|  |  |  |
| --- | --- | --- |
| 《数字逻辑与数字系统》实验报告 |  | 天津大学本科生实验报告专用纸 |
| 学院 智算 年级 2019级 班级 软工一班 姓名 俞林昊 学号 3019207450  课程名称 数字逻辑和数字系统 实验日期 2021.5.20 成绩  同组实验者 | **三．实验原理与步骤（注：步骤不用写工具的操作步骤，而是设计步骤）**  1**. 画出分秒数字钟电路的原理图（模块级别即可，如使能时钟模块、 边沿检测模块等）** 。    **2. 分秒数字钟电路中一共使用了几个计数器，作用分别是什么？**  一共有三个计数器。在计时电路中的计数器根据系统时钟来输出分秒信号。系统时钟周期为40ns，每个周期计数器加1，所以每增加一秒，计数器都要记到25000000。在使能时钟生成器中的计数器根据系统时钟进行分频操作。系统钟周期为40ns，输出周期为1ms，所以每有一个上升沿，计数器都要记到25000。在七段数码管动态扫面显示模块的两位计数器每过1ms（每个上升沿）加1，轮流产生四个不同的数，每个数对应一个数码管，代表此时只有该数码管有效。  **3. 给出分秒数字钟的 SystemVerilog 代码。**  **Clk\_en**  module clk\_en #(parameter N = 25000) (  input clk,  input reset,  output logic clkAfterEn  );  logic [31 : 0] r\_cnt;  always\_ff@(posedge clk) begin  if(!reset) r\_cnt <= 0;  **Timer**  module timer #(parameter MAX\_SEC = 59, MAX\_MIN = 59, MAX\_COUNT = 25000000) (  input clk,  input reset,  input en,  output logic [7:0] min,  output logic [7:0] sec  );    logic [31 : 0] r\_cnt;  logic state;    always\_ff@(posedge clk) begin  if(!reset) state = 1'b0;  else if(en) state = ~state;  end  always\_ff@(posedge clk) begin  // reset  if(!reset) begin  min = 8'd0;  sec = 8'd0;  end  // timing  else if(state) begin  if (r\_cnt < MAX\_COUNT - 1) r\_cnt <= r\_cnt + 1;  else if(r\_cnt == MAX\_COUNT - 1) begin  r\_cnt <= 0;  if(sec == MAX\_SEC) begin  sec <= 0;  min <= min + 1;  end  else if(sec < MAX\_SEC) sec <= sec + 1;  else if(min == MAX\_MIN) min <= 0;  end  end  else if(!state) begin  min <= min;  sec <= sec;  **analyze2a\_to\_g**  module analyze2a\_to\_g(  input [3 : 0] D,  output logic [7 : 0] a\_to\_g  );  always@(\*) begin  unique case(D)  4'd0: a\_to\_g = 8'b11000000;  4'd1: a\_to\_g = 8'b11111001;  4'd2: a\_to\_g = 8'b10100100;  4'd3: a\_to\_g = 8'b10110000;  4'd4: a\_to\_g = 8'b10011001;  4'd5: a\_to\_g = 8'b10010010;  4'd6: a\_to\_g = 8'b10000010;  4'd7: a\_to\_g = 8'b11111000;  4'd8: a\_to\_g = 8'b10000000;  4'd9: a\_to\_g = 8'b10010000;  endcase  end    endmodule  **dig\_clock**  `timescale 1ns / 1ps  module dig\_clock(  input sys\_clk,  input sys\_rst\_n,  input i\_start,  output logic [3 : 0] an,  output logic [7 : 0] a\_to\_g  );    logic clkAfterEn;  logic work;  logic [7 : 0] min, sec, bcd\_min, bcd\_sec;  logic [3 : 0] decimalTime;  clk\_en U0(  .clk (sys\_clk),  .reset (sys\_rst\_n),  **四．仿真与实验结果（注：仿真需要给出波形图截图，截图要清晰，如果波形过长，可以分段截取；实验结果为远程FPGA硬件云平台的截图）**  注：远程FPGA硬件云平台截图只需要一个测试激励即可      **五．实验中遇到的问题和解决办法**  实验中，对于多个模块进行分层设计。在书写代码的时候，不能在定义的时候就把初始值赋给变量。  在设计的时候，在动态扫描之后，又加上了一个将动态扫描的结果作为analyze2a\_to\_g这个模块的输入。这是因为发现scan模块输出的是一个十进制数字，要将其转化为一个能够点亮七段数码管的二进制数字。  具体的实验模块之间的逻辑如原理图所示。先是使能时钟将系统的时间分频，边沿检测电路来检测出时钟的上升沿，之后这两个信号作为计时器模块的输入。计时器模块的输出作为两个bin\_bcd模块的输入，输出十进制的时间。    **教师签字：**  **年 月 日** |
| 实验项目名称 分秒数字钟的设计和实现   1. **实验目的**  |  | | --- | | 1. 掌握基于SystemVerilog HDL 的时序逻辑电路建模方法；  2. 掌握计数器设计方法，并能够使用计数器设计使能时钟（用于时钟分频）；  3. 掌握移位寄存器设计方法，并能够利用移位寄存器设计边沿检测电路；  4. 掌握7 段数码管的动态显示。 |  1. **实验内容**           else if(r\_cnt == N-1) r\_cnt <= 0;  else r\_cnt <= r\_cnt + 1;  end  always\_ff@(posedge clk) begin  if(!reset) clkAfterEn <= 1'b0;  else if(r\_cnt == N-1) clkAfterEn <= 1'b1;  else clkAfterEn <= 1'b0;  end    endmodule  **Edge\_detection**  module Edge\_detection (  input clk,  input reset,  input i\_btn,  output logic pos\_edge  );  logic q1;  logic q2;  always\_ff@(posedge clk)  if(reset) begin  q1 <= i\_btn;  q2 <= q1;  end  else if(!reset) begin  q1 <= 0;  q2 <= 0;  end  assign start\_flag = (~q2) & q1;  endmodule  end  end  endmodule  **Scan**  module scan(  input clk,  input reset,  input en,  input [7 : 0] min,  input [7 : 0] sec,  output logic [3 : 0] decimalTime,  output logic [3 : 0] an  );  always\_ff@(posedge clk)  if(!reset)  an <= 4'd1;  else if(reset && en)  an <= {an[2 : 0], an[3]};    always@(\*) begin  if(an == 1) decimalTime = sec[3 : 0];  else if(an == 2)  decimalTime = sec[7 : 4];  else if(an == 4)  decimalTime = min[3 : 0];  else if(an == 8)  decimalTime = min[7 : 4];  end  endmodule  .clkAfterEn (clkAfterEn)  );  Edge\_detection U1(  .clk (sys\_clk),  .reset (sys\_rst\_n),  .i\_btn (i\_start),  .pos\_edge (work)  );  timer U2(  .clk (sys\_clk),  .reset (sys\_rst\_n),  .en (work),  .min (min),  .sec (sec)  );  bin2bcd\_0 U3(  .bin (min),  .bcd (bcd\_min)  );  bin2bcd\_0 U4(  .bin (sec),  .bcd (bcd\_sec)  );  scan U5(  .clk (sys\_clk),  .reset (sys\_rst\_n),  .en (clkAfterEn),  .min (bcd\_min),  .sec (bcd\_sec),  .decimalTime (decimalTime),  .an (an)  );  analyze2a\_to\_g U6(  .D (decimalTime),  .a\_to\_g (a\_to\_g)  );  endmodule |