

代码清单 6-4 插入编译器屏障 (C++)

```

int v1, v2;
void foo(){
    v1 = v2 + 1;
    __asm__ volatile ("" : : : "memory");
    v2 = 0;
}

```

再次编译后得到如代码清单 6-5 所示的汇编代码：

代码清单 6-5 插入编译器屏障 (汇编)

```

foo:
    mov     eax, DWORD PTR v2[rip]
    add     eax, 1
    mov     DWORD PTR v1[rip], eax
    mov     DWORD PTR v2[rip], 0
    ret

```

在编译后的代码中，v2 先于 v1 赋值，代码没有被编译器重排序，编译器屏障被证明为有效。

6.1.2 处理器重排序

编译器屏障解决了编译器重排序问题，但是并不能完全解决问题，即使消除了编译器重排序，CPU 也可能对指令进行重排序，出现类似编译器重排序后的代码序列。CPU 级的指令重排序又与 CPU 架构相关，具体如图 6-1 所示。

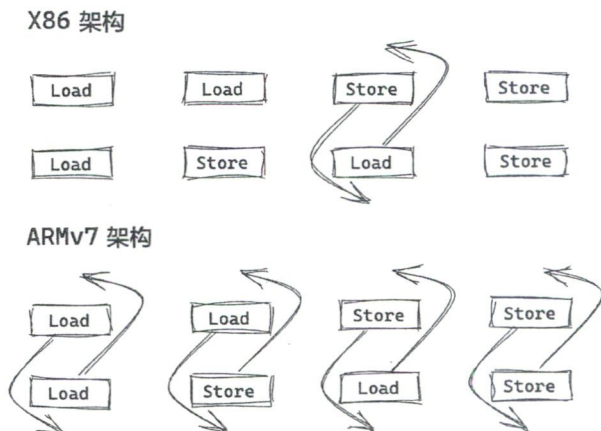


图 6-1 不同 CPU 架构上的处理器重排序规则