

**学生实验实习报告册**

装

订

线

|  |  |
| --- | --- |
| 学年学期： | 2023-2024 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： | 计算机科学与技术学院  计算机科学与技术专业 |
| 班 级： |  |
| 姓 名： |  |
| 学 号： |  |
| 序 号： |  |
| 指导教师： | 黄沛昱 |

**重庆邮电大学教务处制**

**摘要**

首先应用 QuartusII13.1 软件采用模块化设计方法完成了基于 FPGA 的数字电子钟的设计与实现，在实现时钟基本计数的基础上，完成了清零、暂停、整点报时、Led流水灯和星期显示调功能。本报告对该前述工作的实现进行了设计原理上的文字与图片描述，电路图的截取，同时生成了相应模块的时序仿真图。最后，对实现过程中遇到的问题与解决问题的思路和方法进行了梳理总结。

**关键词：**QuartusII FPGA 数字电子钟 原理设计 电路图 时序仿真图

目录

[1 、系统顶层模块设计 3](#_Toc28290)

[1.1 系统功能： 3](#_Toc23017)

[1.2 设计思路： 3](#_Toc27930)

[2 、分频模块电路设计及仿真 5](#_Toc25636)

[3 、计时模块设计及仿真 7](#_Toc9528)

[3.1 分、秒计时模块（模60计数） 7](#_Toc12851)

[3.2 小时计时模块（模24计数） 8](#_Toc14763)

[4 、数码管动态显示模块 1](#_Toc26364)0

[4.1 动态显示模块的设计 1](#_Toc26376)0

[4.2 扫描模块cnt6 1](#_Toc5463)1

[4.3 位选模块dig\_select 1](#_Toc27585)2

[4.4 数据选择模块code\_select 1](#_Toc15985)4

[4.5 译码模块decoder 1](#_Toc5828)5

[4.6 动态显示模块电路图 1](#_Toc26128)7

[5 、其他扩展功能 1](#_Toc21261)7

[5.1 调频模块spead\_select 1](#_Toc4197)7

[5.2 清零功能 1](#_Toc19617)9

[5.3 暂停功能 2](#_Toc30609)1

[5.4 显示星期 2](#_Toc25225)2

[5.5 6、9补段 2](#_Toc25225)4

[5.6 八位数码管显示 25](#_Toc32313)

[5.7 整点报时功能 25](#_Toc30092)

[5.8 Led流水灯功能 27](#_Toc20449)

[6 、系统总体测试 28](#_Toc25003)

[7 、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 28](#_Toc25823)

[7.1 问题1 28](#_Toc8381)

[7.2 问题2 28](#_Toc8381)

[7.3 问题3 28](#_Toc8381)

[7.4 问题4 28](#_Toc8381)

[7.5 问题5 28](#_Toc8381)

[8 、心得体会 29](#_Toc3877)

[9 、参考文献 29](#_Toc18068)

# **、系统顶层模块设计**

表1 按键与功能表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 开关 | 状态 | 功能 |
| 分频模块 | SW1SW2 | 00 | 输入1Hz |
| 01 | 输入500Hz |
| 10 | 输入1Khz |
| 11 | 输入0Hz |
| 清零模块 | SW3 | 1 | 清零 |
| 暂停功能 | SW1SW2 | 11 | 暂停 |

## **系统功能：**

1. 实现了数字电子钟基本的秒、分、小时计数及进位；
2. 可以通过switch开关（S[1]和S[2]）选择输入的时钟频率从而检测电子钟功能；
3. 通过switch开关(S[3])对电子钟进行清零；
4. 通过switch开关（S[1]和S[2]）,当S1,S2均被按下时则进行暂停；
5. 用最左边的一位数码管显示星期一到星期日（0-6）；
6. 对6、9数码管显示进行了补段；
7. 用8位数码管显示，用“-”作为分隔符，显示格式：星期-小时分钟秒；
8. 当时钟上显示的时间为一个整数时，蜂鸣器响1秒。从而实现整个报时的功能；
9. Led灯以1hz的频率移动，即以流水灯的形式展示

## **设计思路：**

1. 将50MHZ时钟信号通过分频器转换为三种频率输出。
2. 将500HZ信号提供给m8和dig\_select\_34模块，相当于进行屏幕刷新，使8个数码管同时显示内容。
3. 添加分频选择模块，将分频器输出的三种情况及一个接地输入根据分频选择switch开关输 入的信号决定频率输出，其中1HZ信号即可当作秒信号用于数字电子钟正常模式，500HZ和1KHZ则分别用来检测时钟信号选择，接地输入即为0HZ，作为暂停功能。
4. 设计模60计数器、模24计数器和模7计数器，将分频选择后输出的信号输入到第一个模60计数器，当计满60个时钟信号，进一位给第二个模60计数器，相当于过了一分钟，第二个模60计数器计满60个时钟信号则相当于过了一小时，进一位给模24计数器，模24计数器计满24个信号，则相当于过了24个小时，进一位给模7计数器用来表示星期。另外注意每次进位清零。
5. 将模60计数器、模24计数器、模7计数器分别添加清零输入，将switch开关（s[3]）同时接三个计数器的清零端，当按下时进行清零。
6. 设计docker模块即有译码模块，用来控制数码管的显示，通过输入的二进制码输出数码管应该显示的内容，同时6，9进行判断，加上与门进行补段，此外规定输入二进制码为1111时输出“-”。
7. 设计数据选择模块，将7组输入以总线的形式输入到数据选择模块，7组输入分别代表最左边的1个数码管和右边6个数码管显示的数字，由地址端控制选择哪组输出（注意当地址端输入0110时，因为其中星期和时间中间为“-”，所以规定输出1111）。
8. 当时钟显示为整点时，分钟和秒钟的个位和十位都为0.所以根据这一点来设计整点报时电路，即应该将秒的个位或十位相或得到，将或非的输出端与时钟信号相与，最后再输出。故而实现了整点报时的功能。而当其他非整点时候则不能满足上述条件，从而蜂鸣器不会叫。
9. 需向Led灯模块提供1hz的频率能实现Led流水灯。

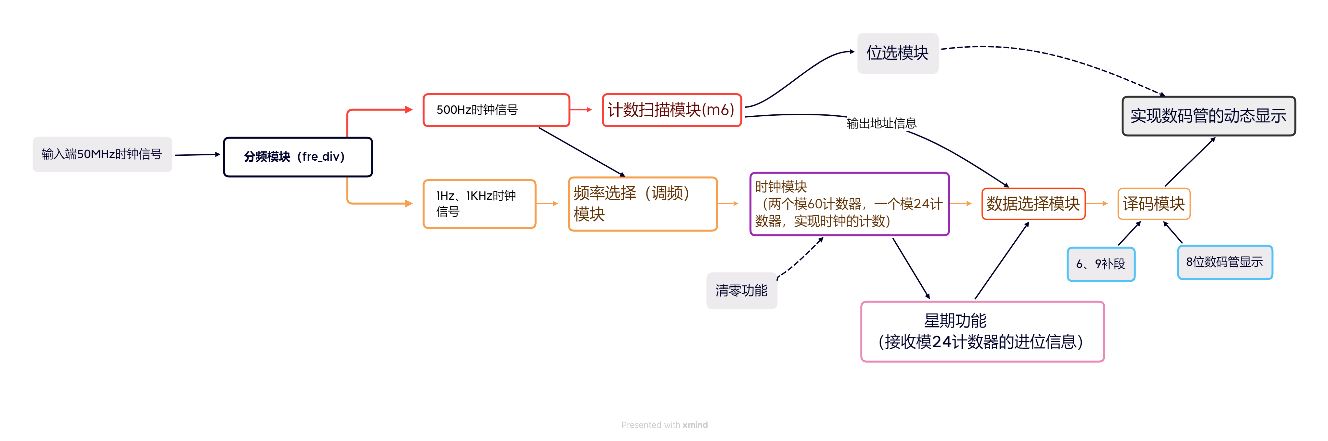


图1 系统顶层模块设计思路

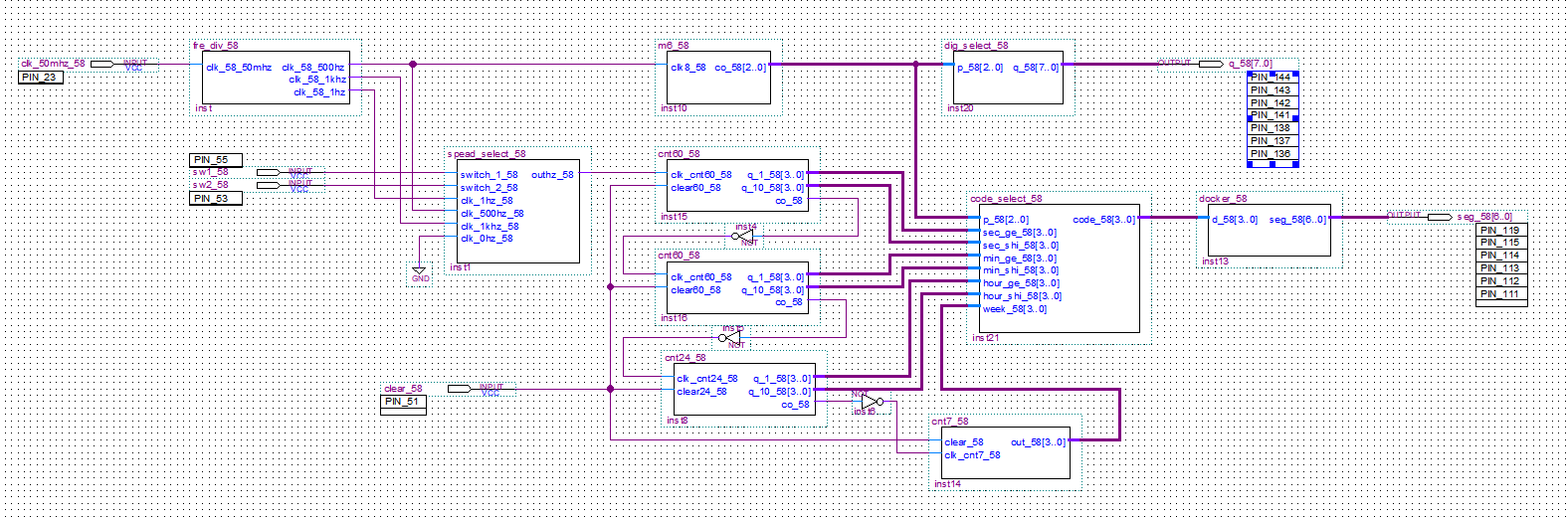


图2 系统顶层电路图

1. **、分频模块电路设计及仿真**

### 模块功能

该模块主要用于数字电子钟的正常走时和调频加速（用于进行时分秒检测）。输入50MHz的时钟信号，输出1Hz、500Hz、1KHz的频率。

### 设计思路

利用74390芯片，将50MHZ的时钟信号经过两次模100（图4）的分频后得到5KHZ的时钟信号，接入74390的1CLKB端，可以通过1QA输出得到500HZ信号，通过1QD输出得到1KHZ信号，将1QA接入2CLKB端，通过模5计数器进行5分频后再经过一个模100（图4）分频即可得到1HZ信号。最终结果见图3。

### 设计结果（电路）

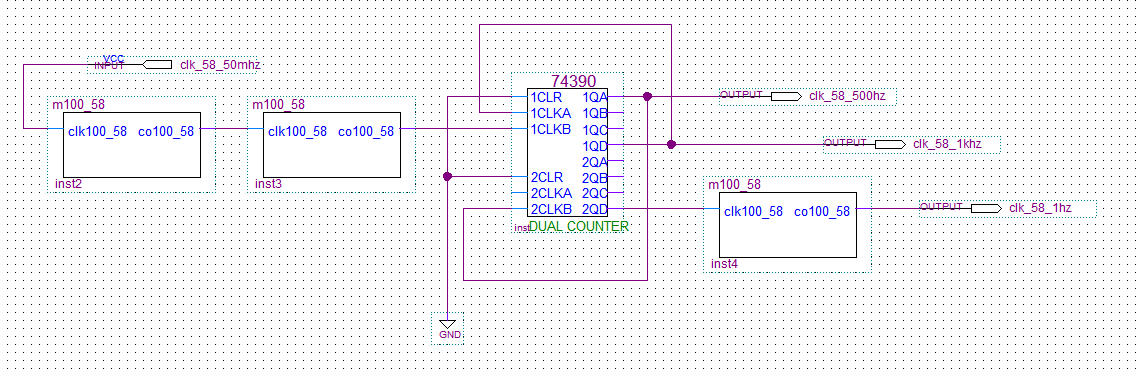


图3 分频模块顶层电路图

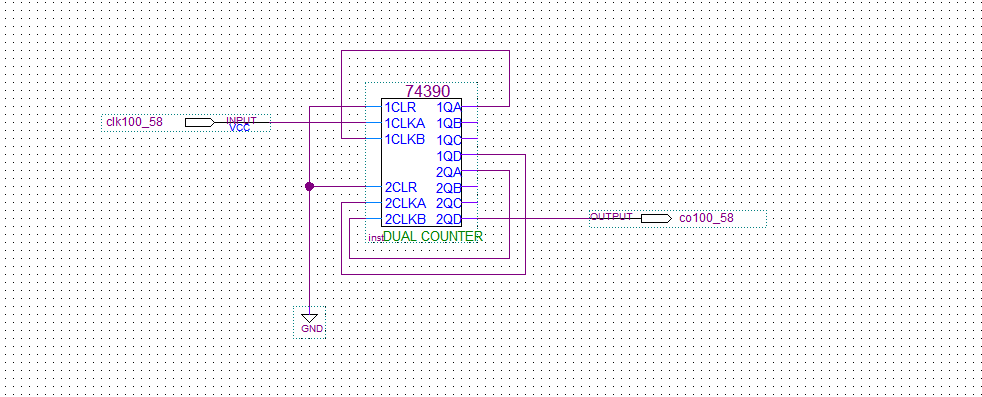


图4 模100计数器电路图

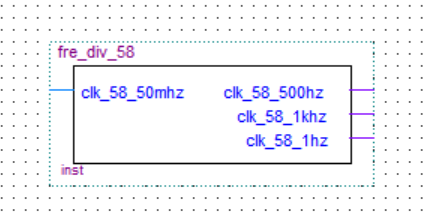


图5 分频模块封装图

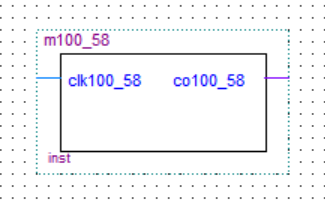


图6 模100计数器封装图

### 测试结果

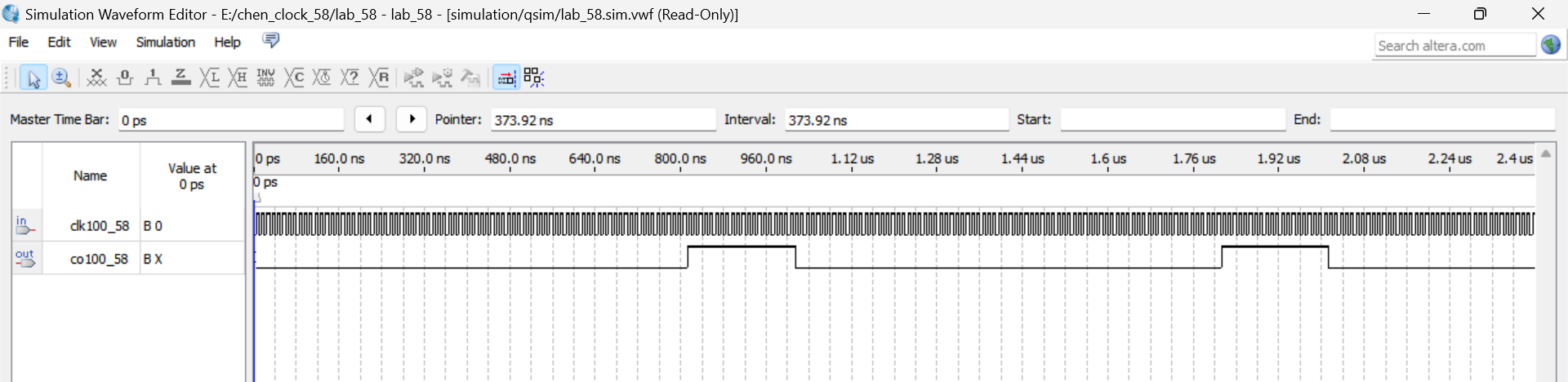


图7 模100 时序仿真波形结果

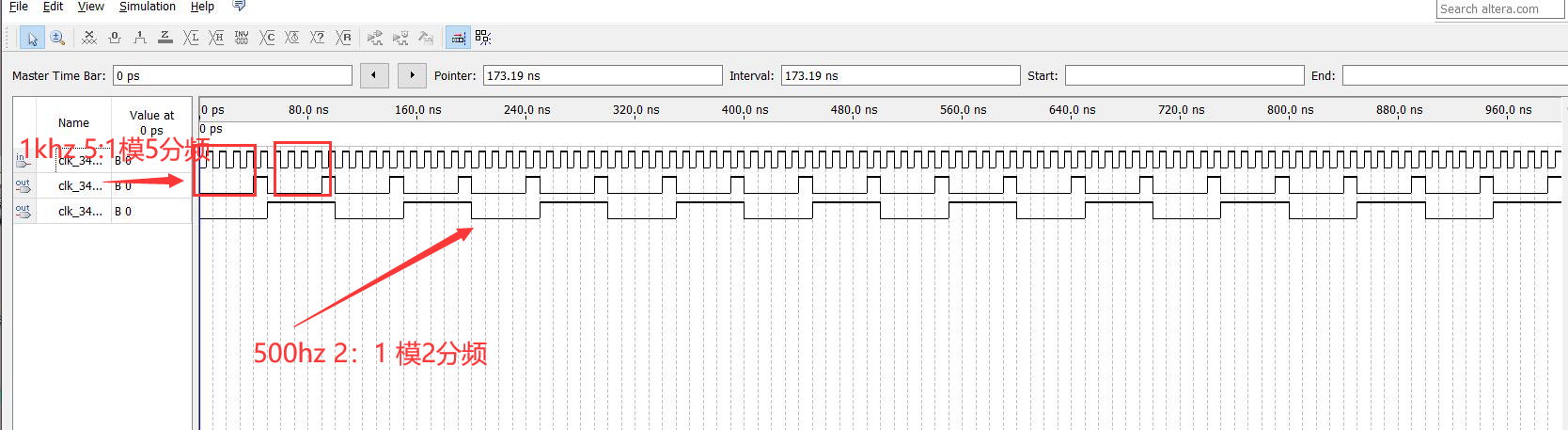


图8 5Khz分频1KHz、500Hz时序仿真波形

### 仿真波形数据分析

由于无法直接仿真50MHz的时钟信号，这里仿真50Mhz信号通过两次m100后成为5Khz再进行分频1kHz、500Hz的模拟。由上图可确定，5kHz信号经过一个模5得到1kHz,之后经过模2得到500hz信号；经过模5和模100分频器得到1kHz。

# **、计时模块设计及仿真**

## **分、秒计时模块（模60计数）**

### 模块功能（计数、进位）

实现模60的计数功能，用来表示分钟和秒及其进位。其中1QD-1QA为个位数字的二进

制数输出，2QD-2QA为十位数字的二进制数输出，当计数达到60时清零，输出进位信号。同时外连清零输入信号，实现手动清零。

### 设计思路

利用74390芯片实现模60计数器的设计。时钟信号接入1CLKA端，1QA连接1CLKB，1QD连接2CLKA，2QA连接2CLKB，1QD-1QA为个位二进制数的输出，2QD-2QA为十位二进制数的输出。当个位为0000以及十位为0110时清零，实现模60计数的功能。74390芯片清零端高电平有效，将十位的2QB和2QC相与后先用DFF进行信号延时，然后接为进位信号输出，再将其与清零输入信号相或后接入1CLR和2CLR，实现当二者其中一个为1时立即清零的功能。

### 设计结果（电路）

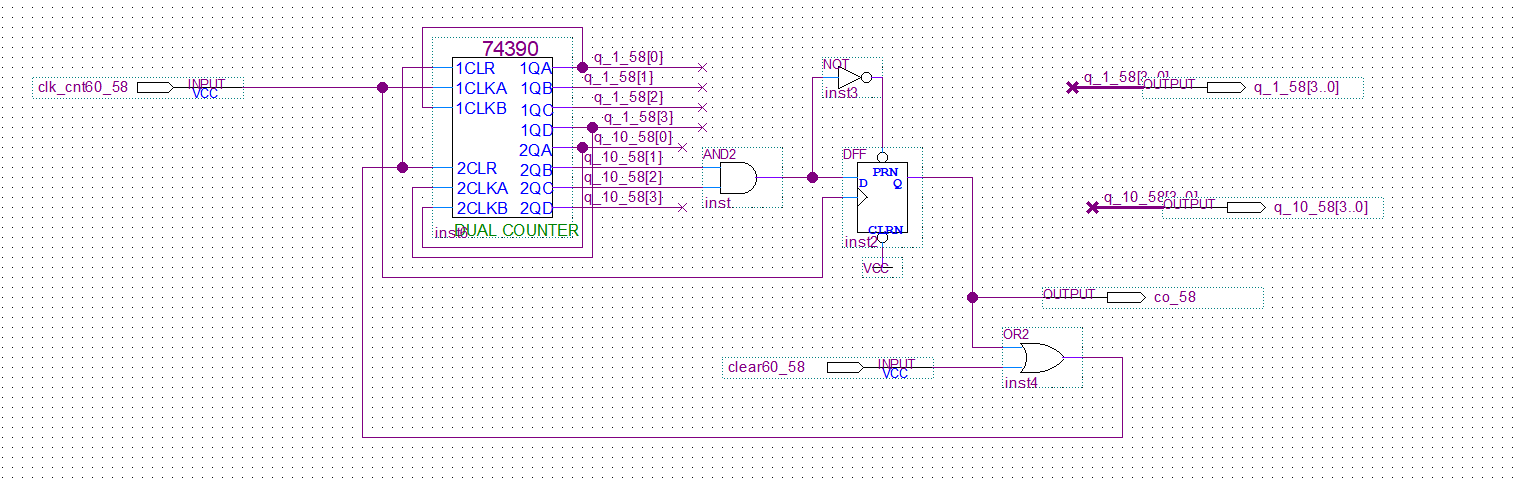


图9 模60计数器电路图

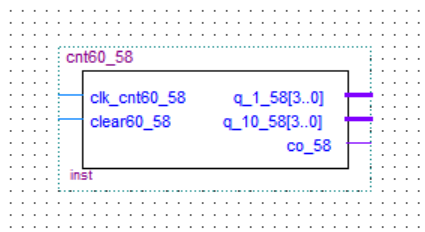


图10 模60计数器封装图

### 仿真测试

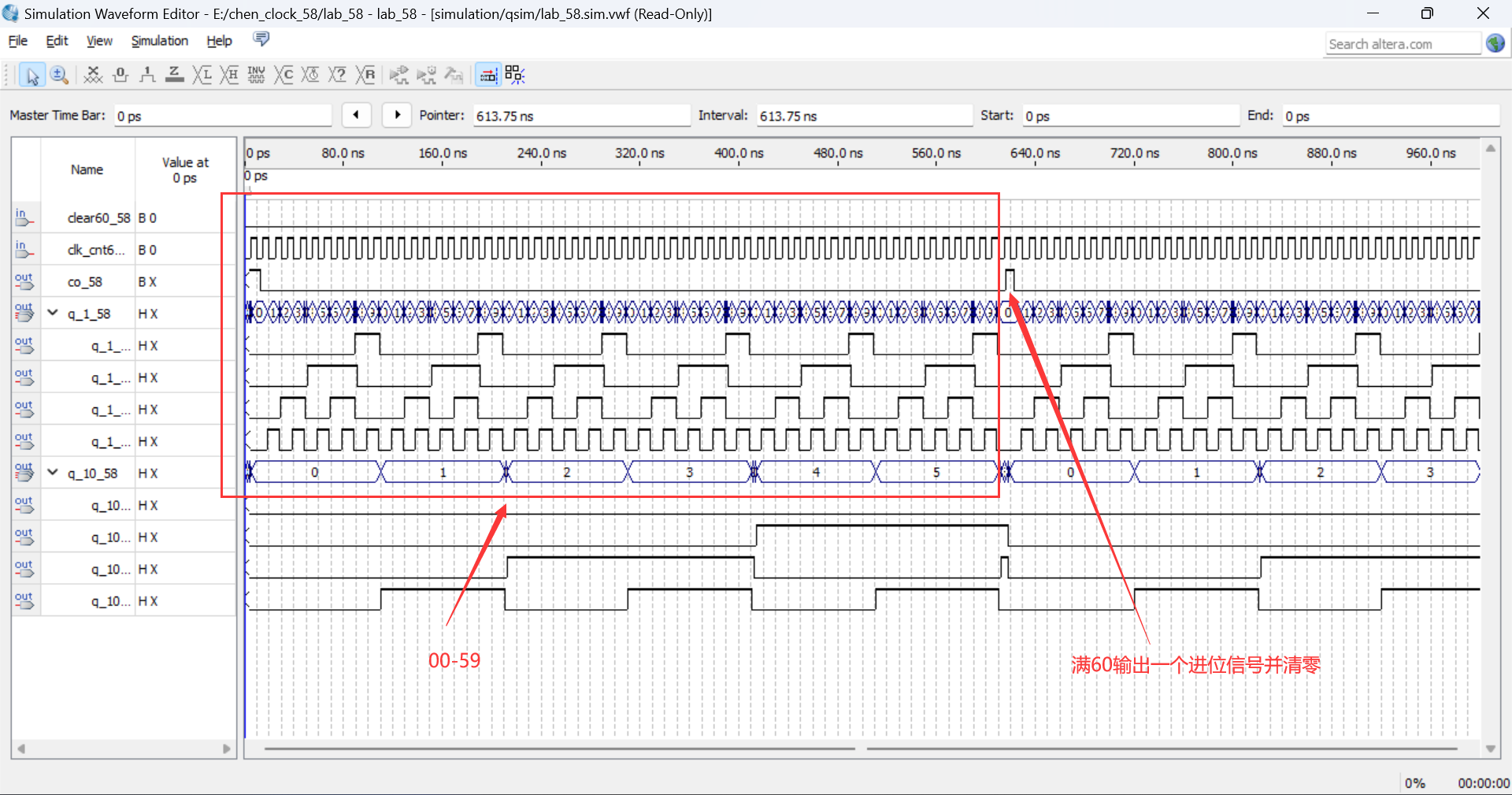


图11 模60时序仿真结果，显示了00-59-00

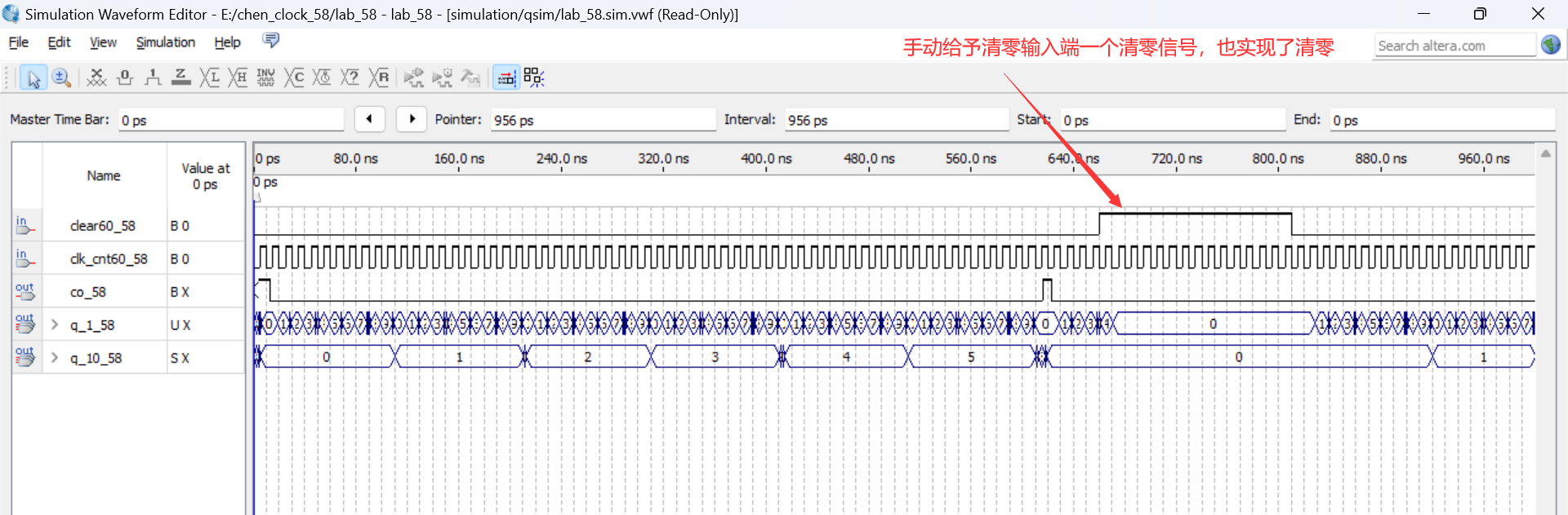


图12 模60时序仿真结果，测试了手动给予清零信号

### 仿真波形数据分析

如图八所示，模60仿真时序测试显示00-59计数，满60输出一个进位信号并清零；图九所示，手动给予清零输入端一个清零信号，也实现了清零。

## **小时计时模块（模24计数）**

### 模块功能

实现模24的计数功能，用来表示小时及其进位作为星期的计数。1QD-1QA为个位数字的二进制数输出，2QD-2QA为十位数字的二进制数输出，当计数达到24时清零，输出进位信号。同时外连清零输入信号，实现手动清零。

### 设计思路

利用74390芯片实现模24计数器的设计。时钟信号接入1CLKA端，1QA连接1CLKB，1QD连接2CLKA，2QA连接2CLKB，1QD-1QA为个位二进制数的输出，2QD-2QA为十位二进制数的输出。当个位为0100并且十位为0010时清零，实现模24计数的功能。将进位(清零)信号与清零输入相或后接入1CLR和2CLR，实现当二者其中一个为1时立即清零的功能。另外其中的D触发器是为了消除实际电路中的竞争冒险现象。

### 设计结果（电路）

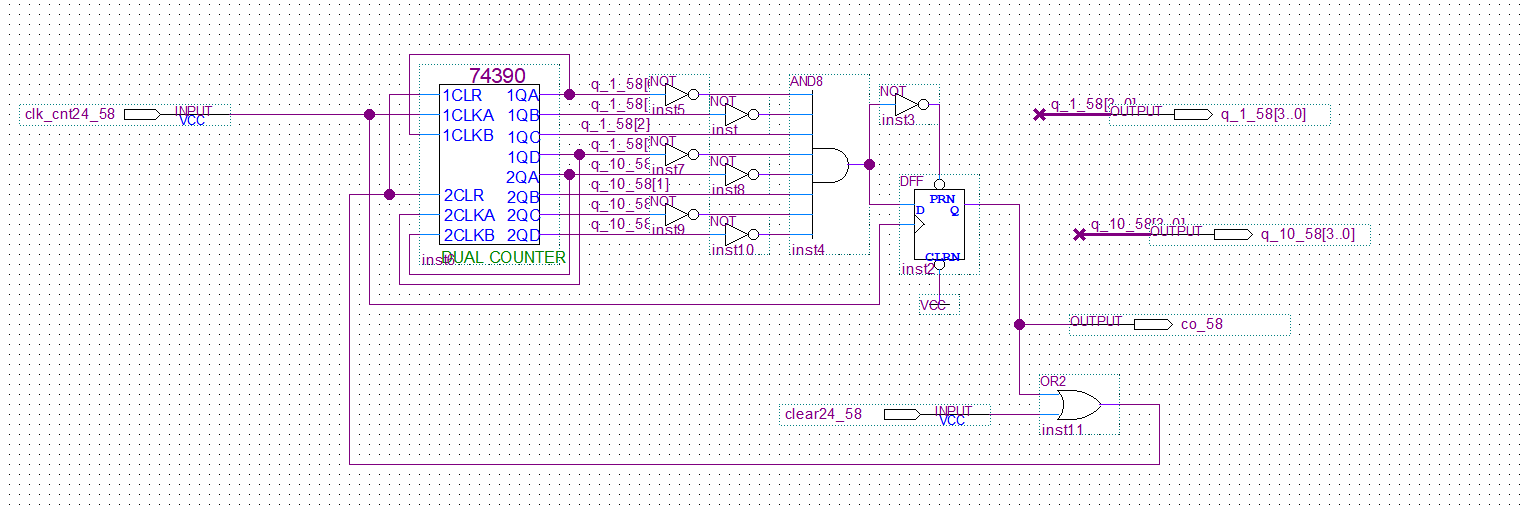


图13 模24计数器电路图

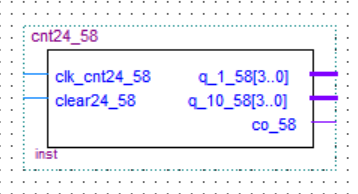


图14 模24计数器封装图

### 仿真测试

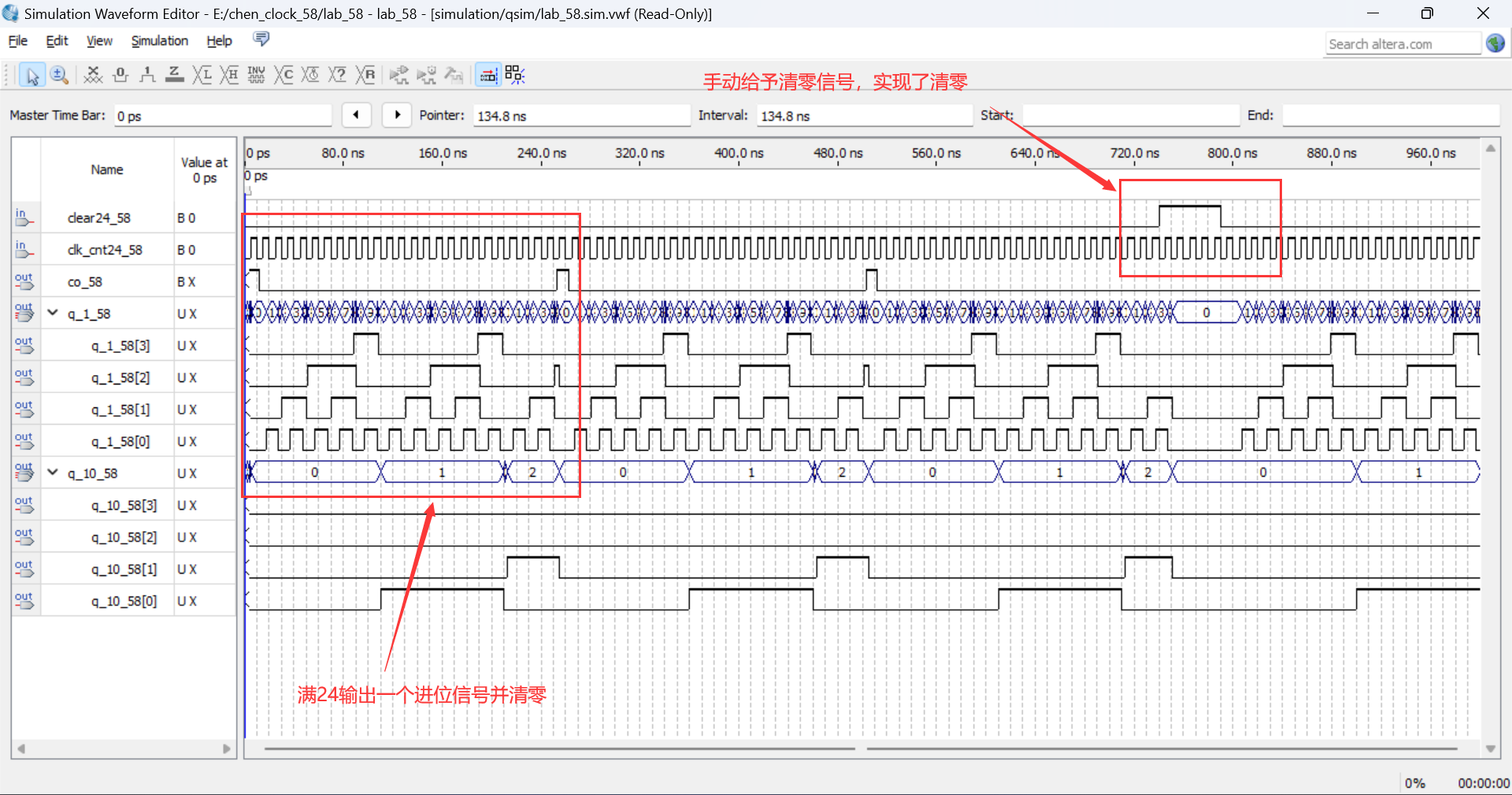


图15 模24仿真波形结果

### 仿真波形数据分析

如图11所示，模24会满24输出一个进位信号并清零，手动给予清零信号也实现了清零。

# **、数码管动态显示模块**

## **动态显示模块的设计**

通过对数码管进行动态扫描，实现8位数码管显示相应结果，从左往右，其中第一个数码管用于显示星期的计数结果，第二个数码管时钟显示“-”，剩余6个数码管显示小时、分钟、秒的计数结果。模块由扫描模块m6(一个模8计数器)、位选模块dig\_select\_58、数据选择模块code\_select、译码模块docker\_58构成。

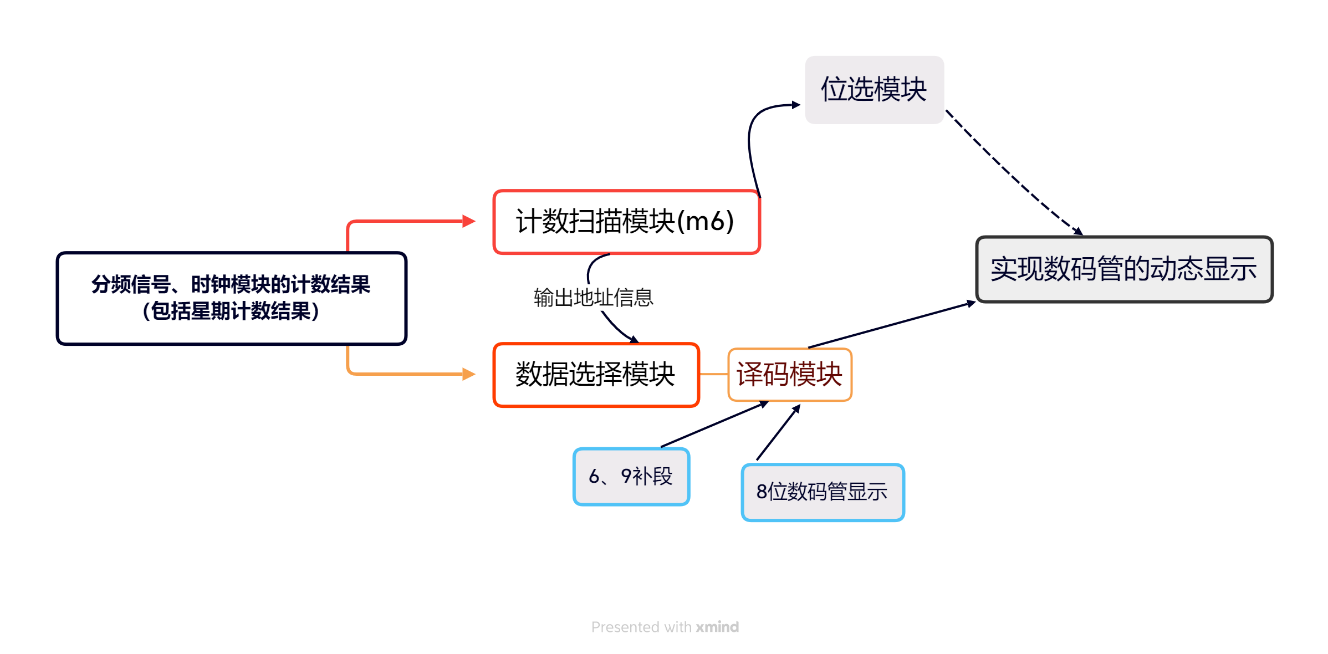


图16 动态显示模块设计框图

## **扫描模块cnt6 (模8计数器)**

### 模块功能

实现了模8计数的功能，当计数达到8时清零，将计数结果作为位选模块dig\_select\_58和数据选择模块code\_select所需的地址信息。可以理解为进行屏幕刷新。

### 设计思路

在实验九中模6计数器的基础上进行了修改，当输出为1000时清零，连接D触发器对信号做延续，然后连接到1CLR端进行清零。从而实现了模8计数的效果。

### 设计结果（电路）

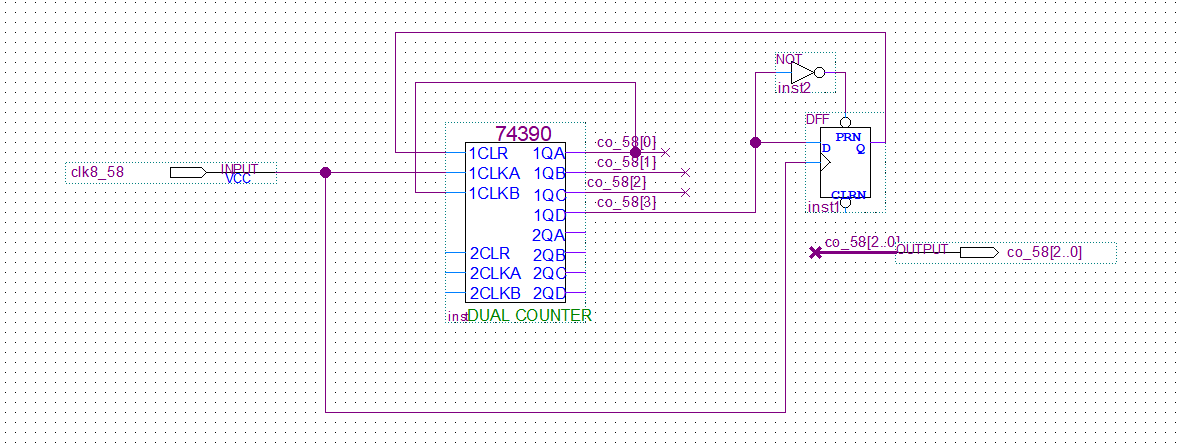


图17 模8计数器电路图

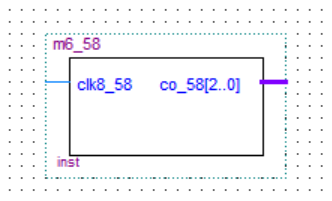


图18 模8计数器封装图

### 仿真测试

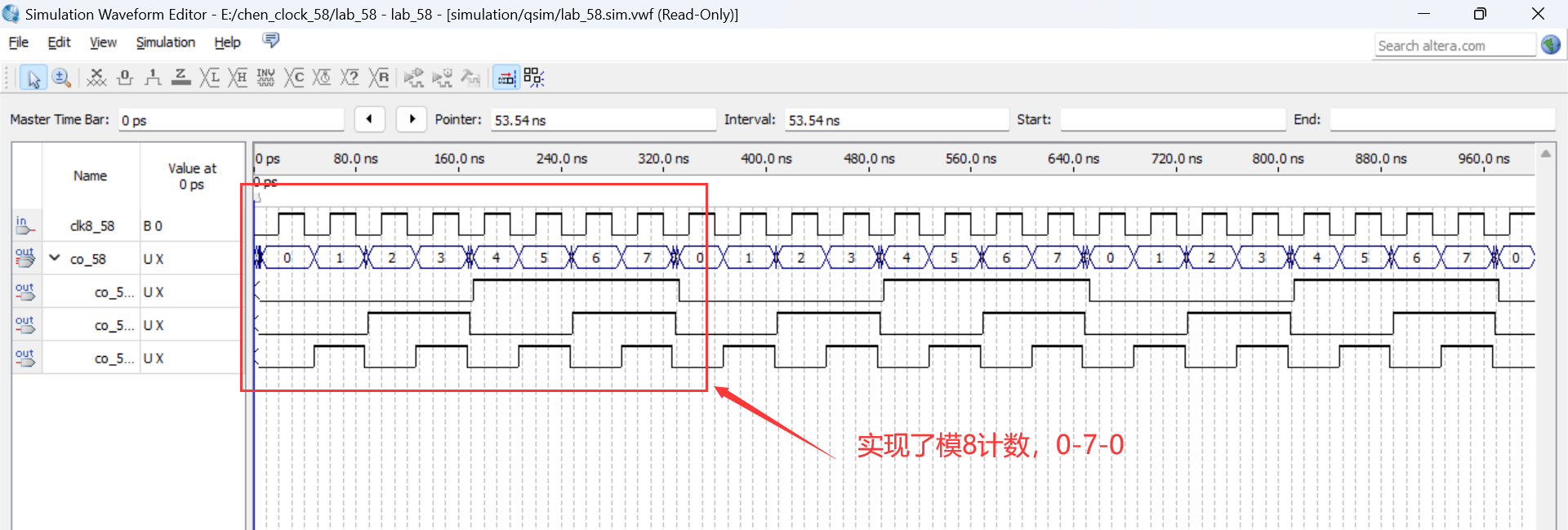


图19 模8计数器仿真结果

### 仿真波形数据分析

如图14所示,模8计数器能实现模8计数，0-7，满7输出一个进位信号并清零。

## **位选模块dig\_select**

### 模块功能

该模块负责选择哪一个数码管显示数字，add端输入二进制码选择第几个数码管显示，由于输入的频率为500Hz，人眼分辨不出来一个个显示的过程，即看起来就是六个数码管同时都在亮。

### 设计思路

选用74138（3线-8线译码器），将扫描模块模8计数器的输出作为为输入信号，当位选信号输入时，输出低电平有效，即使对应数码管工作。

### 设计结果（电路）

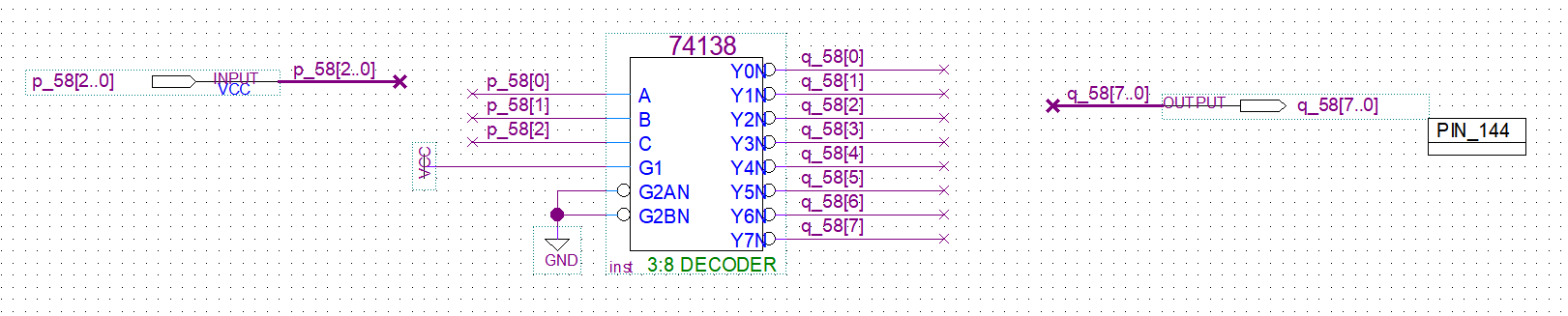


图20 位选模块dig\_select\_58电路图

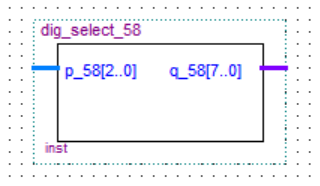


图21 位选模块dig\_select\_58封装图

### 仿真测试

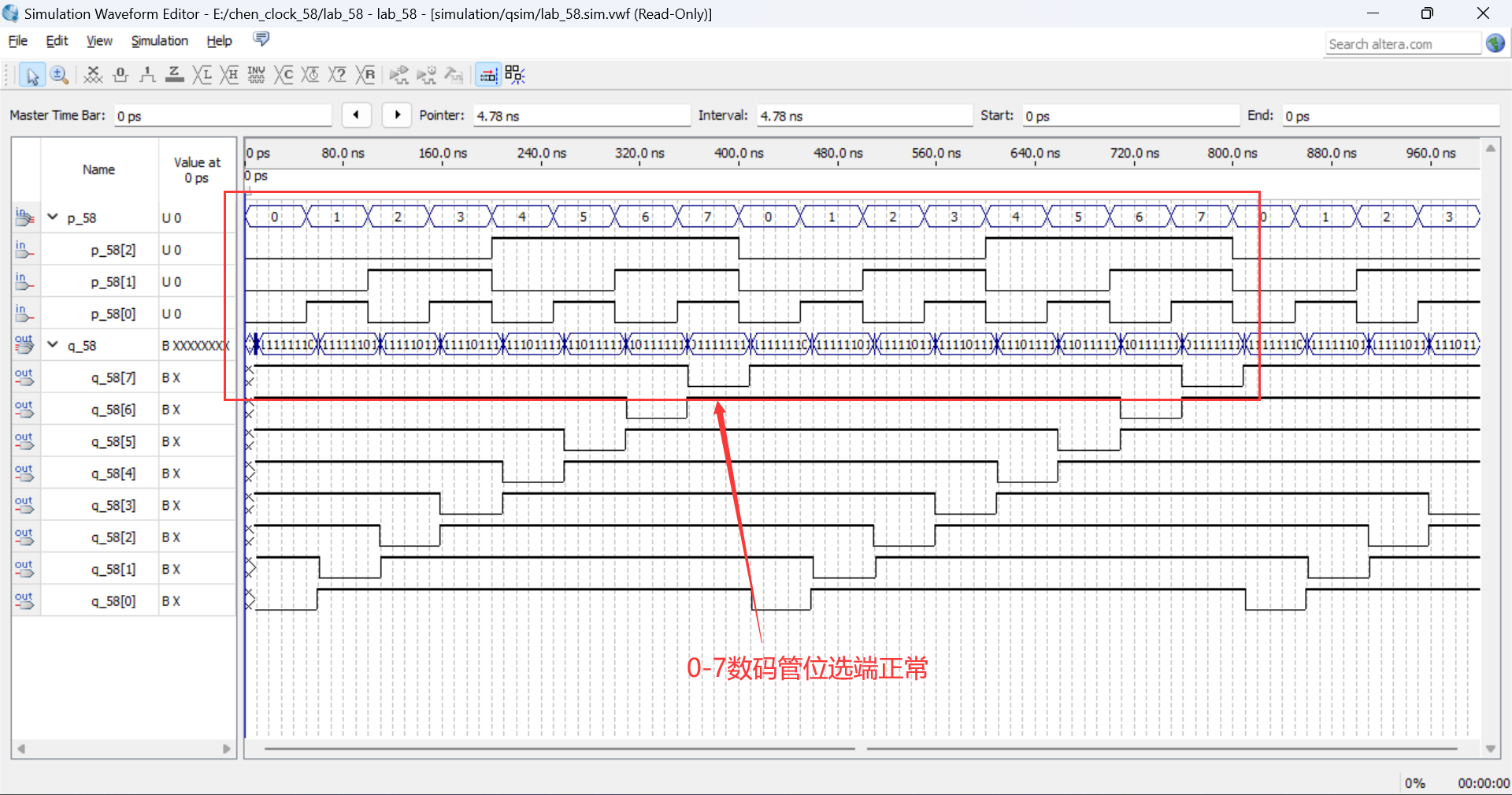


图22 位选模块dig\_select\_58仿真结果

### 仿真波形数据分析

如图22所示，0-7数码管位选端正常，输出端结果从高位到低位为11111110... ...01111111。

## **数据选择模块code\_select**

### 模块功能

从8组数据中选择一组数据输出。

### 设计思路

利用747151芯片的8选1数据选择功能，将每一位的数字的显示功能分开。

### 设计结果

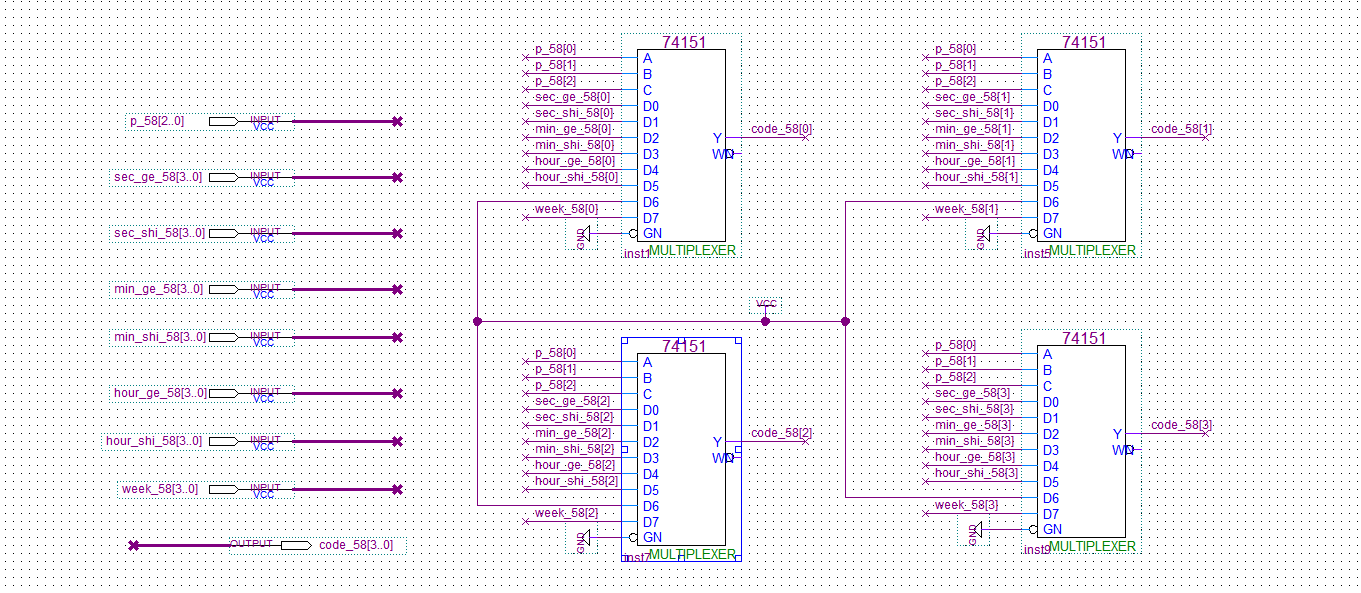


图23 数据选择模块code\_select电路图

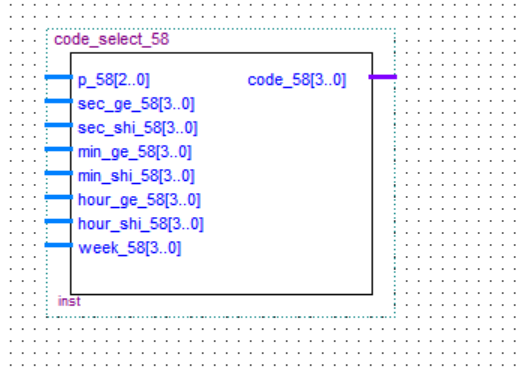


图24 数据选择模块code\_select封装图

### 仿真测试

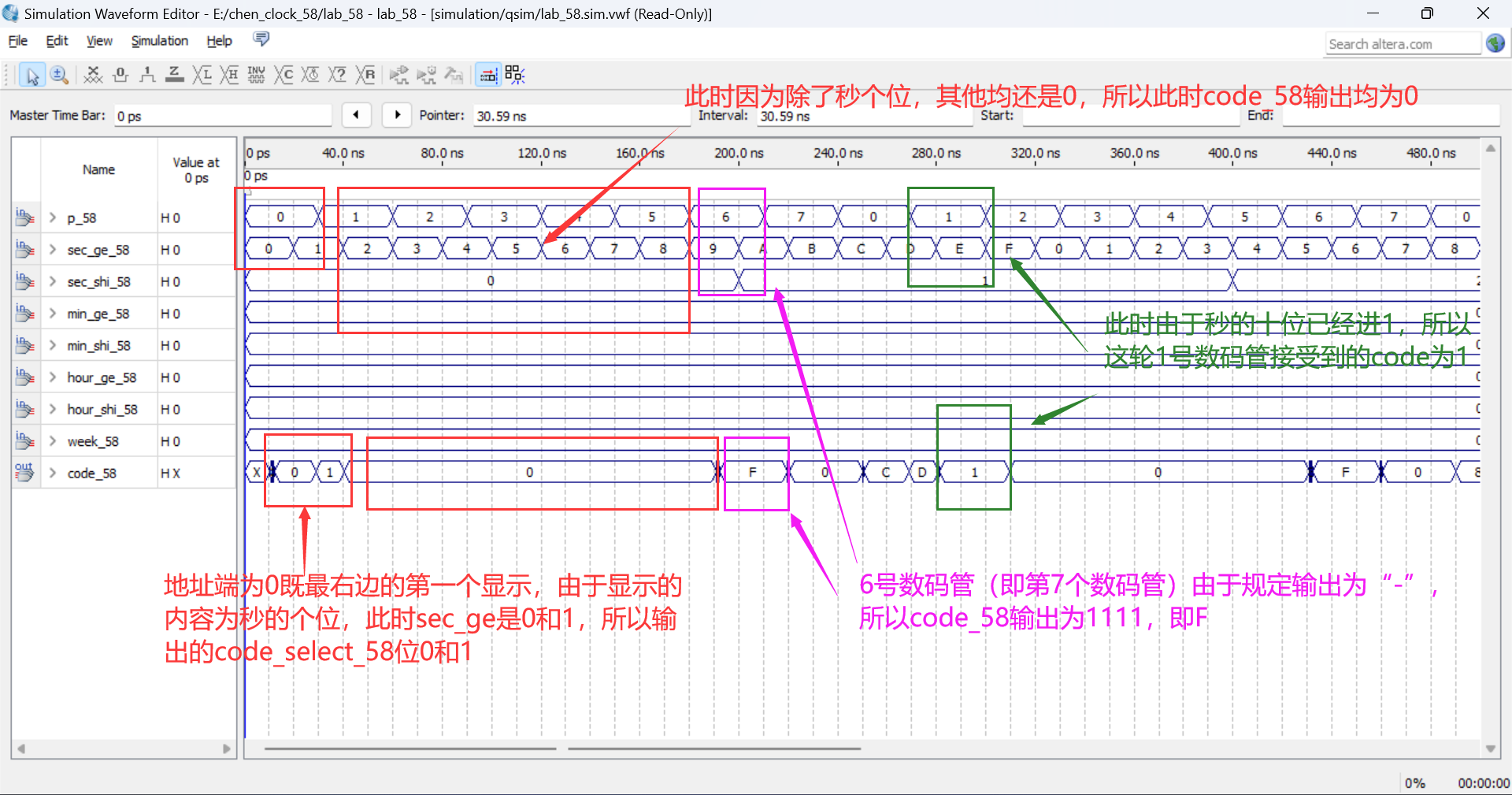


图25 数据选择模块code\_select仿真结果及分析

### 仿真波形数据分析

如图25所示，地址端为0，即最右边的第一个显示，由于显示的内容为秒的个位，此时sec\_ge是0和1，所以输出的code\_select\_58位0和1；刚开始计数，除了秒个位，其余均还是0，所以此时code\_58输出均为0；第七个数码管由于规定输出为“-”，所以code\_58输出为1111，即F；下一轮时秒十位已经进1，所以这轮1号数码管接收到的code为1。

## **译码模块decoder**

### 模块功能

将输入的4位8421BCD码译为七段码，然后输出数码管的七段码，使数码管进行显示数字，另外在正常输出的基础上，对6、9进行补段，并将1111改为输出一条横线。

### 设计思路

利用7448芯片带有的译码功能，在正常输出的基础上，对6、9进行补段（即对0110和1001分别做与判断然后分别与OA端和OD端做或），并将1111改为输出一条横线（即使OG端与ABCD均为1时输出）

### 设计结果

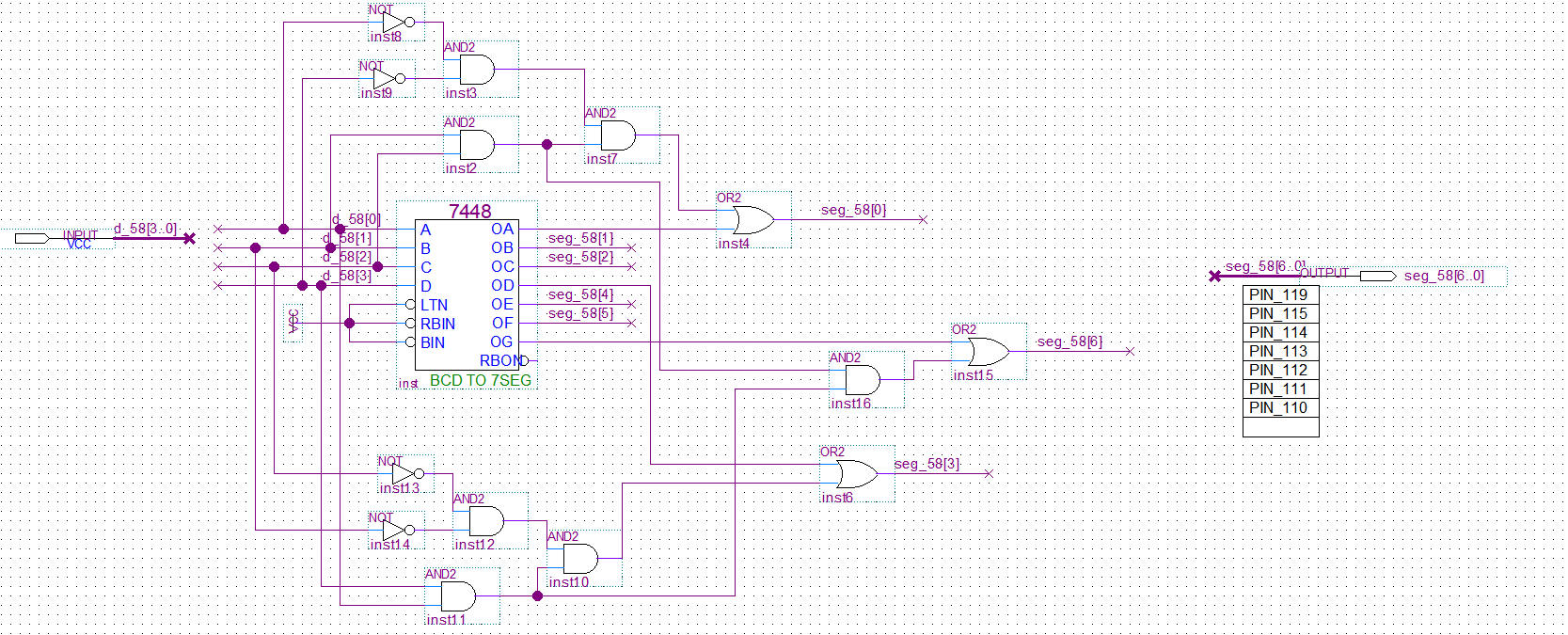


图26 译码模块deocker\_58电路图

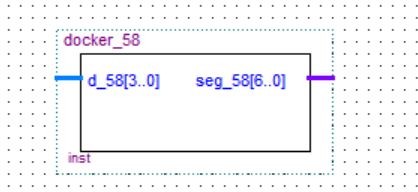


图27 译码模块deocker58封装图

### 仿真测试

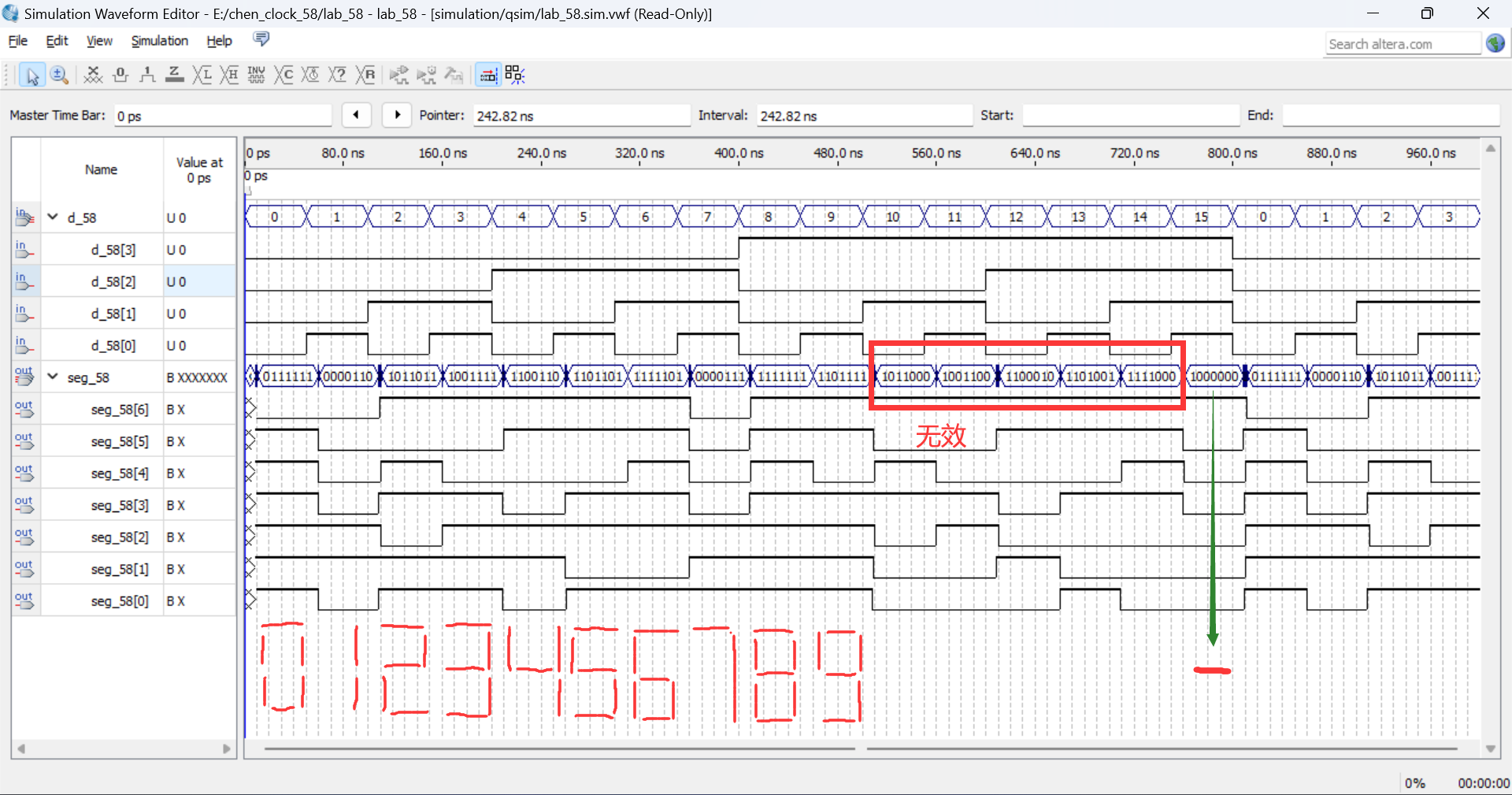


图28 译码模块docker\_58仿真结果

### 仿真波形数据分析

如图28所示，能够进行数字显示，并对“6”、“9”补段，OG端与ABCD均为1时输出“-”。

## **动态显示模块电路图**

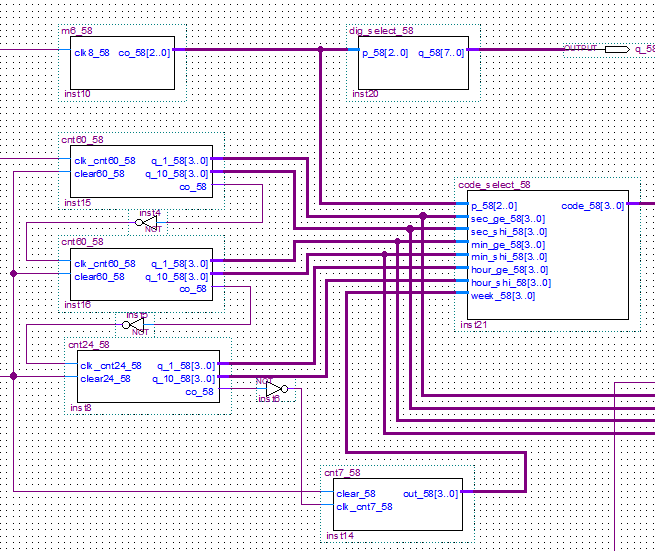


图29 动态显示电路图

# **、其他扩展功能**

## **调频模块spead\_select**

### 模块功能

通过switch开关12来选择输入的时钟频率，进行时分秒检测，检查电子钟的功能。当SW1SW2=00时，输入1hz的时钟信号，正常计时；当SW1SW2=01时，输入500hz的时钟信号,用来检测分钟；当SW1SW2=10时，输入1khz的时钟信号，检查小时；当SW1SW2=11时，输入0hz的时钟信号，实现暂停。

### 设计思路

选用74151芯片，当sw1,sw2分别接AB，1hz、500hz、1khz和0hz分别接入D0、D1、D2和D3。

### 设计结果

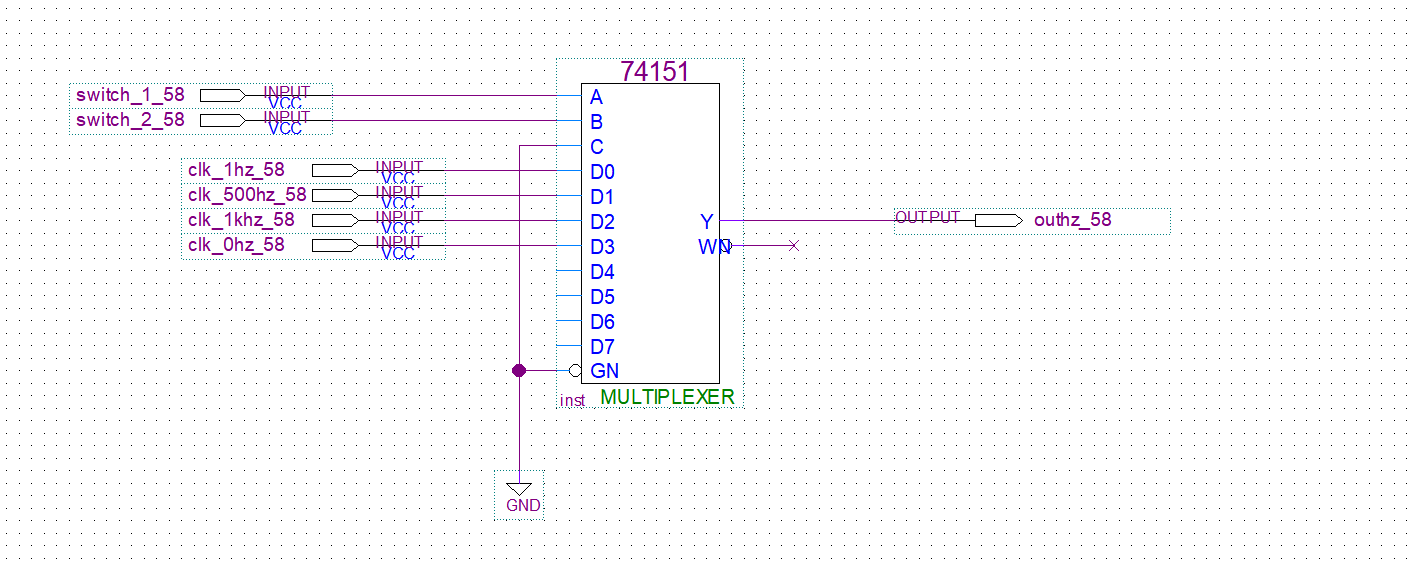


图30 调频模块spead\_select电路图

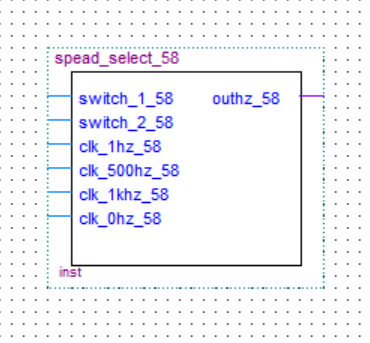
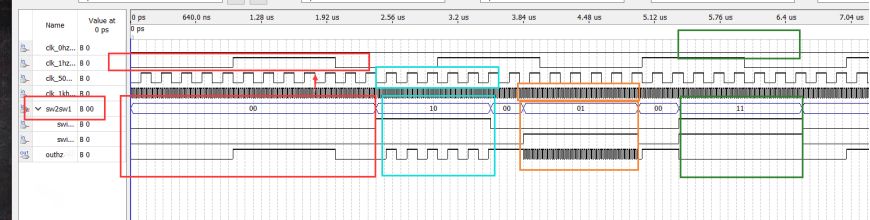


图31 调频模块spead\_select封装图

### 仿真测试

图32 调频模块spead\_select仿真结果

### 仿真波形数据分析

当SW1SW2=00时，输入1hz的时钟信号，正常计时；当SW1SW2=01时，输入500hz的时钟信号,用来检测分钟；当SW1SW2=10时，输入1khz的时钟信号，检查小时；当SW1SW2=11时，输入0hz的时钟信号，实现暂停。

## **清零功能**

### 模块功能

实现时间模块的清零功能

### 设计思路

将两个模60计数器、一个模24计数器、一个模7计数器的的清零端通过或门接到开关，开关输入高电平时清零端共同接到switch开关中的S[3], 则当sw3=1时，输入高低平，即实现清零。

### 设计结果

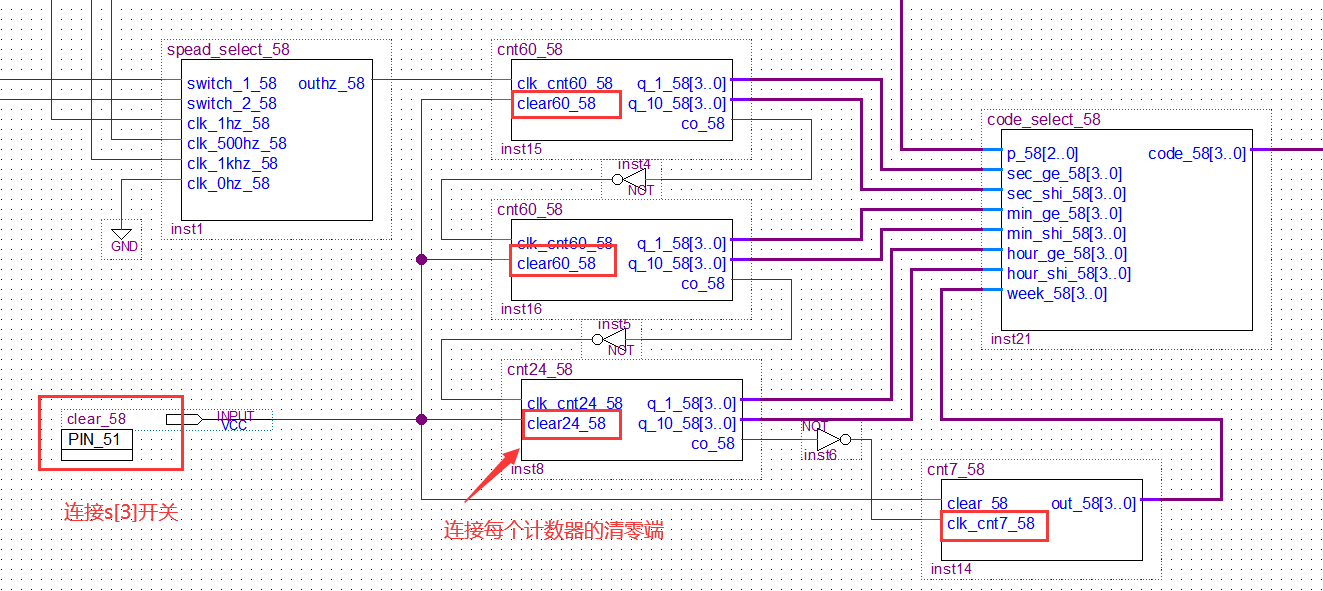


图33 清零功能电路图

### 仿真测试

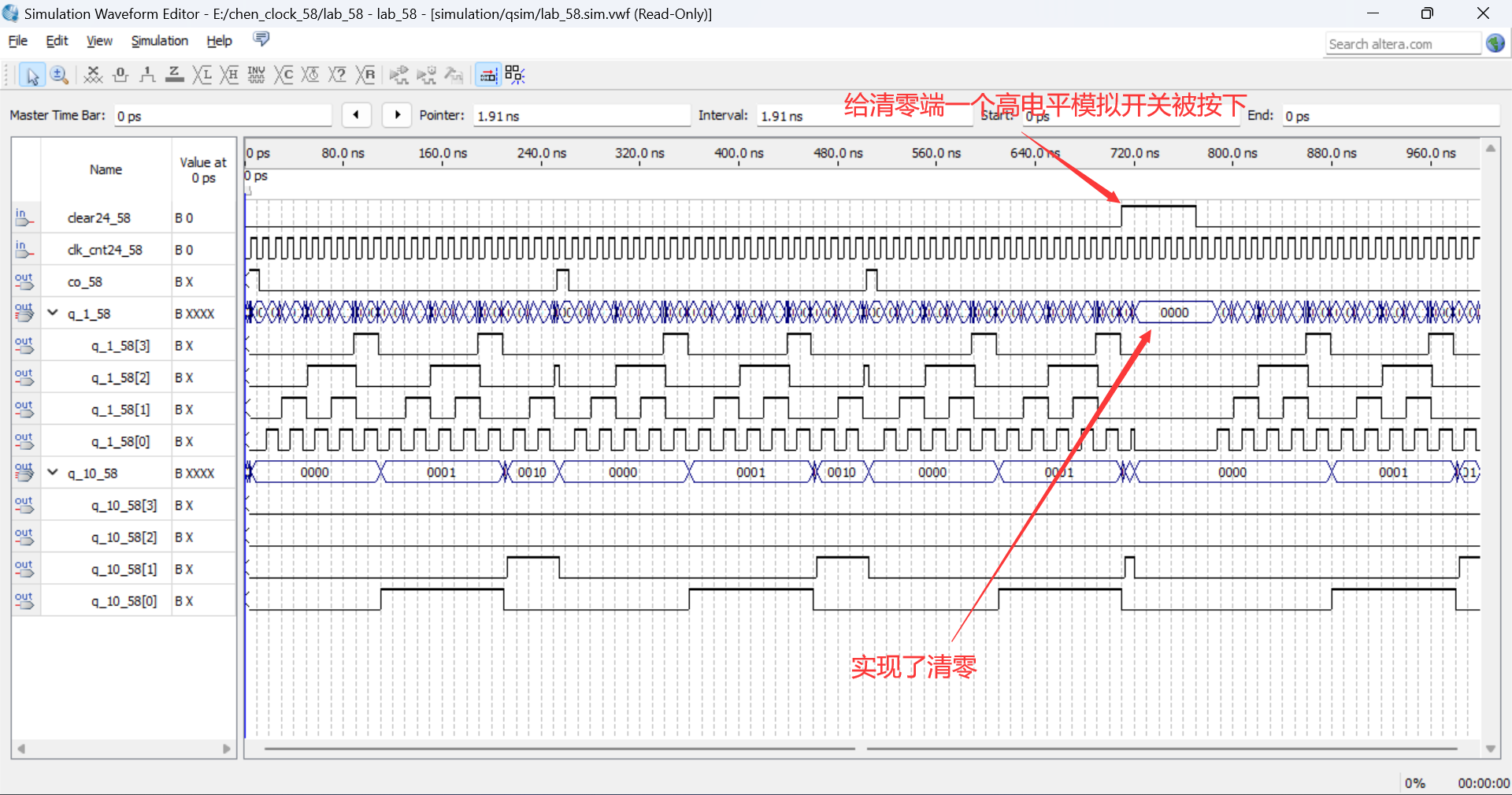


图34 模24计数器仿真手动清零测试波形

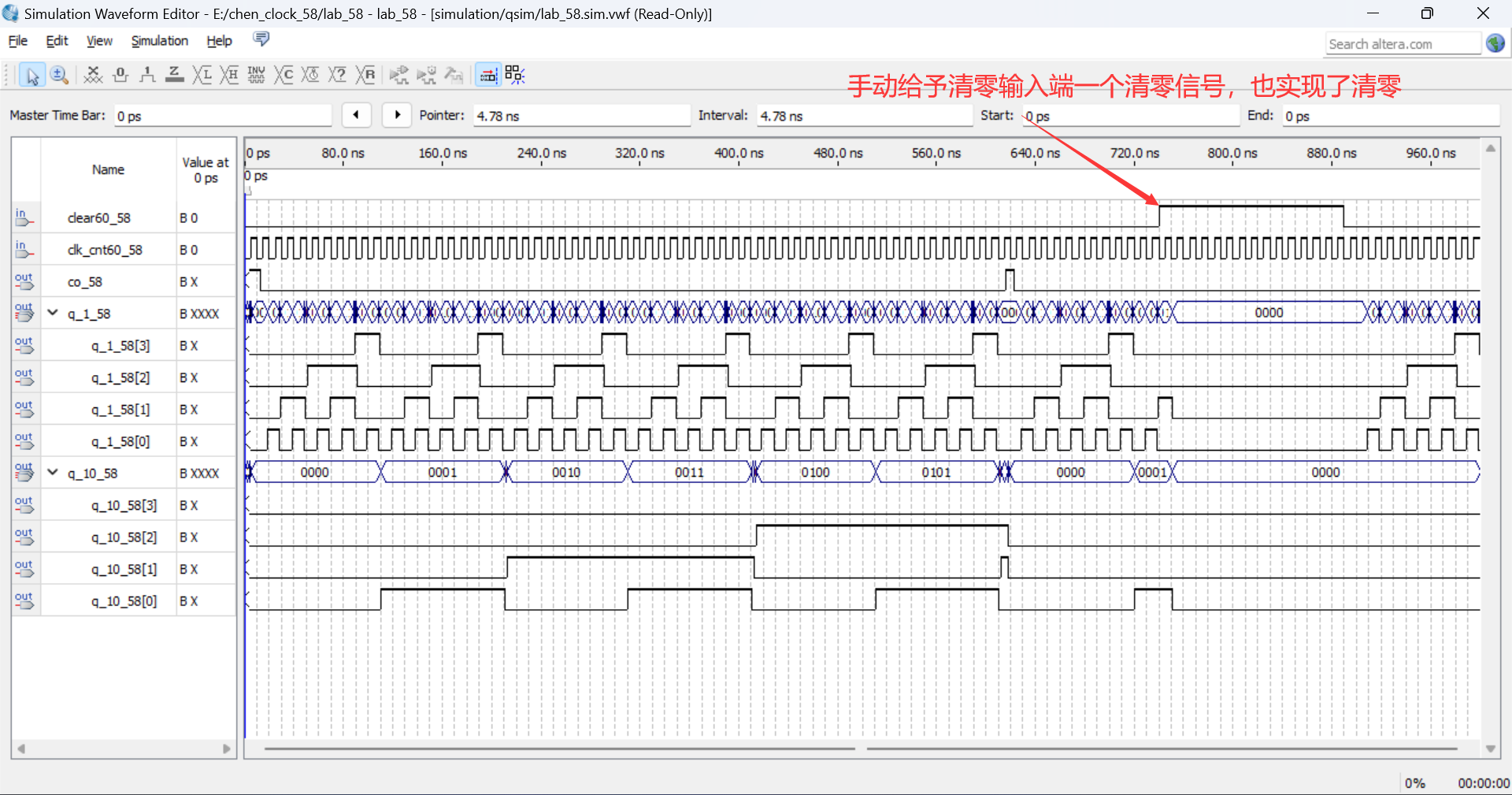


图35 模60计数器测试手动清零信号仿真

### 仿真波形数据分析

如图35、36所示，给清零端一个高电平模拟开关被按下，实现了清零；手动给予清零输入端一个清零信号，也实现了清零。

## **暂停功能**

### 模块功能

通过switch开关，当SW1SW2=11时，输入0hz的时钟信号，实现暂停。

### 设计思路

为给调频模块的0hz输入端接地即可，采用选用74151芯片，当sw1=1,sw2=1时，输出接入D3的0hz信号。

### 设计结果

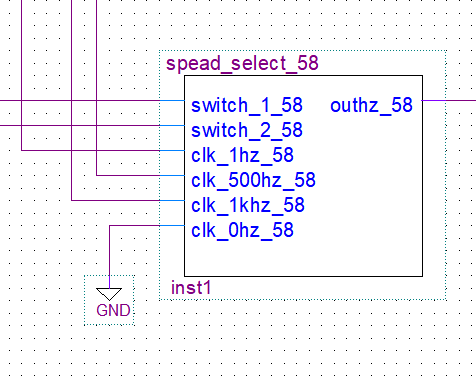


图36 调频模块0hz输入端接地

### 仿真测试

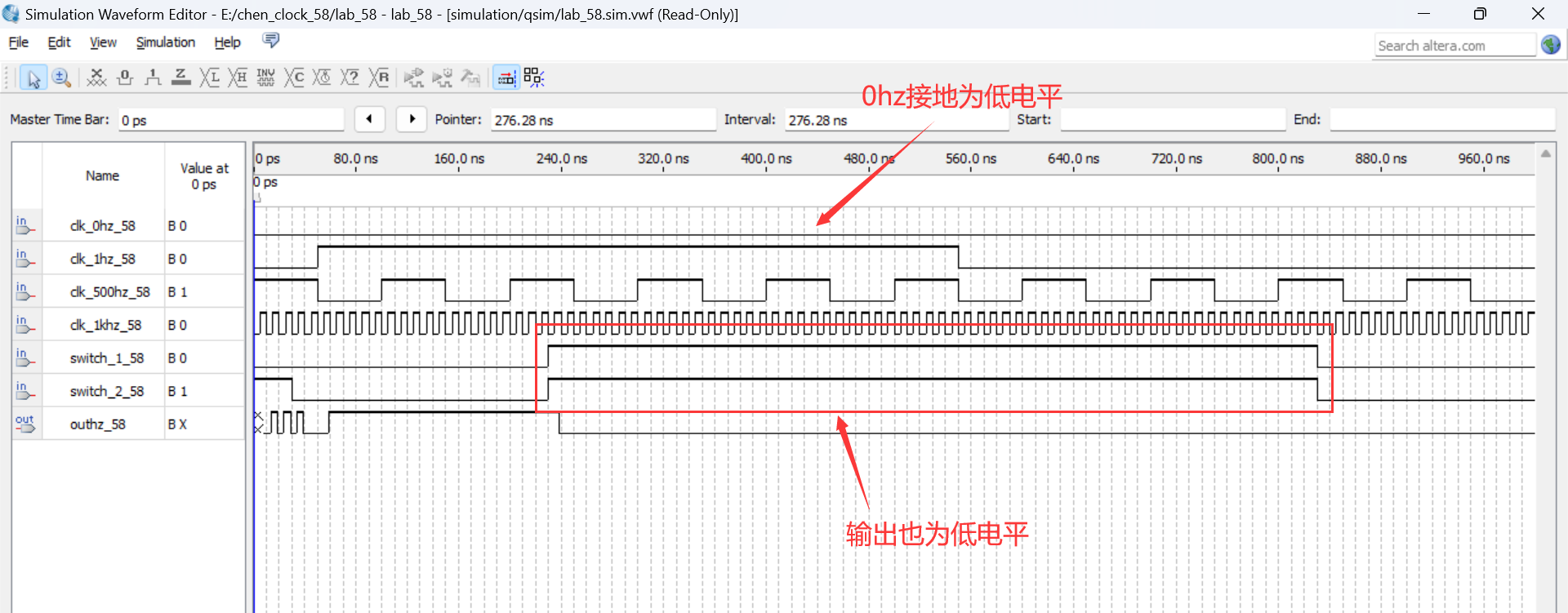


图37 调频模块0hz仿真测试结果

### 仿真波形数据分析

如图38所示，0hz接地为低电平，输出也为低电平。

## **显示星期**

### 模块功能

用最左边的一位数码管显示星期（1-7代表周一到周天），设计一个模7计数器，接收模24计数器的进位信号，同时将该模7计数器计数结果输出到数选模块，数码管动态显示的实现见上面4数码管动态显示模块

### 设计思路

采用74161芯片，设计模7计数器。将表示小时的模24计数器的进位端接到模7计数器上，模7计数器的输出表示星期。

### 设计结果

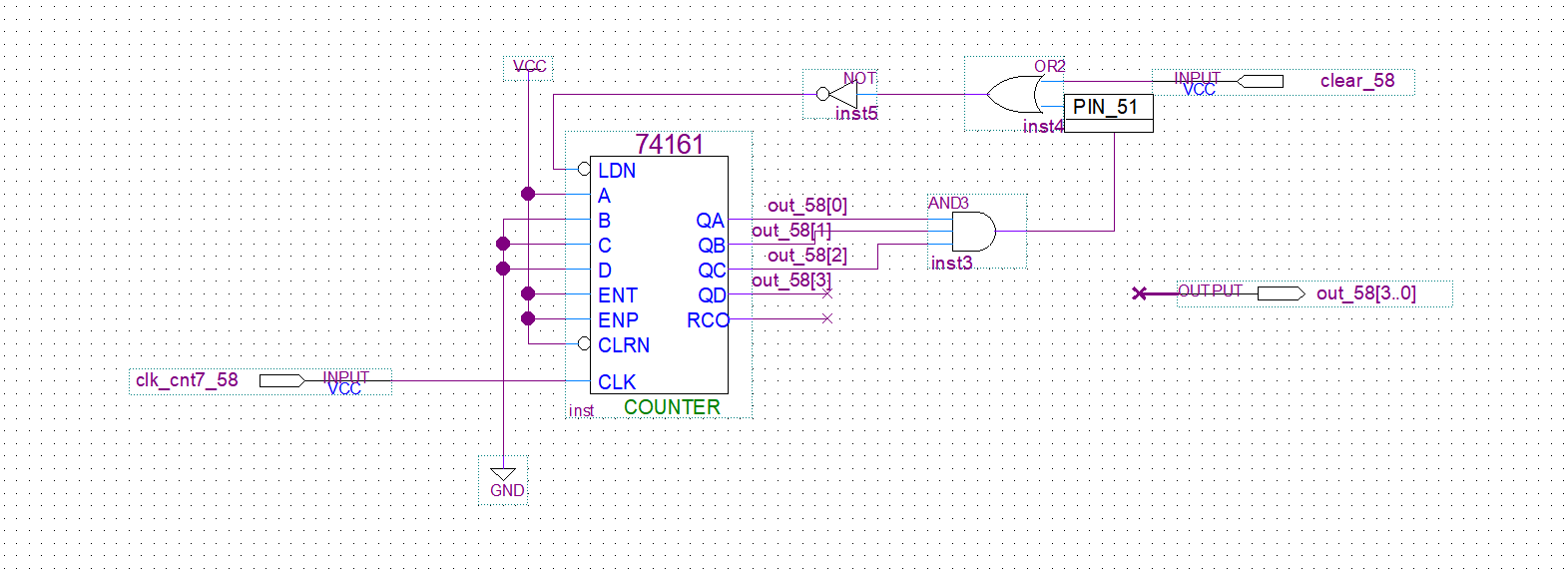


图38 模7计数器电路图

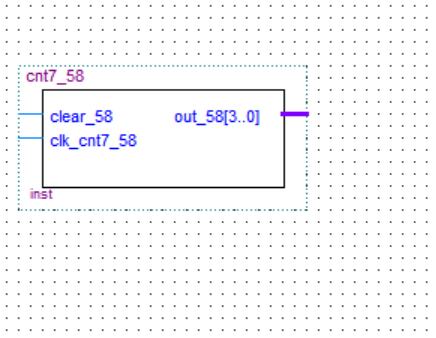


图39 模7计数器封装图

### 仿真测试

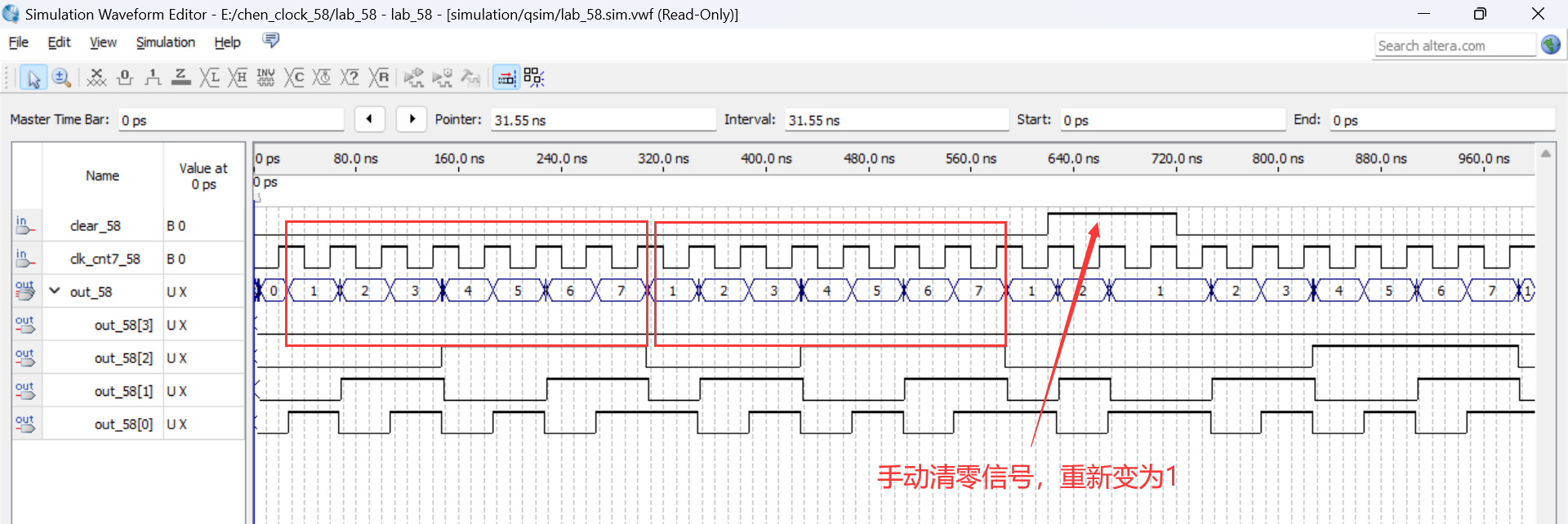


图40 模7计数器仿真结果

### 仿真波形数据分析

如图41所示，星期显示需要1-7，模7计数器一个周期为1-7，7的下一时刻会变成1，实现了星期计数，给一个手动清零信号，重新变为1。

## **6、9补段**

### 模块功能

将数码管上显示不完整的6和9进行补全，即给9补全数码管上的d段，给6补全数码管上的a段。

### 设计思路

因为数码管上的最初设置，6和9的显示是不完整的，并且根据7448译码器功能表显示，当我们输入的信号DCBA为0110时，显示的是不完整的6，即此时显示的是a段没有亮起来的6，这就说明了此时OA的输出是低电平，想要a段亮起来，我们就应读使0A的输出为高电平，并且还要保证，只有当输入信号为0110时才会让0A的输出变为高电平，而在显示其他数字的时候0A正常输出。

### 设计结果

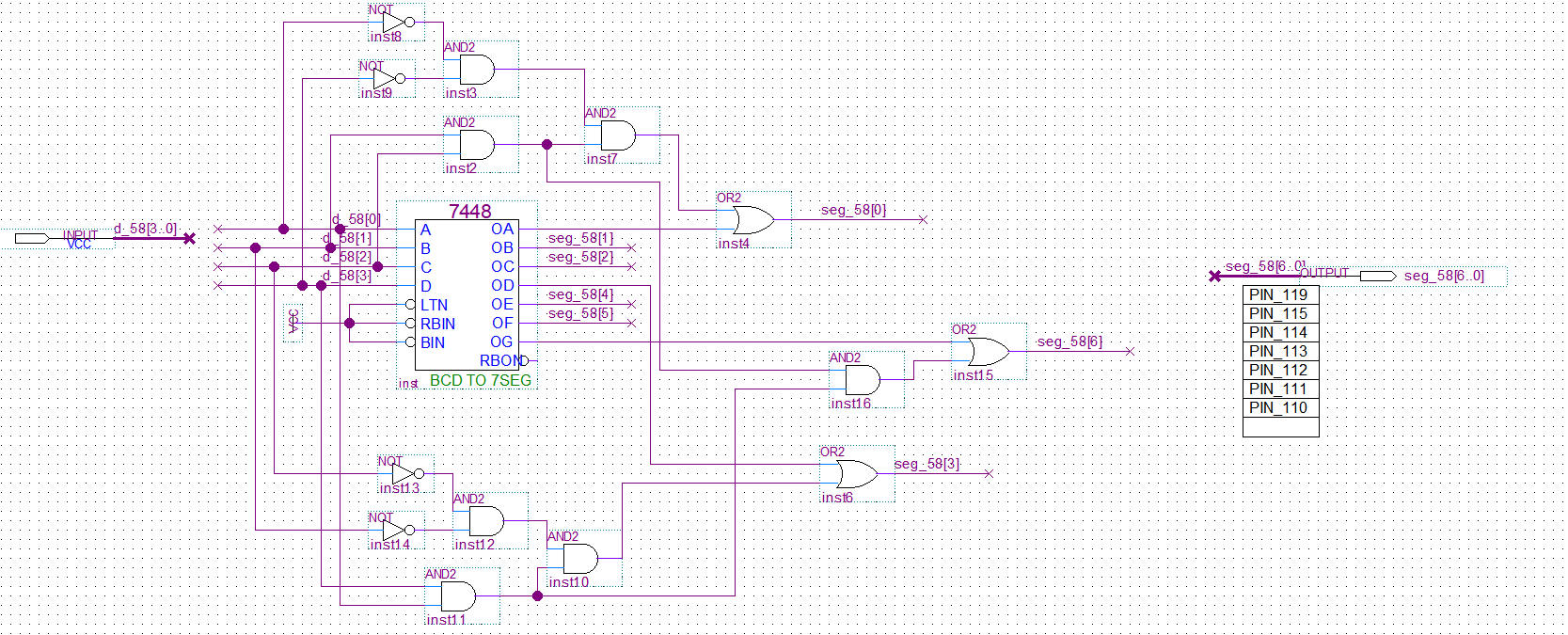


图41 6、9补段电路图

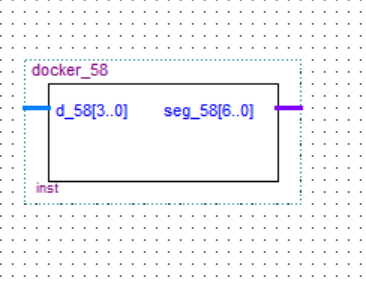


图42 6、9补段封装图

### 仿真测试

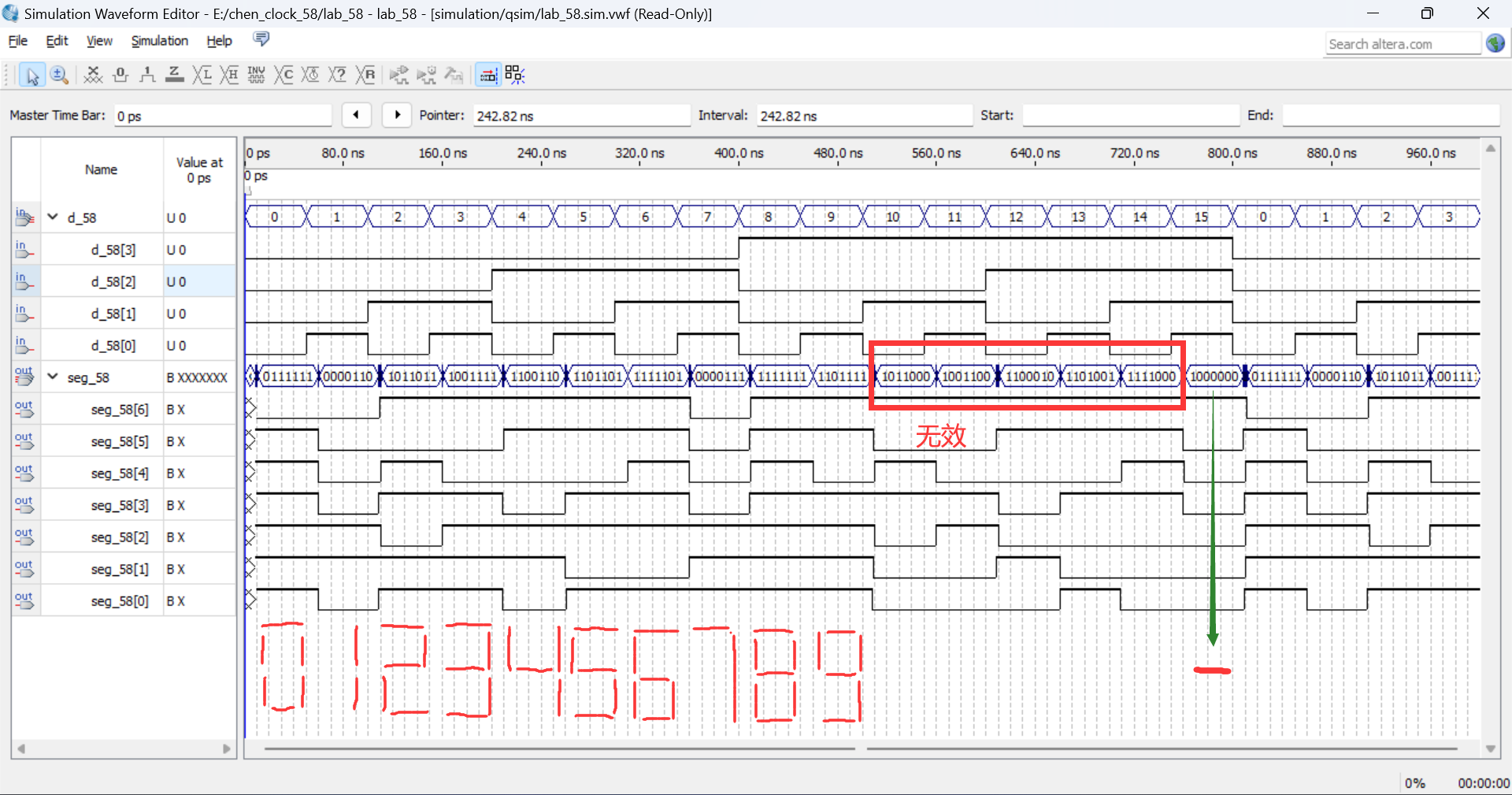


图43 6、9补段仿真结果

### 仿真波形数据分析

如图43所示，能够进行数字显示，并对“6”、“9”补段。

## **八位数码管显示**

用8位数码管显示，用“-”作为分隔符，显示格式：星期-小时分钟秒。设计思路、电路图和仿真结果见数码管动态显示模块

## **整点报时功能**

### 模块功能

当时钟上显示的时间为一个整数时，蜂鸣器响1秒。从而实现整个报时的功能。

### 设计思路

当时钟显示为整点时，分钟和秒钟的个位和十位都为0。所以根据这一点来设计整点报时电路，即应该将秒的个位或十位相或得到：“或1”，同时让分的个位和十位也相或得到：“或2”，然后让“或1”和“或2”进行一次或非。最后再输出。因为在整点的时候，分钟和秒表的个位和十位都为0，则他们相或之后的或1和或2都输出为0，再或非之后输出为1，与500hz的频率相与，从而能够成功实现蜂鸣器的鸣叫。故而实现了整点报时的功能。而当其他非整点时候则不能满足上述条件，从而蜂鸣器不会叫。

### 设计结果

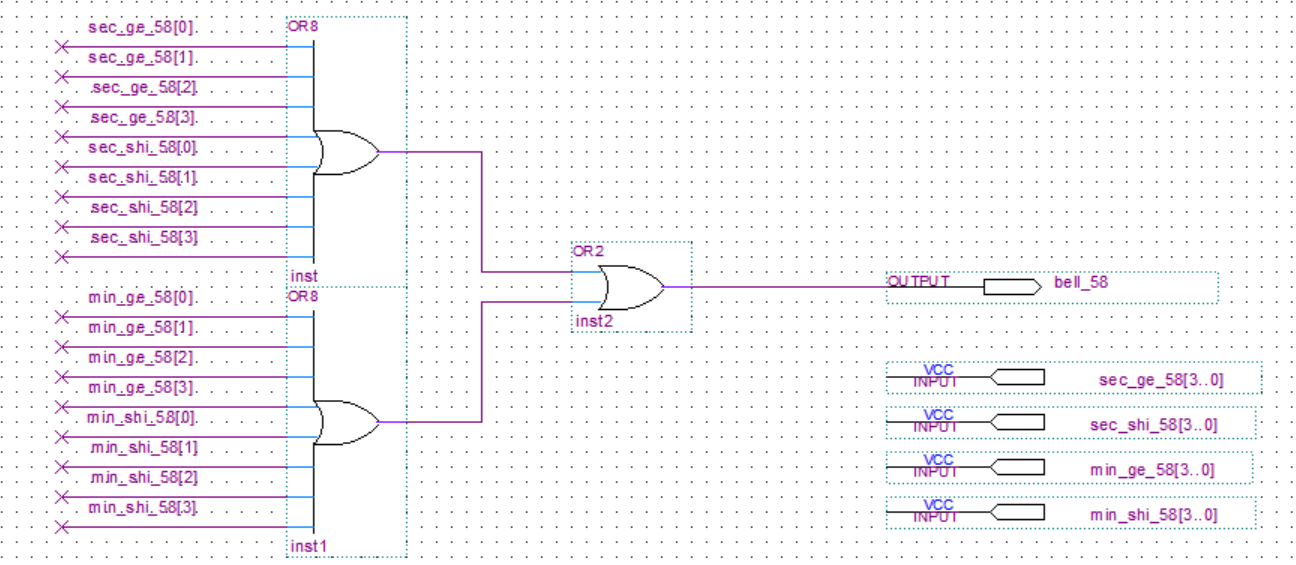


图44 整点报时电路图

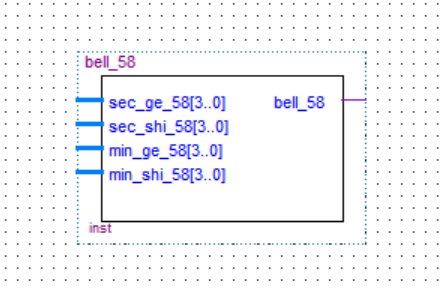


图45 整点报时封装图

### 仿真测试

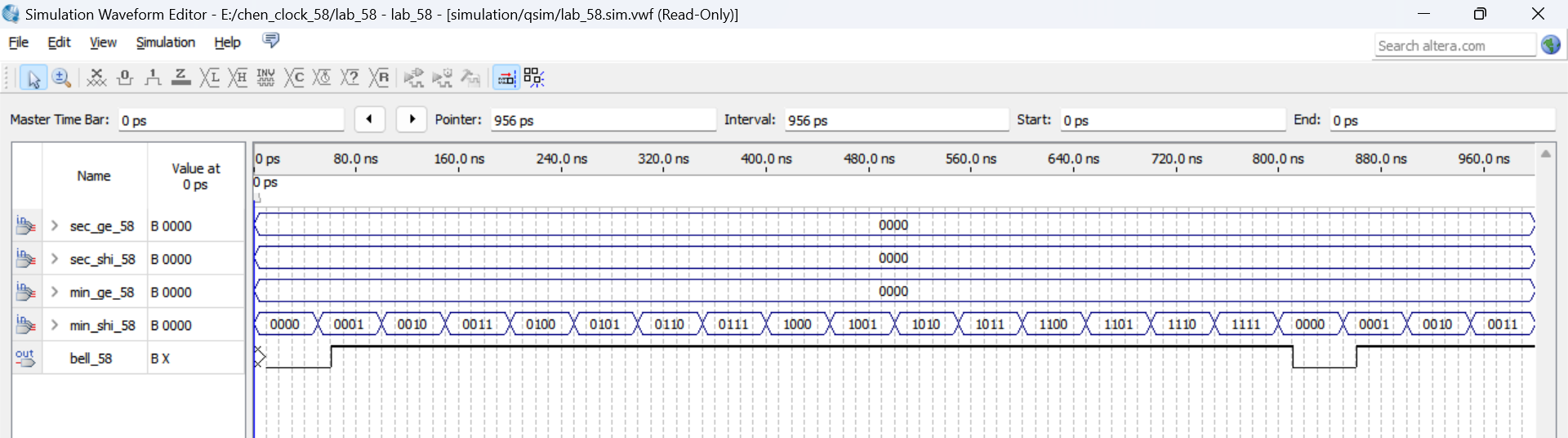


图46 整点报时仿真结果

### 仿真波形数据分析

如图46所示，当时钟上显示的时间为一个整数时，蜂鸣器响1秒。将这个模块放入基本时钟的设计图中，进行编译，然后连接学习板，并且观察整点报时是否能够听见蜂鸣器的声音，或者也可以看清零的时候蜂鸣器是否会鸣叫。检测结果整点或者清零时蜂鸣器鸣叫。

## **Led流水灯功能**

### 模块功能

学习板上的Led灯以1hz的频率移动，即以流水灯的形式展示，使学习板更美观。

### 设计思路

可以观察到在学习板上总共有8个led灯，并且想要灯成为流水灯样式还需要用到一个译码器，因为模8计数器将计数情况输出给74138的A B C端口，然后根据该时刻计数的数值进而选择对谁进行译码，因为译码输出的是低电平，所以该数值对应的灯熄灭,而又根据1h 的时钟信号知道，流水灯一秒移动一下。从面即可设计出led流水灯。

### 设计结果

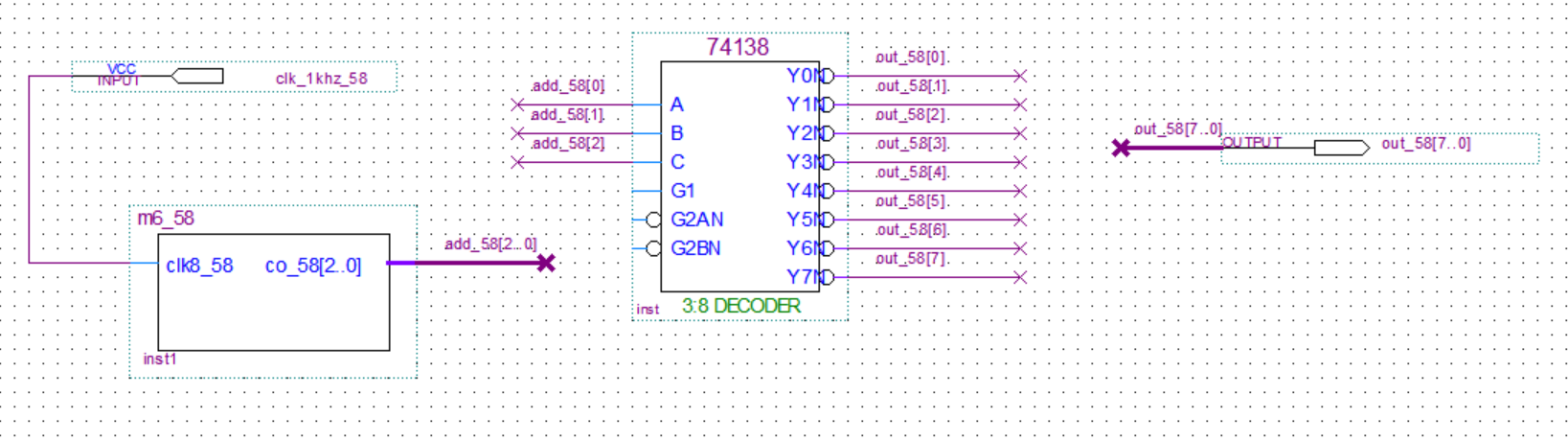


图47 led流水灯电路图

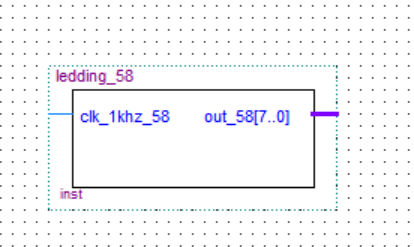


图48 led流水灯封装图

### 仿真测试

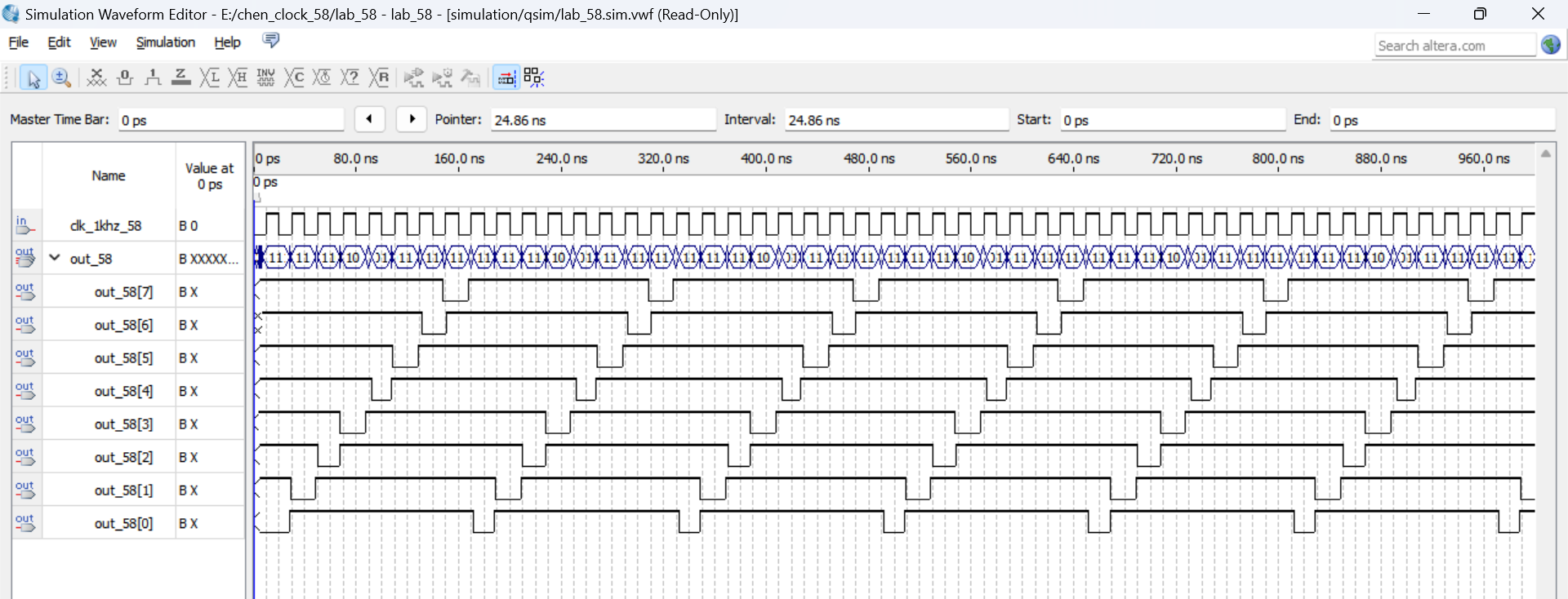


图49 led流水灯仿真结果

### 仿真波形数据分析

如图49所示，out\_58从0到7每个都按顺序显示，一直循环。将 led 流水灯模块放入到设置的时钟设计图中，进行编译，连接学习板上面的灯，再编译， 然后观察学习板上的灯是否能够以流水灯的形式移动。

# **、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”。

# **、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**

**7.1 问题1**

主要问题：编译成功，但最终显示器只出现一位数字 0；

解决思路：经仔细排查电路，发现连线被遗漏；

解决方案：连线，问题解决。

**7.2 问题2**

主要问题：编译成功，但显示始终不对；

解决思路：经仔细排查电路，发现分线与总线名称不匹配；

解决方案：修改总线名称，问题解决。

**7.3 问题3**

主要问题：编译成功，但显示出错；

解决思路：经仔细排查电路，发现两个端口与对应的输入接反；

解决方案：置换输入位置，问题解决。

**7.4 问题4**

主要问题：编译时发生如下报错——Error (275044): Port "IN2" of type AND2 of instance "inst71" is missing sour.

解决思路：经过查询资料，得知报错原因为该端口”IN2”没有连线；

解决方法：检查电路，对该端口连线，问题解决。

**7.5 问题5**

主要问题：编译时发生如下报错——Error (12014): Net "gdfx\_temp0", which fans out to "inst7", cannot be assigned more than one value.

解决思路：经过查询资料，得知报错原因为为模块”inst7”接受了两个没有用逻辑符号解释关系的信号源；

解决方法：检查电路，发现如图所示情况，应为连线时错误多连，删除该结点后重新编译，问题解决。

# **、心得体会**

经过这学期数字电路与逻辑设计实验课上的学习与设计，我加深了对数电知识的了解，通过在实验课上的动手练习，帮助自己有效地巩固了理论课学习的知识。在完成本次数字钟的设计后，很难说自己对 Quartus II 软件熟练度没有较之前大幅度提升，包括对各种操作、细节甚至报错原因的知晓。所以大部分逻辑源于自己思想上的顽强抗争，不停的报错，不停的修改，可能十分不简洁明了，甚至非常冗杂，但却是经过了自己一步一步构想、思考、反省和总结，每当自己的一个设想实现，成就感是相当之高的。如果可以希望温柔的老师能给我一个还不错的分数。

**9、参考文献**

[1]重庆邮电大学，基于 FPGA 的数字电路实验指导书.

[2]邹虹等，数字电路与逻辑设计(第 2 版),人民邮电出版社,2017.