跑表电路的设计

|  |  |
| --- | --- |
| 姓名： | 刘金硕 |
| 学号： | 202428015926051 |

# 目录

[目录 1](#_Toc150846839)

[1 设计规格与要求 2](#_Toc150846840)

[2 算法原理与算法设计 2](#_Toc150846841)

[3 实现架构 2](#_Toc150846842)

[4 仿真结果 3](#_Toc150846843)

[5 综合结果 4](#_Toc150846844)

# 设计规格与要求

设计一个跑表时序逻辑电路，通过按钮控制及数字显示，有时分秒显示，可以清零、开始和暂停。系统主时钟频率为10 MHz。

其中按钮Clear实现清零功能（任意状态按下时分秒值清零并停止计时）、按钮Start/Stop实现开始和暂停功能（若当前状态为停止则按下继续进行计时，若当前状态为计时则按下暂停计时）。

数字显示为XX : XX : XX形式，时分秒各为2位数字。对每位数字使用4位二进制编码输出表示（hr\_h[3:0],hr\_l[3:0] : min\_h[3:0],min\_l[3:0] : sec\_h[3:0],sec\_l[3:0]）。

# 算法原理与算法设计

跑表的实现主要可以分为控制逻辑和计数器两个部分。

跑表的时分秒数据通过计数器cnt来实现。每一个计数器为一个四bit寄存器，对应于时分秒的一位数字。计时器的输入数据有三种情况：1）cnt+1，2）保持，对应于stop状态,3)同时cnt应具有异步clear功能。

控制逻辑用于实现跑表的计时、清零与暂停功能，通过状态机实现，状态编码与状态转换定义如下：

|  |  |
| --- | --- |
| 状态 | 编码 |
| STOP | 0 |
| START | 1 |

当start\_stop==1，状态机在STOP和START之间切换状态，同时修改控制信号cnt\_en来控制cnt的更新。start\_stop和clear信号为上升沿有效，因此需要采用上升沿检测逻辑，检测start\_stop和clear的上升沿。

# 实现架构

本设计主要由控制信号产生模块与跑表各位的计数器组成，其中控制信号由状态机产生，在顶层模块中描述。

**模块端口说明**

**top module：stop\_watch**

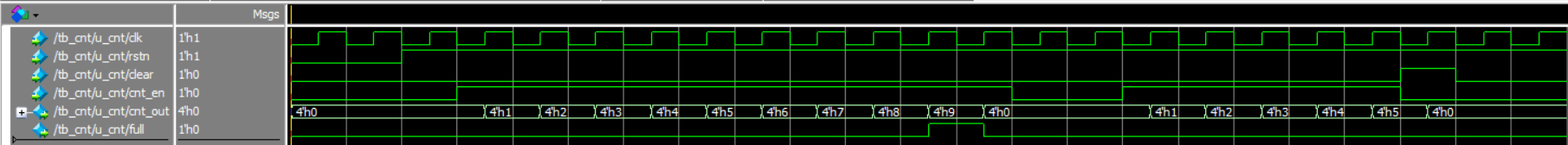
|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **方向** | **位宽** | **描述** |
| clk | I | 1 | 系统时钟，10 MHz |
| rst\_n | I | 1 | 异步复位，低电平有效 |
| clear | I | 1 | 清零按钮，上升沿有效 |
| start\_stop | I | 1 | 开始/暂停按钮，上升沿有效 |
| hr\_h | O | 4 | 时高位输出，取值0~9 |
| hr\_l | O | 4 | 时低位输出，取值0~9 |
| min\_h | O | 4 | 分高位输出，取值0~9 |
| min\_l | O | 4 | 分低位输出，取值0~9 |
| sec\_h | O | 4 | 秒高位输出，取值0~9 |
| sec\_l | O | 4 | 秒低位输出，取值0~9 |

**module: cnt**

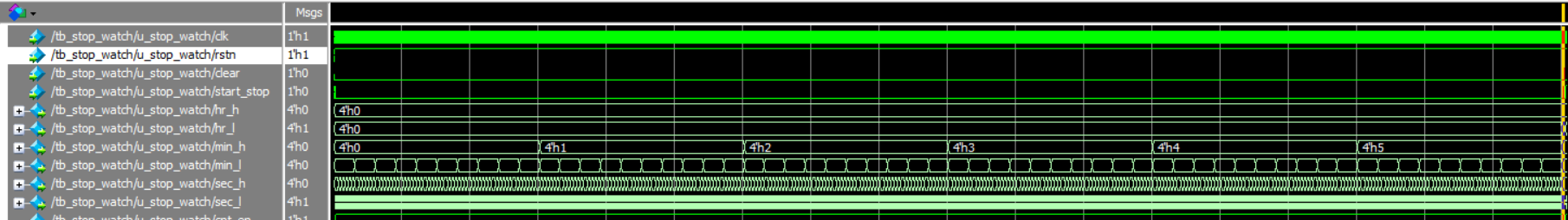
|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **方向** | **位宽** | **描述** |
| clk | I | 4 | 系统时钟，10 MHz |
| rstn | I | 4 | 异步复位，低电平有效 |
| clear | I | 1 | 清零按钮，上升沿有效 |
| cnt\_en | I | 1 | 计数器使能 |
| cnt\_out | O | 4 | 计数器输出 |
| full | O | 1 | 计数器计满 |

# 仿真结果

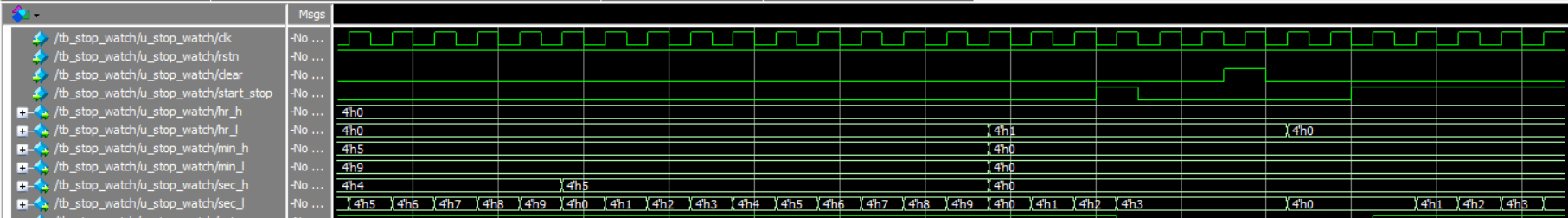
首先仿真cnt.v，testbench中改变控制信号的值，分别验证cnt的复位、清零、使能，modelsim仿真波形如下。可以看到，cnt能够在控制信号控制下计数，停止与清0，功能符合要求。



接下来测试顶层模块的功能。计时1hr（3600时钟周期）左右，跑表时分秒各位波形结果如下，可以看到，min\_h,min\_l,hr\_l均能够正常计时，并在3600时钟周期后计时到1h左右，计时结果准确。



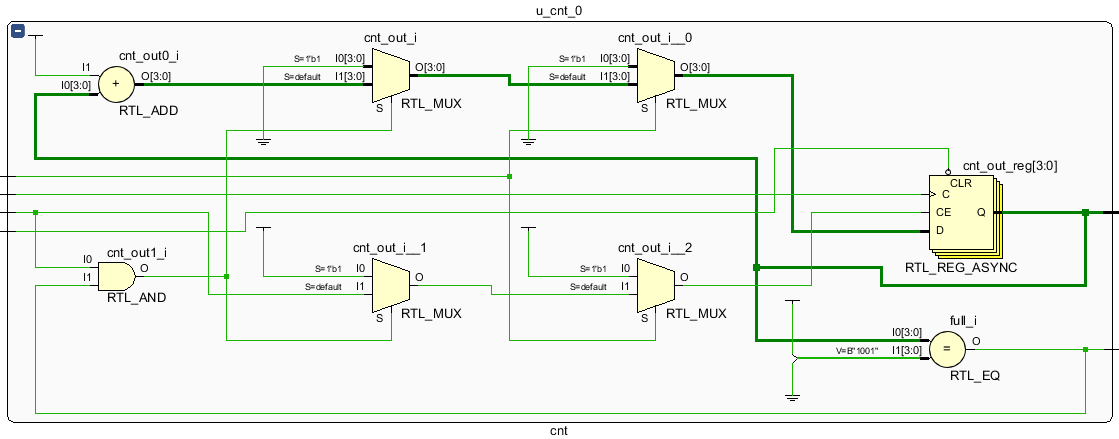
下面来测试clear与stop功能。计时满1h后两个周期产生stop信号，检测到上升沿后的一周期各位cnt即停止计数，而再三个周期后进行clear，一个周期后各位cnt完成清0，证明clear与stop功能正确。综上本设计满足题目要求的功能。



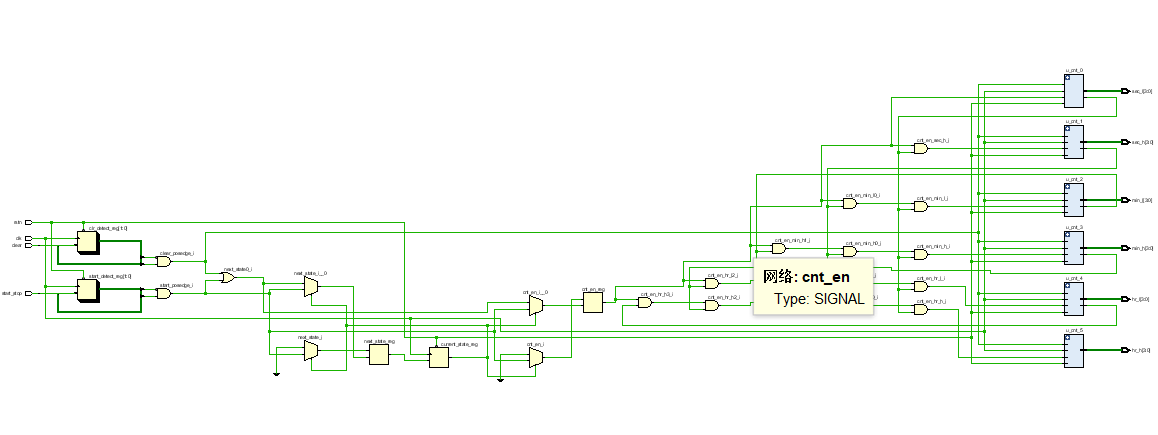
# 综合结果

**RTL SCHEMATIC**

**cnt模块**



**stop\_watch模块**



**FPGA资源使用**

