PA1 必答题

1）我选择的 ISA 是 Riscv32.

2）按照假设来说，我将在调试上面花费：500*0.9*30*20s=75h，如果实现了调试器，花费的时间为 50*0.9*10*20s=25h

3)riscv32 有哪几种指令格式?LUI 指令的行为是什么?mstatus 寄存器的结构是怎么样的?

一共有六种指令格式（^f 之后搜 type 直接寻找），如下图 1 所示。


图 1 寻找 riscv32 的指令格式种类

LUI 作用：把高位的立即数（31-12）位的低 12 位填 0，然后传入目标寄存器 dest 中，如图 2 所示

LUI (load upper immediate) is used to build 32-bit constants and uses the U-type format. LUI places the U-immediate value in the top 20 bits of the destination register *rd*, filling in the lowest 12 bits with zeros.

图 2 寻找 LUI 的作用

mstatus 的结构可以在所给的第二个 riscv 的手册中直接搜索 mstatus 来进行查看，最终搜索的结果如图 3 所示：

The **mstatus** register is an MXLEN-bit read/write register formatted as shown in Figure 3.6 for RV32 and Figure 3.7 for RV64. The **mstatus** register keeps track of and controls the hart's current operating state. Restricted views of the **mstatus** register appear as the **sstatus** and **ustatus** registers in the S-level and U-level ISAs respectively.

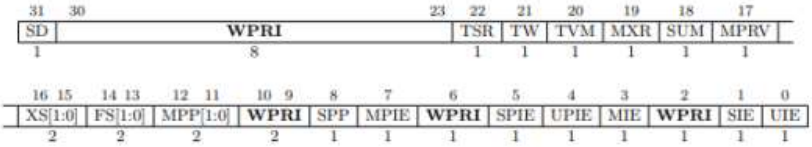


Figure 3.6: Machine-mode status register (**mstatus**) for RV32.

图 3 查看 riscv32 的 mstatus 的结构

4）完成 PA1 的内容之后, nemu/目录下的所有.c 和.h 和文件总共有多少行代码？你是使用什么命令得到这个结果的？和框架代码相比，你在 PA1 中编写了多少行代码？除去空行之外, nemu/目录下的所有.c 和.h 文件总共有多少行代码?

我的所有.c 和.h 文件一共有 5817 行。

命令：(find ./ -name "*.c"; find ./ -name "*.h") | xargs cat | wc -l

和框架代码对比的方法就是用 git 将版本回退之后再运行相同的 shell 命令即可。

去掉空行之后的再统计：(find ./ -name "*.c"; find ./ -name "*.h") | xargs cat | grep -v ^$ | wc -l，此时可以看到代码一共 4815 行。

5）开启了-wall 之后会把一些可以忽视的警告也提示出来（例如类型转化等）,而-error 会把所有的 warning 当成 error，这样我们的程序 make 成功之后就不会有任何 warning 避免了一些因为 warning 带来的错误。