体系结构实习——**UniCore2** 模拟器
实习报告

张番栋 00848180
刘澜涛 00848200
王　沛 00848205
CS08

December 21, 2010

目录

第 1 章　实习内容

1.1　目标

实习的主要内容是用 C 语言编写一个支持 UniCore2 精简指令系统的模拟器，并对其进行测试、验证。在之后的报告中，均称实习所完成的模拟器为 MinSim。MiniSim 的首要目标是进行正确的功能模拟，在此基础上增加对 CPU 流水线以及 Cache 的结构模拟，另外还有对程序动态运行情况的统计。MiniSim 采用五级流水结构，和真实的 UniCore2 处理器并不相同。模拟的 Cache 则具有较灵活的定制性。

1.2　具体实现功能

MiniSim 支持的 UniCore2 指令子集中包含五类指令:

- 数据处理指令
- 乘法和乘加指令
- 跳转切换指令
- 单数据传输指令
- 条件转移指令和带链接条件转移指令。

从中可以看出，MiniSim 并未对处理器特权状态的相关功能进行模拟，进而无法完全支持基于现代操作系统下的大部分程序。针对这个问题，我们做了一些工作，使得 MiniSim 在接受标准 ELF 文件作为输入的情况下，仍然能够完成大部分应用级别的功能模拟。

本次实习是与编译实习联合进行，因此我们在完成实习的过程中做了一些编译器和模拟器之间的协调工作。

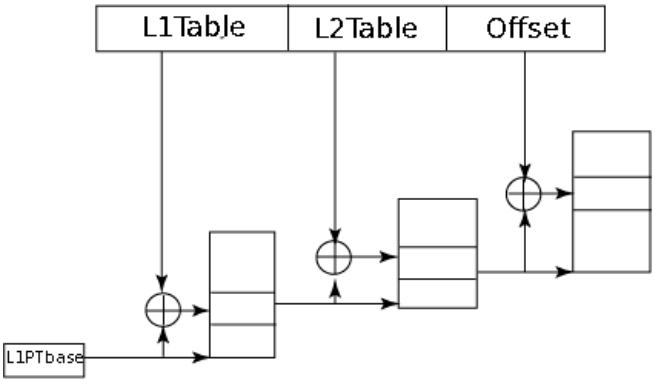
最后，为了方便使用和调试，我们为 MiniSim 编写了一个简单的控制台模块，使得 MiniSim 在运行时可以设置断点，单步执行、查看寄存器、内存和流水线的状态。

第 2 章　模拟器详解

2.1　模拟器的内存管理

MiniSim 在运行时需要在自己的地址空间内同时维护目标程序的地址空间，因此需要建立一套面向目标程序的虚拟内存机制。在这个问题上，MiniSim 采用的是页式内存管理机制。对于目标程序的 32 位地址空间，MiniSim 进行了如下划分

- 31-25 位: 一级页表偏移
- 24-15 位: 二级页表偏移
- 14-0 位: 页内偏移



也就是说，页的大小为 32KB。一级页表常驻内存，共有 128 个页表项，每个页表项大小为 4B，即指向二级页表的指针。每个二级页表含有 1024 个页表项，每个页表项 8B，存储的信息包括其所指向的页的起始地址和该页的读、写、执行属性。这是一个很经典的虚拟内存管理机制，之后

2.2　模拟器运行环境的建立

MiniSim 接受标准 ELF 文件作为输入。在真正模拟运行目标机程序之前，需要做一些初始化工作。换言之，MiniSim 需要完成一部分装载器和操作系统的工作。主要内容有 ELF 文件解析内存环境的建立、代码段和数据段的载入、寄存器堆和流水线的初始化等等。

2.2.1　对系统相关指令、功能的处理

由于 MiniSim 不支持操作系统级别的指令，同时也不需要支持复杂的程序运行环境，所以模拟区间仅限于 main 函数。也就是说，MiniSim 会跳过 main 函数之前的代码段，待 main 函数结束后退出。另外，为了便于检查，MiniSim 还需要支持整型数据的非格式化输出。输出是与系统调用有关的功能，因此需要做一些约定：在给予 MiniSim 的输入中，输出功能要被封装在一个特定的函数中。当模拟进行到这个函数时，模拟器会以宿主机的输出调用替代之，然后结束该函数，从返回地址继续执行。为了达到这个目的，需要在 ELF 文件装载阶段进行一些工作。

2.2.2 ELF 文件解析

需要做的工作主要有两点：

1. 获取 main 和输出封装函数的入口地址。

2. 获得 ELF 文件中代码段和数据段的位置和大小，以备建立模拟内存环境之用。

具体的

2.2.3 寄存器堆和流水线初始化