自律电子时钟

罗华坤 软件 02 2019011799

luohk19@mails.tsinghua.edu.cn

1 设计目的

如今手机上的自律监督学习的软件各式各样，有类似番茄 TODO、Forest 等。

但由于权限原因，人们仍然有各种各样的方式逃脱这种弱监督，以至于最终放弃使用。

并且手机本身作为一个充满诱惑的平台，弱化了自律软件的监督作用，还容易起到相反的作用。

因此一种脱离手机平台具有有效监督功能的计时工具可以提高使用者的注意力，提升学习效率。

并且可以设计成便携式，平时可以用作时钟、闹钟、秒表、光源等，在非学习时间也能派上用场。

2 设计思路

2.1 基础计时功能

能显示当地 24 小时制的时间，包括小时、分钟、秒钟三部分，通过内部的时钟发生器分频得到。

具体时间在数码管上显示。每过一小时，自律电子时钟发出短暂蜂鸣声提示。

功能还包括按时间段置零、设定特定的时间、闹钟。

2.2 秒表部分

类似计时功能部分，包括秒、分钟、时钟三部分。

功能包括暂停（暂停计时）、重置（计时归 0），其余与计时模块相同。

2.3 光源

采用 verilog 写了一个模块，当按下按键时，led 等光源亮 10 秒。

2.4 监督功能

利用秒表模块，当暂停或者重置时，判断用时是否设定时间。

若超过，说明专注时间达到要求，播放美妙音乐（也可替换成蜂鸣，根据场所改变）。否则不发声音。

2.5 其余小模块

2.5.1 分频模块

在 Verilog 中编写简单代码，采用计数器实现时钟信号的分频，分为 800Hz,1Hz 两个频率。

2.5.2 消抖模块

通过 EDA2 的练习，采用 verilog 写一个小的消抖模块，提高输入的准确性。

为了维持界面简洁，后续输入均默认经过消抖处理，具体输入大小可以更改。

```
module xiaodou(input clk, input operate, output reg key_out);
reg old;
reg key_cnt;
parameter DELAY = 15'd5000;
reg count;
always@(posedge clk) begin
   if(operate != old) begin//当当前输入与前输入不同时
      old <= operate;//重置旧输入,重置计数、输入计数
      count <= 0;
      key_cnt <= 0;//输入计数用于显示按钮是否被按下，解决每次按下时间长短问题
   end
   else if(count == DELAY) begin//当达到一定长度后
      key_out <= old;//说明已经稳定
      count <= 0;//重置计数
      key_cnt <= 1;//记已被按下
   end
   else if(key_cnt == 1) begin
      key_out <= 0;//当已被按下而且不发生改变时，视为无按下;
   end
   else begin
      count <= count + 1;//累加计数
   end
end
endmodule
```

Figure 1: 计时电路

2.5.3 显示模块

为了提高引脚、接口的利用率，采用动态显示数码管的方式。

为此需要添加一新分频，用于动态选中数码管、选择显示相应的数据。

3 电路设计

3.1 基础计时模块电路

首先利用分频模块得到的 1Hz 信号，作为计数器的时钟信号，并且采用串行的方式连接电路。

按照时间的进制，秒 60，分钟 60，小时 24，利用课上学到的知识设计对应进制的计数器。

从下图可以看出，采用了异步置零的方式，因此分别在 24、60 的位置置零。

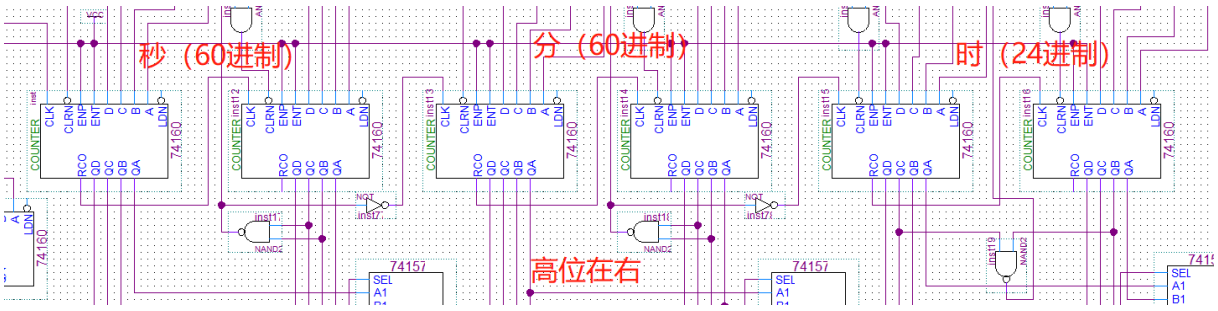

Figure 2: 计时电路

对于设定时间功能，可以利用异步置数的操作，引入对应时间段的三个按钮。

将该按钮与对应时间段的计数器异步置数端相连，当按下时，由于默认为 0，可以达到置零效果。

进一步地可以将其改进为异步置数效果，仅需将各位输入与 0 相与便可。

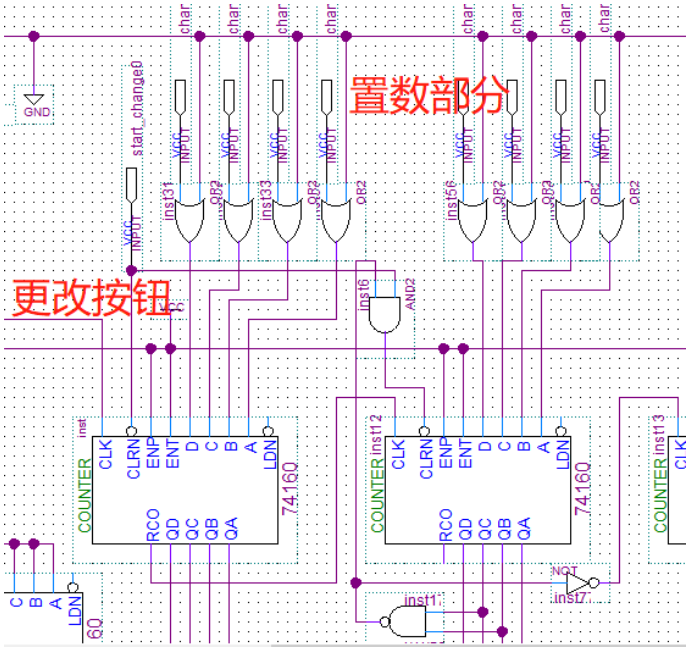

Figure 3: 设定时间功能电路

可以看出，一个更改按钮控制两位数，即两位均异步置数。置数输入可以从 10 进制转 8421BCD 码得到。

为了养成良好习惯，每日的闹钟均固定为 7 点，响铃 10s，可以选择关闭。

选取 7 点第 0 个十秒作为标志，输出接蜂鸣器等，则可以实现 10s 响铃的功能。

后续可以通过利用 ROM 的结构实现自定义闹钟时间的功能。

同时，取 0 分 0 秒作为信号发出蜂鸣以提示过了一小时。

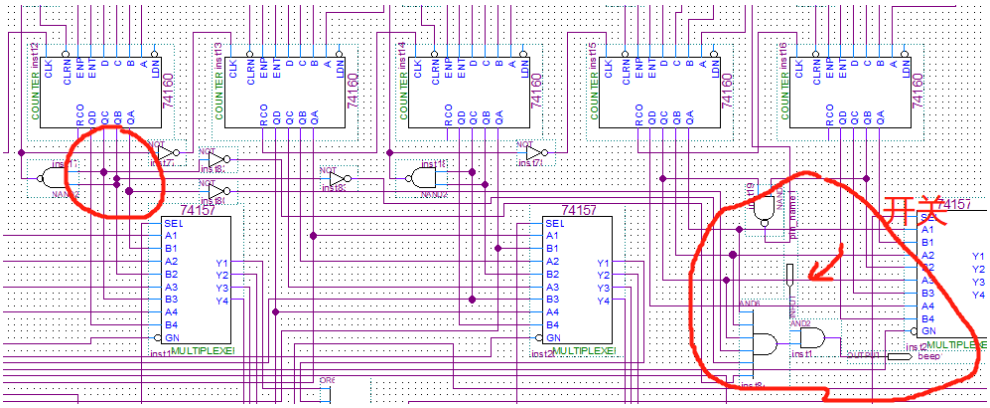

Figure 4: 7 点闹钟

3.2 秒表模块电路

同样类似上述模块，采用 1Hz 时钟信号，与之不同的是，增加了暂停与置 0 功能。

暂停功能通过与时钟相与实现，当按钮按下时，时钟始终为 0，停止技术。

而每次打开秒表、按下置零按钮都需要置零。由于按钮按下为 0，因此采取相与的方式输入异步置数。

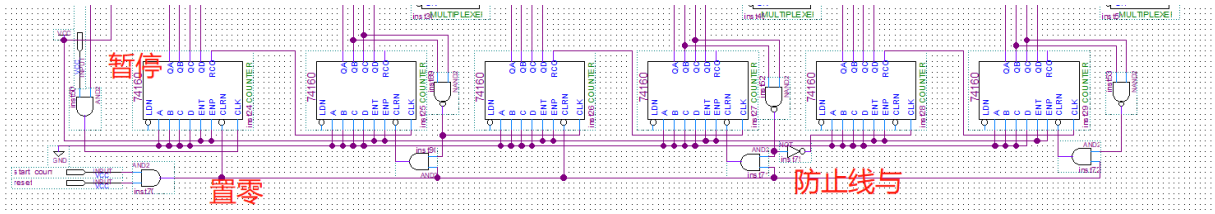

Figure 5: 秒表

3.3 光源电路

采用 Verilog 写了一个模块，将单个时钟信号转化成一个时长 10s 的发光信号。

采用分频模块产生的 1Hz 信号，并将输出连至 LED 等光源接口，具体模块如下。

```
33  module lightControl(input flag, input clk, output reg sign);
34  reg change;
35  reg[4:0] cnt;
36  always@(posedge flag or negedge change) begin
37      if(change == 0)//当读秒不足10s时
38          sign <= 1;
39      else
40          sign <= 0;//不再响铃
41  end
42  always@(posedge clk) begin
43      if(sign == 1 && cnt < 4'd10) begin//当不足10s时
44          cnt <= cnt + 1;
45          change <= 0;
46      end
47      else begin
48          cnt <= 0;
49          change <= 1;
50      end
51  end
52
53  endmodule
```

Figure 6: 光源 Verilog

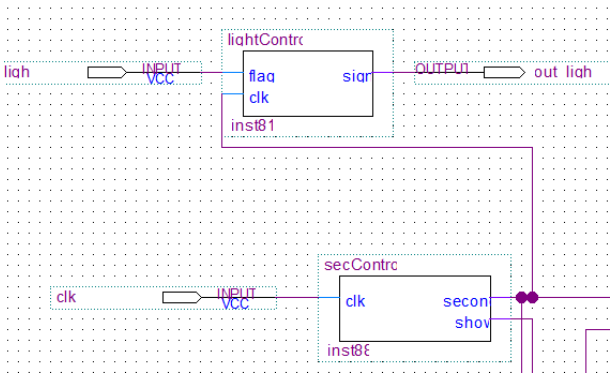

Figure 7: 光源电路

3.4 监督功能模块

监督模块通过比较关闭时秒表的数值与 1 小时的大小，即小时的个位非 0 或十位非 0 便可。

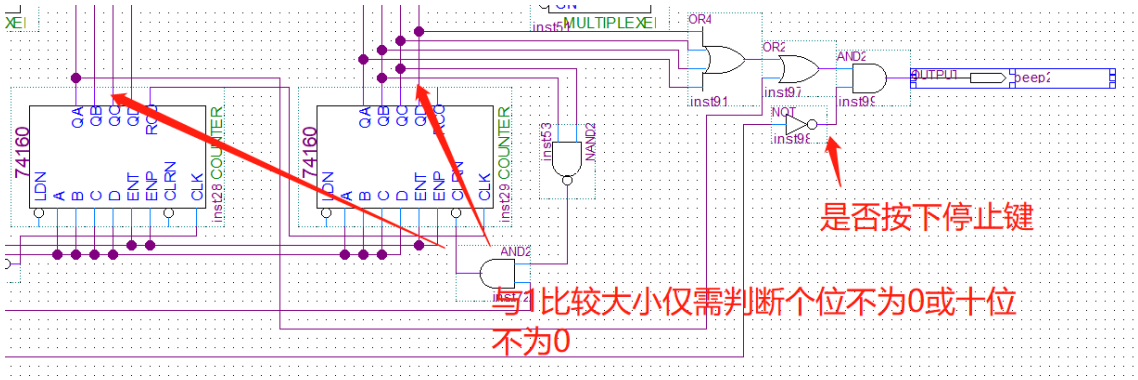

Figure 8: 监督电路 1

进一步改进可以通过引入比较芯片，实现自定义时间目标的功能。输入可以通过十进制转 BCD 码得到。

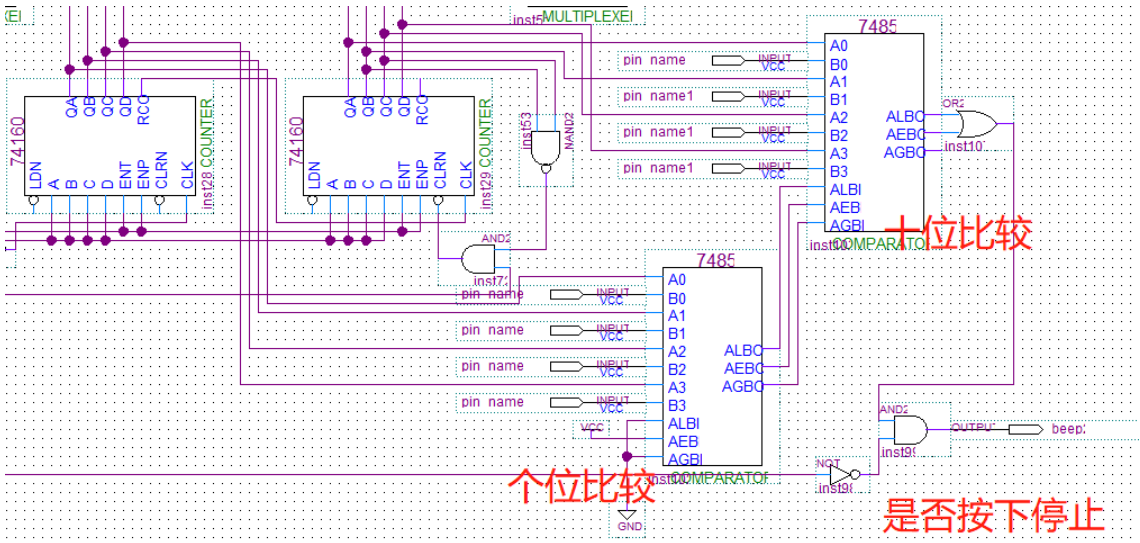

Figure 9: 监督电路 2

3.5 其余小模块

3.5.1 分频模块

采用 Verilog 实现，在内部用计数器实现分频。本设计只需分两种频率，1Hz、800Hz。

1Hz 用于时钟、秒表、光源等需要真实世界 1s 的模块。

而 800Hz 主要用于显示部分，利用视觉残留实现动态显示数码管的功能。

```
33  module lightControl(input flag, input clk, output reg sign);
34  reg change;
35  reg[4:0] cnt;
36  always@(posedge flag or negedge change) begin
37      if(change == 0)//当读秒不足10s时
38          sign <= 1;
39      else
40          sign <= 0;//不再响铃
41  end
42  always@(posedge clk) begin
43      if(sign == 1 && cnt < 4'd10) begin//当不足10s时
44          cnt <= cnt + 1;
45          change <= 0;
46      end
47      else begin
48          cnt <= 0;
49          change <= 1;
50      end
51  end
52
53  endmodule
```

Figure 10: 分频 Verilog

3.5.2 消抖模块

如上第二部分消抖模块，此处不再重复，图中每个输入都经过消抖操作，在图中不再显示。

3.5.3 显示模块

由于引脚数有限，采用数据选择器的方式，尽可能地利用同一个引脚。

利用分频电路得到的 800Hz 电路输入 12 进制计数器，选通对应的 8 选 4 数据选择器。

列出下列真值表，1、2、3、4、5、6 表示选通的数据选择器，out 表示八选四的选择端。

A	B	C	D	Out	1	2	3	4	5	6
0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0
0	0	1	1	1	0	1	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0
0	1	0	1	1	0	0	1	0	0	0
0	1	1	0	0	0	0	0	1	0	0
0	1	1	1	1	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	1	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	0	1
1	0	1	1	1	0	0	0	0	0	1

可以发现，仅需将最后一位连接到每个选择器，而剩余三位输入译码器，输出分别选通选择器。最后将各位相与，如此便得到了 48 选 4 的数据选择器。输入 7447 芯片，控制数码管的 led 灯。

而且，将计数器的四位直接输入 4 位译码器也可作为数码管的选通端，具体电路如下：

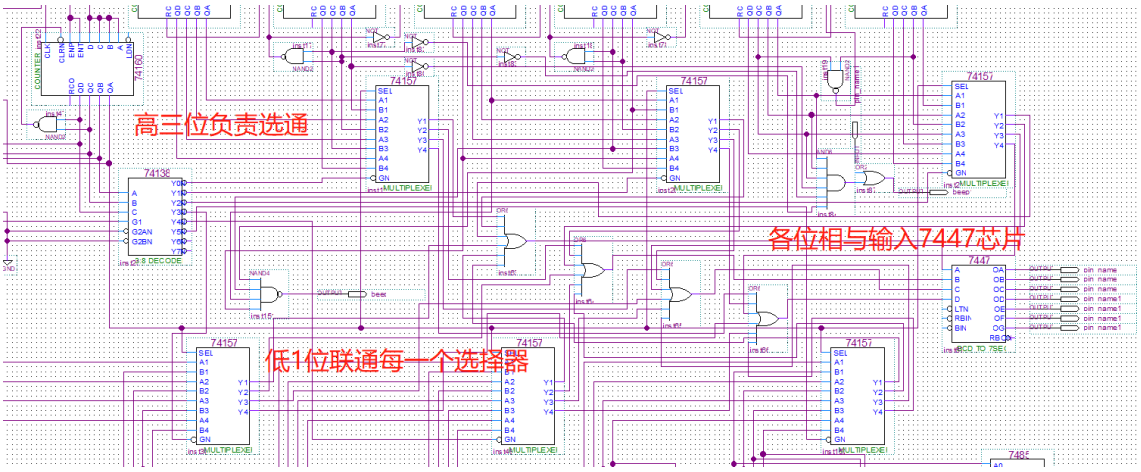

Figure 11: 控制选择器电路

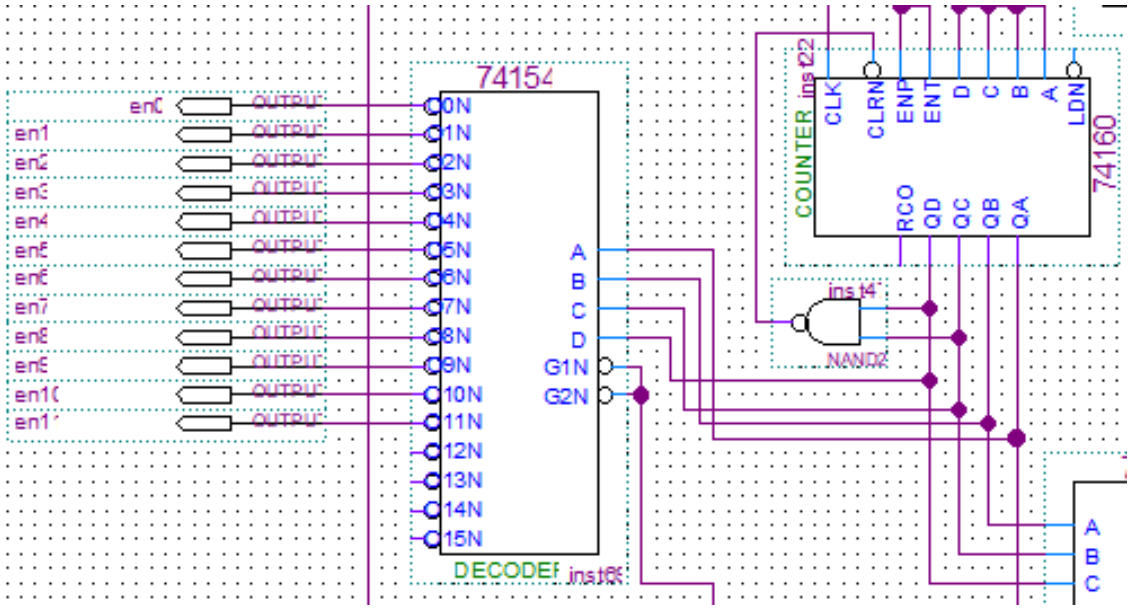

Figure 12: 控制数码管电路

4 整体展示

整体电路如下图，使用 Verilog 的模块（光源、分频、防抖）、纯电路模块（时钟、秒表、监督）。

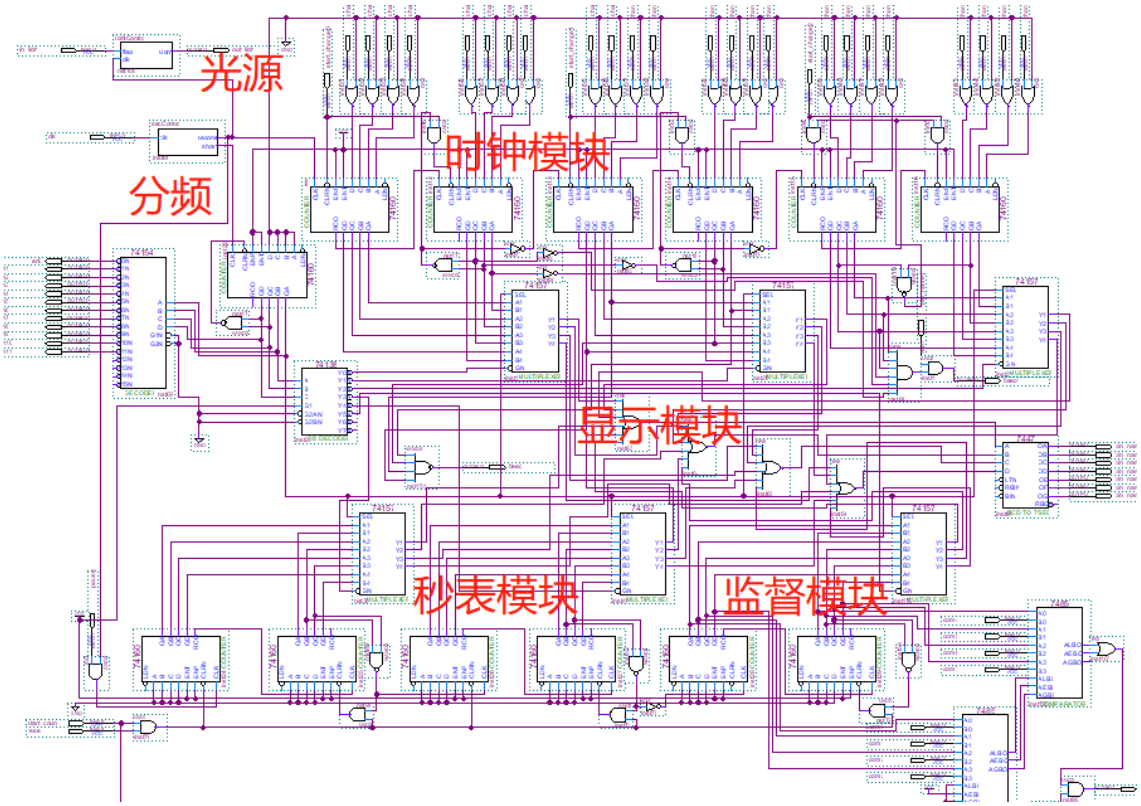

Figure 13: 整体电路

4.1 总体逻辑

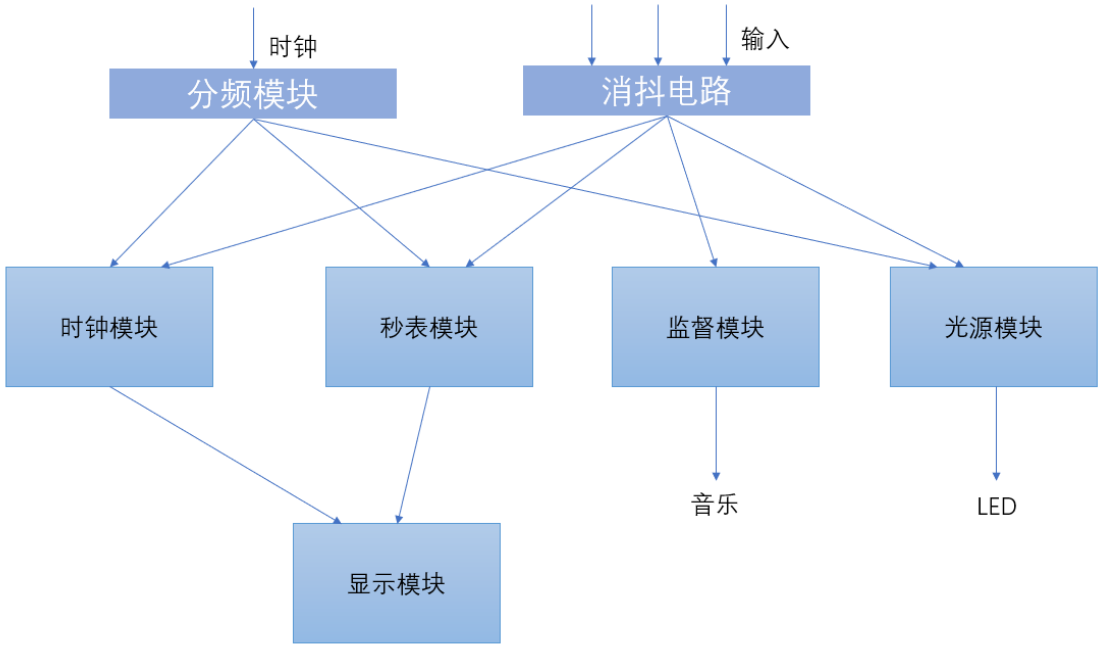

Figure 14: 整体逻辑

4.2 正确性展示

整体功能中，秒表与时钟功能类似，仅展示时钟。此外，还展示本产品的监督功能。

其余由于 quatus 内部无数码管，不予以仿真。

4.2.1 时钟功能

仅取时钟模块进行功能仿真，结果如下：

可以发现个十位都运作良好，在 59 秒后置零，实现 60 进制。由于功能仿真限制，仅展示该部分。

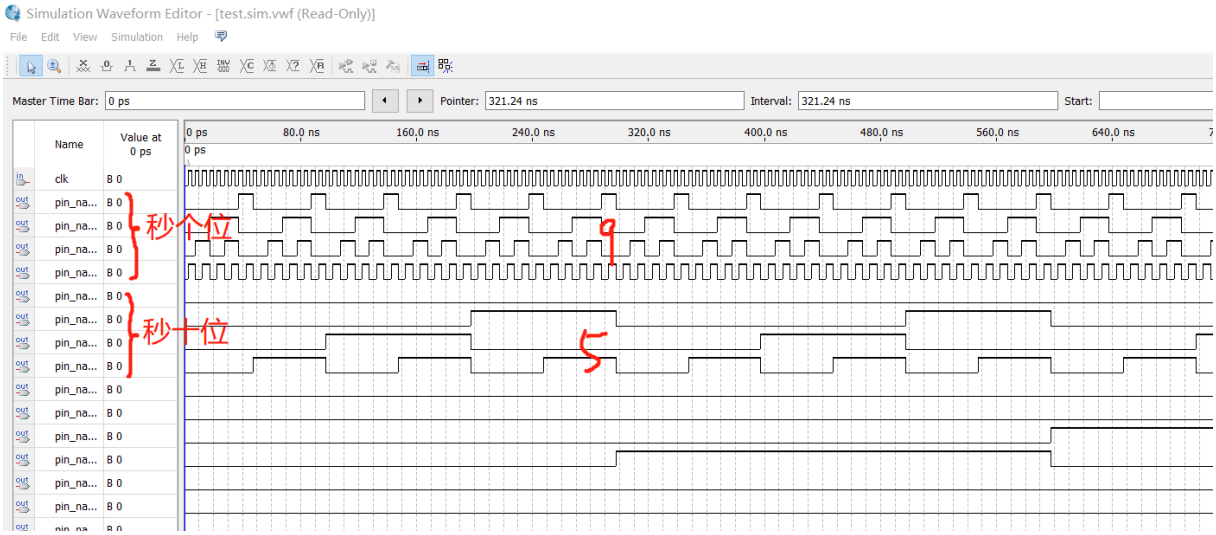

Figure 15: 时钟功能验证

4.2.2 监督功能

监督功能剥离秒表功能后得到下电路，仿真后可以发现，当 $first \geq com$ 时为 1，均正确。

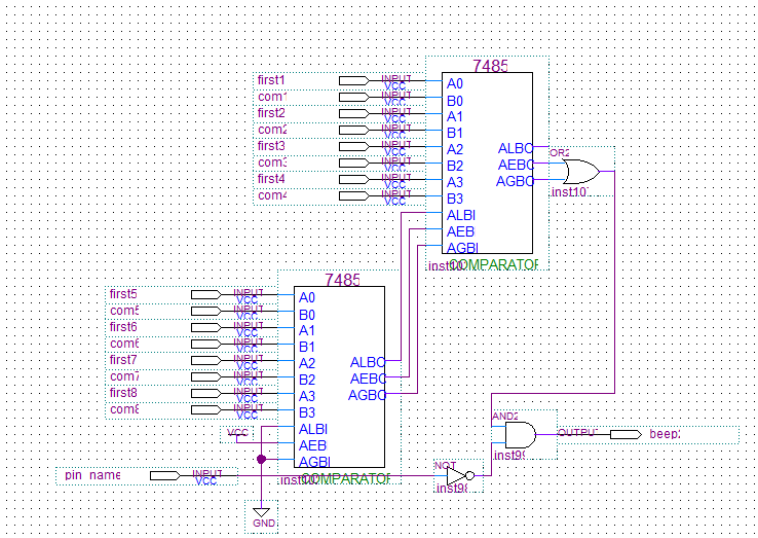

Figure 16: 监督功能验证 1

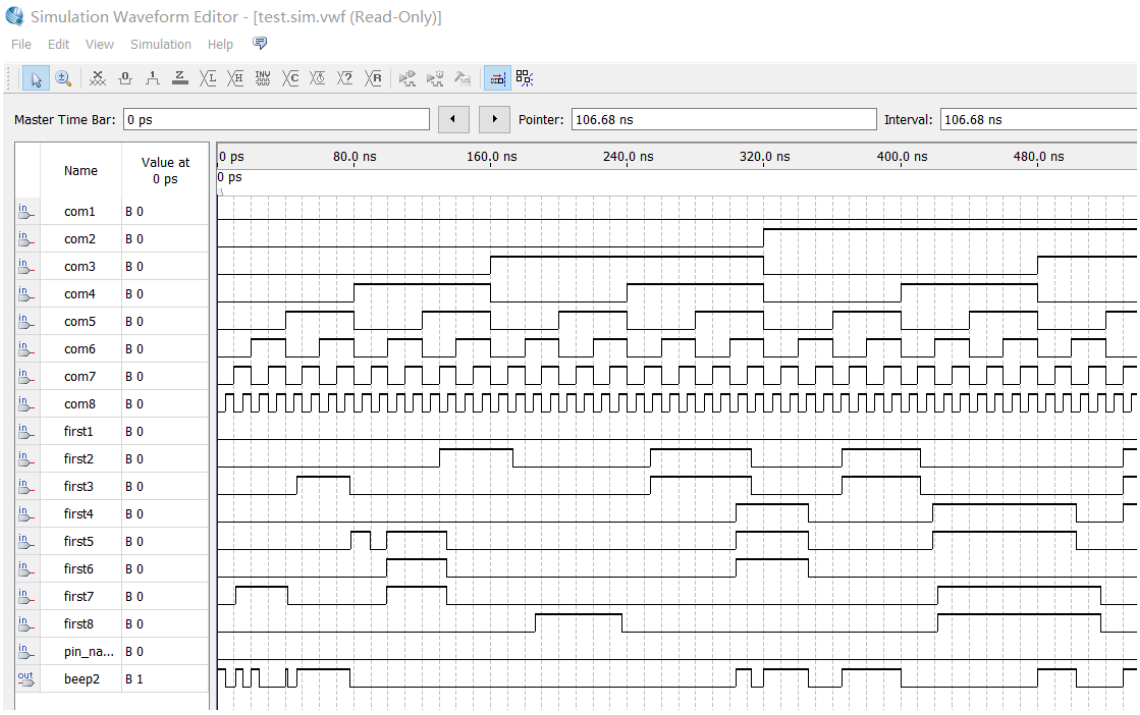

Figure 17: 监督功能验证 2

4.3 使用描述

4.3.1 时钟功能

当通电时，开始显示当前时间。如果想调整，可以打开不同开关，开启不同时间段的置数状态。

此时输入想要的数字便可以根据当时的正确时间给予调整，使用后关闭该开关。

当每天早上 7 点时会发出 10s（可以调整）的蜂鸣声（或者音乐声）。

4.3.2 秒表功能

当进入学习状态时，开启秒表功能，开始计时，设定自己的学习目标时长。

同时时钟功能仍在继续，中间可以随时暂停、清零、关闭秒表模式。

4.3.3 光源功能

按下光源按钮，产生 10s 钟的光源，以备不时之需。

4.3.4 监督功能

当学习结束，按下关闭按钮，装置会将设定时间与实际情况进行比较。

若大于设定时间，将会发出美妙音乐，并提醒使用者休息。

若小于则发出尖锐的声音（10s，可以根据场景关闭）。

5 总结及感想

本产品选择脱离手机平台，实现对学习的有效监督（奖励型），通过播放美妙的音乐、给予提醒、鼓励来提高使用者的使用热情。整体功能较为丰富，并非仅仅作为一个简单秒表，还融合了光源、可调节时钟的功能，让使用者在学习之余也能够让其派上用场。

本产品融合了许多课上学到的知识，消抖电路提高了输入的稳定性，显示电路有效利用芯片的引脚，减少不必要的电量消耗，分频电路同样重复利用了时钟发生器，还实现了 1Hz 的信号。主模块中，采用计数器实现时间的进制，辅助 verilog 进行电路设计。

在设计的过程中，我同时复习了书上的部分知识，包括计数器、比较器、译码器等中小规模集成电路的使用，还利用真值表简化电路。此次大作业我也是首次将硬件知识运用到了现实的实践中，感受到了硬件的趣味之处，发现硬件其实遍布生活的方方面面。并且在未来我也想要利用专业优势，从事软硬件结合领域的工作。

当然在设计中我也遇到了一些困难，例如如何用硬件实现特定的功能、quatus 的使用、verilog 与电路的结合。但是最后都通过查阅网络、书本资料，与同学交流成功解决了这些问题。

上述产品仍存在一些不足，例如无法指定闹钟时间、监督方式还可更多样化等。限于本人的精力原因，上述问题还有待解决，望日后有时间能够进一步改进。

最后，感谢老师一学期以来的教导，让我能够将硬件知识用于实际应用中。