

个人主页 ► 我的课程 ► 2018——2019数字电路与EDA设计 (张锡岭) ►
2018-2019-1《数字电路与EDA设计》考前练习,测试都已开放 ►
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

开始时间 2019年01月5日 星期六 19:06

状态 完成

完成于 2019年01月5日 星期六 19:18

耗时 12 分钟 11 秒

题目1

未回答

满分1.00

设计一个模65的BCD码计数器。qout输出,cout进位,load同步置数,data置数输入,reset同步复位,clk时钟信号,cin前级进位。

```
module count65(qout,cout,data,load,cin,reset,clk);
output[7:0] qout;  output cout;
input[7:0] data;  input load,cin,clk,reset;
reg[7:0] qout;
always@(posedge clk)
begin
if(reset)
qout<=0;
else if(load) qout<=data;
else if(cin) begin
if(qout[3:0]==9&&qout[7:4]!=6)
begin
qout[3:0]<=0 ; qout[7:4]<=qout[7:4]+1;
end
else if(qout[3:0]==4&&qout[7:4]==6)
begin qout[3:0]<=0; qout[7:4]<=0;
end
else
qout[3:0]<=qout[3:0]+1;
end
end
assign cout=((qout==8'h65)&cin)?1:0;
endmodule
```

题目2

未回答

满分1.00

设计一个简单算术逻辑单元。

out为8位输出；a、b为4位操作数，输入；opcode为操作类型，输入。

Opcode实现操作类型：加、减、乘、除。其中减法用任务task调用实现，返回减法的绝对值。

```
module alu(out,opcode,a,b);  
module alu(out,opcode,a,b);  
input[1:0] opcode; input[3:0] a,b; output reg[7:0] out; //数据定义 3分  
always@(a,b,opcode) //逻辑定义 4分  
begin case(opcode)  
2'b00:out=a+b; 2'b01:out=a*b; 2'b10:out=a/b;  
2'b11:sub(a,b,out); default:out=8'hxx;  
endcase end  
  
task sub; //任务定义 3分  
input[3:0] a,b; output[7:0] out;  
if(a>b) out=a-b; else out=b-a;  
endtask  
endmodule
```

题目3

未回答

满分1.00

试用verilog语言描述一个4位移位寄存器，是由四个D触发器（分别设为U1，U2，U3，U4）构成的。其中seri_in是这个移位寄存器的串行输入；clk为移位时钟脉冲输入；clr为清零控制信号输入；Q[1]~Q[3]则为移位寄存器的并行输出。

```
module d_flop(q,d,clr,clk);
output q;
input d,clr,clk;
reg q;
always @(clr)
if(!clr) assign q=0;
else deassign q;
always @(negedge clk)
q=d;
endmodule

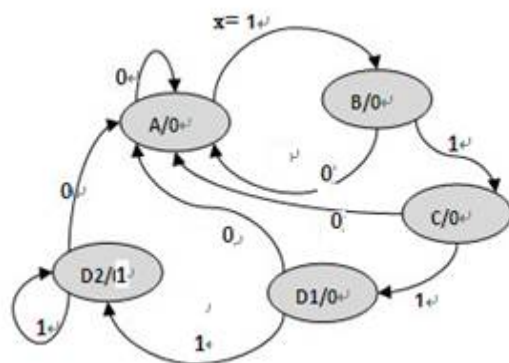
module shifter(seri_in,clk,clrb,Q);
input seri_in,clk,clrb;
output[3:0] Q;
d_flop U1(Q[0],seri_in,clrb,clk),
U2(Q[1],Q[0],clrb,clk),
U3(Q[2],Q[1],clrb,clk),
u4(Q[3],Q[2],clrb,clk);
endmodule
```

题目4

未回答

满分1.00

设计有限状态机。输入x，时钟clk，rst复位信号回到初始状态A。状态机具有五个状态A、B、C、D1、D2，根据输入x为1或0，进行不同转移，如下图。D2状态下，输出y为1，其他状态y为0。



module Detect(x,clk,rst,y); 状态转移图

input x , clk,rst;

output y;

parameter A=3'D0,B=3'D1,C=3'D2,D1=3'D3,D2=3'D4;

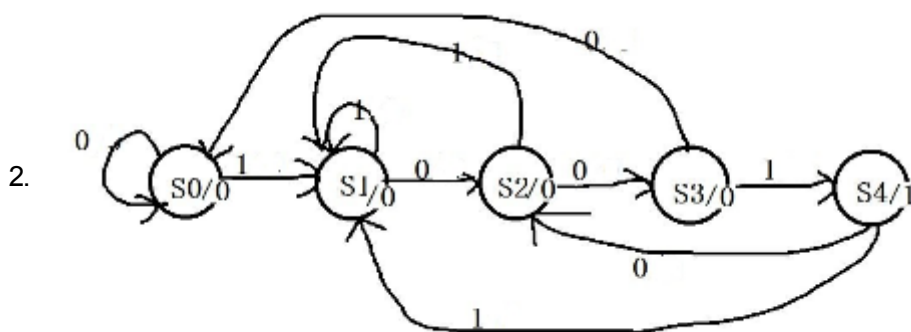
```
reg[2:0] state,next_state;
always@(posedge clk,posedge rst)
if(rst) state<=A;
else state<=next_state;
always@(state,x)
case(state)
A:if(x) next_state<=B;else next_state<=A;
B:if(x) next_state<=C;else next_state<=A;
C:if(x) next_state<=D1;else next_state<=A;
D1:if(x) next_state<=D2;else next_state<=A;
D2:if(x) next_state<=D2;else next_state<=A;
default: next_state<=A;
endcase
always@(state)
case(state)
D2:y<=1;
default:y<=0;
endcase
endmodule
```

题目5

未回答

满分1.00

1. 下图为一个特定序列检测的状态转移图，根据状态转移图写出逻辑设计代码。S0/0标表示S0状态，输出z=0。 clk时钟信号，上升沿；x输入串行序列，reset高电平有效异步复位信号；输出z输出0或1。



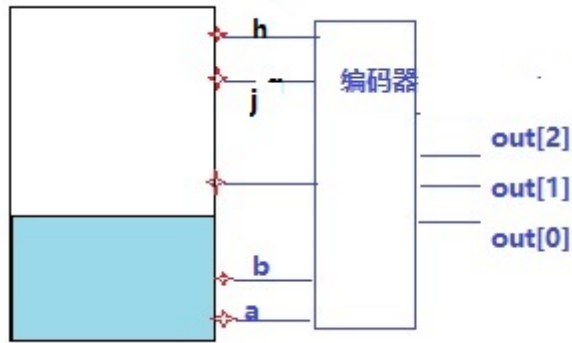
```
module fsm(x , clk , reset , z);
input clk,x,reset;
output reg z;
reg[4:0] state;
parameter s0=5'b00001,s1=5'b00010,s2=5'b00100,s3=5'b01000,s4=5'b10000;
always@(posedge clk,posedge reset)
begin
if(reset)state<=s0;
else
case(state)
s0:begin if(x) state<=s1; else state<=s0; z<=0;end
s1:begin if(x) state<=s1; else state<=s2; z<=0;end
s2:begin if(x) state<=s1; else state<=s3; z<=0;end
s3:begin if(x) state<=s4; else state<=s0; z<=0;end
s4:begin if(x) state<=s1; else state<=s0; z<=1;end
default:begin state<=s0;z<=0;end
end
endmodule
```

题目6

未回答

满分1.00

如下图所以，有一个容器内部安装了a，b，c,d,e,f,j.,h共8个传感器，当水淹过传感器的时
候传感器输出1到译码器。当水淹没a则输出3'b000，若淹过b，则输出3'b001，若淹过
c，则输出3'b010，....一次类推，淹过h，则输出3'b111。设计一个8-3的优先编码器实现
该功能。



```
module encoder8_3(none_on,outcode,a,b,c,d,e,f,g,h);
output none_on;
output[2:0] outcode;
input a,b,c,d,e,f,g,h;
reg[3:0] outtemp;
assign {none_on,outcode}=outtemp;
always@(a or b or c or d or e or f or g or h)
begin
    if(h)                outtemp=3'b111;
    else if(g)            outtemp=3'b110;
    else if(f)            outtemp=3'b101;
    else if(e)            outtemp=3'b100;
    else if(d)            outtemp=3'b011;
    else if(c)            outtemp=3'b010;
    else if(b)            outtemp=3'b001;
    else if(a)            outtemp=3'b000;
end
endmodule
```

题目7

未回答

满分1.00

设计一个20倍 分频器的可综合模型。RESET同步复位信号，低电平。F10M输入时钟，F500K分频后输出时钟信号。

```
module fddivision(RESET,F10M,F500K);  
input F10M,RESET;  
output F500K;  
reg F500K;  
reg [7:0]j;  
always @(posedge F10M)  
if(!RESET) //低电平复位。  
Begin F500K <= 0;j <= 0;end  
Else begin if(j==19)  
Begin j <= 0;  
F500K <= ~F500K;  
end  
else  
j <= j+1;  
end  
endmodule
```


题目8

未回答

满分1.00

设计一个逻辑运算器。宏定义如下

```
`define plus 3'd0
```

```
`define minus 3'd1
```

```
`define band 3'd2
```

```
`define bor 3'd3
```

```
`define unegate 3'd4
```

```
`define plus 3'd0
```

```
`define minus 3'd1
```

```
`define band 3'd2
```

```
`define bor 3'd3
```

```
`define unegate 3'd4
```

```
module alu(out,opcode,a,b);
```

```
output[7:0] out;
```

```
reg[7:0] out;
```

```
input[2:0] opcode;
```

```
input[7:0] a,b; //操作数。
```

```
always@(opcode or a or b) //电平敏感的always 块
```

```
begin
```

```
case(opcode)
```

```
`plus: out = a+b; //加操作。
```

```
`minus: out = a-b; //减操作。
```

```
`band: out = a&b; //求与。
```

```
`bor: out = a|b; //求或。
```

```
`unegate: out=~a; //求反。
```

```
default: out=8'hx; //未收到指令时，输出任意态。
```

```
endcase
```

```
end
```

```
endmodule
```

题目9

未回答

满分1.00

一个简单的状态机设计，功能是检测一个5 位二进制序列“10010”。考虑到序列重叠的可能，有限状态机共提供8 个状态（包括初始状态IDLE）。

```
module seqdet(x,z,clk,rst,state);
input x,clk,rst;
output z;
output[2:0] state;
reg[2:0] state;
wire z;
parameter IDLE='d0, A='d1, B='d2,
C='d3, D='d4,
E='d5, F='d6,
G='d7;
assign z = ( state==E && x==0 )? 1 : 0; //当x=0 时，状态已变为E，
//状态为D 时，x 仍为1。因此
//输出为1 的条件为 ( state==E && x==0 )。
always @(posedge clk)
if(!rst)
begin state <= IDLE;end
else casex(state)
IDLE : if(x==1) beginstate <= A;end
A: if(x==0) begin state <= B;end
B: if(x==0)begin state <= C;end
Else begin state <= F;end
C: if(x==1) begin state <= D; end
Else begin state <= G;end
D: if(x==0) begin state <= E;end
Else begin state <= A;end
E: if(x==0) begin state <= C;end
Else begin state <= A;end
F: if(x==1) begin state <= A;end
Else begin state <= B; end
G: if(x==1) begin state <= F; end
default:state=IDLE; //缺省状态为初始状态。
endcase
endmodule
```

题目10

未回答

满分1.00

设计一个4选1的数据选择器，功能描述如下：

4路单比特数据输入：A、B、C、D；

数据输出：Y

选通控制信号：S0，S1，S2，S3。

当S0=0，Y=A；

当S0不为0，S1=0，Y=B；

当S0，S1都不为0，S2=0，Y=C；

当其他都不为0，S3=0，Y=D；

```
Module mux ( A,B,C,D,S0,S1,S2,S3,Y ) ;
```

```
Input A,B,C,D,S0,S1,S2,S3;
```

```
Output reg Y;
```

```
Always@ ( A,B,C,D,S0,S1,S2,S3 )
```

```
Begin
```

```
  If ( !S0 ) Y=A;
```

```
  Else if ( ! S1 ) Y=B
```

```
    Else if ( ! S2 ) Y=C ;
```

```
      Else if ( ! S3 ) Y=D ;
```

```
        Else Y=x ;
```

```
End
```

```
Endmodule
```

题目11

未回答

满分1.00

设计一个时序电路，双向计数控制功能，功能描述如下。

输入：上升沿触发时钟clk；同步复位信号reset，低电平有效；

同步置数控制信号load，高电平；同步置数数据输入data；

计数控制ud：为1,加法计数；为0，减法计数；

输出：计数值out（0—250计数）

```
module count(out,data,load,reset,clk,ud);
output reg[7:0] out; input[7:0] data; input load,clk,reset,ud;
always @(posedge clk) //clk上升沿触发
    if(!reset) out=8'h00; //同步清0，低电平有效
    else if(load) out=data; //同步预置
    else if(ud) begin
        if(out==250) out=0; else out=out+1;end
        else begin if(out=0) out=250;else out=out-1; end
    end
endmodule
```

题目12

未回答

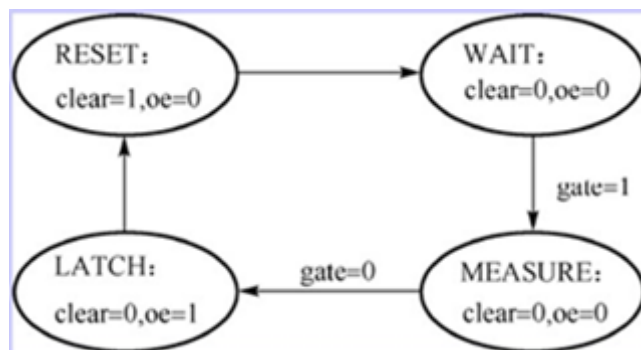
满分1.00

设计一个频率测量系统的控制器，定义如下。

输入：时钟clk，控制gate；

输出：oe和clear

状态转移图如右图所示：



(3) Verilog描述

• 一段式描述

```
module control(clk,reset,gate,oe,clear);
```

```
    input  clk,reset,gate;
```

```
    output reg oe,clear;
```

```
    //状态编码
```

```
    parameter RESET=2'B00,WAIT=2'B01,MEASURE=2'B10,LATCH=2'B11;
```

```
    reg [1:0] state;
```

```
    always @(posedge clk)
```

```
        if (reset) begin state<=RESET;clear<=1;oe<=0; end
```

```
        else
```

```
            case(state)
```

```
                RESET:    begin state<=WAIT; clear<=0; oe<=0;end
```

```
                WAIT:    if(gate) begin state<=MEASURE; clear<=0; oe<=0;end
```

```
                MEASURE:  if(~gate) begin state<=LATCH; clear<=0; oe<=1;end
```

```
                LATCH:    begin state<=RESET; clear<=1; oe<=0; end
```

```
            endcase
```

```
        endmodule
```



第35页

◀ 第二部分 补齐程序、改错，提交答案后或回顾可查看参考答案，5-6

跳至...