

药片装瓶系统 Verilog 设计文档

目录

- 药片装瓶系统 Verilog 设计文档
 - 目录
 - 1. 项目概述
 - 2. 设计要求
 - 3. 系统架构
 - 4. 模块详细说明
 - 4.1 pill_system_top.v
 - pill_system_top 模块
 - 4.2 work_controller.v
 - work_controller 模块
 - 4.3 clock_divider.v
 - clock_divider 模块
 - 4.4 input_validator.v
 - input_validator 模块
 - 4.5 bcd_display.v
 - bcd_display 模块
 - 5. 工作流程 (使用说明)
 - 5.1 系统启动
 - 5.2 参数设置
 - 5.3 开始工作
 - 5.4 工作过程
 - 5.5 完成处理
 - 5.6 异常处理
 - 6. 小组成员分工
 -
 -
 -
 -

1. 项目概述

本项目使用 Verilog HDL 实现了一个基于TEC-8实验台的药片装瓶系统。该系统能够自动计数药片并控制装瓶过程，支持参数化设置（每瓶药片数、装瓶数量），提供直观的数码管显示和蜂鸣器报警功能。设计采用了模块化的方法，将不同的功能单元分解到独立的 Verilog 文件和模块中，确保系统的稳定性和可维护性。

2. 设计要求

● 课设要求基本功能：

- 实现药片自动计数功能；
- 实现装瓶控制功能；
- 实现参数化设置功能（每瓶药片数、装瓶数量）；

- 实现数码管显示功能。

● 项目实现附加功能：

- 实现设置状态时闪烁显示；
- 实现蜂鸣器报警功能；
- 实现防抖动按钮处理；
- 实现BCD格式统一处理；
- 实现智能显示切换功能。

3. 系统架构

系统采用自顶向下的设计方法，包含一个顶层模块 (`pill_system_top`) 和四个功能子模块：工作控制状态机 (`work_controller`)、时钟分频器 (`clock_divider`)、输入验证器 (`input_validator`) 和 BCD显示驱动器 (`bcd_display`)。顶层模块负责实例化所有子模块并建立它们之间的连接。

```
顶层模块 (pill_system_top.v)
├─ 时钟分频模块 (clock_divider.v)
├─ 输入验证模块 (input_validator.v)
└─ 工作控制状态机 (work_controller.v)
└─ BCD显示驱动模块 (bcd_display.v)
```

4. 模块详细说明

4.1 `pill_system_top.v`

文件作用: 定义了整个药片装瓶系统的顶层模块。它负责整合所有子模块，连接输入/输出端口，并协调各个子模块之间的信号传递。

`pill_system_top` 模块

- 模块作用:** 作为设计的顶层实体，实例化所有功能子模块 (`clock_divider`, `input_validator`, `work_controller`, `bcd_display`)。它连接外部输入（时钟、按钮、开关）和内部模块，并将最终的显示数据和蜂鸣器信号输出。
- 输入:**
 - `clk_10kHz`: 10kHz 时钟信号（状态机、显示更新）。
 - `clk_1Hz`: 1Hz 时钟信号（药片脉冲、闪烁控制）。
 - `QD`: 脉冲按钮（用于确认输入、启动工作、继续操作）。
 - `CLR`: 复位按钮（低电平有效）。
 - `switches[10:1]`: 10个开关（第10位模式选择，第9位设置选择，第1-8位数据输入）。
- 输出:**
 - `display1[4:1]`: 第1位数码管BCD码。
 - `display2[4:1]`: 第2位数码管BCD码。
 - `display3[4:1]`: 第3位数码管BCD码。
 - `display4[4:1]`: 第4位数码管BCD码。
 - `display5[4:1]`: 第5位数码管BCD码。
 - `buzzer`: 蜂鸣器控制信号。

- **assign 语句:**
 - `assign mode_switch = switches[10];`: 将第10位开关赋值给模式选择信号，用于指示当前是设置模式还是工作模式。
 - `assign setting_switch = switches[9];`: 将第9位开关赋值给设置选择信号，用于选择设置每瓶药片数还是装瓶数。
 - `assign input_data = switches[8:1];`: 将第1-8位开关作为数据输入。
- **防抖动处理:**
 - 使用同步器对QD和CLR按钮进行防抖动处理。
 - 通过边沿检测生成qd_pressed信号。
- **参数锁存机制:**
 - 在设置模式下，当QD按钮按下且输入有效时，锁存设置值到内部寄存器。

4.2 work_controller.v

文件作用: 定义了控制药片装瓶系统操作流程的核心状态机。

work_controller 模块

- **模块作用:** 实现系统的主要工作逻辑，包括四个状态的转换（设置、工作、暂停、报警），处理药片计数、装瓶控制和蜂鸣器管理。它是整个系统的控制核心。
- **输入:**
 - `clk`: 时钟 (10kHz)。
 - `reset`: 复位信号 (高电平有效)。
 - `mode_switch`: 模式开关 (0-设置模式, 1-工作模式)。
 - `qd_pressed`: QD按钮按下事件。
 - `pill_pulse`: 药片脉冲信号 (1Hz)。
 - `pills_per_bottle[7:0]`: 每瓶药片数设置 (2位BCD)。
 - `target_bottles[7:0]`: 装瓶数设置 (2位BCD)。
- **输出:**
 - `work_state[1:0]`: 工作状态 (00:设置, 01:工作, 10:暂停, 11:报警)。
 - `total_pills_bcd[11:0]`: 已装瓶药片总数 (3位BCD)。
 - `current_pills_bcd[7:0]`: 当前瓶药片数 (2位BCD) 或已装瓶数 (智能切换-装瓶后切换)。
 - `bottle_count_bcd[7:0]`: 已装瓶数 (2位BCD)。
 - `buzzer`: 蜂鸣器控制信号。
- **BCD加1函数:**
 - `bcd_add_one_12bit`: 处理12位BCD数据的加1操作，用于总药片数。
 - `bcd_add_one_8bit`: 处理8位BCD数据的加1操作，用于当前药片数和瓶数。
- **比较逻辑:**
 - `current_bottle_will_be_full`: 预测下一颗药后是否会满瓶。
 - `target_will_be_reached`: 预测下一瓶后是否达到目标。
- **状态转换逻辑:**
 - STATE_SETTING → STATE_WORKING: QD按钮按下且`mode_switch=1`。
 - STATE_WORKING → STATE_PAUSED: 达到目标装瓶数。
 - STATE_PAUSED → STATE_ALARM: QD按钮按下。
 - 任何状态 → STATE_SETTING: 复位信号。

4.3 clock_divider.v

文件作用: 定义了时钟分频和脉冲生成模块。

clock_divider 模块

- 模块作用:** 将1Hz时钟信号转换为药片脉冲信号，解决使用10kHz时钟信号需要编写分频函数的问题。通过边沿检测技术生成单周期脉冲。
- 输入:**
 - clk_10kHz: 10kHz时钟信号。
 - clk_1Hz: 1Hz时钟信号。
 - reset: 复位信号。
- 输出:**
 - pill_pulse: 药片脉冲信号 (1Hz频率的单周期脉冲)。
- 实现原理:**
 - 通过检测1Hz时钟的上升沿生成pill_pulse。
 - 确保每个1Hz周期只产生一个脉冲，避免重复计数。

4.4 input_validator.v

文件作用: 定义了输入数据验证模块。

input_validator 模块

- 模块作用:** 验证输入数据的有效性，确保BCD码格式正确且数值在合理范围内。防止无效输入导致系统异常。
- 输入:**
 - input_data[7:0]: 输入数据 (两位BCD)。
 - setting_switch: 设置选择开关 (0-每瓶药片数, 1-装瓶数)。
- 输出:**
 - input_valid: 输入数据有效信号。
- 验证规则:**
 - 检查BCD码格式：每个半字节必须≤9。
 - 每瓶药片数范围：01-50 (不允许00)。
 - 装瓶数范围：01-18 (不允许00)。
 - 特殊处理零值输入，防止系统异常。

4.5 bcd_display.v

文件作用: 定义了数码管显示控制模块。

bcd_display 模块

- 模块作用:** 根据当前系统状态和模式选择合适的数据源驱动数码管显示。实现设置模式下的闪烁效果，提供直观的数码管显示。
- 输入:**
 - clk: 时钟 (10kHz)。
 - clk_1Hz: 1Hz时钟 (闪烁控制)。
 - reset: 复位信号。
 - mode_switch: 模式开关。

- `setting_switch`: 设置选择开关。
 - `input_data[7:0]`: 输入数据。
 - `input_valid`: 输入数据有效信号。
 - `pills_per_bottle[7:0]`: 每瓶药片数设置。
 - `target_bottles[7:0]`: 装瓶数设置。
 - `total_pills_bcd[11:0]`: 总药片数。
 - `current_pills_bcd[7:0]`: 当前瓶药片数。
 - `bottle_count_bcd[7:0]`: 已装瓶数。
 - `work_state[1:0]`: 工作状态。
- **输出:**
 - `display1[4:1] ~ display5[4:1]`: 五位数码管接收BCD码。
 - **显示逻辑:**
 - 设置模式: 显示参数设置界面, 支持闪烁提示。
 - 工作模式: 显示实时工作状态 (总药片数、当前药片数、瓶数)。
 - 智能切换: 工作时显示当前药片数, 完成时显示瓶数。
 - **闪烁功能:**
 - `setting_switch=0`时, 第4、5位数码管闪烁 (每瓶药片数)。
 - `setting_switch=1`时, 第2、3位数码管闪烁 (装瓶数)。
 - 闪烁时显示0xF (数码管熄灭)。

5. 工作流程 (使用说明)

5.1 系统启动

1. **复位系统:** 按下CLR按钮 (低电平有效) 进行系统复位
2. **进入设置模式:** 系统复位后自动进入设置模式 (STATE_SETTING)
3. **默认参数:** 系统加载默认参数 (每瓶药片数: 10颗, 装瓶数量: 5瓶)

5.2 参数设置

1. **选择设置项:**
 - `switches[9] = 0`: 设置每瓶药片数 (第4、5位数码管闪烁)
 - `switches[9] = 1`: 设置装瓶数量 (第2、3位数码管闪烁)
2. **输入数值:**
 - 使用`switches[8:1]`输入2位BCD数据
 - 每瓶药片数范围: 01-50
 - 装瓶数量范围: 01-18
 - 系统自动验证输入有效性
3. **确认设置:**
 - 按下QD按钮确认当前设置项
 - 参数被锁存到内部寄存器
 - 可重复设置不同参数项
4. **查看设置:**

- 设置模式下数码管实时显示当前设置值
- 闪烁提示当前正在设置的项目

5.3 开始工作

1. **切换工作模式**: 将`switches[10]`设置为1 (工作模式)
2. **启动系统**: 按下QD按钮启动装瓶流程
3. **状态转换**: 系统从`STATE_SETTING`转换到`STATE_WORKING`
4. **计数器清零**: 所有计数器自动清零, 准备开始计数

5.4 工作过程

1. 药片计数:

- 系统接收1Hz药片脉冲信号
- 每个脉冲代表一颗药片通过
- 总药片数和当前瓶药片数同时递增

2. 显示更新:

- 第1位: 总药片数百位
- 第2-3位: 总药片数十位和个位
- 第4-5位: 当前瓶药片数十位和个位

3. 装瓶判断:

- 当前瓶药片数达到设定值时自动换瓶
- 当前瓶药片数清零, 已装瓶数加1
- 总药片数继续累计

4. 进度监控:

- 实时显示装瓶进度
- 用户可随时观察系统状态

5.5 完成处理

1. 达到目标:

- 当已装瓶数达到设定目标时
- 系统自动转换到`STATE_PAUSED`状态
- 蜂鸣器开始鸣响提示

2. 显示切换:

- 第4-5位数码管从显示当前药片数切换到显示已装瓶数
- 用户可清楚看到完成的瓶数

3. 确认完成:

- 按下QD按钮确认任务完成
- 系统转换到`STATE_ALARM`状态

4. 持续监控:

- 在ALARM状态下继续接收和计数额外的药片
- 蜂鸣器持续鸣响
- 超额药片数量被记录

5.6 异常处理

1. 重新设置:

- 任何时候按下CLR按钮都可重置系统
- 系统返回设置模式，重新开始流程

2. 参数修改:

- 在设置模式下可随时修改参数
- 修改后需按QD确认

3. 暂停恢复:

- 工作过程中可通过复位返回设置模式
- 重新设置参数后可重新开始

4. 状态指示:

- 通过数码管显示内容可判断当前系统状态
- 蜂鸣器状态指示系统完成情况

6. 小组成员分工

0 0 0 : 工作状态机、顶层集成、调优

- **主要职责:**

- 工作状态机设计与实现 work_controller.v
- 后续调试与优化

0 0 0 : 时钟分频模块

- **主要职责:**

- 时钟分频模块设计与实现 clock_divider.v

0 0 0 : 输入验证模块

- **主要职责:**

- 输入验证模块设计与实现 input_validator.v

0 0 0 : 数码管显示模块

- **主要职责:**

- 数码管显示模块设计与实现 bcd_display.v