
课后实验二：状态机

预备知识

- 时序逻辑电路的 verilog 实现
 - clk 信号的生成
 - 阻塞赋值与非阻塞赋值
 - always 块的敏感信号的理解，上升沿和下降沿
- always 块中相关语法
 - if else
 - case

实验重难点

- 状态机的 verilog 实现
 - 三段式状态机
 - 状态转移块
 - 状态获取块
 - 结果输出块

实验内容

- 阅读状态机的相关知识
- 阅读示例 Moore 状态机 verilog 代码实现
- 新建工程，将示例状态机模块和 testbench 加入后 run simulation，观察波形

状态机知识点

状态机基础

有限状态机 (Finite-State Machine, FSM)，又成为有限状态自动机，简称状态机，是表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型。

有限状态机的组成

- 状态寄存器：记忆状态机当前所处的状态
- 产生下一状态的组合逻辑：根据输入信号或当前状态，确定下一状态
- 输出逻辑：由当前状态和输入信号，决定当前状态的输出

两类状态机

Moore 状态机

状态机的输出仅仅依赖于当前状态，而与输入条件无关

Mealy 状态机

状态机的输出不仅依赖于当前状态，而且取决于该状态的输入条件

设计状态机（三段式）

- 一个时序逻辑 `always` 块用来描述当前状态转移 (`current_state`)，寄存器的复位和状态转移。即下图由 `z` 到 `q` 的过程。
 - 时序逻辑采用非阻塞赋值 (`<=`)
 - 时序逻辑的敏感信号需要是 信号的上升沿等 (`posedge`、`negedge`)
- 一个组合逻辑 `always` 块用来描述下一个状态的求取 (`next_state`)。即下图组合逻辑电路中求 `z` 的部分。
 - 组合逻辑采用阻塞赋值 (`=`)
 - 组合逻辑的敏感信号需要包含 `always` 块中所有可能会发生变化的变量，简单写法为 `always@(*)`
 - 组合逻辑电路需要处理所有可能的情况，否则会出现锁存器。解决的好办法：`if else` 中的 `else` 处理或者 `case` 中的 `default` 处理
- 一个组合逻辑 `always` 块或者时序逻辑 `always` 块用来描述输出 (`output`)。

实验 tips

- 没有验收和实验报告
- 在第三次实验课前自行学习了解，有问题直接在群里提问
- 示例 Moore 状态机对应的状态转换图如下图所示，示例中提供了一段式、二段式、三段式的实现方式，请认真阅读，理解体会其不同，推荐使用三段式的实现方式。

-
- 后续会有状态机的相关实验，因此，本实验内容请务必掌握，否则后续实验寸步难行

