算法设计与分析第一次作业

李春奇　　彭焯　　华连盛　　王衍

December 24, 2010

Contents

1 项目概述 3

2 模块说明 3

2.1 模块整体视图 3

2.2 CPU模块 4

1 项目概述

本项目实现了一个unicore32体系结构的模拟器，模拟器实现了CPU的五级流水、分立的i-cache和d-cache、内存管理、动态指令统计等模块，实现的指令集是unicore32体系结构指令集的一个子集，与同组的四个人所做的编译器实现的c语言子集相对应。最终实现的目标是能够使得实验室提供的unicore-gcc编译得到的ELF可执行文件能够正确在模拟器上运行，同组的编译器能够正确的在模拟器上运行。

本项目由于和编译器同时展开，所以同组共有四名同学。在编译器和模拟器的分工上，有着一定的偏重，其中李春奇、彭焯同学偏重于模拟器的实现，华连盛、王衍同学偏重于编译器的实现。但是在设计阶段和后期编译目标代码生成、编译优化、模拟器综合验证部分是由四名同学一起来完成的，即李春奇、彭焯同学同时也参与了编译器方面的工作，华连盛、王衍同学也同时参与了模拟器的验证工作。

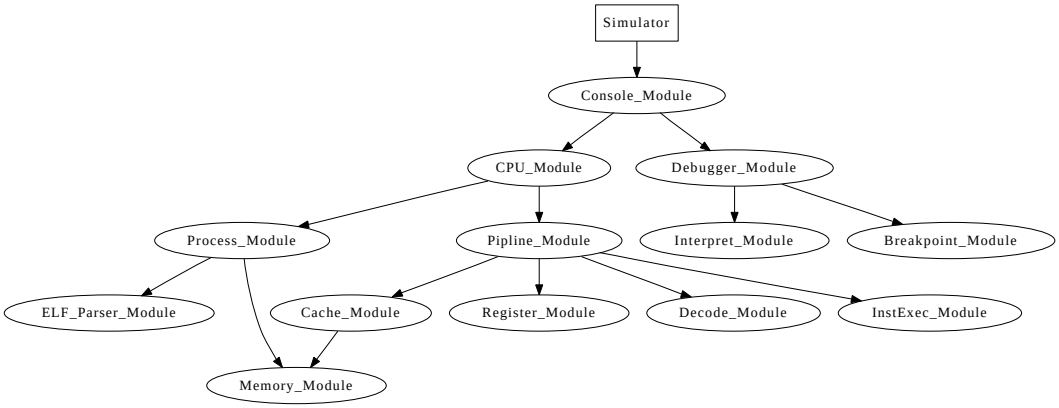
项目最终实现的模拟器能够在linux环境下正确运行，需要外部libelf库，请在项目svn的lib文件夹下解压安装。

项目svn地址: https://minic.googlecode.com/svn/trunk/simulator

2 模块说明

2.1 模块整体视图

模块的视图如下:



通过模块视图可以看到，处于第零层的模块是Console模块，即所有模块最终被封装到了一个控制台上，这样Simulator解析命令行参数之后只需要通过调用控制台即可完成下面所有的调用工作。处于第一层的模块有CPU模块和Debugger模块，他们分别被Console Module调用，完成各自的功能。CPU模块是模拟CPU的模块，其子模块分为Process模块和Pipline模块，分别是模拟进程和流水线的模块。Process模块负责ELF文件的解析

和对进程内存的初始化工作，所以起子模块是ELF Parser模块和Memory模块。Pipline模块则是模拟了CPU的五级流水，在五级流水中包括Cache模块、Register模块、Decode模块和指令执行模块。第一层的模块中另外一部分是Debugger模块，这一部分主要负责调试的工作，包含的子模块有Interpret模块（负责将机器码反汇编成汇编语言指令）和Breakpoint模块（设置和维护断点）。

以上就是项目的模块试图，每个模块对应项目中的一个文件，整体结构清晰明了，模块内部紧凑，模块之间相互独立，模块功能实现正交，便于实现和维护。下面就针对一些重要的模块（组）来进行说明，虽然在实现的过程中为了便于测试我们是采用自底向上的方式来实现，但是在说明的时候我们采用自顶向下的方式来说明，以便更清晰的说明层次关系。

2.2 CPU模块