**《数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | *孙莹莹* | | **年级** | | **2020级** |
| **学号** | | **20204104** | | **专业、班级** | | *计算机科学与技术卓越一班* |
| **实验名称** | **实验八 乘除法器的实现** | | | | | |
| **实验时间** | **2021.11.15** | | **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  设计实现一个8位的乘法器和除法器，并进行仿真验证和下板验证。 | | | | | | |
| 二、实验项目内容  1.乘法器  **（1）乘法器设计**  **（2）乘法器代码实现**  **（3）乘法器仿真**  **（4）乘法器下板验证**  2.除法器  **（1）除法器设计**  **（2）除法器代码实现**  **（3）除法器仿真**  **（4）除法器下板验证** | | | | | | |
| 1. 实验设计（实验原理、真值表、原理图等）   **1.乘法器**  （1）实验原理  两个8位整数相乘，乘积不会超过16位。假设乘数为x，被乘数为y，乘积为z，两个8位整数相乘，实际上进行了8次加法操作，乘法实现过程如下所示。每次判断y的0位是否为1，若为1则z加上x；反之若y的0位不为1，则不执行操作。每次判断后x左移一位，y右移一位。重复以上步骤8次，完成乘法操作。  （2）乘法器原理图    **2.除法器**  （1）实验原理  假设被除数为a，除数为b，均是8位整数，则它们相除的商和余数一不会超过8位。除法器其实是通过减法和移位操作实现除法功能的，实现过程如下。  首先将a转换为16位数tempa，tempa高8位为0，低8位为a；将b转换为tempb，tempb高8位为b，低8位为0。每个周期开始之前将tempa左移一位，然后跟tempb比较，若大于tempb，则tempa赋值为tempa-tempb+1，否则继续往下执行。重复以上比较执行的步骤8次，完成除法操作。最后tempa高8位是余数，低8位是商。  （2）除法器原理图 | | | | | | |
| 四、实验过程或算法  1.乘法器  （1）乘法器核心代码  module mutl\_machine  #(parameter SIZE = 8)  (a,b,out);  input wire [SIZE-1:0] a;  input wire[SIZE-1:0] b;  output reg [2\*SIZE-1:0] out;  reg [2\*SIZE-1:0] tempa;  reg [SIZE-1:0] tempb;    always@(\*)  begin  tempa = a;  tempb = b;  out = 16'b0000\_0000\_0000\_0000;  repeat(SIZE)  begin  if(tempb[0])  out = out + tempa;  tempa = {tempa[14:0],1'b0};  tempb = {1'b0, tempb[7:1]};  end  end  endmodule  （2）七段译码管控制代码  module show(  input clk,  input [15:0] data,  output [3:0] sm\_wei,  output [7:0] sm\_duan  );  //分频  integer clk\_cnt;  reg clk\_400Hz;  always@(posedge clk)  begin  if(clk\_cnt==32'd100000)  begin  clk\_cnt<=1'b0;  clk\_400Hz<=~clk\_400Hz;  end  else begin  clk\_cnt<=clk\_cnt+1'b1;  end  end  //位控制  reg [3:0]wei\_ctrl=4'b1110; //共阳极  always@(posedge clk\_400Hz)  begin  wei\_ctrl<={wei\_ctrl[2:0],wei\_ctrl[3]};  end  //段控制  reg[3:0] duan\_ctrl;  always@(wei\_ctrl)begin  case(wei\_ctrl)  4'b1110:duan\_ctrl=data[3:0];  4'b1101:duan\_ctrl=data[7:4];  4'b1011:duan\_ctrl=data[11:8];  4'b0111:duan\_ctrl=data[15:12];  default:duan\_ctrl=4'hf;  endcase  end  //解码模块  reg[7:0] duan;  always@(duan\_ctrl)  begin  case(duan\_ctrl)  4'h0:duan=8'b1100\_0000; //h-a  4'h1:duan=8'b1111\_1001;  4'h2:duan=8'b1010\_0100;  4'h3:duan=8'b1011\_0000;  4'h4:duan=8'b1001\_1001;  4'h5:duan=8'b1001\_0010;  4'h6:duan=8'b1000\_0010;  4'h7:duan=8'b1111\_1000;  4'h8:duan=8'b1000\_0000;  4'h9:duan=8'b1001\_0000;  4'ha:duan=8'b1000\_1000;  4'hb:duan=8'b1000\_0011;  4'hc:duan=8'b1100\_0110;  4'hd:duan=8'b1010\_0001;  4'he:duan=8'b1000\_0110;  4'hf:duan=8'b1000\_1110;  default:duan=8'b1100\_0000;  endcase  end  assign sm\_wei = wei\_ctrl;  assign sm\_duan = duan;  endmodule  （3）顶层模块调用乘法器和七段译码管  module multitude(  input clk,  input wire [7:0] a,  input wire [7:0] b,  output [15:0] out,  output [3:0] sm\_wei,  output [7:0] sm\_duan  );    mutl\_machine tryMachine(a,b,out);  show tryShow(clk,out,sm\_wei,sm\_duan);  endmodule  （4）管脚约束文件  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[15]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[14]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[13]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[12]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[11]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[10]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[9]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[8]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {out[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_duan[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_wei[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_wei[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_wei[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {sm\_wei[0]}]  set\_property PACKAGE\_PIN W7 [get\_ports {sm\_duan[0]}]  set\_property PACKAGE\_PIN W6 [get\_ports {sm\_duan[1]}]  set\_property PACKAGE\_PIN U8 [get\_ports {sm\_duan[2]}]  set\_property PACKAGE\_PIN V8 [get\_ports {sm\_duan[3]}]  set\_property PACKAGE\_PIN U5 [get\_ports {sm\_duan[4]}]  set\_property PACKAGE\_PIN V5 [get\_ports {sm\_duan[5]}]  set\_property PACKAGE\_PIN U7 [get\_ports {sm\_duan[6]}]  set\_property PACKAGE\_PIN V7 [get\_ports {sm\_duan[7]}]  set\_property PACKAGE\_PIN U2 [get\_ports {sm\_wei[0]}]  set\_property PACKAGE\_PIN U4 [get\_ports {sm\_wei[1]}]  set\_property PACKAGE\_PIN V4 [get\_ports {sm\_wei[2]}]  set\_property PACKAGE\_PIN W4 [get\_ports {sm\_wei[3]}]  set\_property PACKAGE\_PIN R2 [get\_ports {a[0]}]  set\_property PACKAGE\_PIN T1 [get\_ports {a[1]}]  set\_property PACKAGE\_PIN U1 [get\_ports {a[2]}]  set\_property PACKAGE\_PIN W2 [get\_ports {a[3]}]  set\_property PACKAGE\_PIN R3 [get\_ports {a[4]}]  set\_property PACKAGE\_PIN V2 [get\_ports {a[7]}]  set\_property PACKAGE\_PIN T3 [get\_ports {a[6]}]  set\_property PACKAGE\_PIN T2 [get\_ports {a[5]}]  set\_property PACKAGE\_PIN W13 [get\_ports {b[0]}]  set\_property PACKAGE\_PIN W14 [get\_ports {b[1]}]  set\_property PACKAGE\_PIN V15 [get\_ports {b[2]}]  set\_property PACKAGE\_PIN W15 [get\_ports {b[3]}]  set\_property PACKAGE\_PIN W17 [get\_ports {b[4]}]  set\_property PACKAGE\_PIN W16 [get\_ports {b[5]}]  set\_property PACKAGE\_PIN V16 [get\_ports {b[6]}]  set\_property PACKAGE\_PIN V17 [get\_ports {b[7]}]  set\_property PACKAGE\_PIN L1 [get\_ports {out[0]}]  set\_property PACKAGE\_PIN P1 [get\_ports {out[1]}]  set\_property PACKAGE\_PIN N3 [get\_ports {out[2]}]  set\_property PACKAGE\_PIN P3 [get\_ports {out[3]}]  set\_property PACKAGE\_PIN U3 [get\_ports {out[4]}]  set\_property PACKAGE\_PIN W3 [get\_ports {out[5]}]  set\_property PACKAGE\_PIN V3 [get\_ports {out[6]}]  set\_property PACKAGE\_PIN V13 [get\_ports {out[7]}]  set\_property PACKAGE\_PIN V14 [get\_ports {out[8]}]  set\_property PACKAGE\_PIN U14 [get\_ports {out[9]}]  set\_property PACKAGE\_PIN U15 [get\_ports {out[10]}]  set\_property PACKAGE\_PIN W18 [get\_ports {out[11]}]  set\_property PACKAGE\_PIN V19 [get\_ports {out[12]}]  set\_property PACKAGE\_PIN U19 [get\_ports {out[13]}]  set\_property PACKAGE\_PIN E19 [get\_ports {out[14]}]  set\_property PACKAGE\_PIN U16 [get\_ports {out[15]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports clk]  set\_property PACKAGE\_PIN W5 [get\_ports clk]   1. 除法器   （1）除法器代码  module devide #(parameter size=8)(  input [size-1:0]a,  input [size-1:0]b,  output [size-1:0]consult,  output [size-1:0]remainder  );  reg [2\*size-1:0] a\_16;  reg [2\*size-1:0] b\_16;  always@(\*)begin  if(b==0)begin  a\_16={16'b1111\_1111\_1111\_1111};  end  else begin  a\_16={8'b0000\_0000,a};  b\_16={b,8'b0000\_0000};    repeat(size)begin  a\_16=a\_16<<1;  if(a\_16>=b\_16)begin  a\_16=a\_16-b\_16+1'b1;  end  else begin  a\_16=a\_16;  end  end  end  end  assign consult=a\_16[size-1:0];  assign remainder=a\_16[size\*2-1:size];  endmodule  （2）七段译码管控制代码  module led#(parameter size=8)(  input clk,  input [size-1:0]consult,  input [size-1:0]remainder,  output[3:0]wei\_show,  output[7:0]duan\_show  );  //分频  integer cnt;  reg clk\_400hz;  always@(posedge clk)begin  if(cnt==32'd100000)begin  cnt<=0;  clk\_400hz=~clk\_400hz;  end  else begin  cnt<=cnt+1'b1;  end  end    reg[3:0]wei\_ctrl=4'b1110;  always@(posedge clk\_400hz)begin  wei\_ctrl<={wei\_ctrl[2:0],wei\_ctrl[3]};  end    reg [3:0]duan\_ctrl;  always@(wei\_ctrl)begin  case(wei\_ctrl)  4'b1110: duan\_ctrl = consult[3:0];  4'b1101: duan\_ctrl = consult[7:4];  4'b1011: duan\_ctrl = remainder[3:0];  4'b0111: duan\_ctrl = remainder[7:4];  default: duan\_ctrl = 4'hf;  endcase  end    reg[7:0]duan;  always@(duan\_ctrl)begin  case(duan\_ctrl)  4'h0:duan=8'b1100\_0000; //h-a  4'h1:duan=8'b1111\_1001;  4'h2:duan=8'b1010\_0100;  4'h3:duan=8'b1011\_0000;  4'h4:duan=8'b1001\_1001;  4'h5:duan=8'b1001\_0010;  4'h6:duan=8'b1000\_0010;  4'h7:duan=8'b1111\_1000;  4'h8:duan=8'b1000\_0000;  4'h9:duan=8'b1001\_0000;  4'ha:duan=8'b1000\_1000;  4'hb:duan=8'b1000\_0011;  4'hc:duan=8'b1100\_0110;  4'hd:duan=8'b1010\_0001;  4'he:duan=8'b1000\_0110;  4'hf:duan=8'b1000\_1110;  default:duan=8'b1100\_0000;  endcase  end    assign wei\_show = wei\_ctrl;  assign duan\_show = duan;  Endmodule  （3）顶层模块调用除法器和七段译码管  module main #(parameter size=8)(  input clk,  input [size-1:0]a,  input [size-1:0]b,  output [size-1:0]consult,  output [size-1:0]remainder,  output [3:0]wei\_show,  output[7:0]duan\_show  );  devide demo1(.a(a),.b(b),.consult(consult),.remainder(remainder));  led demo2(.clk(clk),.consult(consult),.remainder(remainder),.wei\_show(wei\_show),.duan\_show(duan\_show));  endmodule  （4）管脚约束文件  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {a[0]}]  set\_property PACKAGE\_PIN R2 [get\_ports {a[0]}]  set\_property PACKAGE\_PIN T1 [get\_ports {a[1]}]  set\_property PACKAGE\_PIN U1 [get\_ports {a[2]}]  set\_property PACKAGE\_PIN W2 [get\_ports {a[3]}]  set\_property PACKAGE\_PIN R3 [get\_ports {a[4]}]  set\_property PACKAGE\_PIN T2 [get\_ports {a[5]}]  set\_property PACKAGE\_PIN T3 [get\_ports {a[6]}]  set\_property PACKAGE\_PIN V2 [get\_ports {a[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {b[0]}]  set\_property PACKAGE\_PIN W13 [get\_ports {b[0]}]  set\_property PACKAGE\_PIN W14 [get\_ports {b[1]}]  set\_property PACKAGE\_PIN V15 [get\_ports {b[2]}]  set\_property PACKAGE\_PIN W15 [get\_ports {b[3]}]  set\_property PACKAGE\_PIN W17 [get\_ports {b[4]}]  set\_property PACKAGE\_PIN W16 [get\_ports {b[5]}]  set\_property PACKAGE\_PIN V16 [get\_ports {b[6]}]  set\_property PACKAGE\_PIN V17 [get\_ports {b[7]}]  set\_property PACKAGE\_PIN L1 [get\_ports {remainder[0]}]  set\_property PACKAGE\_PIN P1 [get\_ports {remainder[1]}]  set\_property PACKAGE\_PIN N3 [get\_ports {remainder[2]}]  set\_property PACKAGE\_PIN P3 [get\_ports {remainder[3]}]  set\_property PACKAGE\_PIN U3 [get\_ports {remainder[4]}]  set\_property PACKAGE\_PIN W3 [get\_ports {remainder[5]}]  set\_property PACKAGE\_PIN V3 [get\_ports {remainder[6]}]  set\_property PACKAGE\_PIN V13 [get\_ports {remainder[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {remainder[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {consult[0]}]  set\_property PACKAGE\_PIN V14 [get\_ports {consult[0]}]  set\_property PACKAGE\_PIN U14 [get\_ports {consult[1]}]  set\_property PACKAGE\_PIN U15 [get\_ports {consult[2]}]  set\_property PACKAGE\_PIN W18 [get\_ports {consult[3]}]  set\_property PACKAGE\_PIN V19 [get\_ports {consult[4]}]  set\_property PACKAGE\_PIN U19 [get\_ports {consult[5]}]  set\_property PACKAGE\_PIN E19 [get\_ports {consult[6]}]  set\_property PACKAGE\_PIN U16 [get\_ports {consult[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {duan\_show[0]}]  set\_property PACKAGE\_PIN W7 [get\_ports {duan\_show[0]}]  set\_property PACKAGE\_PIN W6 [get\_ports {duan\_show[1]}]  set\_property PACKAGE\_PIN U8 [get\_ports {duan\_show[2]}]  set\_property PACKAGE\_PIN V8 [get\_ports {duan\_show[3]}]  set\_property PACKAGE\_PIN U5 [get\_ports {duan\_show[4]}]  set\_property PACKAGE\_PIN V5 [get\_ports {duan\_show[5]}]  set\_property PACKAGE\_PIN U7 [get\_ports {duan\_show[6]}]  set\_property PACKAGE\_PIN V7 [get\_ports {duan\_show[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {wei\_show[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {wei\_show[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {wei\_show[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {wei\_show[0]}]  set\_property PACKAGE\_PIN U2 [get\_ports {wei\_show[0]}]  set\_property PACKAGE\_PIN U4 [get\_ports {wei\_show[1]}]  set\_property PACKAGE\_PIN V4 [get\_ports {wei\_show[2]}]  set\_property PACKAGE\_PIN W4 [get\_ports {wei\_show[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports clk]  set\_property PACKAGE\_PIN W5 [get\_ports clk] | | | | | | |
| 五、实验过程中遇到的问题及解决情况  1.自带的时钟电频太快，无法看清数字。因此在七段译码管中添加了分频器的代码，以便能稳定地看到数字。  2.整数型为4字节，即为32位2进制，但由于板子资源不足，因此我们将其简化为8位二进制。且由于27\*27=16384，超过四位七段数码管所能表示的范围，因此将其转化为16进制再显示。  3.除法器最初思路为简单的用两个寄存器表示结果，其中一个表示商s，另一个为余数r，设除数为x,被除数为y,如果x>=y,则用x=x-y,s++，直到x<y,此时余数r=x,商为s。但这个想法会造成资源不必要的浪费，因此经过查阅资料后改进方法。  4.在最初设想中，除0时led应为ffff，ld1-15输出为1，但在实际中却非如此，后来我将h15\_15\_15\_15改为了b1111\_1111\_1111\_1111便成功了。 | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程   1.乘法器  （1）仿真文件  module Try();  reg [7:0] a;  reg [7:0] b;  wire [15:0] out;  reg clk;    mutl\_machine tryone(a,b,out);    always  #50  clk = ~clk;    initial  begin  a = 8'b00000001;  b = 8'b00000000;  clk = 0;  end    always@(posedge clk)  begin  a = {$random%256};  b = {$random%256};  end  endmodule   1. 仿真结果      1. 电路      1. 资源利用      1. 上板测试结果 2. 输入为1\*1， 结果为1   C:/Users/LUY/AppData/Local/Temp/picturecompress_20211115110212/output_1.jpgoutput_1   1. 输入为27\*27，结果为4000（16进制）=16384   C:/Users/LUY/AppData/Local/Temp/picturecompress_20211115110407/output_1.jpgoutput_1  2.除法器  （1）仿真文件  module sim();  reg clk;  reg [7:0] a;  reg [7:0] b;  wire [7:0] consult;  wire [7:0] remainder;  main utt(.a(a),.b(b),.consult(consult),.remainder(remainder));  initial  begin  clk = 0;  a = 0;  b = 0;  end  always #0.001 clk=~clk;  always@(posedge clk)begin  a = {$random%256};  b = {$random%256};  end  endmodule  （2）仿真结果    （3）电路图    （4）资源利用    （5）上板测试   1. b=0        1. a=0,b=1      1. a=1,b=1      1. a=1,b=12      1. a=12,b=5      1. a=120,b=10     七、小组分工情况说明  孙莹莹：乘法器的设计，上板，实验报告的攥写。  刘淑文：除法器的设计，上板，实验报告的攥写。 | | | | | | |