

PA1 实验报告

王卫东 221900332

2023 年 10 月 12 日

一 必做题

1 程序是个状态机

对于 $(PC, r1, r2)$:

$(0, x, x) \rightarrow (1, 0, x) \rightarrow (2, 0, 0) \rightarrow (3, 0, 1) \rightarrow (4, 1, 1)$
 $\rightarrow (5, 1, 1) \rightarrow (6, 1, 2) \rightarrow (7, 3, 2) \rightarrow (8, 3, 2) \rightarrow \dots$
 $\rightarrow (39, 78, 13) \rightarrow (40, 91, 13) \rightarrow (41, 91, 13) \rightarrow (42, 91, 14)$
 $\rightarrow (43, 105, 14) \rightarrow (44, 105, 14)$

2 理解基础设施

会花费 75h, 可以节省 50h.

3 RTFM

1. 在手册 P32-RV32I 基础指令集第一小节中可以知道 riscv32 有六种基本指令格式, 分别是: 用于寄存器-寄存器操作的 R 类型指令, 用于短立即数和访存 load 操作的 I 型指令, 用于访存 store 操作的 S 型指令, 用于条件跳转操作的 B 类型指令, 用于长立即数的 U 型指令和用于无条件跳转的 J 型指令。
2. 在手册 P27-2.4 整数计算中可知 lui 为加载立即数到高位。(将 20 位常量加载到寄存器的高 20 位)
3. 在手册 P101-10.3 机器模式下的异常处理中可知 mstatus 寄存器为控制状态寄存器 (CSR), 保存全局中断使能, 以及许多其他的状态。

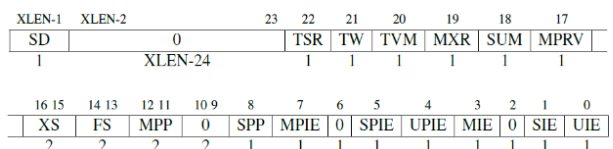


图 10.4: mstatus 控制状态寄存器。在仅有机器模式且没有 F 和 V 扩展的简单处理中，有效的域只有全局中断使能、MIE 和 MPIE（它在异常发生后保存 MIE 的旧值）。RV32 的 XLEN 是 32，RV64 是 40。
 （来自[Waterman and Asanovic 2017]中的表 3.6；有关其他域的说明请参见该文档的第 3.1 节。）

4 shell 命令

- 使用命令 `find . -type f -name "*.ch"|xargs cat| wc -l` 统计得到行数为 24289。
- 先通过 `git checkout pa0` 切换到 pa0 分支，统计代码行数为 23775，再回到 pa1 分支，统计行数为 24289，计算得到 514 行代码。
- 通过 `find . -type f -name "*.ch"|xargs cat|grep -v -E \s*$| wc -l` 统计得到 21084 行。

5 shell 命令

- `-Wall` 选项告诉编译器开启所有的警告信息。这会使得编译器对代码中潜在的问题（如未使用的变量、可能导致错误的语法）等发出警告。它会提供一些额外的警告，而不仅仅是那些明显错误的警告。举例来说，如果你有一个未使用的变量，使用了未初始化的变量，或者存在一些可能导致错误的语法，`-Wall` 会帮助你发现这些问题。
- `-Werror` 选项将所有的警告都视为错误。也就是说，一旦编译器发出任何警告，它会停止编译过程并将其视为一个错误。这迫使开发者在编译之前解决所有的警告信息。
- 方便生成错误信息，减少不必要的调试时间。

二 实验心得

整个国庆节都奉献给 PA1 了 QAQ...

在 PA1 里，让我受益匪浅的是 RTFSC，大概花了我 1 天的时间了解整个框架和阅读宏与各种函数，发现过于麻烦之后，在黄文睿同学的帮助下配置了 nvim，用 `gd` 指令就可以跳转到函数的 `def` 部分。

此后，我花了 3 天时间完成表达式求值，遇到的 bug 也是不胜其数，了解了不少 C 语言的知识。其中，de 的最久的 bug 是没有正确定义 16 进制数（万恶的 `0xf` 样例和 `$$0` 样例），然后遇到的一些 bug：输入 `'` 返回 0（没有正确的使用指针 `bool`，没有解引用导致不报错 `return 0`），在定义 `bool` 变量前没有初始化（非常不好的习惯，不能即时地返回报错）。通过这些过程，我逐渐学会了使用 `gdb`。

此后我花了 3-4h 完成了 `watchpoint` 的实现，恶补了一下 C 语言链表的知识点))。遇到了一个 bug，发现是因为我的 `cmd_w` 函数开始忘记使用 `strtok` 忽略空格导致直接 `return` 了。

总而言之，是一次很不错很有收获的经历。（没有想到其他的形容词了。。）