

ALU相关部分Prompt分析

Multi-Cycle planning和ALU部分

- 根据初始ISA架构生成寄存器定义部分包含一次Restart
- 将ALU生成opcode表格部分包含一次Restart
- 在出现偏差的修正部分，都使用了**中级以上的人工反馈**，从始至终**没有采用**直接反馈给chatGPT错误信息的**工具反馈**方法

USER

中级人工反馈

But if PC is updated in the FETCH cycle, then the math is still off? Because in FETCH the PC would be PC+1, and then in execute then it might be +2, making it overall +3 which is wrong. Perhaps we could just update PC in the Execute stage? What else would that change in your list of instructions?

- 在最终阶段给出了推理链条+具体修改位置的中/高级人工反馈，但Assistant并未采纳，而是保留了FETCH部分的PC更新，并修改了EXECUTE环节的PC更新值，达到了正确的逻辑要求

USER

- Makes sense to just have one comprehensive ALU, so can you add support for the missing instructions e.g. ROL/ROR etc?
- 同时有一个非错误，但出于丰富ALU功能要求的指令，加入ROL/ROR指令，ASSISTANT只用一步完成了任务
- 出现了**错误两次**，都在一步中/高级人工反馈内完成了修正

ALU优化

- 给出了直接的优化要求，一次成功

总结

1. 结合章节三中的严格遵循框架的对话流程，在初始生成出现错误后，`tool feedback`并不能取得很好的结果，而结合了人类知识的中/高级反馈效果较好，在FREE CHAT中测试工程师完全放弃了工具反馈，直接地给出自己的指导
2. 初始prompt生成后能够完成verilog代码的大部分构建，后续进行细节调整
3. ChatGPT可以部分领会人类的指令，但有时会有不同的理解和偏差，例如上文中对EXECUTE部分中PC的更新
4. 在生成大段综合性的表格文字阶段，往往需要restart

USER MESSAGE COUNT
Restart: 2
修正错误: 2
增加/补充需求: 3

ISA 至 ALU OPCODE

- 被打断一次，在生成操作码部分将所有操作码都编为8位，为了避免继续错误的输出，人类工程师进行了中断，ChatGPT给出了及时的修正
- 遗漏操作码，且从上一阶段的错误输出的指令操作码中进行了抽取，说明其虽然进行了修正，但并未意识到之前的错误对后续的影响
- 再次遗漏了ADDI指令

总结

1. 输出中的中断反馈可以帮助ChatGPT纠错
2. ChatGPT很可能不能记住刚刚犯过且已经改正的错误
3. 遗漏需求较多，反映出initial Prompt不够完整

USER MESSAGE COUNT
Restart: 0

修正错误: 3
增加/补充需求: 0

Python汇编程序

- ADDI指令又发生了错误，且是类似的位长问题，指出错误并RESTART一次后才成功

总结

1. 同样的错误反复出现，某些类型具有很高的错误概率
2. 初始initial prompt生成重启了两次
USER MESSAGE COUNT
Restart: 3
修正错误: 1
增加/补充需求: 1

整体总结

- 初始prompt的生成质量直接影响后续调整难易程度
- 有些高频错误会在多个设计部分中重复出现，即使两个对话线程没有直接联系，说明错误与某些设计部分直接相关
- ChatGPT并不能深度理解对话内容
- 汇总性信息生成往往需要Restart多次