

开始时间 2019年01月5日 星期六 18:58

状态 完成

完成于 2019年01月5日 星期六 19:06

耗时 7 分钟 59 秒

题目1

未回答

满分1.00

```
module shifter(ctr,din,clk,clr,dout);  
input din,clk,clr;//din数据输入, clk时钟信号, clr高电平异步复位  
input [1:0] ctr;//移位方式控制输入  
output[7:0] dout;  
reg[7:0] dout;  
always @(posedge clk, posedge clr)  
begin  
if (clr) dout<= 8'b0; //异步清0, 高电平有效  
else case (ctr)  
2'b00 : dout<={dout,din};//左移  
2'b01 : _____ (1) _____ //右移, din存入dout最高位  
2'b10 : dout<={dout,dout[7]};//循环左移1位  
2'b11 : _____ (2) _____ ; //循环右移1位  
default:dout<=8'bxxxxxxxx;  
endcase  
end  
endmodule
```

(1)dout<={din,dout[6:0]};(2)dout<={dout[0],dout[7:1]}

题目2

未回答

满分1.00

阻塞赋值和非阻塞赋值。若当前 $q_0=1$; $q_1=0$; $q_2=1$; $q_3=0$; $din=1$.则时钟上升沿后。

```
module block(q0,q1,q2,q3,din,clk);  
input clk,din; output reg q0,q1,q2,q3;  
always@(posedge clk)  
begin q0=din; q1=q0; q2=q1; q3=q2; end  
endmodule
```

则时钟上升沿后 , $\{q_0,q_1,q_2,q_3\}=4'b$ ____ (1) ____

```
module block(q0,q1,q2,q3,din,clk);  
input clk,din; output reg q0,q1,q2,q3;  
always@(posedge clk)  
begin q0<=din; q1<=q0; q2<=q1; q3<=q2; end  
endmodule
```

则时钟上升沿后 , $\{q_0,q_1,q_2,q_3\}=4'b$ ____ (2) ____

(1) 1111

(2) 1101

题目3

未回答

满分1.00

```
module alutask ( opcode,a,b,c ) ;
input[1:0] opcode ; input[7:0] a,b; output reg[7:0] c;
always@(a,b,opcode)
case(opcode)
  2'b00: c=Andab ( ina , inb ) ; //调用函数，实现ina、inb相与，结果赋值给c
  2'b01: _____ ( 1 ) _____ ; //任务调用，实现ina、inb相或，结果赋值给c
  2'b10: _____ ( 2 ) _____ ; //模块XORab中的值out传递给c，实现异或逻辑
  2'b11:c=~a&b;
  default:c=9'bx;
endcase
_____ ( 3 ) _____ ; //out变量定义
XORab x0 ( ina , inb , out ) ; //模块调用
_____ ( 4 ) _____ ; //补齐函数定义Andab

input[7:0] ina,inb;
add=ina&inb;

endfunction

task ORab;//任务定义
input[7:0] ina,inb; output[7:0] out;

out=ina|inb ;

endtask

endmodule

module XORab(ina,inb,out);//子模块定义，实现异或功能
input[7:0] ina,inb;
output[7:0] out;
assign out=ina^inb;
endmodule
```

(1) ORab(ina,inb,c) (2) c=out (3) wire [7:0] out (4) function [7:0] Andab

题目4

未回答

满分1.00

任务调用

```
module sort4(ra,rb,rc,rd,a,b,c,d);  
output[3:0] ra,rb,rc,rd; input[3:0] a,b,c,d;  
____( 1 )____; //补齐ra,rb,rc,rd定义  
____( 2 )____ //补齐va,vb,vc,vd定义  
always @ (a or b or c or d)  
begin  
{va,vb,vc,vd}=____( 3 )____ ; //将a、b、c、d的值分别赋给va,vb,vc,vd  
____( 4 )____; //va 与vc 互换。调用任务sort2，完成va和vc交换  
略 //vb 与vd 互换。va 与vb 互换。vc 与vd 互换。vb 与vc 互换。  
{ra,rb,rc,rd}={va,vb,vc,vd};  
end  
____( 5 )____; //任务sort2定义  
inout[3:0] x,y; reg[3:0] tmp;  
if(x>y)begin  
tmp=x; //x 与y 变量的内容互换，要求顺序执行，所以采用阻塞赋值方式。  
x=y;y=tmp;end  
endtask  
endmodule
```

reg[3:0] ra,rb,rc,rd (1);

reg[3:0] va,vb,vc,vd; (2)

{a,b,c,d} (3)

sort2(va,vc) (4);

task sort2 (5);

题目5

未回答

满分1.00

下面是15输入裁判电路，使用函数、任务和模块调用实现功能。

定义：函数count0，统计0（不同意）的个数sum0；任务count1统计1（同意）的个数sum1；将sum0和sum1通过译码模块showHex调用显示在hex0和hex1上；结果输出到ledr，若同意数sum1>sum0,ledr输出1，否则ledr输出0。

module test(in15,hex0,hex1,ledr); input[14:0] in15; output[6:0] hex0,hex1; output ledr; <u>(9)</u> sum0,sum1;//补齐数据类型定义 showHex s0(sum0,hex0);//显示 showHex s1(sum1,hex1);//显示 assign <u>(10)</u> ;//函数调用计算sum0 always@(in15) <u>(11)</u> ;//任务调用，计算sum1 assign ledr=(sum1>sum0)?1:0; //下面为函数定义	<u>(12)</u> ;//函数声明 input[14:0] in15; integer i; for(i=0;i<15;i=i+1) if(~in15[i]) count0=count0+1; endfunction //下面为任务定义 <u>(13)</u> ;//任务声明 input[14:0] in15; output[3:0] sum; integer i; for(i=0;i<15;i=i+1) if(in15[i]) sum=sum+1; endtask endmodule
//下面是子模块，译码显示电路 module showHex(in4,hex); input[3:0] in4; output reg[6:0] hex; //逻辑实现略 endmodule	

(9)	wire[3:0]
(10)	sum0=count0(in15)
(11)	count1(in15,sum1)
(12)	function[3:0] count0
(13)	task count1

题目6

未回答

满分1.00

下面代码为运算器逻辑代码，当opcode=2'b11的时候完成输入a，b差的绝对值的计算的功能，c的赋值可以用任务、函数、模块方式实现。完成下面代码

<pre>module cputask(opcode,a,b,c); input[1:0] opcode; input[7:0] a,b; output reg[8:0] c; reg f; //(7) 本行有错，请修改 always@(opcode,a,b) begin case(opcode) 2'b00:c=a&b; 2'b01:c=a b; 2'b10:c=a+b; 2'b11: ourtak(a,b,c);//任务调用 或 c=ourfun(a,b);//函数调用</pre>	<pre>或 c=f;//通过模块产生输出f default:c=9'bx; endcase end (8) //模块调用，模块输出f (9) //完成任务定义 input[7:0] in1,in2; output[8:0] out1; if(in1>in2) out1=in1-in2; else out1=in2-in1; endtask</pre>
<pre>_(10)_ //函数定义 input[7:0] in1,in2; reg[8:0] out1; if(in1>in2) out1=in1-in2; else out1=in2-in1; ourfun=out1; endfunction endmodule</pre>	<pre>module ourm(in1,in2,out1); //子模块定义 input[7:0] in1,in2; output reg[8:0] out1; always@(ina,inb) begin if(in1>in2) out1=in1-in2; else out1=in2-in1; end endmodule</pre>

<pre>7 wire[8:0] f;</pre>	<pre>8 outm a0(a,b,f);</pre>
<pre>9 task ourtak;</pre>	<pre>10 function[8:0] ourfun;</pre>

题目7

未回答

满分1.00

数据比较器

```
module compare(equal,a,b);
```

```
input a,b;
```

```
output equal;
```

```
assign equal=_____ ( 1 ) ;
```

```
//a 等于b 时 , equal 输出为1 ; a 不等于b 时 , equal 输出为0。
```

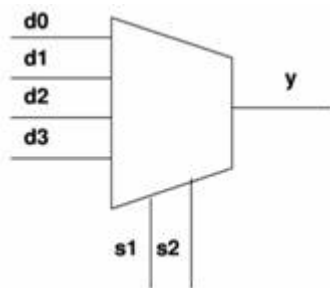
```
endmodule
```

```
(a==b)?1:0 ( 1 ) ;
```

题目8

未回答

满分1.00



利用行为描述方法设计如下四选一的数据选择题，完成程序填空。

```
module mux4_to_1(____(1)____);  
    output y;  
    reg    y;//定义数据类型  
    ____ ( 2 ) ____ d0,d1,d2,d3,s1,s2;//定义数据类型  
    always@(*);  
    begin  
        case(____ ( 3 ) ____)  
            2'b00: y=d0;  
            2'b01: y=d1;  
            2'b10: y=d2;  
            2'b11: y=d3;  
            default: y=0;  
        endcase  
    end  
endmodule
```

```
(d0,d1,d2,d3,y,s1,s2 (1)____);  
    wire ( 2 ) ____  
    {s1 , s2} ( 3 ) ____
```


题目9

未回答

满分1.00

下面是一个8位累加器ACC，对输入的8位数据进行累加。

```
module cc(accout,cout,accin,cin,clk,clear);//累加器顶层模块
  ( 1 ) ;//写出8位累加和输出变量accout的定义
  output cout; input[7:0] accin; input cin,clk,clear;
  ( 2 ) ;//补齐中间变量sum的定义

  assign { cout ,sum}=add8(accout,accin,cin); //调用add8函数
  reg8 accreg8(accout,sum,clk,clear); //寄存器子模块

  定义函数 add8，实现累加操作。
  ( 3 ) //补齐
  input[7:0] a,b; input cin;
  add8=a+b+cin;
  endmodule

  module reg8(qout,in,clk,clear);
  ( 4 ) //补齐8位输出qout的定义
  input[7:0] in;input clk,clear;
  always @( ( 5 ) )//clk上升沿，异步clear
  if(clear) qout<=0; else qout<=in;
  endmodule
```

(1)	output[7:0] accout
(2)	wire[7:0] sum
(3)	function[8:0] add8; 位宽>=7
(4)	output reg[7:0] qout;
(5)	posedge clk or posedge clear

题目10

未回答

满分1.00

输入一个6位2进制序列，在七段管显示其十进制形式。代码如下。

顶层模块	module calcut (in6,shi,ge) ;
module show(in6,hex0,hex1);	input[5:0] in6;
input[5:0] in6;	output[3:0] shi,ge;
output reg[6:0] hex0,hex1;	assign shi=in6/10;
____(13)____; //补齐shi_ge定义	assign ge=in6%10;
calcut c0(in6,shi,ge); //模块调用	endmodule
always@(in6)	
begin	
____(14)____; //调用任务，输出shi到hex1	
.....//其他代码略	
end	
//下面七段管译码任务定义	
____(15)____; //任务名shw7	
output[6:0] hex;	
input[3:0] in4;	
case(in4)	
4'd0=7'b1000000;	
.....//其他略	
endcase	
endtask	
endmodule	

(13)	wire[3:0] shi,ge
(14)	shw7(hex1,shi)或shw7(shi,hex1)
(15)	task shw7;

题目11

未回答

满分1.00

若定义一个模块，模块名为My_M，输入in4和 con1；输出为5位out5，内部4位宽变量temp。根据要求补齐代码。

```
module__ ( 1 ) _____; //补齐模块声明
input[3:0] in4;
input con1;
____ ( 2 ) _____; //补齐输出变量out5定义,一句或两句代码都可以
____ ( 3 ) _____ //补齐临时变量temp定义
assign temp=(con1)?4'b0001:4'b1000;
always@(in4)
    out5=temp+in4;
endmodule
```

(1)	My_M(in4,con1,out5)
(2)	output reg[4:0] out5
(3)	wire[3:0] temp;

题目12

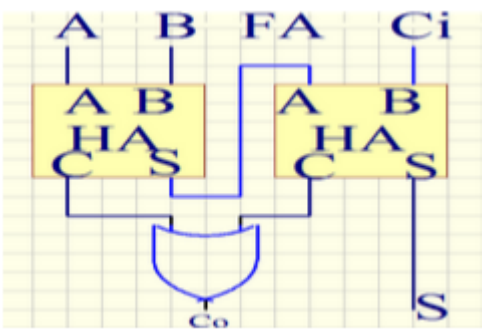
未回答

满分1.00

HA模块程序如下，写出引用HA模块描述FA模块的Verilog程序。

```
Module HA(A,B,S,C);
input A,B;
output S,C;
assign __ (13) __=A+B;
endmodule

module FA(A,B,Ci,Co,S);
input A,B,Ci;
output Co,S;
__ (14) __S1,C1,C2;
HA a(A,B,S1,C1);
HA b(S1,Ci,C2,S);
Assign__ (15) __//定义Co.;
endmodule
```



(13)	{C,S}
(14)	wire
(15)	Co=C1 C2

题目13

未回答

满分1.00

一个序列状态检测程序，三段式。上升沿时钟信号clk；异步低电平有效复位信号clr；串行输入序列x；检测到某X序列输出z=1，否则z=0.

```

module fsmseq (clk,clr,x,z);
input clk,clr,x;
( 6 ) //定义输出z
reg[1:0] state , next_state;//定义现态state和次态next_state。
( 7 ) /*状态S0,S1,S2,S3编码，一位热码编码方式*/ 补齐状态编码定义

always @(posedge clk , negedge clr) /*该过程定义当前状态*/
begin  if((8)___ ) state<=S0; //clr复位
else state<=next_state; end

always @(_(9)_____) /*该过程定义次态*/
case (state)
    S0:begin if(x) next_state<=S1;
    .....//其他代码略
    Endcase

assign z=(state==S3)?1:0; /*输出逻辑*/

endmodule

```

(6)	output z;
(7)	parameter S0=4'b0001,S1=4'b0010, S2=4'b0100,S3=4'b1000;或二进制 编码
(8)	!clr或~clr
(9)	state or x

题目14

未回答

满分1.00

下面是一个2级流水线实现的8位加法器。

```
module adder_pipe2(cout,sum,ina,inb,cin,clk);
input[7:0] ina,inb; input cin,clk;
output reg[7:0] sum; output reg cout;
reg[7:0] tina,tinb; //缓冲级变量tina , tinb
reg tcin;//缓冲级tcin
(10)_____ ;//定义第一级流水线变量tempa , tempb , firsts
reg firstc;
always@ ( posedge clk ) //数据缓冲级
begin tina<=ina; tinb<=inb; tcin<=cin;end
always @(posedge clk)//第一级
begin {firstc,firsts}=tina[3:0]+tinb[3:0]+tcin;
(11)_____ //补齐代码，缓存未运算数据
end
always @(posedge clk)//第二级
begin {cout,sum[7:4]}=tempa+tempb+firstc;
sum[3:0]=firsts;
//{cout,sum}= ( 12 ) _____ //将上面两条语句合并成一句代码
end
endmodule
```

(10)	reg[3:0] tempa,tempb,firsts
(11)	tempa=tina[7:4]; tempb=tinb[7:4]
(12)	{{(tempa+tempb+firstc),firsts}}

题目15

未回答

满分1.00

下面代码为2级流水线实现的4位加法器，按要求完成代码。

```
module kaoshi(cout,sum,ina,inb ,clk);
input[3:0] ina,inb;  input clk; output reg cout;
output [3:0] sum;// ( 1 ) 本行代码有误，请修改
reg[1:0] tempa,tempb,firsts;  reg firstc;
always @(posedge clk)
begin {firstc,firsts}<=ina[1:0]+inb[1:0];
tempa<=ina[3:2]; tempb<=inb[3:2]; end
always @(posedge clk)
begin {cout,sum[3:2]}<=     (2)    ;//请补齐代码
sum[1:0]<=   ( 3 )   ;//补齐代码
end
endmodule
```

1	output reg[3:0] sum;	3	firsts
2	tempa+tempb+firstc		

题目16

未回答

满分1.00

下列为一个4选1的数据选择器代码。

```
module select4to1(in1,in2,in3,in4,out,sel);
input[3:0] in1,in2,in3,in4;
input[1:0] sel;
output[3:0] out;
reg[3:0] out
always@(__ ( 4 )__); //补齐always过程定义
    case(sel)
        2'b00:out=in1;
        2'b01:out=in2;
        2'b10:out=in3;
        2'b11:out=in4;
        __ ( 5 )__; //从组合电路条件完备性角度，补齐代码，
    endcase
endmodule
```

(4)	in1,in2,in3,in4,sel或*
(5)	default:out=4'bx

题目17

未回答

满分1.00

带优先级5-3编码器和数据选择器，代码如下。

```
module code83(in1,in2,in3,in4,in5,out3);
input in1,in2,in3,in4,in5;
output reg[2:0] out3;
always@(in8)
begin if(in3) out3=3'b101;
  else if(in4) out3=3'b100;
else if(in5) out3=3'b011;
  else if(in2) out3=3'b010;
  else if(in1) out3=3'b001;
    ( 11 ) //该电路为组合电路，从电路完备性方面考虑，补齐代码
end
in1---in5按照优先级低到高的顺序为： ( 12 )
endmodule
```

module mux_casex(out,a,b,c,d,sel); //带优先级的数据选择器，补齐代码。

```
input a,b,c,d,sel;//a,b,c,d为四路输入
input[3:0] sel; //sel为选择控制信号
output reg out;
always@(a or b or c or d or sel)
begin
  ( 13 ) //补齐代码
  4'b???1:out=a;
  4'b??1?:out=b;
  4'b?1??:out=c;
  4'b1???:out=d;
endcase
end
```

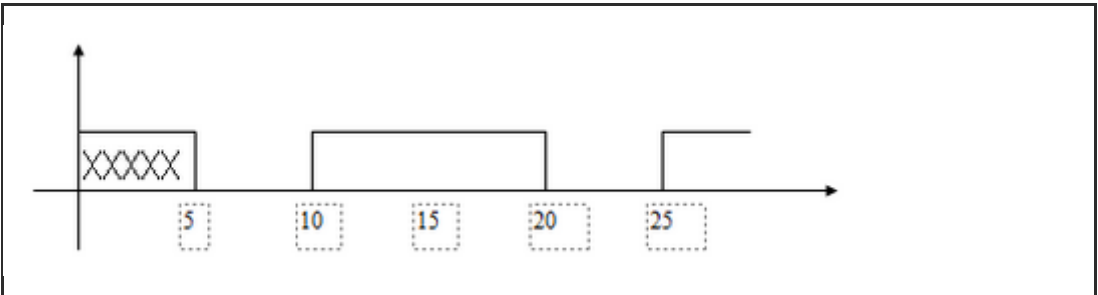
11	default:out3=3'bx;	13	casez(sel)
12	in1,in2,in5,in4,in4,in3		

题目18

未回答

满分1.00

如下图所示r的波形，补齐时延数据。



```
fork
#20 r=1' b0;
#10 r=1' b1;
# ( 11 )    r=1' b1;
#25 r=1' b1;
# ( 12 )    r=1' b0;
join
```

(11)	15
(12)	5

题目19

未回答

满分1.00

阻塞与非阻塞。当前a=1 ; b=2 ; c=3。经过一个时钟周期后。

<pre>module blocking(clk,a,b,c); output [3:0] b,c;input [3:0] a;input clk; reg [3:0] b,c; always @(posedge clk) begin b = a; c = b;end endmodule</pre>	<pre>module non_blocking(clk,a,b,c); output [3:0] b,c;input [3:0] a;input clk;reg [3:0] b,c; always @(posedge clk) begin b <= a;c <= b; end endmodule</pre>
(1) a、b、c的值为_____	(2) a、b、c的值为_____

(1) a、b、c的值为 1、1、1 (2) a、b、c的值为 1、1、2

题目20

未回答

满分1.00

两段代码是阻塞和非阻塞赋值，若当前q=0,则clk出现一个上升沿后，对应的a、q的值

<pre>module serial1(q,a,clk); output reg q,a; input clk; always @(posedge clk) begin q=~q; a=~q; end endmodule</pre>	<pre>module serial1(q,a,clk); output reg q,a; input clk; always @(posedge clk) begin q<=~q; a<=~q; end endmodule</pre>
{a , q}=2'b____(1)____	{a , q}=2'b____(2)____

(1)	01
(2)	11

题目21

未回答

满分1.00

带同步清零、同步置数的模256计数器，根据提示补齐代码。

```
module count(out,data,load,reset,clk);  
output[7:0] out; input[7:0] data;  
input load,clk,reset;  
____ ( 1 ) ____ ;//补齐out数据类型定义  
always @( ____ ( 2 ) ____ ) //clk上升沿触发  
begin  
if(____ ( 3 ) )out=8'h00; //reset同步清0，低电平有效  
else if(load) out=data; //同步预置  
else out=out+1; //计数  
end  
endmodule  
1. reg[7:0] out; ( 2 ) posedge clk ( 3 ) !reset
```

(2) 下面是一个1/2 分频器的可综合模型。

```
module half_clk(reset,clk_in,clk_out);  
input clk_in,reset; output clk_out;  
____ ( 1 ) ____ ;//补齐clk_out的数据类型定义  
always @(____ ( 2 ) ____)//clk_in上升沿  
begin if(____ ( 3 ) ) clk_out=0; //reset同步触发，低电平有效。  
else clk_out=~clk_out; end  
endmodule
```

```
reg clk_out ( 1 )  
posedge clk_in ( 2 )  
!reset ( 3 )
```

题目22

未回答

满分1.00

下面D触发器的逻辑代码，请按要求完成。

```
module DFF2(q,qn,d,clk,set,reset);
input d,clk,set,reset;//clk使用上升沿触发。set是同步置位信号，reset是异步复位信号
output reg q,qn;
always @(( 1 ))//补齐代码
begin
if_(( 2 )) begin q <= 0; qn <= 1; end //补齐代码，reset异步复位0，低电平有效
    else if (( 3 )) begin q <=1; qn <=0; end //补齐代码，set同步置 1，高电平有效
        else  begin q <= d; qn <= ~d; end
    end
endmodule
```

1) posedge clk,negedge reset	
2) !reset	3) set

题目23

未回答

满分1.00

采用参数定义的 8位 数据比较器。

```
module compare_w(a,b,larger,equal,less);  
_____( 1 )_____ //参数定义，定义符号常量SIZE  
input[SIZE-1:0] a,b;  
output larger,equal,less;  
_____( 2 )_____ //定义equal的数据类型;  
assign larger=(a>b);  
always@( _____ ( 3 ) _____ )//补齐敏感列表a,b  
    equal=(a==b);  
assign less=(a<b);  
endmodule
```

(1) parameter SIZE=8; (2) reg equal; (3) a or b

题目24

未回答

满分1.00

用for循环的移位相加实现2个8位数相乘

```
module mult_for(outcome,a,b);  
    ( 1 ) ; //定义符号常量size  
input[size:0] a,b;  
output reg [2*size+1:0] outcome;  
integer i;  
always @(a or b)  
begin outcome=0;  
for(i=1; i<= ( size+1); i=i+1)  
    ( 2 ) //补齐代码  
end  
endmodule
```

1) parameter size=7;

2) if(a[i-1]) coutcome=outcome+b<<(i-1)
或if(b[i-1]) coutcome=outcome+a<<(i-1)

题目25

未回答

满分1.00

下面是一个johnson计数器的代码，按要求补齐代码。clk为时钟信号，上升沿；clr为**异步**复位信号。

计数状态为：00000à00001à00011à...à10000à00000

```
module johnson(clk,clr,qout);
  ( 6 ) ;//根据上下文，定义符号常量
input clk,clr;
output reg[(WIDTH-1):0] qout;
always @( ( 7 ) )//补齐信号列表
begin
  if( ! clr)
    qout<=0;
  else begin
    qout<=qout<<1;//语句A
    qout[0]<=~qout[WIDTH-1];//语句B
    ( 8 ) ; //用连接运算符{}，将A、B合并成一句。
  end
end
endmodule
```

(6)	parameter WIDTH=5
(7)	posedge clk or negedge clr
(8)	qout<={qout,~ out[WIDTH-1]}

题目26

未回答

满分1.00

```
module test ( clk,a,b,c,d ) ;  
input clk;  input[2:0] a;  
output[2:0] b,c,d ;  
wire c;____ //(1)本行有错，请将正确语句写出来  
____ //( 2 ) b、d数据定义错误，将漏掉的数据类型定义补全。  
assign c=a&b ;  
always@ ( a )  
    begin    if(a>5) b=a;    else  b=a-1;  
    end  
always@(posedge clk)  
    begin    d<=d+1;  end  
endmodule
```

1. wire[2:0] c;
2. reg[2:0] b,d;

题目27

未回答

满分1.00

下面代码为字符串“HE_”移位电路，en为1时，循环左移，每次移位一个七段管；en为0时，循环右移，每次移位一个七段管。分频模块，输入时钟50Hz，输出时钟信号为1Hz。

```
module lab008(HEX0,HEX1,HEX2,HEX3,CLK,en);
input en, CLK; output[6:0] HEX0,HEX1,HEX2,HEX3;
reg[27:0] temp=28'b0001001_0000110_1110111_1110111;//HE_ _
divclk (CLK,clock_out);//分频模块
assign {HEX3,HEX2,HEX1,HEX0}=temp;
always@(posedge clock_out)
begin
    if(en) temp<={temp,temp[27:21]};
    else temp<={temp[6:0],temp};// ( 9 ) 本行代码有误
end
endmodule

module divclk(c_in,c_out);//分频模块，分频比为50
input c_in;
output reg c_out;
integer i;
always @(posedge c_in)
begin
    if(i==50)_ // ( 10 ) 本行代码有误。
begin c_out<=~c_out ; i<=0;end
    else i<=i+1;
end
endmodule
```

(9)	temp<={temp[6:0],temp[27:7]}
(10)	if(i==24)

题目28

未回答

满分1.00

偶数分频(8分频，占空比50%)

```
module count5(reset,clk,out)
input clk,reset;
output out;
reg _count;//错误8
always@(posedge clk)
if(reset) begin count<=0; out<=0; end
else if( _count==4 ) //错误9
begin out<=!out: end//此处漏掉一句，请补充 错误10
else count<=count+1;
endmodule
```

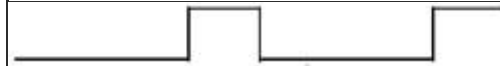
(8)	reg[1:0] count
(9)	Count==3
(10)	count<=0;

题目29

未回答

满分1.00

下面代码为40倍分频电路，占空比30%（高电平占整个周期的比例），波形如下



```
module divclk(clk40,clk1,key0);
input clk40; output reg clk1; reg[6:0] count;
always@(posedge clk80 , negedge key0)
begin
    if(!key0) count<=0;
    else begin
        if(count==39) count<=0; else count=count+1;// ( 19 ) 本行代码有错误
        if ( ( 20 ) ) clk1<=0;//补齐代码
    else clk1<=1;
    end
end
endmodule
```

19 cout<=cout+1;

20 cout<=27或cout<28

题目30

未回答

满分1.00

用函数和 case 语句描述的编码器,输入din高位优先级高。

```
module code_83(din,dout);
input[7:0] din;
output reg[2:0] dout; //(1)本代码有错误
assign dout = code(din) ; //函数调用
function code; //(2)函数定义，本行代码有错误
input[7:0] din;
case (din) __ ( 3 ) 本行代码有错误
8'b1xxx_xxxx : code = 3'h7;
.....//代码略
default: code = 3'hx;
endcase
endfunction
endmodule
```

(1)	output [2:0] dout
(2)	function[2:0] code
(3)	casex (din)或casez (din)

题目31

未回答

满分1.00

下面代码为可置数的模256计数器，并将计数值以十进制形式在三个七段管上显示。

clk为时钟信号，上升沿；load为**同步**置数信号，低电平有效；clr为**异步**复位信号，**高电平**有效；data为8位置数输入；hex2、hex1、hex0为七段管输出。

```
module first(clk,load,clr,data,hex0,hex1,hex2);
input clk,clr,load; input[7:0] data;
output reg [6:0] hex0,hex1,hex2; // ( 4 ) 本行代码有错
reg[7:0] sum;
wire s2,s1,s0; // ( 5 ) 本行代码有错
always@(_posedge clk,posedge clr,posedge load) //(6)黑体代码有错
begin
// A行
if(clr) sum<=0;
else if(load) sum<=data; else sum<=sum+1; // ( 7 ) 黑体代码有误
seprate se0(sum,s2,s1,s0); // ( 8 ) 代码报错，请问移到A行还是B行
end
// B行
decode d0(s0,hex0); decode d1(s1,hex1); decode d2(s2,hex2);
endmodule
```

子模块，将数值拆分为百、十、个位

```
module seprate(in8,s2,s1,s0);
input[7:0] in8; output[3:0] s2,s1,s0;
具体代码略
endmodule
```

子模块，七段管译码电路

```
module decode(ina, out);
input [3:0] ina; output reg [6:0] out;
具体代码略
endmodule
```

(4)	output [6:0] hex0,hex1,hex2;
(5)	wire[3:0] s2,s1,s0
(6)	posedge clk,posedge clr
(7)	!load或~load
(8)	B

题目32

未回答

满分1.00

根据提示将正确代码或错误原因写入指定位置。

```
一个可预置初值的7进制循环计数器
module count(clk,reset,load,date,out);
input load,clk,reset;
input[3:0] date;
output reg[3:0] out;
parameter [3 : 0] WIDTH=4'd7;//错误4
always@(clk or reset)//错误5
begin
    if(reset) out<=4'd0;
    else if(!_load_) //load低电平有效，错误6
out<=date;
    else if(out==WIDTH-1) out<=4'd0;
    else out=out+1;//错误7
end
endmodule
```

(4)	parameter WIDTH=4'd7;或去掉[3:0]
(5)	Posedge clk , posedge reset
(6)	!load
(7)	Out<=out+1或<=

题目33

未回答

满分1.00

根据提示将正确代码或错误原因写入指定位置。

```
module addsub
(  input [7:0] dataa, input [7:0] datab,
  input reg add_sub, //错误 ( 1 )
  input clk,
  output [8:0] result ); //错误2
  always @ (posedge clk)
  begin   if (add_sub)
assign result <= dataa + datab; //错误3
  else   result <= dataa - datab;
        end
endmodule
```

(1)	<u>Input add_sub</u> 或去掉reg
(2)	output reg [8:0] result 或添加reg
(3)	<u>result <= dataa + datab;</u> 或去掉assign

题目34

未回答

满分1.00

```
module decode4_7(decodeout,indec);
*****错误（1）数据类型定义错误，如有遗漏补全代码*****

output[6:0] decodeout;
*****错误（1）*****

input[3:0] indec;
always @(indec)
case(indec) //用case语句进行译码
4'd0:decodeout=7'b1111110;
4'd1:decodeout=7'b0110000;
4'd2:decodeout=7'b1101101;
4'd3:decodeout=7'b1111001;
4'd4:decodeout=7'b0110011;
4'd5:decodeout=7'b1011011;
4'd6:decodeout=7'b1011111;
4'd7:decodeout=7'b1110000;
4'd8:decodeout=7'b1111111;
4'd9:decodeout=7'b1111011;
(2) //代码有错误，不符合组合电路条件完备性，请补齐
endcase
endmodule
```

(1) reg[6:0] decodeout; (2) default: decodeout=7'bx;

题目35

未回答

满分1.00

(1) module test (in4,hex1,hex0) ;

input[3:0] in4;

output reg[6:0] hex1,hex0; // (1) 该行有错误，请修改

_____ //错误 (2) ge、sh默认类型和位宽无法实现设计功能。写出正确代码

assign ge=in4%10;

assign shi=in4/10;

bto7 b0(ge,hex0);

bto7 b1(shi,hex1);

endmodule

module bto7(in4,out7)//bcd-七段管译码模块

.....代码略

Endmodule

1. output[6:0] hex1,hex0;

2. wire[3:0] ge,shi;

题目36

未回答

满分1.00

根据提示修改错误。

```
module AAA ( a ,b );  
    output a ;  
    input [6:0] b ;  
    reg __sum;__//错误 ( 1 )  
    integer i;  
    reg a ;  
    always @ (b)  
    begin    sum = 0;  
        for(i = 0;i<=6;i = i+1)  
            if(b[i]) sum +=1;__//错误 ( 2 )  
                if(sum[2]) a = 1;        else    a = 0;  
    end  
endmodule
```

reg[2:0] sum;__//错误 (1)

sum =sum+1;__//错误 (2)

题目37

未回答

满分1.00

两个4位二进制数相加，七段管输出5位的和。。

```
module add4(sw, hex1,hex0);  
input[7:0] sw;  
output reg[6:0] hex1,hex2,hex0;// ( 1 ) //错误1  
wire[4:0] sum;  
( 2 ) //错误2，漏掉数据定义，导致显示0或1数据  
assign sum=sw[3:0]+sw[7:4];  
assign ge2=sum%4'd10 ;  
assign shi2=sum/4'd10;  
BCDto7 b3(shi2,hex1);BCDto7 b2(ge2,hex0);  
endmodule
```

(1)	output [6:0] hex1,hex2,hex0;或去掉reg
(2)	wire [3:0] ge2,shi2;

题目38

未回答

满分1.00

下面是一个模60的二进制加法计时器。

```
module lab4_2(KEY0,CLOCK_50,HEX1,HEX0);  
input KEY0; input CLOCK_50;  
output reg[6:0] HEX1,HEX0;// ( 16 ) 本行有误 , 请修改  
reg_sec0 , sec1 ; // ( 17 ) 本行有错误 , 请修改  
reg[5:0] sum;  
assign sec0=sum%10; assign sec1=sum/10;  
BCDto7 a0(sec0,HEX0); BCDto7 a1(sec1,HEX1);  
divclk1hz c1(CLOCK_50,clk1);//分频电路 , 实现略  
always@(posedge clk1,negedge KEY[0])  
begin if(!KEY[0]) sum=0;  
    else if(( 18 )) sum=0;//补齐代码  
    else sum=sum+1;  
end endmodule
```

16 output[6:0] HEX1,HEX0;	17 wire[3:0] sec0,sec1;
18 sum==59	

题目39

未回答

满分1.00

根据提示改正错误。

```
module tryfunc(clk,n,result,reset);
```

```
output[31:0] result; input[3:0] n;
```

```
input reset,clk;
```

```
reg result;//错误 ( 1 )
```

always @(posedge clk , posedge reset) //clk上沿触发，reset同步。错误 (2)，只改横线部分

```
begin
```

```
if(!reset) //reset 为低时复位。
```

```
result<=0;
```

```
else begin result <= n * factorial(n)/((n*2)+1); end
```

```
end
```

```
function factorial; //函数定义，返回32位计算结果。错误 ( 3 )
```

```
input [3:0] operand;
```

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

```
always@ ( posedge clk ) //错误 ( 4 )
```

```
begin
```

```
factorial = operand ? 1 : 0;
```

```
for(index = 2; index <= operand; index = index + 1)
```

```
factorial = index * factorial;
```

```
end
```

```
endfunction
```

```
endmodule
```

reg[31:0] result;//错误 (1)

posedge clk错误 (2)

function [31:0]//错误 (3)

去掉always@ (posedge clk) //错误 (4)

题目40

未回答

满分1.00

下面代码为流水灯电路，在18个led等上循环右移。可以通过按键调节速度。

```
module shiftc(clk_50,ledr,key);
input clk_50; output[17:0] ledr; reg state;
reg[17:0] temp=18'b101010101001001001;
div_clock_1hz d1(clk_50,clk1);
div_clock_10Hz d2(clk_50,clk10);
assign clk=state?clk1:clk10;//频率切换
always(negedge key)//key为de2开发板按键
begin
    if(!key) state<=~state; ( 3 ) //错误3
end
always@(posedge clk)
begin
    temp<={temp[1:0],temp} ( 4 ) ;//实现序列循环右移,每次移动2位操作，但是有错误
end
assign ledr=temp;
endmodule
```

下面代码为分频代码（之一），错误5所示代码区间，逻辑功能不能实现，如果代码错误或缺少，请将正确代码写入答题纸。

```
module div_clock_1hz(clock_in,clock_out,clr);//50Mhz分频为1Hz
input clock_in,clr;
output reg clock_out;
integer i;
always@(posedge clock_in or posedge clr)
begin
    if(clr) i<=0;
    else if(i==24999999)
begin //*****错误5，代码段开始*****
    clock_out<=~clock_out;
end //*****错误5，代码段结束*****
    else i<=i+1;
end
endmodule
```

(3)	state<=~state;或去掉if(!key)
-------	---------------------------

(4)	temp<={temp[1:0],temp[17:2]}
(5)	添加 i<=0;

◀ 第二部分 Verilog基础语法 (选择题) 4-5号开放

跳至... ▼

第三部分 程序设计，提交答案后或回顾可查看参考答案，5-6 ▶