时的 CPE 等于 2.33 还要差。

IA32 可用寄存器实际只有 6 个，而三路展开需要 i, limit, xdata, vdata, 以及存储 xdata[i], vdata[i] 的寄存器，所以有些循环变量可能会溢出到寄存器，进而影响效率。