目前毕设还有以下几部分内容需要完成：

硬件方面：

1. 代码整理，就是将修改后的sis代码模块化。全系统模拟平台主要由以下几部分组成：指令执行，时间系统（执行异步事件），可执行程序的加载，命令行界面，dbg的支持（包括反汇编），各外设模块的建模；主要外设模块有总线(对单处理器而言，就是地址译码)，中断控制器，存储器，uart，timer等；
2. 基于leon3的最小系统，包括其中存储器映射空间的分配，涉及到的外设等；

软件方面：

1. Linux0.11内核理解，这里面涉及几个基本问题：1）X86保护模式和实模式的运行机制；2）内核中，与硬件部分直接相关的内容的独立；3）在内核之上的系统服务的操作原理，也就是这类软件的设计原则，及与内核是如何交互的；
2. 然后才是内核的移植，从最简单的显示A,B开始；接着再移植0.11的内核，最后争取能够将2.0的内核移植到上面去；

2016.3.11 总结

到目前为止，硬件平台已经有了两个版本：

1、sis的修改版，其主要分为以下几个模块：

1）、命令行执行参数的解释，也就是填充那几个全局标志变量的值；全局变量全在func.c文件中定义，在其他文件中使用前需要用extern进行声明；

2）、指令历史的记录。这部分就是readline库里面的history相关函数的使用，但在sis中，并没有实现可示的功能，而且在使用remove\_history函数的时候，没有释放动态分配的内存；要添加功能很简单，具体可参考同级目录下的history.c文件；

3）、命令的读取与执行。读取实在sis.c文件中实现，而执行则是调用func.c中的exe\_cmd函数来实现的；几乎所有的操作都是在这个函数中组装定义的；

4）、指令的trace；是通过history命令来实现；history命令再第一次执行时，给出一个trace长度参数，在后序调用history命令时，就会给出最近执行的指令；

5）、指令的反汇编；

6）、指令的执行，在sparc.c文件中实现，同时在该文件中实现了对中断的检测，同时实现了trap执行的函数；

7）、时间机制，在时间前进的同时，会执行等待在相应时间点的事件；

8）、断点的添加，删除，和检测；

9）、外部中断该进程的执行的处理；signal()函数来实现的；可参考signal介绍的pdf；

几乎所有的帮助函数，初始化函数等都在func.c文件中实现；现在的系统是只有存储器，其他的外设都不存在，外部中断也没有；也就是一个裸的ISS；

2、sis和utm组合版，其特点如下：

1）、没有命令的读取和执行过程；上来之后直接运行；因为utm时间系统，开始运行后时钟周期就不断前进，不会中途停下来；

2）、在sparc.c文件中，实现了run\_cycle函数，也就是每周期处理器要干的活儿；

系统没必要的东西都删除了，只剩下一个时间驱动，ISS和存储器；