

专业 人工智能 班级 2202 日期 2024.3.18 成绩  
实验组别 10 第 四 次实验 指导教师 潘晓明  
学生姓名 秦明远

## 实验名称

FPGA综合设计

## 一、实验任务

- (必做) 基本功能
  - 准确计时，以数字形式（小时是“24 进制”，分是“60 进制”）显示小时和分钟的时间
  - 校时：能调整小时、分钟的时间
- (选做) 提高
  - 任意闹钟
  - 小时为 12/24 进制可切换
  - 报正点数（几点钟 LED 闪烁几下）

## 二、设计平台

## 1. 软件设计流程及作用

- 编写Verilog HDL设计源文件，对要实现电路进行语言描述，并对程序进行编译检查；
- 编写仿真文件，对源文件生成的电路进行测试，通过波形图等方式，检验其逻辑功能是否符合要求。利用软件进行电路设计，改错成本较低，同时也排除了因为接线等原因产生的错误。

## 2. 硬件平台简要介绍

- 通过编辑文本文件，确定接口位置，即将FPGA板的引脚编号与软件设计中的各输入、输出量对应起来；

## 三、设计源文件及注释

//-----顶层模块

```
`timescale 1ns / 1ps
module
top_clock(CLK_50M,nCR,Adj_Min,Adj_Hour,EN,Change,Switch0,Segout,AN,Second);
    input CLK_50M;
    input nCR; //总开关=sw0

    input Adj_Min; //sw2

    input Adj_Hour; //sw4

    input EN; //sw1

    input Change; //sw7

    input Switch0; //BTN0
```

```

output [7:0] Segout; //段码

output [3:0] AN; //位码

output [7:0] Second; //表示秒数的BCD码

    //supply1 Vdd;
wire MinL_EN,MinH_EN,Hour_EN;

wire CLK_1k,CLK_1;

wire [15:0] BCD;

wire [7:0] Hour,Minute;

//进行分频操作
divided50MHz D1(CLK_1k,nCR,CLK_50M);
divided1Hz D2(CLK_1,nCR,EN,CLK_1k);

//60进制秒计数器：调用10进制和6进制底层模块构成
counter10 U1(CLK_1,nCR,EN,Second[3:0]);
counter6 U2(CLK_1,nCR,(Second[3:0]==4'h9),Second[7:4]);

//产生分钟计数器使能信号。Adj_Min=1，校正分钟；Adj_Min=0，分钟正常计时
assign MinL_EN=Adj_Min? 1'b1:(Second==8'h59);
assign
MinH_EN=(Adj_Min&&(Minute[3:0]==4'h9))|((Minute[3:0]==4'h9)&&(Second==8'h59));
//60进制分钟计数器：调用10进制和6进制底层模块构成
counter10 U3(CLK_1,nCR,MinL_EN,Minute[3:0]);
counter6 U4(CLK_1,nCR,MinH_EN,Minute[7:4]);

//产生小时计数器使能信号。Adj_Hour=1，校正小时；Adj_Hour=0，小时正常计时
assign Hour_EN=Adj_Hour? 1'b1:((Minute==8'h59)&&(Second==8'h59));

//24进制小时计数器：调用24进制底层模块构成
counter24 U5(CLK_1,nCR,Hour_EN,Change,Hour[3:0],Hour[7:4]);

//进行数码管显示操作
assign BCD = (Switch0) ? {Minute, Second} : {Hour, Minute};
RippleDisp TO(CLK_50M, nCR, BCD, Segout, AN);

Endmodule

```

```
//-----数码管显示模块
```

```
module RippleDisp (CLK_50M,CLR,BCD,Segout,AN);
```

```
input CLK_50M;
```

```
input CLR;
```

```
input [15:0] BCD;
```

```
output reg [7:0] Segout;
```

```
output reg [3:0] AN;
```

```
wire S1,S0;
```

```
wire [3:0] En;
```

```
reg [3:0] InDigit;
```

```
reg [19:0] Count;
```

```
//时钟分频器
```

```
always @(posedge CLK_50M or negedge CLR)
```

```
begin
```

```
    if(~CLR)
```

```
        Count<=0;
```

```
    else
```

```
        Count<=Count+1;
```

```
end
```

```
assign {S1,S0}=Count[19:18];
```

```
//刷新整个周期, T=20.97ms, T/4=5.24ms
```

```
//四位4选1数据选择器, 控制具体哪个数码管发光
```

```
always @(*)
```

```
begin
```

```
    case ({S1,S0})
```

```
        2'b00: InDigit=BCD[3:0];
```

```
        2'b01: InDigit=BCD[7:4];
```

```
        2'b10: InDigit=BCD[11:8];
```

```
        2'b11: InDigit=BCD[15:12];
```

```
    endcase
```

```
end
```

```
//七段显示译码器模块
```

```
always @(InDigit)
```

```
    case (InDigit)
```

```
        0: Segout=8'b11000000; //显示0,最高位为小数点, 后七位从右到左为a,b,c,d,e,f,g
```

```
        1: Segout=8'b11111001; //显示1
```

```
        2: Segout=8'b10100100; //显示2
```

```
        3: Segout=8'b10110000; //显示3
```

```
        4: Segout=8'b10011001; //显示4
```

```

5: Segout=8'b10010010; //显示5
6: Segout=8'b10000010; //显示6
7: Segout=8'b11111000; //显示7
8: Segout=8'b10000000; //显示8
9: Segout=8'b10010000; //显示9
default: Segout=8'b11000000; //不显示
endcase

```

//2-4线译码器模块

```

assign En[3]=1; //最左边一位为0
assign En[2]=1; //最左边两位为0
assign En[1]=1; //最左边三位为0
assign En[0]=1; //最低位不灭0

```

```

always @(*)
begin
    AN=4'b1111;
    //if(En[{S1,S0}]==1)
    AN[{S1,S0}]=0;
end
endmodule

```

//-----六进制计数器

```

module counter6(CP,nCR,EN,Min);
input CP;
input nCR; //异步清零信号
input EN; //数字钟使能信号
output reg[3:0] Min;

always @(posedge CP or negedge nCR)
begin
    if(~nCR) Min<=4'b0000; //nCR=0, 计数器被异步清零
    else if(~EN) Min<=Min; //EN=0, 暂停计数
    else if(Min==4'b0101) Min<=4'b0000;
    else Min<=Min+1'b1; //计数器增1计数
end
endmodule

```

//-----十进制计数器

```

module counter10(CP,nCR,EN,Min);
input CP;
input nCR; //异步清零信号
input EN; //数字钟使能信号

```

```

output reg[3:0] Min;

always @(posedge CP or negedge nCR)
begin
    if(~nCR) Min<=4'b0000; //nCR=0, 计数器被异步清零
    else if(~EN) Min<=Min; //EN=0, 暂停计数
    else if(Min==4'b1001) Min<=4'b0000;
    else Min<=Min+1'b1; //计数器增1计数
end
endmodule

```

//-----二十四/十二进制计数器

```

module counter24(CP, nCR, EN, Change, HrL, HrH);
input CP;
input nCR;
input EN;
input Change;
output reg [3:0] HrL;
output reg [7:4] HrH;

always @(posedge CP or negedge nCR) begin
    if (nCR == 0) begin
        HrH <= 4'b0000; // 异步清零
        HrL <= 4'b0000; // 异步清零
    end
    else if (~EN) begin
        // 使能En为0时保持原有状态, 无需做任何操作
    end
    else begin
        if (Change) begin
            if ((HrH > 2) || (HrL > 9) || ((HrH == 2) && (HrL >= 3))) begin
                HrH <= 8'h00; // 当满足条件时清零
                HrL <= 8'h00; // 当满足条件时清零
            end
            else if ((HrH == 2) && (HrL < 3)) begin
                HrL <= HrL + 1'b1; // 个位加1
            end
            else if (HrL == 9) begin
                HrL <= 4'b0000; // 个位清零
                HrH <= HrH + 1'b1; // 十位加1
            end
            else begin
                HrL <= HrL + 1'b1; // 其他情况, 个位加1
            end
        end
    end
end

```

```

end
else begin
    if ((HrH > 1) || (HrL > 9) || ((HrH == 1) && (HrL >= 1))) begin
        HrH <= 8'h00; // 当满足条件时清零
        HrL <= 8'h00; // 当满足条件时清零
    end
    else if ((HrH == 1) && (HrL < 1)) begin
        HrL <= HrL + 1'b1; // 个位加1
    end
    else if (HrL == 9) begin
        HrL <= 4'b0000; // 个位清零
        HrH <= HrH + 1'b1; // 十位加1
    end
    else begin
        HrL <= HrL + 1'b1; // 其他情况，个位加1
    end
end
end
end
endmodule
//-----50Mhz到1khz分频
module divided50MHz(CLK_1k,nCR,CLK_50M);
input CLK_50M,nCR; //异步清零端
output reg CLK_1k;
reg [15:0]q; //用于存储计数数字

always @(posedge CLK_50M,negedge nCR)
begin
    if(~nCR) //异步清零
    begin
        q<=15'd0;
        CLK_1k<=1'b0;
    end //当计数到24999时，1khz变量翻转，因为50mhz频率是1000hz的50000倍

    else if(q==15'b110000110100111)
    begin //1khz的周期是50mhz的50000倍，24999个周期翻转一次，翻转两次构成一个周期，恰好满足50000倍的周期关系，也就达到分频的效果
        q<=15'd0;
        CLK_1k<=(~CLK_1k);
    end

    else
    begin
        q<=q+1'b1; //正常加1
    end
end
end

```

```
        end
    end
endmodule
```

```
//-----1khz到1hz分频
```

```
module divided1Hz(CLK_1,nCR,EN,CLK_1k);
```

```
    input CLK_1k,nCR,EN;
```

```
    output CLK_1; //分频器的结果输出，即不同频率的信号
```

```
    wire [11:0] q; //用于获取分频信号的变量，在不同位上取，就可获得不同的频率信号
```

```
    wire EN1,EN2; //使能信号
```

```
    //通过使用三个十进制计数器完成1000进制的计数器扩展
```

```
    counter10 du0(CLK_1k,nCR,EN,q[3:0]);
```

```
    counter10 du1(CLK_1k,nCR,EN1,q[7:4]);
```

```
    counter10 du3(CLK_1k,nCR,EN2,q[11:8]);
```

```
    assign EN1=(q[3:0]==4'd9);
```

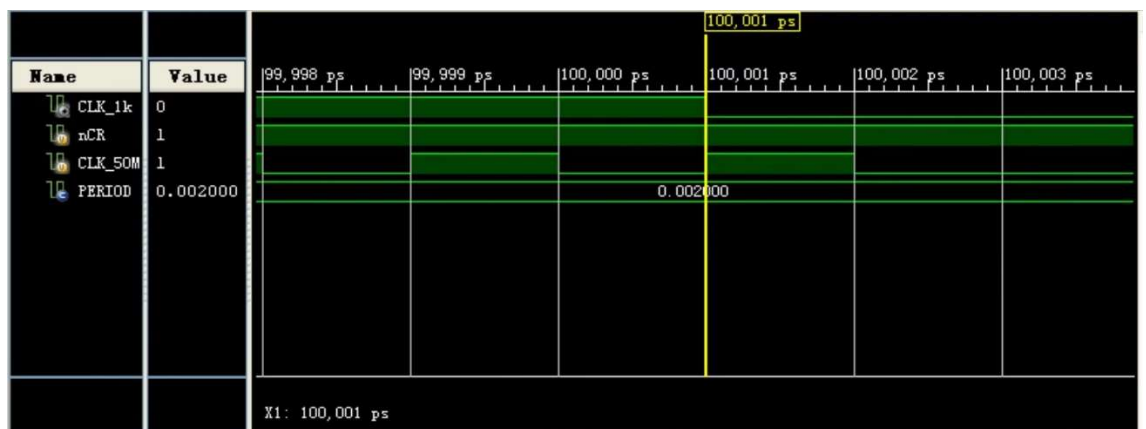
```
    assign EN2=(q[7:4]==4'd9)&(q[3:0]==4'd9);
```

```
    assign CLK_1=q[11]; //最低位实现二分频，位往高位移动，依次类推
```

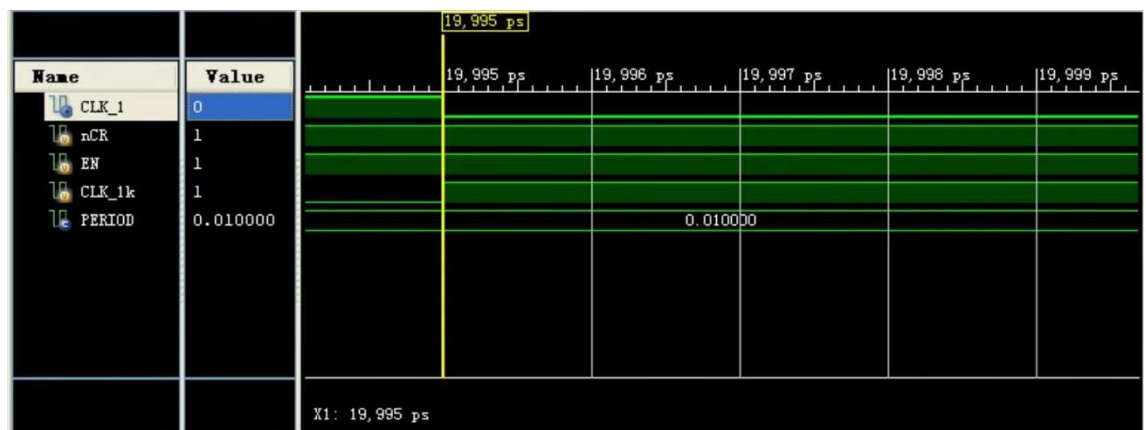
```
endmodule
```







- 1kHz 到 1Hz 分频



- 整体仿真

```

module test(
    input CLK_50M,
    input Adj_Min,
    input Adj_Hour,
    input Change,
    input QMY,
    output [7:0] Segout,
    output [3:0] AN,
    output [7:0] Second
);
    parameter PERIOD = 20;
    reg CLK;

```

```

always
begin
    CLK = 1'b0;
    #(PERIOD/2) CLK = 1'b1;
    #(PERIOD/2);
end

reg nCR;
wire EN;

top_clock top_clock1( CLK_50M, nCR, Adj_Min, Adj_Hour, EN, Change, QMY,
Segout, AN, Second);

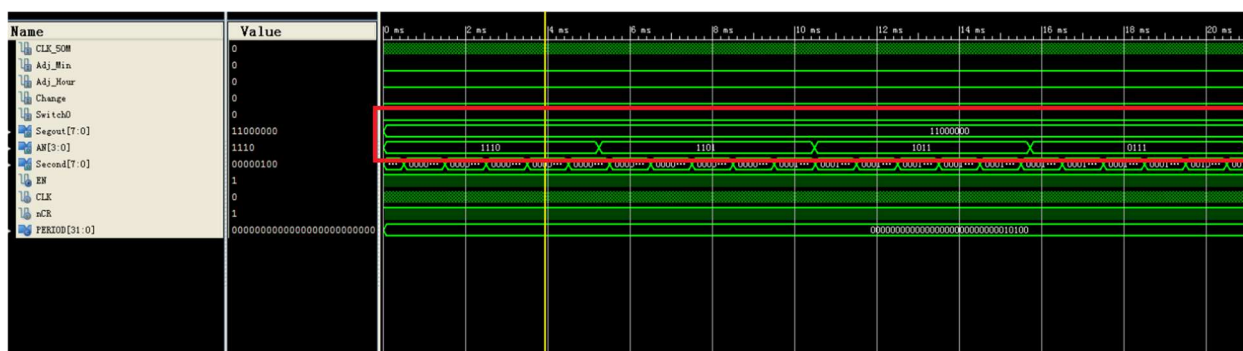
assign EN = 1;
assign CLK_50M = CLK;
assign Adj_Min = 0;
assign Adj_Hour = 0;
assign Change = 0;
assign QMY = 0;

initial
begin
    nCR = 0;
    #10000
    nCR = 1;
end

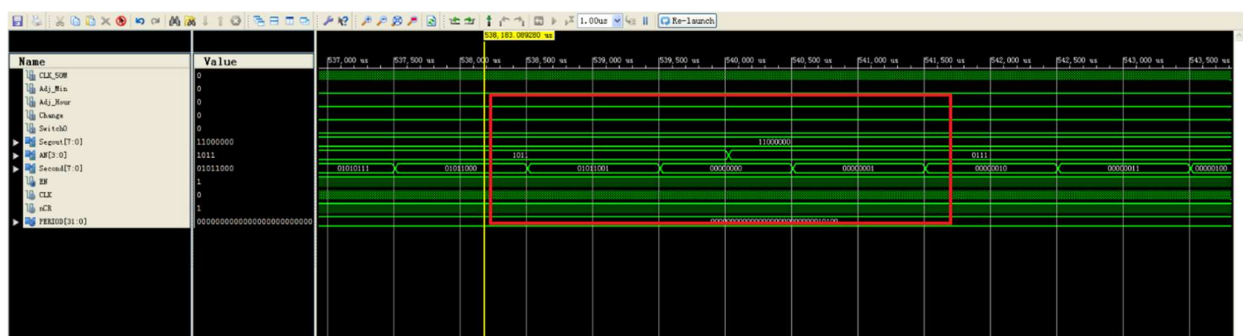
endmodule

```

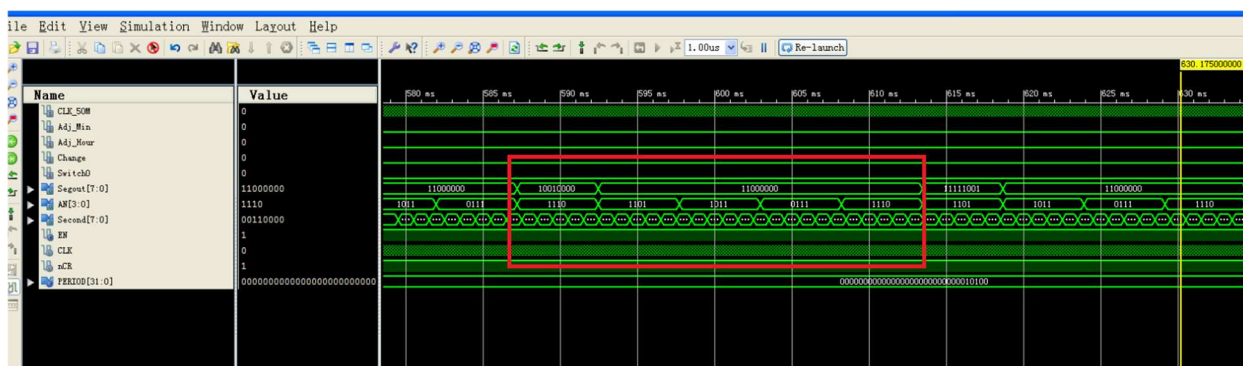
整体仿真时，我将时钟频率只分频到 1khz



如图，当原始状态，四个数码管都为 0，段码为 0，位码逐次遍历四个数码管，使其显示为 0



当秒钟计到 60，重新计数，分钟加 1。



此时第一个段码为 1，其余段码显示 0，如图所示。

其余小时，分钟进制也同理

## 五、 反思及总结

通过本次实验，我第一次接触了 Verilog HDL 语言进行大规模电路设计和开发。在任务指导下，我复习了基础的语法知识，包括门级、数据流、行为级三种描述方式，同时也学习了仿真文件的编写以及 ISE 开发平台的基础操作。最令我兴奋的是将设计的程序下载到 FPGA 板子上进行实际应用，这种实践感受非常神奇。

