

《数字电路与逻辑设计》课程设计报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

日期： 2014. 09. 24 分数： 指导教师： 胡迎松

一、课程设计内容

实现以下三个控制器中的至少一个：

- (1) 智能交通灯信号控制器；
- (2) 洗衣机控制器；
- (3) 智能电梯控制器。

二、 课程设计目的

通过硬件描述语言 Verilog HDL 的编程，深入了解并掌握可编程逻辑芯片的设计技术，加强学生对数字电路与逻辑设计知识的综合运用能力，培养学生创造性思维能力和独立解决实际问题的能力。

三、课程设计环境

- (1) 计算机 1 台；
- (2) Basys2 开发板（芯片为 XC3S100E，封装为 CP132）1 套；
- (3) Xilinx_ISE 设计仿真软件 1 套。

四、课程设计要求

- (1) 能够较全面地应用课程中所学的基本理论和基本方法，完成从设计逻辑电路到设计简单数字系统的过渡。
- (2) 能够独立思考、查阅资料，独立完成规定的设计任务。
- (3) 能够熟练使用 Verilog HDL 进行电路设计，并在指定可编程芯片上进行实现。

五、课程设计任务

- (1) 制定出详细设计方案；
- (2) 通过 Verilog HDL 完成规定的设计任务，然后进行编译和仿真，保证设计的正确性；
- (3) 生成熔丝图文件，下载到 FPGA 芯片，进行验证；
- (4) 对复杂系统的设计采取模块化、层次化的设计方法；
- (5) 撰出设计报告并对存在的问题进行分析、提出改进意见。

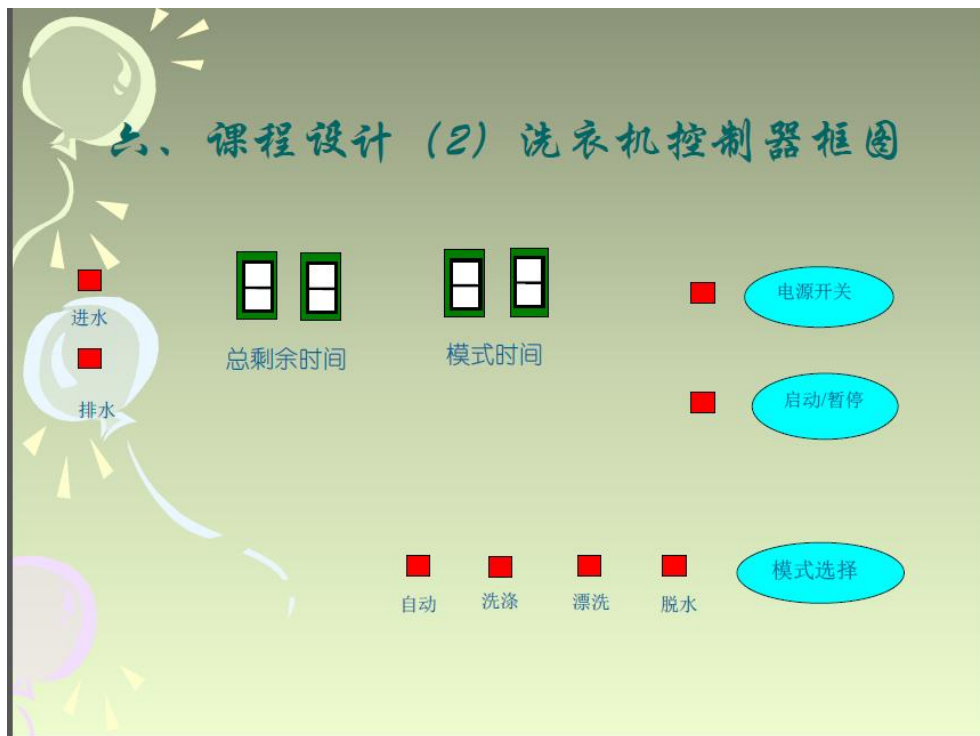
六、 方案设计

洗衣机控制器

[要求：给出详细的设计过程，包括相关说明、模块图、流程图，可续页]

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205



相关设计要求：

(1) 输入（开关）

- (A) 电源开关；
- (B) 启动/暂停开关；
- (C) 模式选择开关。

(2) 输出（Led8 盏、7 段代码显示灯 4 个）

(A) Led 指示灯

分别表示：

电源开关、启动/暂停、进水、排水、脱水、自动、洗涤、漂洗共 8 盏指示灯。

(B) 七段代码显示管

2 位表示洗涤剩余总时间，2 位表示当前模式剩余时间，采用倒计时显示。

(3) 电源开关

实际为电路总清零信号，初始状态为 0ff；拨动“电源开关”，洗衣机电路进入工作状态，电源指示灯亮。

默认为标准洗涤（含洗、漂、脱）程序。

(4) 启动/暂停开关

(A) 启动

在电源打开，“启动”指示灯不亮的情况下，按一下“启动/暂停”，“启动”指示灯亮，开始按设定的程序模式运行；

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

(B) 暂停

“启动”指示灯亮的情况下，按一下“启动/暂停”，此时洗衣机处于“暂停”状态，正在工作所对应的“灯”停止闪烁，其他的灯依旧，等待重新“启动”，继续工作。

(5) 模式选择开关（模式指示）

模式开关可选择 4 种工作模式，由 4 盏对应的 Led 灯分别表示，即：标准、洗涤、漂洗、脱水。

(A) 选择模式时，被选模式对应的指示灯亮”；

(B) 洗衣机工作时，模式中的（洗涤、漂洗、脱水）所对应的当前工作模式灯“闪烁”，未完成的模式灯“亮”，已未完成的模式灯“灭”；

(C) 设定

进水 3 分钟；洗涤 9 分钟；出水 3 分钟；漂洗 6 分钟；脱水 3 分钟。

(D) 标准洗：洗涤（进水、洗涤、排水、脱水）、漂洗（进水、漂、排水、脱水）、脱水，共 36 分钟；

(E) 洗涤：（进水、洗涤、排水、脱水），共 18 分钟；

(F) 漂洗：（排水、脱水、进水、漂、排水、脱水），共 21 分钟；

(G) 单脱：3 分钟。

各模块功能及说明：

1. 输入输出变量说明

```
//////////////////////////////////////  
module laundry(  
    input clk,  
    input power_button,  
    input pause_button,  
    input mode_button,  
    output reg power_led,  
    output pause_led,  
    output auto_led,  
    output [4:0] wash_led,  
    output reg [3:0] an,  
    output reg [7:0] ag  
);
```

clk: 时钟脉冲，与开发板上的接口 B8-50MHZ 对应

power_button, pause_button, mode_button 为电源、暂停、模式选择按键，对应的有相关的输出的灯，值得一提的是这里标准模式的灯单独列为一个变量，而后面进水、出水、洗、漂、脱由于后三盏灯既可作为模式选择时亮的灯又可作为过程指示时亮的灯故用 wash_led [4:0] 变量表示。最后的 an 和 ag 分别控制显示管的序号和控制相应的 8 位显示管。

2. clk 及 real_clk 模块

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
always @(posedge clk)
begin
    ns_timer = ns_timer+1'b1;
    if (ns_timer == 50000000)
    begin
        ns_timer = 0;
    end
    if (ns_timer < 25000000)
    begin
        real_clk = 1'b1;
    end
    else
    begin
        real_clk = 1'b0;
    end
end
end
```

此处为了节约时间将课设中要求的分钟缩小为秒钟用以快速计时，即每秒输出一个 real_clk 脉冲。

3. 7 位显示管显示模块

```
always @(posedge clk)
begin
    if (power_led)
    begin
        an_timer = an_timer+1'b1;
        if (an_timer == 50000)
        begin
            an_timer = 0;
        end
        if (an_timer < 12500)
```

此处代码只截取了一部分作为样例，详细代码参考附录。此模块利用了视觉暂留的原理，在一秒内多次反复显示四个显示管来造成实际上四个显示管同时显示的效果，其实显示管不可能同时显示不同数字。

4. 洗涤完成自动关机模块

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
always @(posedge pause_button or posedge wash_done or posedge power_button)
begin
    if (power_button)
    begin
        shutdown_start = 1'b0;
    end
    else if (wash_done)
    begin
        if (pause_save)
        begin
            shutdown_start = 1'b1;
        end
    end
    else
    begin
        if (pause_save)
        begin
            shutdown_start = 1'b0;
        end
    end
end

always @(posedge real_clk)
begin
    shutdown = 1'b0;

    if (shutdown_start == 1'b1 && shutdown_count == 5'b10100)
    begin
        shutdown = 1'b1;
    end
end
|
always @(posedge real_clk or posedge wash_done)
begin
```

此处也截取的部分代码用以示例，当洗涤过程结束时，shutdown_start 改变赋值致使 shutdown_count 开始计数，满足一定时间限制后开始 shutdown 改变赋值，自动关机。

5. 模式选择模块

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
always @(posedge mode_button or posedge wash_done or posedge power_button)
begin
    if (power_button)
    begin
        mode_save = 3'b000;
    end
    else if (wash_done)
    begin
        mode_save = 3'b000;
    end
    else
    begin
        if (power_led && pause_save)
        begin
            mode_save = mode_save+1'b1;
            if (mode_save == 3'b100)
            begin
                mode_save = 3'b000;
            end
        end
    end
end

always @(mode_save)
```

此模块约束了选择模式键对于选择模式的控制以及相应的模式时长及特征。其中值得一提的是在 mode_save 中的 mode_choose_state 和 auto_save 分别为洗漂脱三盏灯的一灯两用作为辨识变量提供了支持。在模式选择时和工作时这些变量将被赋予不同值。

6. 倒计时模块

```
always @(posedge real_clk or posedge mode_button or posedge power_button)
begin
    if (power_button)
    begin
        time_led = 6'b100100;
    end
end
```

此模块用于控制工作过程中的倒计时以及对工作模式下相应灯的控制，其中 auto_state 和 process_state 用于约束一灯两用功能时灯的状态。

7. 洗衣机过程亮灯及闪烁控制模块

```
always @(time_led)
begin
    case (mode_save)
        3'b000 : begin
            if (time_led > 6'b100001)
            begin
                wash_temp = 5'b11111;
                blink = 5'b01111;
            end
            else if (time_led > 6'b011000)
            begin
                wash_temp = 5'b11111;
            end
        end
    end
end
```

此处用 wash_temp 和 blink 两个变量来控制灯的亮灭及是否闪烁。之后

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

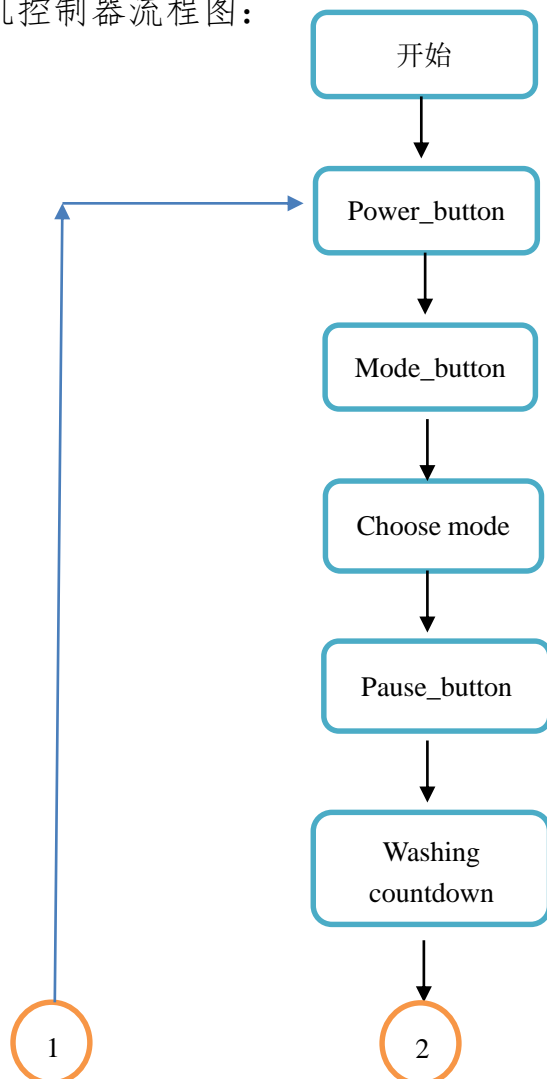
wash_led 工作模式下的约束主要跟这两个变量相关。

8. 灯及脉冲约束模块

```
assign wash_led = ((wash_temp & {real_clk | pause_led, real_c  
  
assign pause_led = pause_save & power_led;  
assign power_pulse = power_button | shutdown;  
assign pause_pulse = pause_button | wash_done;  
assign timer_pulse = real_clk | mode_button;  
  
assign auto_led = auto_state & auto_save & power_led;
```

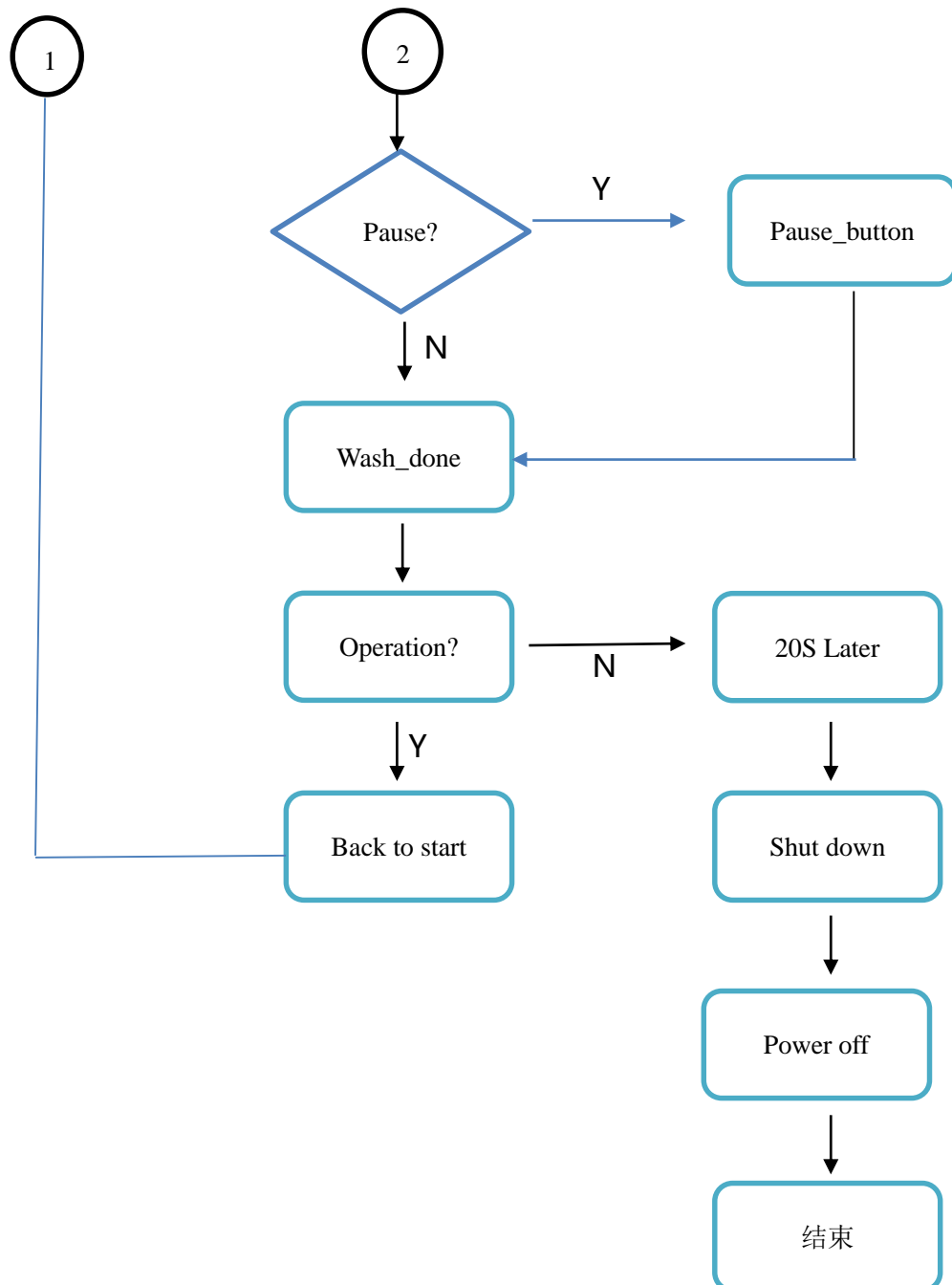
此处为代码核心设计，尤其为对 wash_led 的约束。通过对 wash_temp、real_clk、pause_led、blink、power_led、process_state、mode_choose_state 的约束来实现不同模式下的灯的控制。

洗衣机控制器流程图：



《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205



七、实验步骤与观察记录

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

1. 实验准备与需求分析

本次实验的选题有三个，分别是电梯控制器，洗衣机和交通灯，想到比较贴近生活的是洗衣机，我们就选择了这个选题。一开始只做过数电的几个比较简单的实验，还从来没有尝试编写工作量这么大的课设。但是在认真学习了老师下发的课设要求和夏宇闻老师编著的 Verilog 教程后，我们开始了对整个代码整体的架构。首先需要考虑的是数码管的显示问题，我们知道显示管上的接口只有一个，也就是说理论上的同时显示四个显示管只能显示一个数字或者图形，在查阅了相关资料后我们知道可以利用四个显示管交替反复显示的方式来达到视觉暂留而让用户看起来有同时显示的效果。

接下来的工作就是对整个程序的一些逻辑变量和输入输出的设置了。由于之前我们对每个模块都有比较详尽的规划，所以这个工作其实可以在一个模块实现的过程中在代码编写过程中添加。

在看到了课设过程记录时需要增加一些创意进去的时候，我们选择了室友建议的增添一个洗衣机完成后的定时关机功能。

2. 代码编写、调试及仿真

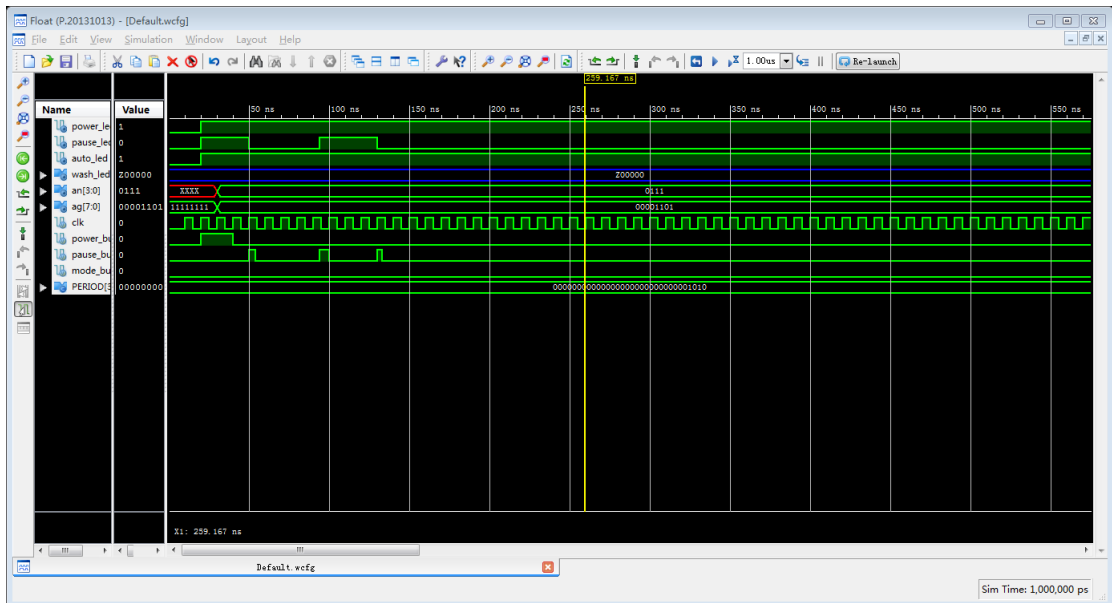
代码的编写其实是整个课设过程的最复杂的部分，因为我们数电课设时并没有大量的接触与课设紧密相关的 Verilog 语言，好在之前有其他软件类编程语言的基础，学习起来也不是很困难。在完成对整个过程的模块化处理后，我们比较顺利的完成了整个代码的构思。但是比较矛盾的地方是一开始的构思和 always 语句的实现是有一些冲突的，最突出的地方在于实现洗漂脱三个灯的不同工作模式下的不同作用，由于 always 语句的特殊性，我们编写最开始的几个模块时对其触发条件和里面实现的一些要求忽略了很多，导致编译时出现各种奇奇怪怪的错误，百度查找了一些方案后问题得到了部分解决。幸运的是室友之前也遇到了大量这一方面的问题，于是我看了他的一些对 always 语句的总结觉得比较深刻，在一灯两用的症结上也找到了突破方法，即增加一些关于不同模式的状态变量用以控制灯的闪烁方式和亮灭规则。

最后我们增加了定时关机功能，如果在洗衣过程结束后没有任何操作，则开始倒计时自动关机，这个过程的实现跟几个控制自动关机的变量 shutdown_start、shutdown_count、shutdown 密切相关，把握好这几个变量之间的相互约束以及洗涤完成时变量的变化，完成过程比较轻松。

以下是在 ISE14.7 上的仿真：

《数字电路与逻辑设计》实验报告

学生姓名: 赵子昂 学号: U201214946 所在班级: CS1205



可以看到仿真过程正常。

3. Basys2 上的实现

在仿真过程顺利实现后我们将 Bit 在 Basys2 上进行了 programming, 启动了电源后可以选择四种模式, 此时模式灯在选择模式的时候分别对应选择的模式, 在选定了模式后我们按 pause 键, 所有的除了 auto_led 之外的模式灯全部变为过程灯, 当前过程下对应的灯闪烁, 未完成的过程亮, 而完成的灯将会熄灭, 整个过程随时可以暂停。洗涤结束后不进行操作则在一段时间后会自动关机。整个过程圆满成功!

八、对实验过程的分析、处理及讨论

本次课设总体来说是比较成功的，比较成功的模块化处理和对 always 语句的使用以及对一些变量的巧妙设置是成功的关键，在这里我想引用我室友的一段总结作为对 always 语句的一些见解，在他当时做数电课设时我与其讨论并查找相关资料收获很大。引述如下：

1.always 语句包含多个敏感事件时（下面省略这句话），内部需要对触发该 always 的事件做分开处理，用 if-else 就很好；

2. 在 `always` 里面进行赋值的寄存器类型，必须从逻辑上与每个触发事件都要有相关性，否则不能写到该 `always` 语句中，应当为这个寄存器类型再开一个 `always`，只将与其有关的事件作为敏感对象；举个例子：比如 `always @(posedge a or posedge b)` 中，我在 `if (a)` 时对某寄存器 `temp` 赋值，但是却没有在 `if (b)` 中对 `temp` 赋值，那么 `temp` 其实与 `b` 无关，这样极有可能出错，应该为 `temp` 重新写一个 `always @(posedge a)` 来触发。至于产生问题的原因，应当归结于 Verilog 语言的特点以及真实电路的逻辑实现问题。如果一个输入与输出无关，虽然语法正确，但电路似乎无法生成，因此会返回

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

一个错误提示，表示不符合 FF template 什么的，无法生成电路。

3. 除了要满足赋值的变量与所有事件都有相关性之外，还要注意内部的 if-else 对事件点评的判断顺序；应该将 if 为真时间短放在前面，比如 if (clk)，其次再放 if 为真时间长的比如 if (mode_button)；这样做的原因是，如果你 always @(posedge clk or posedge mode_button)，如果你把 mode_button 放在前面判断，那么当你按下 mode_button 的时候，由于 clk 脉冲很短很快，在你按下的时候会触发很多次这个 always 块，同时 if (mode_button) 为真，那么就执行了 if (mode_button) 语句块的内容，导致表现出来的结果是 mode_button 的结果乱变就像按了无数次按钮，因而出错；

4. 最后是 always 语句使用时要注意的地方：不能在两个 always 语句块中对同一个变量赋值，理由很简单，在电路中不可能存在两根线可以改同一个输出的。但是，将变量拿来当判断条件是没有限制的，其实也就是拿来当电路模块的输入。应当灵活使用这一点，将 always 语句中的多个触发事件逐一拆开，一直拆到符合上述的 1~3 规范为止，这样的电路就肯定没有跳变沿触发上面的问题。

可以说整个洗衣机控制电路的实现都离不开上面的这些小细节，也正是 always 这一系列单调的触发语句才构成了一个完整的洗衣机控制电路。

除此之外，整个实验过程中代码编写是比较繁杂的，尤其是对显示管控制电路的实现和对过程灯的闪烁和明灭的 wash_temp 和 blink 的控制，需要很大的代码量来实现。

整个逻辑电路的每一个部分都是值得思考的，可以说没有任何一个部分能够那么能够轻易的实现。

九、心得体会

本次数电课设总体来说是非常能够反映我们对数电这门课程的理解的一次考验。从前期的控制电路整体的架构到后期每一点细枝末节的处理我们都需要花费很多的功夫。一开始我们对实验任务书的理解并非那么透彻，这让我们对模块的划分和阶段的处理有了比较多的分歧和误解，但是仔细研读后发现其实这是有据可依的。到后来自动关机功能的加入丰富了洗衣机的功能，也让我们新鲜感自豪感增添了不少。在这里我想特别感谢我的室友给了我很多的指导和帮助，在遇到瓶颈时他也很乐意为我解答一些问题研究一些策略，最后一灯两用功能的完美实现里面也有他的功劳。其实课设训练我们的不仅是工程素养，更是一种乐于并且执着于去解决问题的能力，还有就是一种让人振奋的团队精神。以后我们能学到的还有很多！

十、附录及参考文献

源代码：

文件名：laundry.v

```
`timescale 1ns / 1ps
```

```
////////////////////////////////////////////////////////////////
```

```
// Company:
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
// Engineer:
//
// Create Date:    01:31:07 09/22/2014
// Design Name:
// Module Name:    laundry
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////
module laundry(
    input clk,
    input power_button,
    input pause_button,
    input mode_button,
    output reg power_led,
    output pause_led,
    output auto_led,
    output [4:0] wash_led,
    output reg [3:0] an,
    output reg [7:0] ag
);

    reg [5:0] time_led;
    reg [4:0] blink;
    reg [2:0] mode_save;
    reg [4:0] wash_temp;
    reg [5:0] time_limit;

    reg [4:0] shutdown_count;
    reg wash_done;
    reg shutdown_start;
    reg shutdown;
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
reg [25:0] ns_timer;
reg real_clk;
reg [15:0] an_timer;
reg [5:0] ag_helper;
reg pause_save;
wire power_pulse;
wire pause_pulse;
wire timer_pulse;
reg [4:0] process_state;
reg [4:0] mode_choose_state;
reg auto_save;
reg auto_state;

initial
begin

    time_led = 6'b100100;
    mode_save = 3'b000;
    power_led = 0;
    pause_save = 1;
    wash_temp = 5'b000000;
    blink = 5'b000000;
    time_limit = 6'b100100;

    wash_temp = 5'b111111;

    blink = 5'b111111;
    wash_done = 1'b0;
    shutdown_count = 5'b000000;
    shutdown_start = 1'b0;
    shutdown = 1'b0;
    process_state=5'b000000;
    mode_choose_state=5'b000000;
    auto_save=1'b1;
    auto_state=1'b1;

    ns_timer = 0;
    real_clk = 1'b1;
    an_timer = 0;
    ag = 8'b11111111;

end
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
always @(posedge clk)
begin
    ns_timer = ns_timer+1'b1;
    if (ns_timer == 50000000)
    begin
        ns_timer = 0;
    end
    if (ns_timer < 25000000)
    begin
        real_clk = 1'b1;
    end
    else
    begin
        real_clk = 1'b0;
    end
end

always @(posedge clk)
begin
    if (power_led)
    begin
        an_timer = an_timer+1'b1;
        if (an_timer == 50000)
        begin
            an_timer = 0;
        end
        if (an_timer < 12500)
        begin
            an = 4'b0111;
            if (time_limit > 29)
            begin
                ag = 8'b00001101;
            end
            else if (time_limit > 19)
            begin
                ag = 8'b00100101;
            end
            else if (time_limit > 9)
            begin
                ag = 8'b10011111;
            end
        end
    end
end
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        else
        begin
            ag = 8'b00000011;
        end
    end
else if (an_timer < 25000)
begin
    an = 4'b1011;
    if (time_limit > 29)
    begin
        ag_helper=time_limit-30;
    end
    else if (time_limit > 19)
    begin
        ag_helper = time_limit-20;
    end
    else if (time_limit > 9)
    begin
        ag_helper = time_limit-10;
    end
    else
    begin
        ag_helper = time_limit;
    end
    case (ag_helper)
        6'b000000 : ag = 8'b00000011;
        6'b000001 : ag = 8'b10011111;
        6'b000010 : ag = 8'b00100101;
        6'b000011 : ag = 8'b00001101;
        6'b000100 : ag = 8'b10011001;
        6'b000101 : ag = 8'b01001001;
        6'b000110 : ag = 8'b01000001;
        6'b000111 : ag = 8'b00011111;
        6'b001000 : ag = 8'b00000001;
        6'b001001 : ag = 8'b00001001;
    endcase
end
else if (an_timer < 37500)
begin
    an = 4'b1101;
    if (time_led > 29)
    begin
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        ag = 8'b00001101;
    end
    else if (time_led > 19)
    begin
        ag = 8'b00100101;
    end
    else if (time_led > 9)
    begin
        ag = 8'b10011111;
    end
    else
    begin
        ag = 8'b00000011;
    end
end
else
begin
    an = 4'b1110;
    if (time_led > 29)
    begin
        ag_helper=time_led-30;
    end
    else if (time_led > 19)
    begin
        ag_helper = time_led-20;
    end
    else if (time_led > 9)
    begin
        ag_helper = time_led-10;
    end
    else
    begin
        ag_helper = time_led;
    end
    case (ag_helper)
        6'b000000 : ag = 8'b00000011;
        6'b000001 : ag = 8'b10011111;
        6'b000010 : ag = 8'b00100101;
        6'b000011 : ag = 8'b00001101;
        6'b000100 : ag = 8'b10011001;
        6'b000101 : ag = 8'b01001001;
        6'b000110 : ag = 8'b01000001;
```


《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        6'b000111 : ag = 8'b00011111;
        6'b001000 : ag = 8'b00000001;
        6'b001001 : ag = 8'b00001001;
    endcase
end
end
else
begin
    ag = 8'b11111111;
end
end

always @(posedge power_pulse)
begin
    power_led = ~power_led;
end

always @(posedge pause_pulse or posedge power_button)
begin
    if (power_button)
    begin
        pause_save = 1'b1;
    end
    else
    begin
        pause_save = ~pause_save;
    end
end

always @(posedge pause_button or posedge wash_done or posedge power_button)
begin
    if (power_button)
    begin
        shutdown_start = 1'b0;
    end
    else if (wash_done)
    begin
        if (pause_save)
        begin
            shutdown_start = 1'b1;
        end
    end
end
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        else
        begin
            if (pause_save)
            begin
                shutdown_start = 1'b0;
            end
        end
    end

    always @(posedge real_clk)
    begin
        shutdown = 1'b0;

        if (shutdown_start == 1'b1 && shutdown_count == 5'b10100)
        begin
            shutdown = 1'b1;
        end
    end

    always @(posedge real_clk or posedge wash_done)
    begin
        if (wash_done)
        begin
            shutdown_count = 5'b00000;
        end
        else
        begin
            if (shutdown_start && power_led && pause_save)
            begin
                shutdown_count = shutdown_count+1'b1;
                if (shutdown_count == 5'b10101)
                begin
                    shutdown_count = 5'b00000;
                end
            end
        end
    end

    always @(posedge mode_button or posedge wash_done or posedge power_button)
    begin
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        if (power_button)
        begin
            mode_save = 3'b000;
        end
        else if (wash_done)
        begin
            mode_save = 3'b000;
        end
        else
        begin
            if (power_led && pause_save)
            begin
                mode_save = mode_save+1'b1;
                if (mode_save == 3'b100)
                begin
                    mode_save = 3'b000;
                end
            end
        end
    end

always @(mode_save)
begin
    case (mode_save)
        3'b000 : begin
                                time_limit = 6'b100100;
                                mode_choose_state=5'b00000;
                                auto_save=1;
                            end
        3'b001 : begin
                                time_limit = 6'b010010;
                                mode_choose_state=5'b00100;
                                auto_save=0;
                            end
        3'b010 : begin
                                time_limit = 6'b010101;
                                mode_choose_state=5'b00010;
                                auto_save=0;
                            end
        3'b011 : begin
                                time_limit = 6'b000011;
                                mode_choose_state=5'b00001;

```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        auto_save=0;
    end

endcase
end

always @(posedge real_clk)
begin
    wash_done = 1'b0;
    if (time_led == 0)
    begin
        wash_done = 1'b1;
    end
end

always @(posedge real_clk or posedge mode_button or posedge power_button)
begin
    if (power_button)
    begin
        time_led = 6'b100100;
    end
    else if (mode_button)
    begin
        if (power_led && pause_save)
        begin
            time_led = time_limit;
        end
    end
    else
    begin
        if (power_led && ~pause_save)
        begin
            if (time_led > 0)
            begin
                time_led = time_led-1'b1;
                process_state=5'b11111;
                auto_state=1'b0;
            end
            else
            begin
                time_led = 6'b100100;
                process_state=5'b00000;
                auto_state=1'b1;
            end
        end
    end
end
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        end
    end
end

always @(time_led)
begin
    case (mode_save)
        3'b000 : begin
            if (time_led > 6'b100001)
            begin
                wash_temp = 5'b11111;
                blink = 5'b01111;
            end
            else if (time_led > 6'b011000)
            begin
                wash_temp = 5'b11111;
                blink = 5'b11011;
            end
            else if (time_led > 6'b010101)
            begin
                wash_temp = 5'b11011;
                blink = 5'b10011;
            end
            else if (time_led > 6'b010010)
            begin
                wash_temp = 5'b11011;
                blink = 5'b11010;
            end
            else if (time_led > 6'b001111)
            begin
                wash_temp = 5'b11011;
                blink = 5'b01011;
            end
            else if (time_led > 6'b001001)
            begin
                wash_temp = 5'b01011;
                blink = 5'b01001;
            end
            else if (time_led > 6'b000110)
            begin
                wash_temp = 5'b01001;
            end
        end
    end
end
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
        blink = 5'b00001;
    end
    else if (time_led > 6'b000011)
    begin
        wash_temp = 5'b00001;
        blink = 5'b00000;
    end
    else
    begin
        wash_temp = 5'b00001;
        blink = 5'b00000;
    end
    end
end
3'b001 : begin
    if (time_led > 6'b001111)
    begin
        wash_temp = 5'b11101;
        blink = 5'b01101;
    end
    else if (time_led > 6'b000110)
    begin
        wash_temp = 5'b01101;
        blink = 5'b01001;
    end
    else if (time_led > 6'b000011)
    begin
        wash_temp = 5'b01001;
        blink = 5'b00001;
    end
    else
    begin
        wash_temp = 5'b00001;
        blink = 5'b00000;
    end
    end
end
3'b010 : begin
    if (time_led > 6'b010010)
    begin
        wash_temp = 5'b11011;
        blink = 5'b10011;
    end
    else if (time_led > 6'b001111)
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
begin
    wash_temp = 5'b11011;
    blink = 5'b11010;
end
else if (time_led > 6'b001100)
begin
    wash_temp = 5'b11011;
    blink = 5'b01011;
end
else if (time_led > 6'b000110)
begin
    wash_temp = 5'b01011;
    blink = 5'b01001;
end
else if (time_led > 6'b000011)
begin
    wash_temp = 5'b01001;
    blink = 5'b00001;
end
else
begin
    wash_temp = 5'b00001;
    blink = 5'b00000;
end
end
3'b011 : begin

begin
    wash_temp = 5'b00001;
    blink = 5'b00000;
end

end

endcase
end

assign wash_led = ((wash_temp & {real_clk | pause_led, real_clk | pause_led, real_clk | pause_led,
real_clk | pause_led, real_clk | pause_led}) | blink) & {power_led, power_led, power_led, power_led,
power_led} & (process_state | mode_choose_state);
```

《数字电路与逻辑设计》实验报告

学生姓名： 赵子昂 学号： U201214946 所在班级： CS1205

```
assign pause_led = pause_save & power_led;

assign power_pulse = power_button | shutdown;

assign pause_pulse = pause_button | wash_done;

assign timer_pulse = real_clk | mode_button;

assign auto_led = auto_state & auto_save & power_led;
endmodule
```

参考文献

1. 欧阳明星. 《数字逻辑》. 4 版. 武汉: 华中科技大学出版社. 2007
2. [美] J.BHASKER 著. 夏宇闻 甘伟 译. 《A Verilog HDL Primer》, Third Edition
3. 华科数字逻辑课程组. 《数字逻辑实验指导书》