

装

订

线

**学生实验实习报告册**

|  |  |
| --- | --- |
| 学年学期： | 2022-2023 学年☑春🞏秋学期 |
| 课程名称： | 数字电路与逻辑设计实验A |
| 实验项目： | 基于FPGA的数字电子钟的设计与实现 |
| 学院和专业： | 通信与信息工程学院 |
| 班 级： |  |
| 姓 名： |  |
| 学 号： |  |
| 序 号： |  |
| 指导教师： |  |

**重庆邮电大学教务处制**

1. 系统顶层模块设计

表1.1 数字钟总体功能表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 按键/开关 | 状态 | 功能 |
| 分频模块 | SW1 SW0 | 00 | 输出1hz |
| 01 | 输出5khz |
| 10 | 输出1khz |
| 11 | 输出0hz |
| 清零模块 | Key0 | 1 | 清零 |
| 校准时间模块 | SW5 | 1 | 暂停校时 |
| 0 | 开始计时 |
| Key2 | 1 | 校秒 |
| Key3 | 1 | 校分 |
| Key4 | 1 | 校时 |
| 闹钟模块 | SW6 | 1 | 进入设置闹钟模式 |
| 0 | 退出设置闹钟模式 |
| Key2 | 1 | 设置闹钟秒种 |
| Key3 | 1 | 设置闹钟分钟 |
| Key4 | 1 | 设置闹钟时钟 |
| SW2 | 1 | 关闭闹钟 |
| 0 | 开启闹钟 |
| 秒表模块 | SW7 | 1 | 进入秒表模式 |
| 0 | 退出秒表模式 |
| Key6 | 1 | 开始计时/结束计时 |
| Key7 | 1 | 清零 |

图示

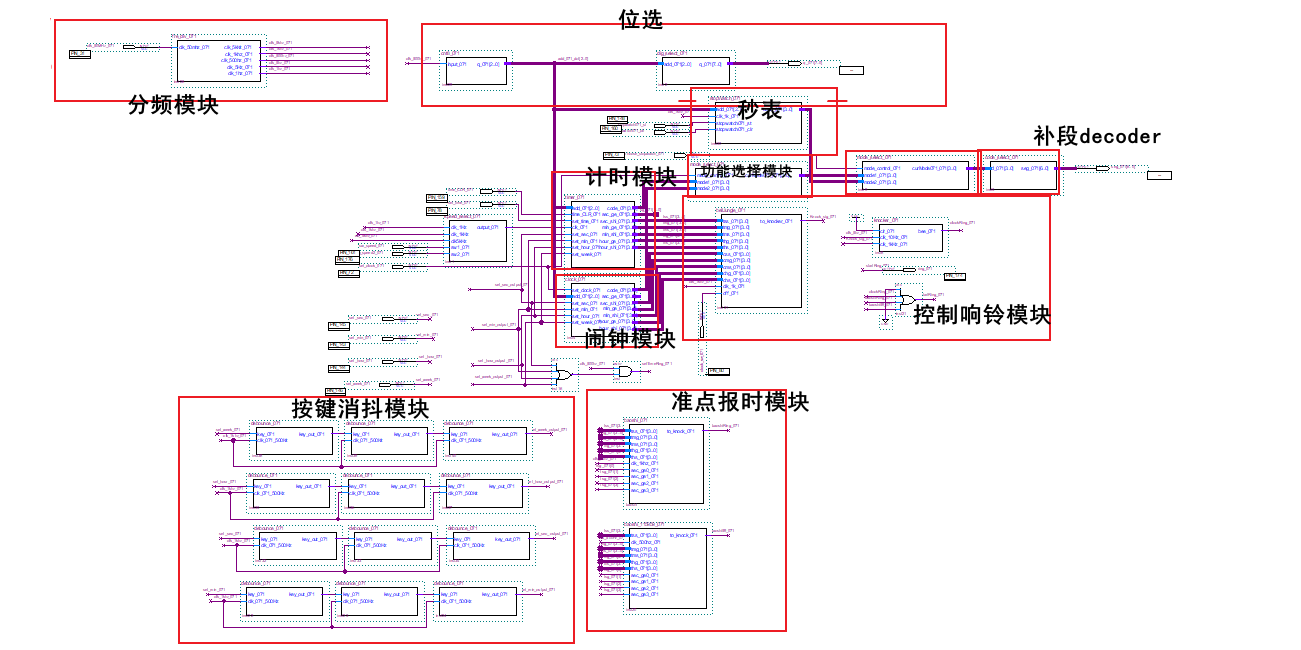
描述已自动生成

图1.1 总体设计框图

图示, 示意图

描述已自动生成

图1.2 顶层设计框图



**控制时钟频率模块**

图1.3 顶层设计框图说明

图片包含 图表

描述已自动生成

图1.4 顶层 RTL Viewer 视图

1. 分频模块电路设计及仿真
2. 模块功能

将50MHz时钟分频为5KHz、1KHz、500KHz、5hz、1hz共5种频率。

1. 设计思路

实验平台上提供频率为50MHz的时钟信号，通过2、5、100分频，对信号进行逐级分频。5、100分频功能采用74390实现。

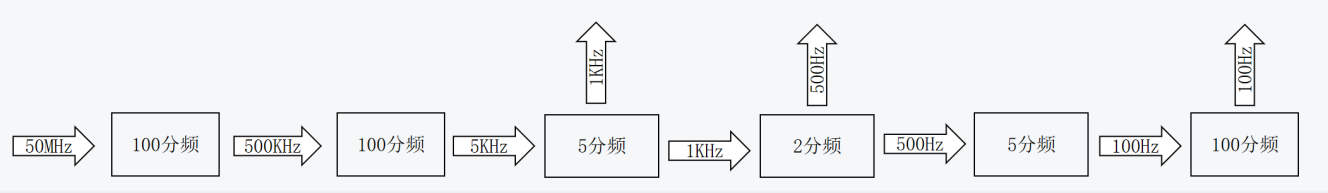


图2.1 分频器设计框图

1. 设计结果（电路）

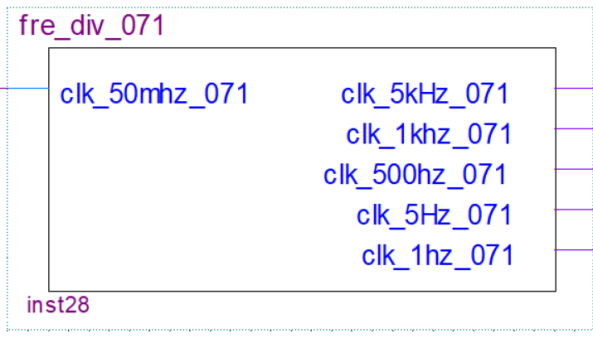


图2.2 分频模块封装设计

图示

描述已自动生成

图2.3 分频模块设计电路

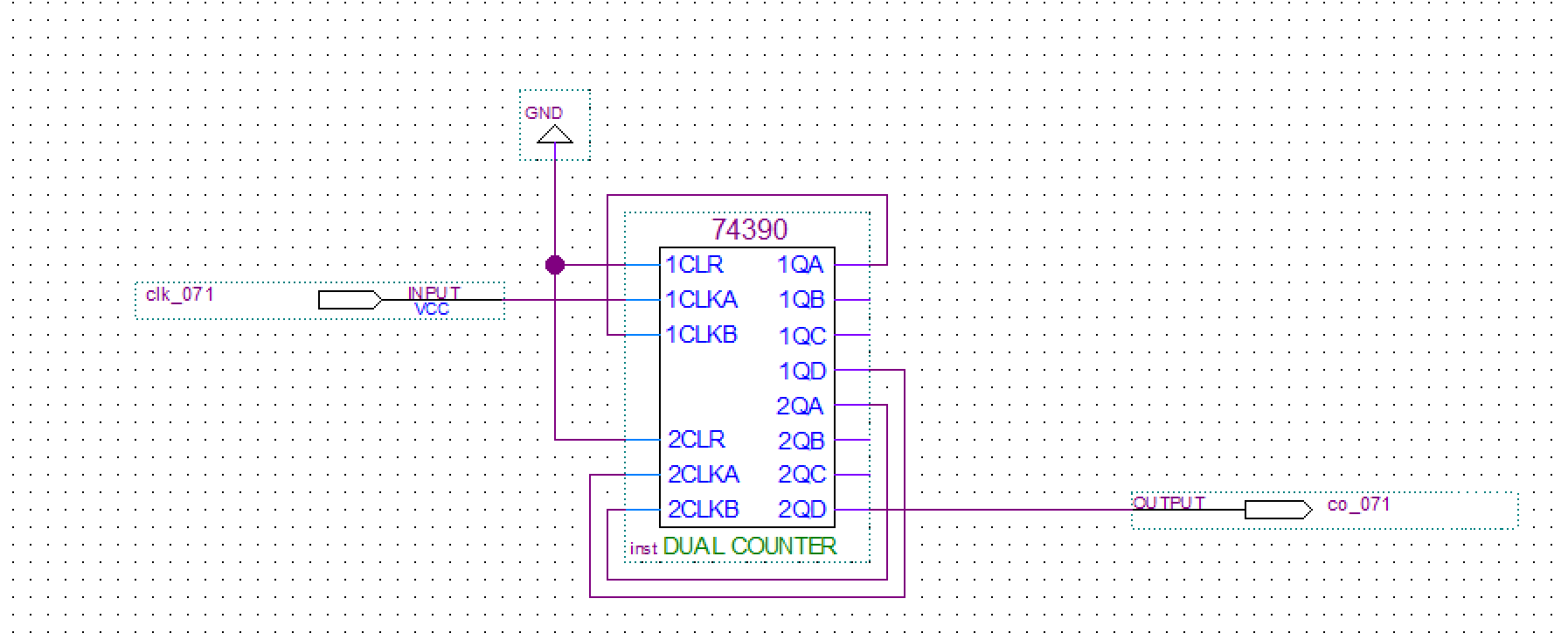


图2.4 m100计数器模块设计电路

1. 测试结果

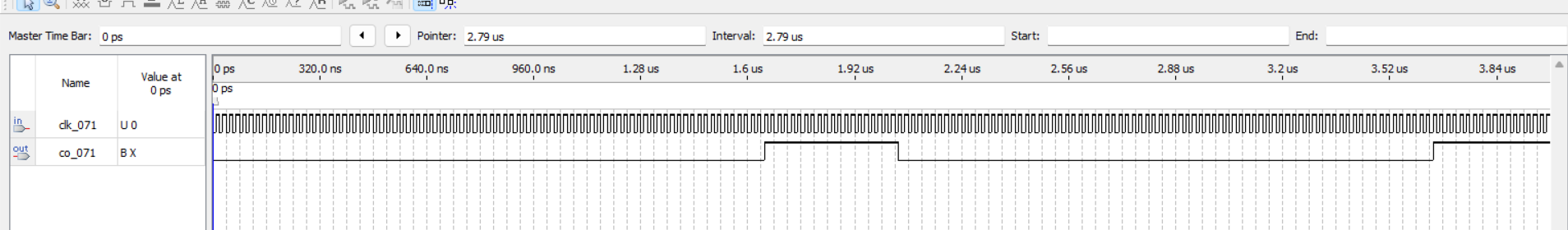


图2.5 m100模块波形仿真测试结果

1. 仿真波形数据分析

0-79计数状态CO输出为0，计数状态为80-99时CO输出为1。20ns一个时钟周期。大约1.6us时CO升为高电平。一直持续到2us左右CO再次降为低电平。时序仿真结过正确

1. 计时模块设计及仿真
   1. 分、秒计时模块（模60计数）
2. 模块功能（计数、进位）

输入CLK和清零信号。当清零型号为0时，模块将CLK输入信号进行60分频，分别输出个位和10位的8421BCD码和进位信号。

1. 设计思路

使用74390搭建模值为60的计数器。计数状态如下：

表3.1 模60计数器计数状态表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M60 | 2QD | 2QB | 2Q | 2QA | 1QD | 1QC | 1QB | 1QA | 2CLR | 1CLR |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 12 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 14 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 16 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 18 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 20 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 22 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 23 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 24 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 26 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 27 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 30 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 32 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 33 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 34 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 35 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 36 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 37 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 38 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 39 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 40 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 41 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 42 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 43 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 44 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 45 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 46 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 47 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 48 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 49 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 51 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 52 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 53 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 54 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 55 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 56 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 57 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 58 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 59 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 60 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

由计数状态表得到清零端表达式：2CLR = 1CLR = 2QB \* 2QA

1. 设计结果（电路）

文本

描述已自动生成

图3.1 模60计数器封装设计

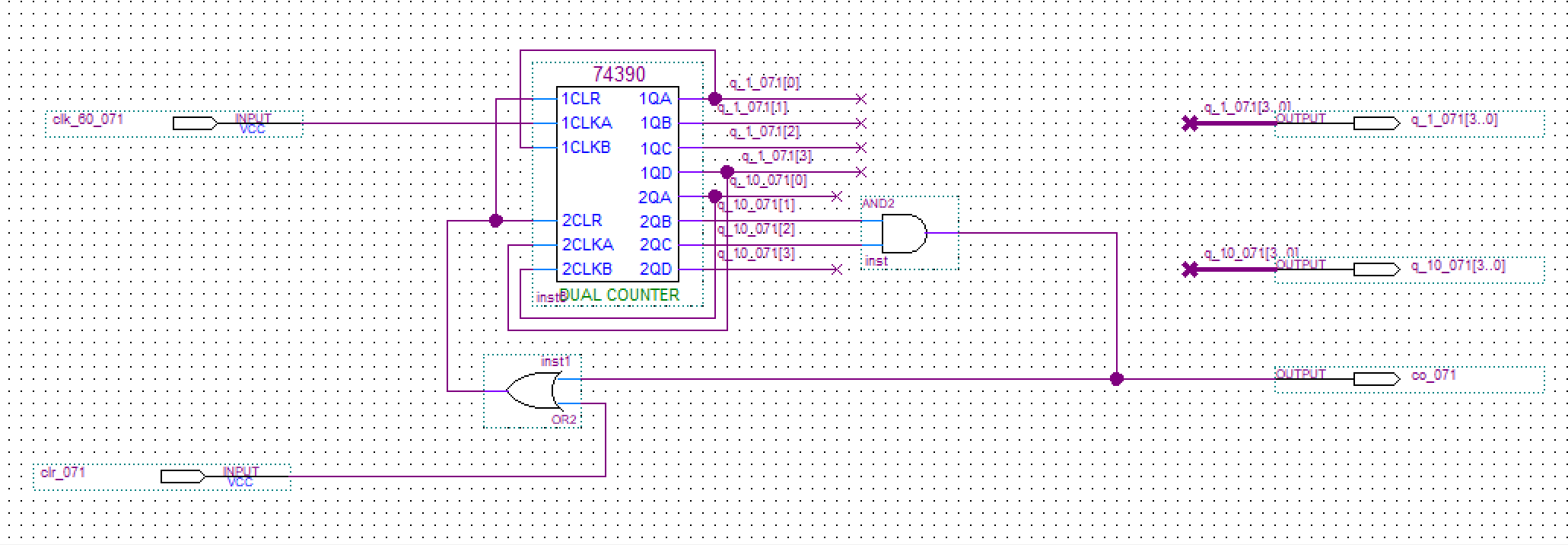


图3.2 模60计数器电路图

1. 仿真测试

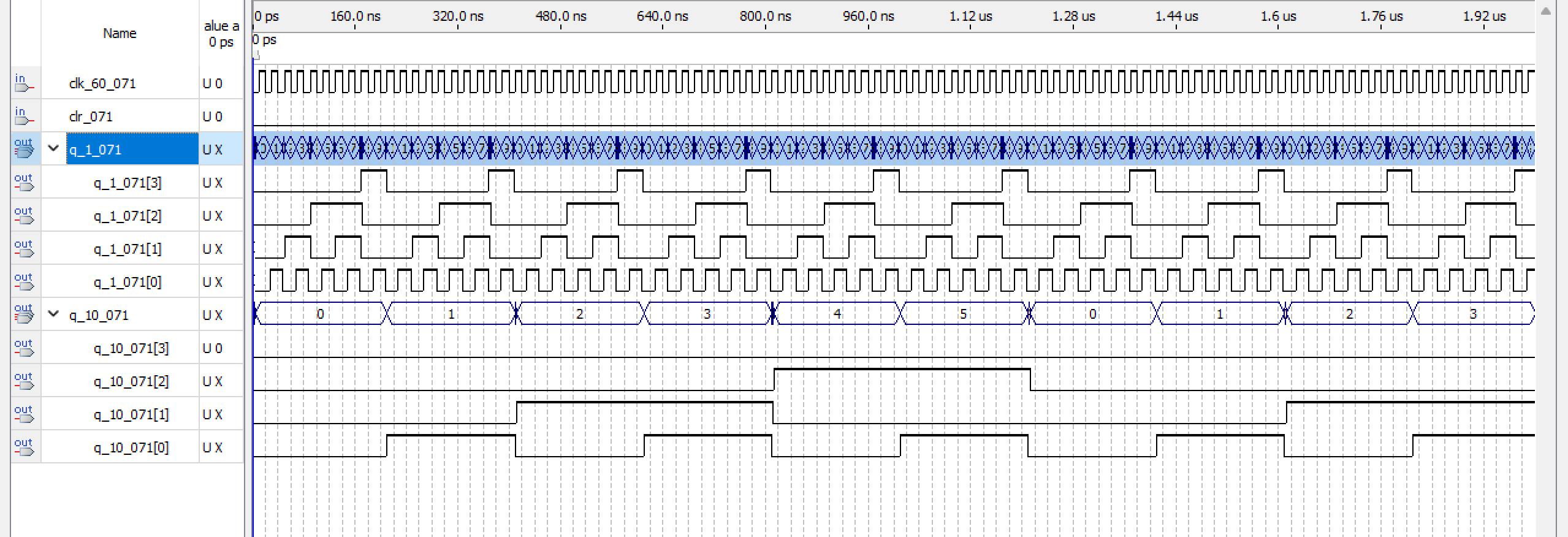


图3.3 模60计数器波形仿真图

1. 仿真波形数据分析

由图3.3可看出，每经过60个计数状态，计数状态回到第一个状态。计数状态为59的下一个计数状态为00。时序仿真结果正确。

* 1. 小时计时模块（模24计数）

1. 模块功能

模块实现模24计数功能。输出计数结果10位和各位的8421BCD码。

1. 设计思路

使用74390搭建模值为24的计数器。计数状态如下：

表3.2 模24计数器计数状态表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M24 | 2QD | 2C | 2QB | 2QA | 1QD | 1QC | 1QB | 1QA | 2CLR | 1CLR |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 12 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 14 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 16 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 18 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 20 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 22 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 23 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 24 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

由计数状态表得到清零端表达式：2CLR = 1CLR = 1QC \* 2QB

1. 设计结果（电路）

文本, 信件

描述已自动生成

图3.4 模24计数器封装设计

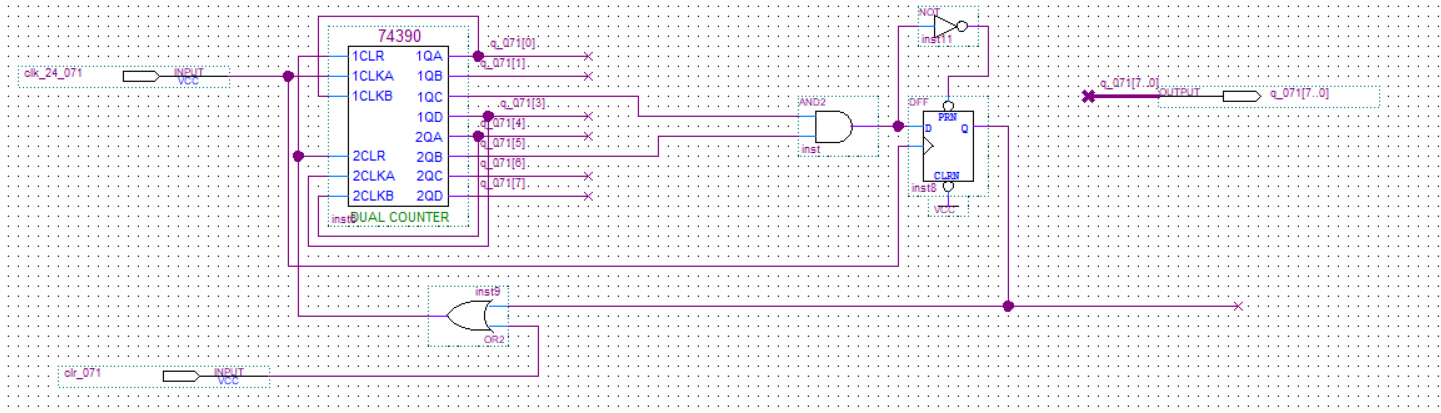


图3.5 模24计数器电路图

1. 图形用户界面, 应用程序

   中度可信度描述已自动生成仿真测试

图3.6 模24计数器波形仿真图

1. 仿真波形数据分析

由图3.6可看出，每经过24个计数状态，CO输出一次1。计数状态为23的下一个计数状态为00。时序仿真结果正确。

* 1. 星期计时模块（模7计数）

1. 模块功能

模块实现模7计数功能。输出计数结果10位和各位的8421BCD码。

1. 设计思路

使用74161搭建模值为7的计数器。计数状态如下：（因为星期没有星期0因此计数状态从1到7）

表3.3 模7计数器计数状态表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M7 | 1QC | 1QB | 1QA | 1CLR |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 |

1CLR=1CLR=1QC\*1QB\*1QA.

1. 设计结果（电路）

图片包含 文本

描述已自动生成

图3.7 模7计数器封装设计

图示, 示意图

描述已自动生成

图3.8 模7计数器电路图

1. 仿真测试

图片包含 日程表

描述已自动生成

图3.9 模7计数器波形仿真图

1. 仿真波形数据分析

由图3.9可看出，每经过7个计数状态，同步置数一次。计数7状态为的下一个计数状态为1。时序仿真结果正确。

1. 数码管动态显示模块
   1. 动态显示模块的设计

该模块主要功能是通过数码管的动态扫描实现8位数码管显示计数结果，本模块由扫描模块cnt8,位选控制模块dig\_select，数据选择模块sec\_select以及译码模块code\_select构成。

图示

描述已自动生成

图4.1 数码管动态显示模块功能框图

* 1. 扫描模块cnt8

1. 模块功能

产生数据选择模块和位选模块所需要的地址信息，扫描时钟决定位选信号和数据切换的速度。本设计用到了8位数码管，因此该模块实现模8计数器的功能，输出计数状态8421BCD码。

1. 设计思路

使用74390搭建模值为8的计数器。计数状态如下

表4.1 模8计数器计数状态表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| M8 | 1QD | 1QC | 1QB | 1QA | 1CLR |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 |

由计数状态表可知1CLR = 1QD

1. 设计结果（电路）

图片包含 文本

描述已自动生成

图4.2 扫描模块封装设计

图示, 示意图

描述已自动生成

图4.3 模8计数器电路图

1. 图形用户界面, 应用程序, 表格, Excel

   描述已自动生成仿真测试

图4.4 模8计数器波形仿真图

1. 仿真波形数据分析

由图可看出，模块实现模8计数器功能。计数状态为7的下一个计数状态为0。时序仿真结果正确。

* 1. 位选模块dig\_select

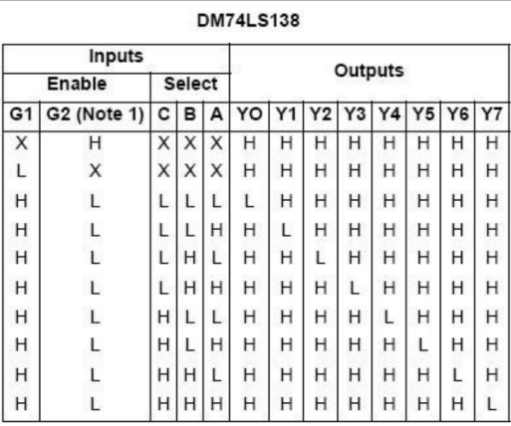
1. 模块功能

在地址端的控制下，产生位选信号。输入三位地址码add[2..0]，输出七位控制信号q[7..0],控制对应的数码管。

1. 设计思路

使用74138译码器实现模块功能。输入三位地址码add[2..0]，输出七位对应的控制信号q[7..0]。

表4.2 74138功能表



1. 设计结果（电路）

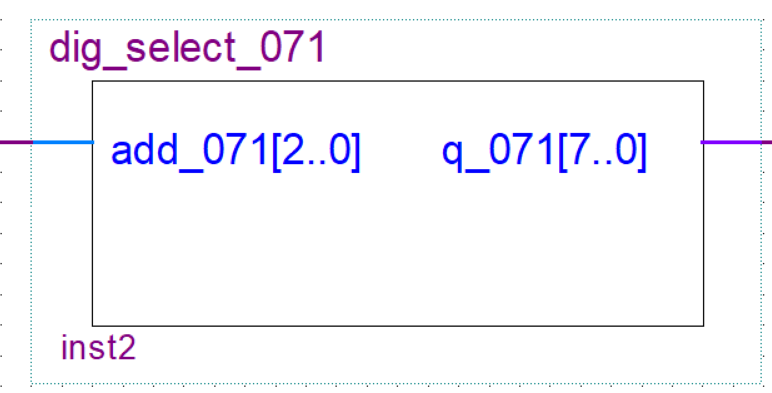


图4.5 位选模块封装设计

图示, 示意图

描述已自动生成

图 4.6 位选模块dig\_select电路图

1. 仿真测试

图4.7 位选模块dig\_select形仿真图

1. 仿真波形数据分析

由图4.7所示，输入0—7时，输出端结果从高位到低位分别是11111110、11111101……到01111111。符合74138功能表对应的输出结果。符合模块需求。时序仿真结果正确。

* 1. 数据选择模块sec\_select

1. 模块功能

输入7组数据，每组数据4bit，本模块完成在地址端的控制下从7组数据中选择1组输出。

1. 设计思路

用74151实现每一位bit位的选择。本数字钟设计第8位数码管显示星期，第6到1位数码管分别显示时、分、秒。其中第7位数码管始终显示横杠，因此数据选择器的D6均连接VCC，以配合译码模块显示数码管的g段。

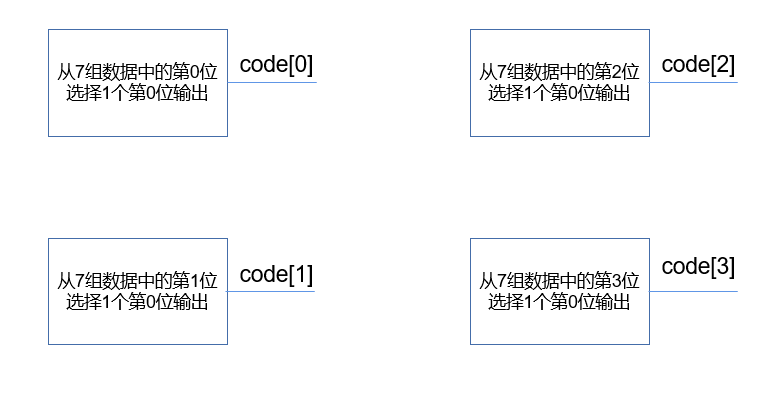


图 4.8 数据选择模块sec\_select功能框图

1. 设计结果

图片包含 示意图

描述已自动生成

图4.9 数据选择模块sec\_select电路图

1. 仿真测试

图4.10 数据选择模块sec\_select波形仿真图

1. 仿真波形数据分析

测试数据按星期、时、分、秒的顺序，分别是作者的学号后7位：2、21、00、71（实际上数字钟并不会显示71秒，此处仅用学号做仿真测试）。从第8个输出状态到第1个输出状态分别是：2、F、2、1、0、0、7、1。与输入数据一致。其中第7个状态控制显示第七位数码管的横杠，因此设置为输出F（1111）。该时序仿真结果正确。

* 1. 译码模块decoder

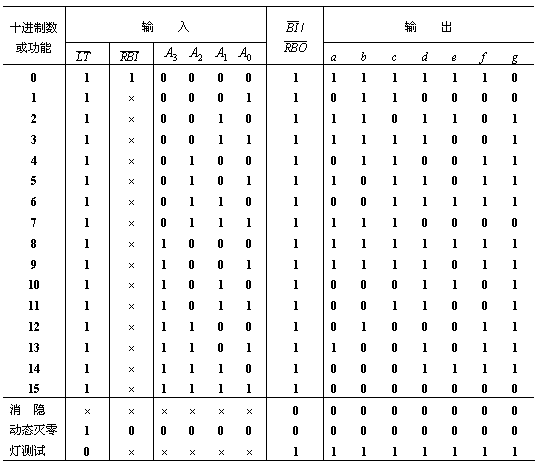
1. 模块功能

将4位8421BCD码译位7位段码，驱动共阴数码管以十进制数形式显示数字字形。当输入1111时，仅点亮g段以显示横杠。同时该模块对7448译码结果中的“6”、“9”进行补段。

1. 设计思路

使用7448译码器，配合一些门电路实现补段的译码模块。

表4.3 7448译码器功能表



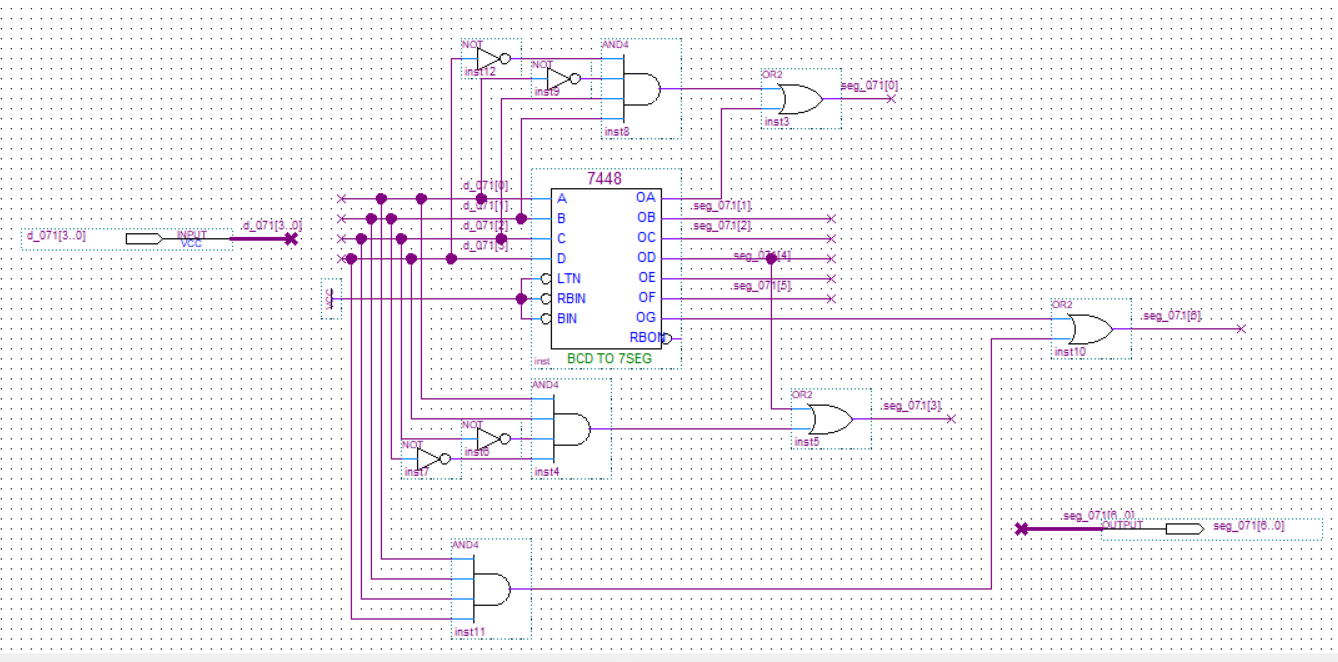
1. 设计结果

图4.11 译码模块decoder电路图

1. 仿真测试

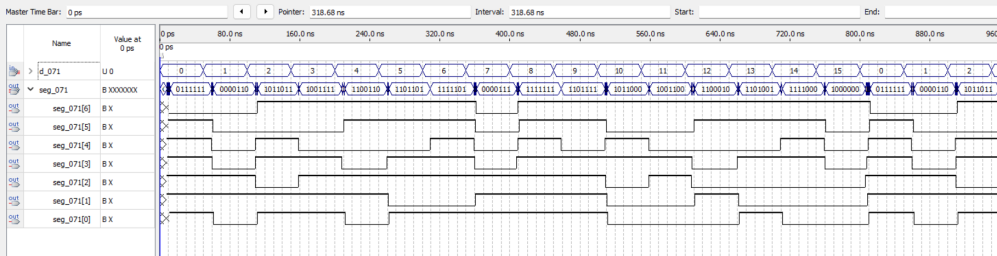


图4.12 译码模块decoder波形仿真图

1. 仿真波形数据分析

输入0、1、2、3、4、5、7、8时，输出结果与7448功能表一致。输入6，输出结果比7448功能表多输出一位Qa以实现补段功能。输入9，输出结果比7448功能表多输出一位Qd以实现补段功能。输入15时，仅输出Qg显示横杠。

* 1. 动态显示模块电路图

图示

描述已自动生成

图4.13 动态显示模块电路图

1. 其他扩展功能
   1. 设置闹钟模块
2. 模块功能

该模块实现了设置闹钟并按时响铃的功能。其中SW6控制进出闹钟模式。进入闹钟模式后，Key2、Key3、Key4可分别设置闹钟的秒钟，分钟，时钟。SW2可控制开启或关闭闹钟功能（即关闭闹钟功能后，到点不响铃）。

模块包含7个输入端，分别为add[2..0]、time\_CLR、set\_time、clk、set\_sec、set\_min、set\_hour。功能分别是：地址信号、清零信号、设置时间按钮，时钟信号、设置秒钟按键、设置分钟按键、设置时钟按键。

1. 设计思路

用两个cnt60模块输出秒钟和分钟的十位和个位的8421BCD码。用一个cnt24模块输出时钟的十位和各位的8421BCD码。用一个cnt7（模7计数器）输出星期的8421BCD码。cnt60模块、cnt24模块、cnt7模块设计思路在前文3.1、3.2、3.3已写明，此处不再赘述。输入端连接对应的cnt模块的时钟输入端，以达到每按下一次按键，设置的时间BCD码加一的功能。

表5.1.1 闹钟相关模块功能表

|  |  |  |  |
| --- | --- | --- | --- |
| 闹钟模块 | SW6 | 1 | 进入设置闹钟模式 |
| 0 | 退出设置闹钟模式 |
| Key2 | 1 | 设置闹钟秒种 |
| Key3 | 1 | 设置闹钟分钟 |
| Key4 | 1 | 设置闹钟时钟 |
| SW2 | 1 | 关闭闹钟 |
| 0 | 开启闹钟 |

1. 设计结果

a文本

描述已自动生成

图5.1.1 设置闹钟模块封装设计图

图示

描述已自动生成

图5.1.2 设置闹钟模块电路图

1. 仿真测试

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

图5.1.3 time\_CLR输入低电平的设置闹钟模块波形仿真图

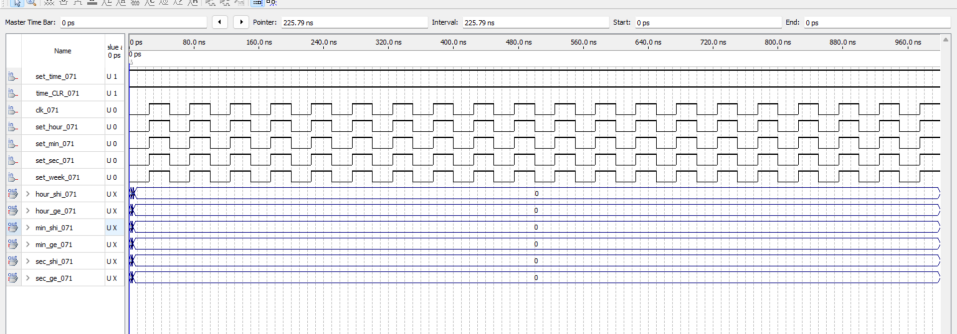


图5.1.4 time\_CLR输入高电平的设置闹钟模块波形仿真图

1. 仿真波形数据分析

由图所示，在set\_time为高电平，time\_CLR为低电平时，set\_hour、set\_min、set\_sec、每输入一次高电平，hour\_ge,min\_ge,sec\_ge输出的8421BCD码分别加1。hour\_ge,min\_ge,sec\_ge为9后再次加一输出0。hour\_ge,min\_ge,sec\_ge每经过10个状态，hour\_shi,min\_shi,sec\_shi输出的8421BCD码分别加一。

由图2所示，当time\_CLR输入端为高电平的时候，无论set\_hour、set\_min、set\_sec经过几个高电平。输出均为0。清零端信号工作正常。

* 1. 闹钟输出模块

1. 模块功能

该模块输入当前时间8421BCD码与设置闹钟时间的8421BCD码，比对当前时间8421BCD码与设置闹钟时间的8421BCD码。若一致，输出响铃信号。若不一致，则输出低电平。时间设置最小精确到某时某分的某10秒。响铃时间为10秒。此模块还有off输入端，off端输入高电平闹钟功能关闭。

1. 设计思路

输入当前时间8421BCD码与设置闹钟时间的8421BCD码，通过20个同或门分别比较设置的时间BCD码和当前时间的BCD码是否一致。用与门将所有同或门的输出相与，输入到数据选择器74151的最低位A。B、C均连接GND。若A输入为1，则数据选择器输出clk信号，否则输入低电平。

为实现蜂鸣器发出“滴滴滴”的间隔声音，数据选择器输出的clk信号会输入到另一片74151数据选择器的D1,D2,D5数据端。使用一个计数器，该计数器输入端连接10hz时钟信号，输出000-111，共8个计数状态到74151的地址输入线。由此实现了间隔声音“滴滴滴”的声响。

1. 设计结果

文本, 表格

中度可信度描述已自动生成

图5.2.1 闹钟输出模块封装设计图

图片包含 图表

描述已自动生成

图5.2.2 闹钟输出模块电路图（上）

示意图

中度可信度描述已自动生成图5.2.3 闹钟输出模块电路图（中）

图片包含 图形用户界面

描述已自动生成

图示, 示意图

描述已自动生成图5.2.4 闹钟输出模块电路图（下）

图5.2.5 控制“滴滴滴”间隔声响电路图

1. 仿真测试

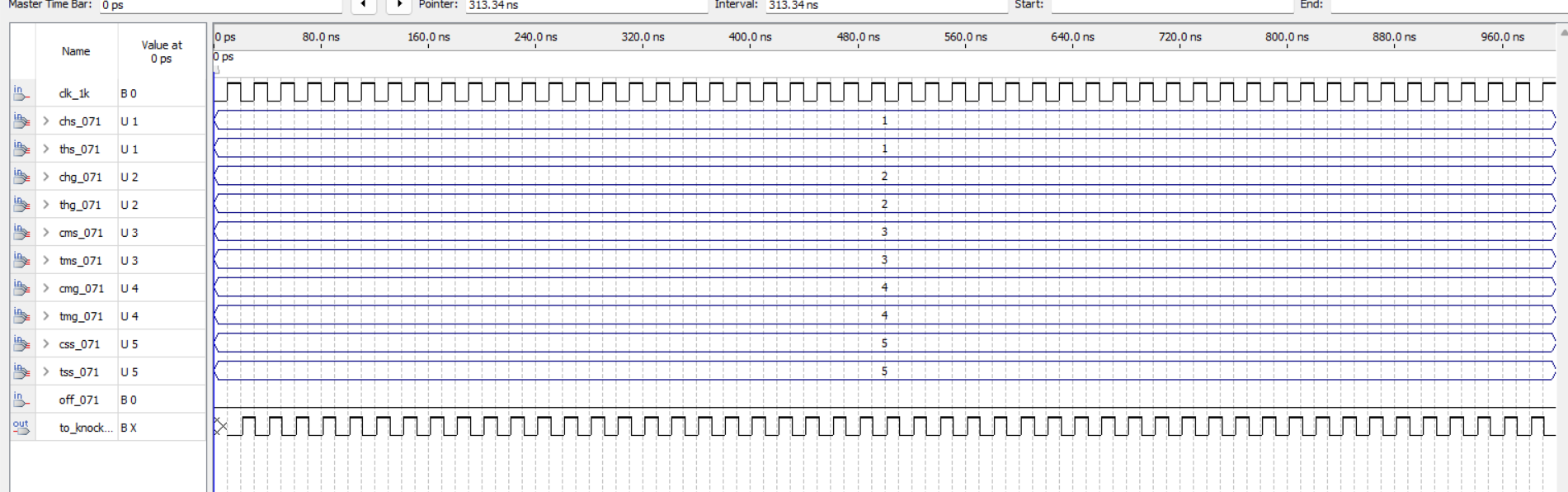


图5.2.6 到点时波形仿真

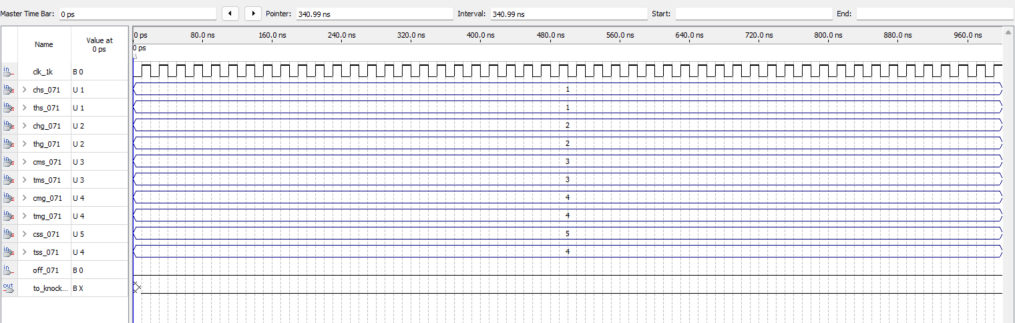


图5.2.7 未到点时波形仿真

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

图5.2.8 闹钟功能关闭时波形仿真

图片包含 图示

描述已自动生成

图5.2.9 控制“滴滴滴”间隔声响波形仿真

1. 仿真波形数据分析

如5.2.6所示，设置闹钟时间为12点34分50秒，当前时间为12点34分50秒。（也有可能是12点34分51秒到12点34分59秒中的任意某刻，因为该模块不输入秒钟的个位，时钟、分钟、秒钟十位一致即响铃）to\_knock输出端输出时钟信号。功能正常。

如图5.2.7所示，设置的闹钟时间为12点34分50秒，当前时间为12点34分40秒。to\_knock输出端输出低电平。

如图5.2.8所示，设置时间和当前时间一致。但是off输入端为高电平，闹钟功能被关闭。输出低电平。

如图5.2.9所示，clr为1时（清零信号低电平有效），输入由第一片74151输出的时钟信号后，输出波形如图所示。可以实现“滴滴滴”的间隔响声。

* 1. 整点报时

1. 模块功能

该模块实现在12点整响铃报时的功能。其中，11点59分59秒会以500Hz频率响铃一秒，12点整会以1KHz频率响铃1秒。

1. 设计思路

整点报时功能可以看作设定了特殊的闹钟。该功能创建了两个模块，一个等效于设置了11点59分59秒的闹钟，控制11点59分59秒以500Hz频率响铃一秒，一个控制12点整会以1KHz频率响铃1秒。

由于5.2所示的闹钟输出模块不进行秒钟个位的比较。因此，需要保证在确定的某一秒内响铃，需要用数据选择器控制闹钟模块的off段（使能端，低电平有效）。

sec\_ge3,sec\_ge2,sec\_ge1,sec\_ge0分别为秒钟8321BCD码。

12点整报时的真值表如下：（仅在秒钟个位数为0的时候为输出0）

表5.3.1 12点整报时模块真值表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | sec\_ge3 | sec\_ge2 | sec\_ge1 | sec\_ge0 | off |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 |

注：实际上sec\_ge3,sec\_ge2,sec\_ge1,sec\_ge0只会输入0-9的组合。但为了更好的展示使用两片3线-8线数据选择器级联成的4线-16线数据选择器的连线方式，列出了16种全部的组合。

11点59分59秒报时的真值表如下：

表5.3.2 11点59分59秒报时模块真值表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | sec\_ge3 | sec\_ge2 | sec\_ge1 | sec\_ge0 | off |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 |

为了方便连接电路，减少重复操作，还另外设计了four\_gnd模块，输出4条低电平的总线。此模块较为简单，此处不赘述。

1. 设计结果

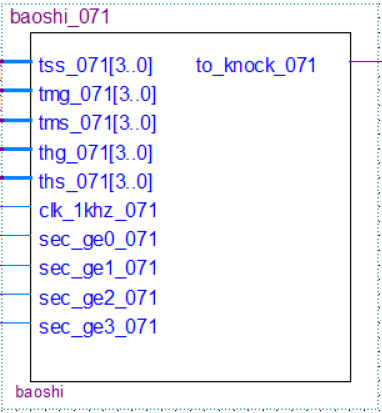


图5.3.1 12点整报时模块封装设计

电脑萤幕画面

描述已自动生成

图5.3.2 12点整报时模块电路图（上半）

图示

中度可信度描述已自动生成

图5.3.3 12点整报时模块电路图（下半）

表格

描述已自动生成

图5.3.4 11点59分59秒报时模块封装设计

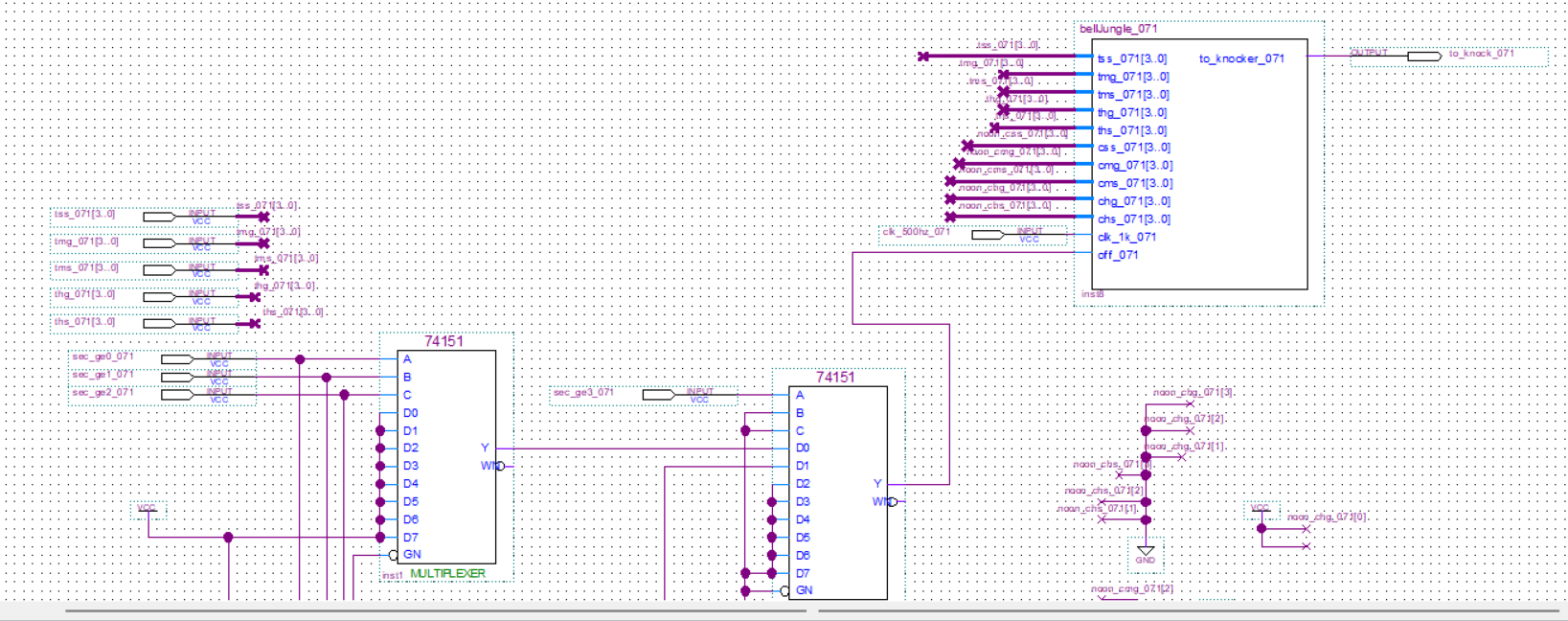


图5.3.5 11点59分59秒报时模块电路图（上半）

图片包含 示意图

描述已自动生成

图5.3.6 11点59分59秒报时模块电路图（下半）

1. 仿真测试

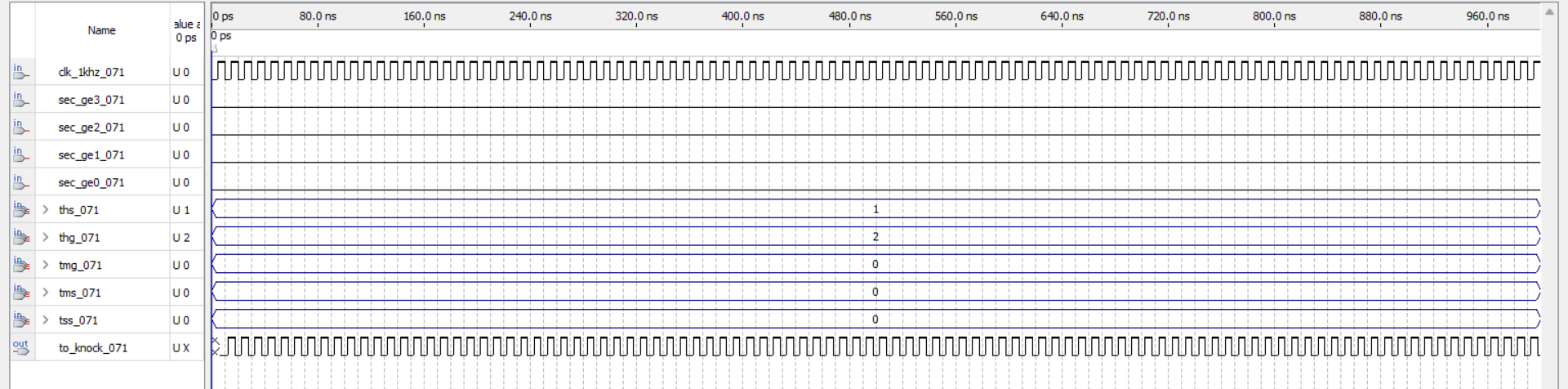


图5.3.7 12点整报时模块仿真

图片包含 应用程序

描述已自动生成

图5.3.8 12点0分1秒时波形仿真

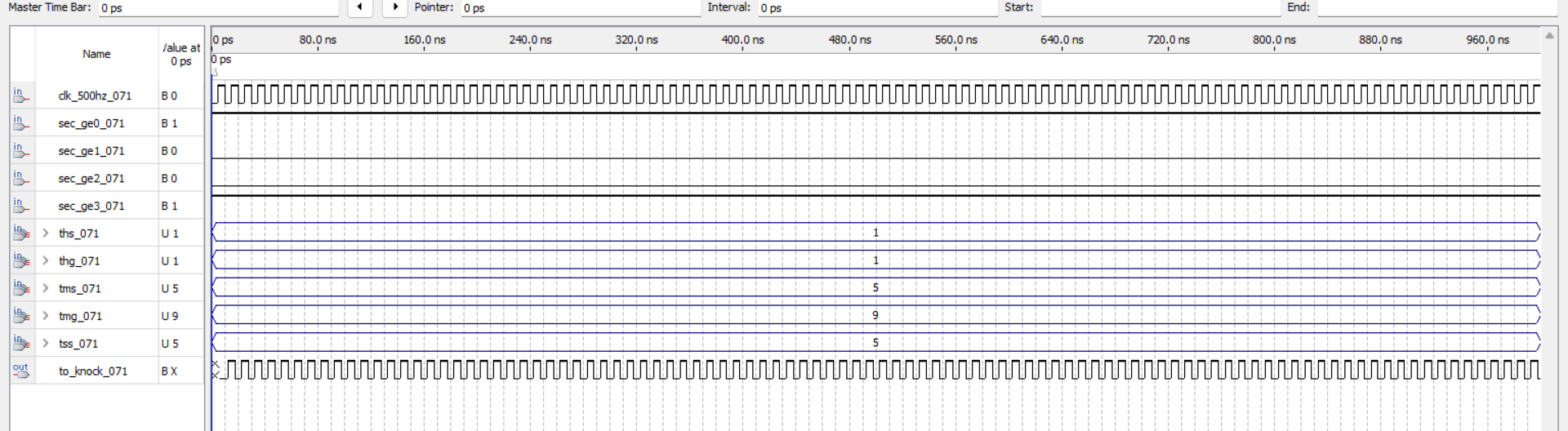


图5.3.9 11点59分59秒报时模块仿真

应用程序, 表格

中度可信度描述已自动生成

图5.3.10 11点59分58秒时波形仿真

1. 仿真波形数据分析

如图5.3.7所示，当时间为12点00分00秒时，模块输出时钟信号。发出蜂鸣声报时。

如图5.3.8所示，当时间为12点00分01秒时，模块输出低电平，蜂鸣器不发声音。

因此，12点整报时功能正确实现。

如图5.3.9所示，当时间为11点59分59秒时，模块输出时钟信号。发出蜂鸣声报时。

如图5.3.10所示，当时间为11点59分58秒时，模块输出低电平，蜂鸣器不发声音。

因此，11点59分59秒时响铃功能正确实现。

通过连接不同频率的clk输入，即可控制11点59分59秒以500Hz频率响铃一秒，控制12点整会以1KHz频率响铃1秒。

* 1. 手动校时

1. 模块功能

实现手动校准时间的功能，SW5开关置1时，计数模块停止计时，通过按键可分别使时钟、分钟、秒钟数值加一。SW5开关置0时，计数模块继续计时。

表5.4.1 手动校时模块功能表

|  |  |  |  |
| --- | --- | --- | --- |
| 校准时间模块 | SW5 | 1 | 暂停校时 |
| 0 | 开始计时 |
| Key2 | 1 | 校秒 |
| Key3 | 1 | 校分 |
| Key4 | 1 | 校时 |

1. 设计思路

该功能集成到了计时模块之中。set\_time输入端取反与clk相与。使得计时功能停止。通过set\_hour,set\_min,set\_sec按键产生的上升沿可以

1. 设计结果

图示

中度可信度描述已自动生成

图示, 示意图

描述已自动生成图5.4.1 手动校时电路图（上半）

图5.4.2 手动校时电路图（下半）

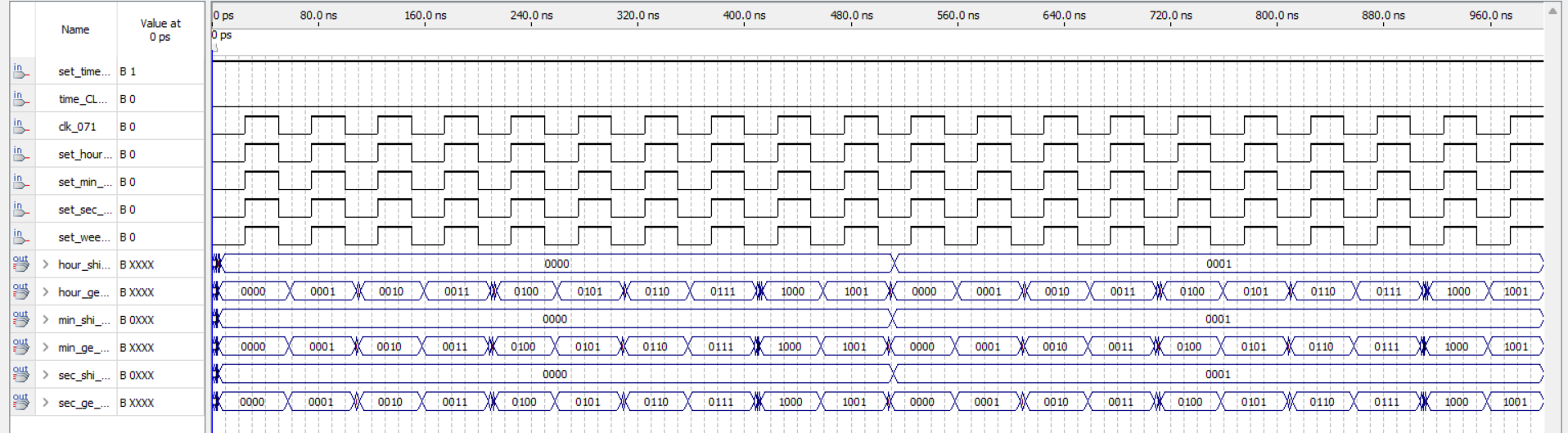
1. 仿真测试

图5.4.3 手动校时功能波形仿真

1. 仿真波形数据分析

由图5.4.3所示，在set\_time为高电平时，set\_hour、set\_min、set\_sec、每输入一次高电平，hour\_ge,min\_ge,sec\_ge输出的8421BCD码分别加1。hour\_ge,min\_ge,sec\_ge为9后再次加一输出0。hour\_ge,min\_ge,sec\_ge每经过10个状态，hour\_shi,min\_shi,sec\_shi输出的8421BCD码分别加一。

* 1. 秒表

1. 模块功能

该模块实现秒表计时功能，SW7输入1进入秒表模式，输入0退出秒表模式。

进入秒表模式后，按key6可以开始\结束计时。按key7键清零。

表5.5.1 秒表模块功能表

|  |  |  |  |
| --- | --- | --- | --- |
| 秒表模块 | SW7 | 1 | 进入秒表模式 |
| 0 | 退出秒表模式 |
| Key6 | 1 | 开始计时/结束计时 |
| Key7 | 1 | 清零 |

1. 设计思路

按键连接控制秒表开始\结束的模块。开始\结束模块由按键消抖动模块，模8计数器以及74151数据选择器组成。模8计数器的输出端接入数据选择器的输入端，数据选择器数据端D0,D2,D4,D6接地，D1,D3,D5,D7接vcc。由此实现了不断按下按键，模块交替输出低电平和高电平的功能，符合秒表使用逻辑。秒表计时功能由模100计数器，两个模60计数器组成。模100计数器输出10分秒，1分秒信号。两个模60计数器分别输出分钟和小时的信号。

1. 设计结果

文本

描述已自动生成

图5.5.1 秒表模块封装设计

图示, 示意图

描述已自动生成

图5.5.2 秒表模块电路图（左半）

日程表

低可信度描述已自动生成

图5.5.3 秒表模块电路图（右半）

文本

描述已自动生成

图5.5.4 秒表开始\停止模块封装设计

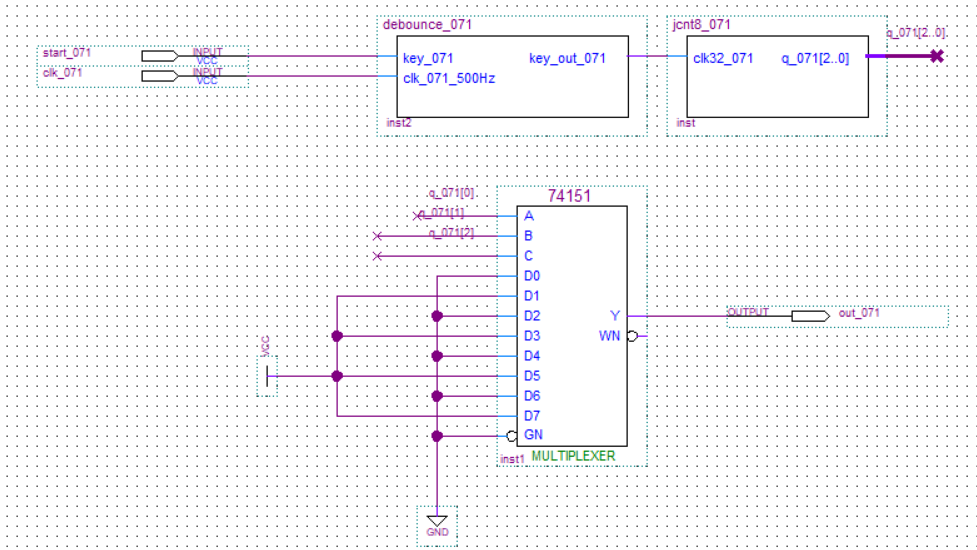


图5.5.5 秒表开始\停止模块电路图

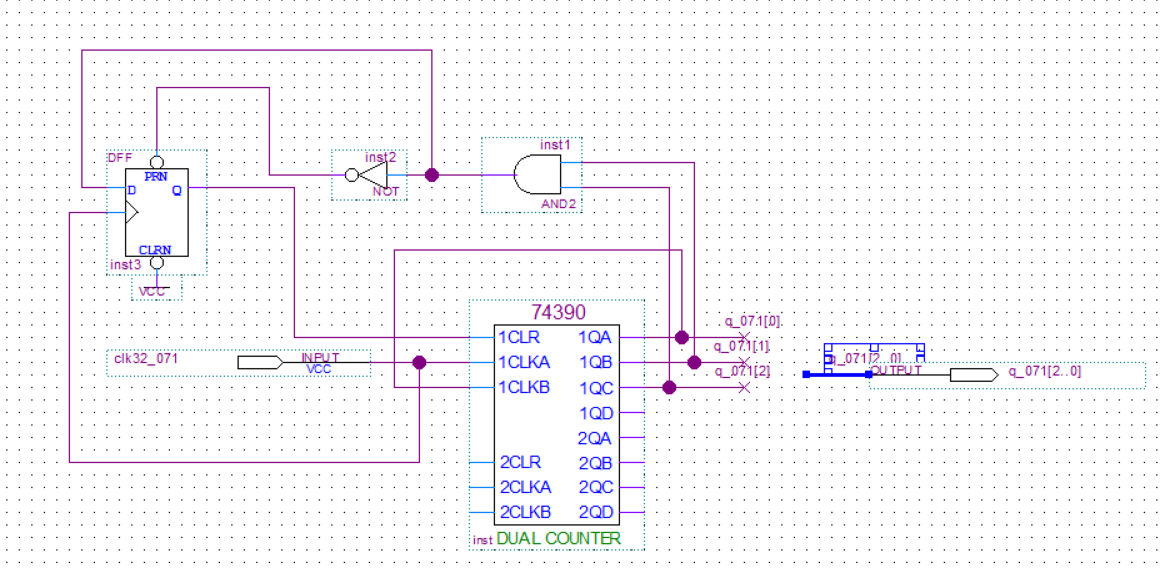


图5.5.6 模8计数器电路图

1. 图形用户界面, 应用程序, 表格, Excel

   描述已自动生成仿真测试

图5.5.7 秒表开始\停止模块波形仿真

图片包含 表格

描述已自动生成

图5.5.8 秒表模块波形仿真

图片包含 日程表

描述已自动生成

图5.5.9 秒表模块波形仿真（清零）

1. 仿真波形数据分析

如图5.5.7所示，按下按键后输出电平由低电平变为高电平。再次按下后由高电平转变为低电平。满足秒表开始\停止交替的功能需求。

如图5.5.8所示，当位选为000时（即输出1分秒位的数值），每过0.01秒code输出端输出的值加1。

如图5.5.9所示，当有清零信号时，秒表之后从0开始计时。

* 1. 模式选择模块

1. 模块功能

接收不同模块输出的code[3..0]数据输出信号，以及开关信号。以达到切换功能的效果。

1. 设计思路

表5.6.1 与模式选择模块相关的功能表

|  |  |  |
| --- | --- | --- |
| SW5 | 1 | 暂停校时 |
| 0 | 开始计时 |
| SW6 | 1 | 进入设置闹钟模式 |
| 0 | 退出设置闹钟模式 |

接收不同模块输出的code[3..0]数据输出信号，以及开关信号。以达到切换功能的效果。SW5,SW6均为0时为计时模式。

1. 设计结果

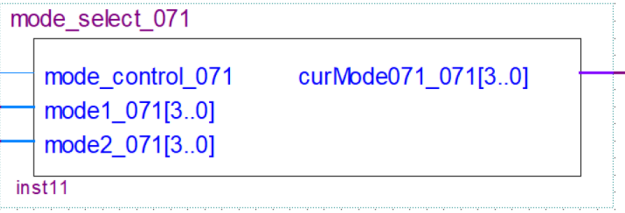


图5.6.1 模式选择模块封装设计

图示, 表格

描述已自动生成

图5.6.2 模式选择模块电路图

1. 仿真测试

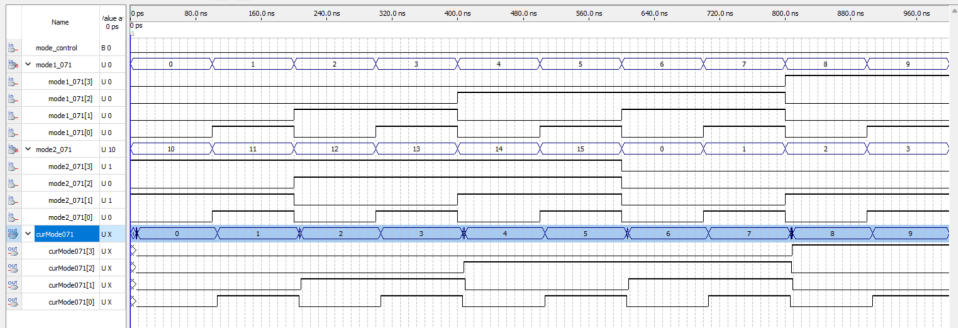


图5.6.3 模式选择波形仿真（输出模式1）

图片包含 日程表

描述已自动生成

图5.6.4 模式选择波形仿真（输出模式2）

1. 仿真波形数据分析

如图1、2，输入两种不同的数据信号mode1,mode2。mode\_control输入端为0时,输出为mode1的波形。mode\_control输入端为1时，输出为mode2的波形。

* 1. 按键防抖模块

1. 模块功能

消除按键的电平抖动，使按下一次按键只产生一次电平变化。

1. 设计思路

利用D触发器的延时特性，将每一个D触发器（延时不同）的输出端相与，仅当所有D触发器都输出为1的时候，消抖模块才输出为1，这样就可以将按键抖动的部分（有1也有0的部分）跳过，达成消抖的目的。

1. 设计结果

图形用户界面, 文本, 应用程序

描述已自动生成  
图5.7.1 按键消抖封装设计

图示

描述已自动生成

图5.7.2 按键消抖模块电路图

1. 仿真测试

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

图5.7.3 按下时间较长时按键消抖模块波形仿真

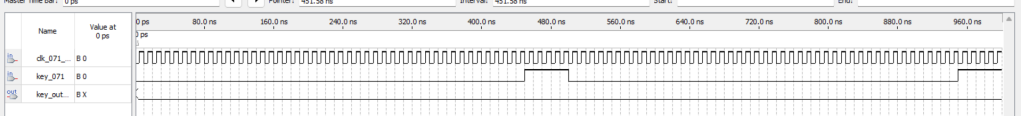


图5.7.4 按下时间较短时按键消抖模块波形仿真

1. 仿真波形数据分析

如图5.7.3、图5.7.4所示，当按键输入连续80ns以上，输出端才输出高电平。模块由8个D触发器组成，每个D触发器延时10ns。波形仿真正确。

1. 系统总体测试

系统总体测试结果如“附件一 评分细则及测试原始数据记录”。

1. 系统设计实现过程中遇到的主要问题、解决思路和解决方案

在设计的过程中，遇过不少问题。

①多个输出端相连，一个输入端连接多条线路导致的报错。解决问题的重要途径是阅读报错信息，定位具体的错误位置，检查是哪些输入端连接错误。

②在双击模块的时候就很容易误操作，导致被点击的模块连接上不必要的线路，这种错误常常是在不自知的情况下产生的，非常隐蔽，因此更需要阅读报错提示，细致地重连电路。

③修改模块后，出现编译错误。这种错误常常是因为修改模块后没有修改对应的原理图和顶层设计导致的。应当仔细阅读报错信息，将出错位置的元件更改为修改后的文件。封装设计上的端口名要与内部端口名一致。

④软件中选中的芯片型号与所用的不同。该错误可能导致无法设定真确的引脚序号。通常情况下，软件会对此问题进行报错。应当选择正确的芯片型号后重新编译。

⑤此外，还有一些未知随机问题有时出现，比如无法将设计下载到开发板上，或者下载到开发板上的结果与预期不符但设计本身没有问题。这类情况可能是由软件bug或者一些较为隐蔽的问题导致的。面对这类问题应当沉着冷静，通过控制变量的方法确定出现问题的设备。比如，可以通过更换连接开发板的电脑USB接口解决USB接触不良的问题。重新编译或重新选择下载到开发板的文件以避免下载了旧版本或者错误的文件到开发板上。

心得体会

FPGA技术为我们提供了更加灵活的设计方案，可以快速地实现各种功能。在电子钟的设计过程中，需要首先确定其基本功能，例如显示时间、设置闹钟等。然后，需要考虑如何集成这些功能，并确保它们能够稳定地运行在FPGA芯片上。在测试和调试阶段，需要通过仿真和实际测试来验证每个模块的功能和正确性，并进一步优化和完善系统的性能。设计这个电子钟的过程中经历了很多的工作和思考，从中也获得了不少经验和收获。