**《数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | *李佳明* | | **年级** | | *2020* |
| **学号** | | *20204254* | | **专业、班级** | | *计算机科学与技术（卓越）* |
| **实验名称** | **实验七 数字钟设计** | | | | | |
| **实验时间** | **2021.11.18** | | **实验地点** | | **DS1410** | |
| **实验成绩** |  | | **实验性质** | | **□验证性 □设计性 □综合性** | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  评语：  评价教师签名（电子签名）： | | | | | | |
| 一、实验目的  通过实验，充分理解和掌握数字钟的原理，理解多进制计时器和数码管的工作原理，学会设计自己的数字钟。 | | | | | | |
| 二、实验项目内容   1. 实现一个六十进制数字时钟，秒到60则归零重加，同时让分加1，分加到60归零重加，并让小时加1，小时加到24归零重加。 2. 设计用于数字钟的计数器（60进制，24进制）并编写仿真文件，观察其波形图并验证其正确性。 3. 将设计好的计数器连接数码管并显示数值，数码管1，0显示秒值，数码管2，3显示分值，数码管4，5显示小时（可以十六进制显示在led灯上）。 4. 综合、实验、生成bit流，下载到开发板进行验证。 | | | | | | |
| 三、实验设计  **设计思路**  首先将整个实验分成三大部分，一是分频模块，二是计数模块，三是显示模块。  分频：  因为代码中的时钟信号clk频率过高，与数字时钟的要求不匹配。在代码中采用设置parameter分频获得clk2，计数时以clk2为时钟信号进行计算。  计数：  计数模块钟分别设计一个60进制和24进制的计数器，用来计算时分秒及其进位。为方便显示模块使用，  将秒拆分为Second\_2,Second\_1；  将分拆分为Minute\_2,Minute\_1；  将时拆分为Hour\_2,Hour\_1；  则Second\_1,Minute\_1为十进制计数器，Second\_2,Minute\_2为六进制计数器，Hour\_2为3进制计数器，Hour\_1为10进制计数器（最后为5进制计数器，特殊情况：23:59:59）  确定各位进位以及归零情况，见代码。  仿真波形图如下：    显示：  使用七段数码管进行显示。  其中低电平为有效。为了实现4个数码管显示不同内容，利用人眼的视觉暂留，对4个气短数码管进行分时复用。当数码管的刷新频率高于人眼的视觉暂留时，人眼就无法察觉数码管信息的改变。  reg[10:0] display  使用11位的display对数码管进行输出，前4位为使能端，后七位与七段数码管相对应。 | | | | | | |
| 四、实验过程或算法  **设计总文件clock.v**  `timescale 1ps / 1ps  *//////////////////////////////////////////////////////////////////////////////////*  *// Company:*  *// Engineer:*  *//*  *// Create Date: 2021/11/14 00:22:39*  *// Design Name:*  *// Module Name: clock*  *// Project Name:*  *// Target Devices:*  *// Tool Versions:*  *// Description:*  *//*  *// Dependencies:*  *//*  *// Revision:*  *// Revision 0.01 - File Created*  *// Additional Comments:*  *//*  *//////////////////////////////////////////////////////////////////////////////////*  module clock(      input clk,rst,second\_select,      output reg[10:0] display      );      reg[3:0] Hour\_2,Hour\_1,Minute\_2,Minute\_1,Second\_2,Second\_1;      reg[26:0] count = 0;      reg[15:0] count2 = 0;      reg[2:0] sel = 0;      parameter T = 10000000;      parameter T2 = 50000;      reg clk2 = 1'b0;  *//时间*      always @(posedge clk) begin          if(rst)              count<='b0;          else if(count == T)          begin              clk2 <= ~clk2;              count <= 0;          end          else count <= count + 'b1;      end  *//Second\_1 % 10*      always @(posedge clk2) begin  *//rst*          if(rst)              Second\_1 <= 4'b0000;  *//进位*          else if(Second\_1==4'b1001)              Second\_1 <= 4'b0000;          else              Second\_1 <= Second\_1 + 4'b0001;      end  *//Second\_2 % 6*      always @(posedge clk2) begin  *//rst*          if(rst)              Second\_2 <= 4'b0000;  *//进位*          else if(Second\_1==4'b1001)*//Second\_1进位*              begin                  if(Second\_2 == 4'b0101)*//Second\_2==5*                      Second\_2<=4'b0000;                  else                      Second\_2<=Second\_2+4'b0001;              end      end  *//Minute\_1 % 10*      always @(posedge clk2) begin          if(rst)              Minute\_1 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001)              begin                  if(Minute\_1==4'b1001)                      Minute\_1 <= 4'b0000;                  else                      Minute\_1 <= Minute\_1+4'b0001;              end      end  *//Minute\_2 % 6*      always @(posedge clk2) begin          if(rst)              Minute\_2 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001)              begin                  if(Minute\_1==4'b1001)                  begin                      if(Minute\_2==4'b0101)                          Minute\_2<=4'b0000;                      else                          Minute\_2 <= Minute\_2+4'b0001;                  end              end      end  *//Hour\_1 % 10 最后为4*      always @(posedge clk2) begin          if(rst)              Hour\_1 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001&&Minute\_2==4'b0101&&Minute\_1==4'b1001)              begin                  if(Hour\_2==4'b0010&&Hour\_1==4'b0011)*//23:59:59*                      Hour\_1 <= 4'b0000;                  else if(Hour\_1 == 4'b1001)                      Hour\_1 <= 4'b0000;                  else                      Hour\_1 <= Hour\_1 + 4'b0001;              end      end  *//Hour\_2 % 3*      always @(posedge clk2) begin          if(rst)              Hour\_2 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001&&Minute\_2==4'b0101&&Minute\_1==4'b1001)              begin                  if(Hour\_2==4'b0010&&Hour\_1==4'b0011)*//23:59:59*                      Hour\_2 <= 4'b0000;                  else if(Hour\_1 == 4'b1001)                      Hour\_2 <= Hour\_2+4'b0001;              end      end  *//显示*      always @(posedge clk) begin          if(rst)              count2<='b0;          else if(count2==T2)              begin                  count2<=0;                  sel<=sel+1;                  if(sel==4)                      sel<=0;              end          else              count2 <=count2+1;      end      always @(posedge clk) begin  *//显示秒*          if(second\_select)              begin                  case(sel)  *//Second\_1 0~9*                  0,1: begin                      case(Second\_1)                          4'b0000:display = 11'b1110\_0000001;                          4'b0001:display = 11'b1110\_1001111;                          4'b0010:display = 11'b1110\_0010010;                          4'b0011:display = 11'b1110\_0000110;                          4'b0100:display = 11'b1110\_1001100;                          4'b0101:display = 11'b1110\_0100100;                          4'b0110:display = 11'b1110\_0100000;                          4'b0111:display = 11'b1110\_0001111;                          4'b1000:display = 11'b1110\_0000000;                          4'b1001:display = 11'b1110\_0000100;                      endcase                  end  *//Second\_2 0~5*                  2,3:begin                      case(Second\_2)                          4'b0000:display = 11'b1101\_0000001;                          4'b0001:display = 11'b1101\_1001111;                          4'b0010:display = 11'b1101\_0010010;                          4'b0011:display = 11'b1101\_0000110;                          4'b0100:display = 11'b1101\_1001100;                          4'b0101:display = 11'b1101\_0100100;                      endcase                  end                  endcase              end  *//显示小时和分钟*          else              begin                  case(sel)  *//Minute\_1 0~9*                  0:begin                      case(Minute\_1)                          4'b0000:display = 11'b1110\_0000001;                          4'b0001:display = 11'b1110\_1001111;                          4'b0010:display = 11'b1110\_0010010;                          4'b0011:display = 11'b1110\_0000110;                          4'b0100:display = 11'b1110\_1001100;                          4'b0101:display = 11'b1110\_0100100;                          4'b0110:display = 11'b1110\_0100000;                          4'b0111:display = 11'b1110\_0001111;                          4'b1000:display = 11'b1110\_0000000;                          4'b1001:display = 11'b1110\_0000100;                      endcase                  end  *//Minute\_2 0~5*                  1:begin                      case(Minute\_2)                          4'b0000:display = 11'b1101\_0000001;                          4'b0001:display = 11'b1101\_1001111;                          4'b0010:display = 11'b1101\_0010010;                          4'b0011:display = 11'b1101\_0000110;                          4'b0100:display = 11'b1101\_1001100;                          4'b0101:display = 11'b1101\_0100100;                      endcase                  end  *//Hour\_1 0~9*                  2:begin                      case(Hour\_1)                          4'b0000:display = 11'b1011\_0000001;                          4'b0001:display = 11'b1011\_1001111;                          4'b0010:display = 11'b1011\_0010010;                          4'b0011:display = 11'b1011\_0000110;                          4'b0100:display = 11'b1011\_1001100;                          4'b0101:display = 11'b1011\_0100100;                          4'b0110:display = 11'b1011\_0100000;                          4'b0111:display = 11'b1011\_0001111;                          4'b1000:display = 11'b1011\_0000000;                          4'b1001:display = 11'b1011\_0000100;                      endcase                  end  *//Hour\_2*                  3:begin                      case(Hour\_2)                          4'b0000:display = 11'b0111\_0000001;                          4'b0001:display = 11'b0111\_1001111;                          4'b0010:display = 11'b0111\_0010010;                      endcase                  end                  endcase              end      end  endmodule  **计数模块设计文件clock\_count.v**  `timescale 1ns / 1ps  *//////////////////////////////////////////////////////////////////////////////////*  *// Company:*  *// Engineer:*  *//*  *// Create Date: 2021/11/18 18:36:58*  *// Design Name:*  *// Module Name: clock\_conut*  *// Project Name:*  *// Target Devices:*  *// Tool Versions:*  *// Description:*  *//*  *// Dependencies:*  *//*  *// Revision:*  *// Revision 0.01 - File Created*  *// Additional Comments:*  *//*  *//////////////////////////////////////////////////////////////////////////////////*  module clock\_conut(      input clk,rst,      output reg[3:0] Hour\_2,Hour\_1,Minute\_2,Minute\_1,Second\_2,Second\_1      );  *//Second\_1 % 10*      always @(posedge clk) begin  *//rst*          if(rst)              Second\_1 <= 4'b0000;  *//进位*          else if(Second\_1==4'b1001)              Second\_1 <= 4'b0000;          else              Second\_1 <= Second\_1 + 4'b0001;      end  *//Second\_2 % 6*      always @(posedge clk) begin  *//rst*          if(rst)              Second\_2 <= 4'b0000;  *//进位*          else if(Second\_1==4'b1001)*//Second\_1进位*              begin                  if(Second\_2 == 4'b0101)*//Second\_2==5*                      Second\_2<=4'b0000;                  else                      Second\_2<=Second\_2+4'b0001;              end      end  *//Minute\_1 % 10*      always @(posedge clk) begin          if(rst)              Minute\_1 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001)              begin                  if(Minute\_1==4'b1001)                      Minute\_1 <= 4'b0000;                  else                      Minute\_1 <= Minute\_1+4'b0001;              end      end  *//Minute\_2 % 6*      always @(posedge clk) begin          if(rst)              Minute\_2 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001)              begin                  if(Minute\_1==4'b1001)                  begin                      if(Minute\_2==4'b0101)                          Minute\_2<=4'b0000;                      else                          Minute\_2 <= Minute\_2+4'b0001;                  end              end      end  *//Hour\_1 % 10 最后为4*      always @(posedge clk) begin          if(rst)              Hour\_1 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001&&Minute\_2==4'b0101&&Minute\_1==4'b1001)              begin                  if(Hour\_2==4'b0010&&Hour\_1==4'b0011)*//23:59:59*                      Hour\_1 <= 4'b0000;                  else if(Hour\_1 == 4'b1001)                      Hour\_1 <= 4'b0000;                  else                      Hour\_1 <= Hour\_1 + 4'b0001;              end      end  *//Hour\_2 % 3*      always @(posedge clk) begin          if(rst)              Hour\_2 <= 4'b0000;          else if(Second\_2 == 4'b0101&&Second\_1==4'b1001&&Minute\_2==4'b0101&&Minute\_1==4'b1001)              begin                  if(Hour\_2==4'b0010&&Hour\_1==4'b0011)*//23:59:59*                      Hour\_2 <= 4'b0000;                  else if(Hour\_1 == 4'b1001)                      Hour\_2 <= Hour\_2+4'b0001;              end      end  endmodule  **仿真文件clock\_tb.v**  `timescale 1ns / 1ps  *//////////////////////////////////////////////////////////////////////////////////*  *// Company:*  *// Engineer:*  *//*  *// Create Date: 2021/11/14 00:48:21*  *// Design Name:*  *// Module Name: clock\_tb*  *// Project Name:*  *// Target Devices:*  *// Tool Versions:*  *// Description:*  *//*  *// Dependencies:*  *//*  *// Revision:*  *// Revision 0.01 - File Created*  *// Additional Comments:*  *//*  *//////////////////////////////////////////////////////////////////////////////////*  module clock\_tb();      reg clk,rst;      wire[3:0] h2,h1,m2,m1,s2,s1;      clock\_conut u1(clk,rst,h2,h1,m2,m1,s2,s1);      always #5 clk=~clk;      initial      begin          clk=0;          rst=1;          #10 rst=0;      end  endmodule  **约束文件clock\_xdc.xdc**  set\_property IOSTANDARD LVCMOS33 [get\_ports clk]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[8]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[9]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {display[10]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports rst]  set\_property IOSTANDARD LVCMOS33 [get\_ports second\_select]  set\_property PACKAGE\_PIN W5 [get\_ports clk]  create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports clk]  set\_property PACKAGE\_PIN R2 [get\_ports second\_select]  set\_property PACKAGE\_PIN U18 [get\_ports rst]  set\_property PACKAGE\_PIN W4 [get\_ports {display[10]}]  set\_property PACKAGE\_PIN V4 [get\_ports {display[9]}]  set\_property PACKAGE\_PIN U4 [get\_ports {display[8]}]  set\_property PACKAGE\_PIN U2 [get\_ports {display[7]}]  set\_property PACKAGE\_PIN W7 [get\_ports {display[6]}]  set\_property PACKAGE\_PIN W6 [get\_ports {display[5]}]  set\_property PACKAGE\_PIN U8 [get\_ports {display[4]}]  set\_property PACKAGE\_PIN V8 [get\_ports {display[3]}]  set\_property PACKAGE\_PIN U5 [get\_ports {display[2]}]  set\_property PACKAGE\_PIN V5 [get\_ports {display[1]}]  set\_property PACKAGE\_PIN U7 [get\_ports {display[0]}] | | | | | | |
| 五、实验过程中遇到的问题及解决情况(主要问题及解决情况) | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  因timescale中的设置，使得clk过快，故要采用分频器，使得时间变化尽量与现实生活中的一秒接近（为了使上板测试效果明显，时间的变化还是快于一秒）。  只有4个七位数码管，但要显示小时分钟秒需要六个数码管，故引进一个开关second\_select，使得选择显示小时分钟或者秒，方便了上板显示。  将秒拆成Second\_1和Second\_2编写两个计数器，共同作用，从而完成秒的计数。小时和分钟同理。  七、小组分工情况说明  *李佳明：Verilog代码文件的编写，仿真测试，上板测试，撰写实验报告* | | | | | | |