الناز رضایی 98411387

امتحان میان‌ترم درس سیستم‌های دیجیتال

پاسخ 1:

```
`timescale 1ns / 1ps
module Soal1(
    input c,
    input [1:0] s,
    input [7:0] b0,
    input [7:0] b1,
    input [7:0] b2,
    input [7:0] b3,
    input [7:0] a,
    input clk,
    output [15:0] x,
    wire [7:0] muxOut,
    wire ff1Out,
    wire [7:0] multOut1,
    wire ignore,
    wire [7:0] ff2Out,
    wire notc,
    wire [7:0] ff3out,
    wire [7:0] ff4out,
    wire [7:0] multOut2,
    wire [15:0] last
);

    always @(*)
    begin
        ignore <= 0;
        MUX8bit_4to1 e(b0,b1,b2,b3,s,muxOut);
        ripplemod e1(a,muxOut,0,multOut1,ignore);
        D_Flipflop e2(c,clk,ff1Out);
        DFlipflop e3(multOut1,clk,0,1,ff2Out);
        notc <= !ff1Out;
        DFlipflop e4(ff2Out,clk,0,ff1Out,ff3out);
        DFlipflop e5(ff2Out,clk,0,notc,ff4out);
        ripplemod e6(ff3out,ff4out,0,multOut2);
        multiplierr e7(last,multOut2,a,clk,0);
        x <= last;
    end
```

پاسخ 1:

```
endmodule
module MUX8bit_4to1(
    input [7:0] in_0,
    input [7:0] in_1,
    input [7:0] in_2,
    input [7:0] in_3,
    input [1:0] sel,
    output reg [7:0] out
    );
always @(in_0 or in_1 or in_2 or in_3 or sel) begin
        case (sel)
                2'b00: out <= in_0;
                2'b01: out <= in_1;
                2'b10: out <= in_2;
                2'b11: out <= in_3;
        endcase
end
endmodule
module DFlipflop(Din,clk,clear,enable,Q);
input [7:0]Din;
input clk,clear,enable;
output reg [7:0] Q;
always@(posedge clk)
if(enable)
begin
if(clear)
Q<=0;
else
Q<=Din;
end
endmodule
module ripplemod(a, b, cin, sum, cout);
input [07:0] a;
input [07:0] b;
input cin;
output [7:0]sum;
output cout;
wire[6:0] c;
fulladd a1(a[0],b[0],cin,sum[0],c[0]);
fulladd a2(a[1],b[1],c[0],sum[1],c[1]);
```

پاسخ 1:

```
fulladd a3(a[2],b[2],c[1],sum[2],c[2]);
fulladd a4(a[3],b[3],c[2],sum[3],c[3]);
fulladd a5(a[4],b[4],c[3],sum[4],c[4]);
fulladd a6(a[5],b[5],c[4],sum[5],c[5]);
fulladd a7(a[6],b[6],c[5],sum[6],c[6]);
fulladd a8(a[7],b[7],c[6],sum[7],cout);
endmodule
module fulladd(a, b, cin, sum, carry);
input a;
input b;
input cin;
output sum;
output carry;
assign sum=(a^b^cin);
assign carry=((a&b)|(b&cin)|(a&cin));
endmodule
module multiplierr(
output reg [15:0] product,
input [7:0] multiplicand, multiplier,
input clock, clear);
integer i, j ;
wire [63:0] s ,c ;
reg p [7:0][7:0];
always@(multiplier, multiplicand)
begin
for (i = 0; i <= 7; i = i + 1)
for (j = 0; j <= 7; j = j + 1)
p[j] <= multiplicand[j] & multiplier;
end
fulladd ha_11 ( .sum(s[0]), .carry(c[0]), .a(p[1][0]), .b( p[0][1]) , .cin(0) );
fulladd fa_11
( .sum(s[1]), .carry(c[1]), .a(p[2][0]), .b( p[1][1]), .cin( p[0][2]) );
fulladd fa_12
( .sum(s[2]), .carry(c[2]), .a(p[3][0]), .b( p[2][1]), .cin( p[1][2]) );
fulladd fa_13
( .sum(s[3]), .carry(c[3]), .a(p[4][0]), .b( p[3][1]), .cin( p[2][2]) );
fulladd fa_14
( .sum(s[4]), .carry(c[4]), .a(p[5][0]), .b( p[4][1]), .cin( p[3][2]) );
fulladd fa_15
( .sum(s[5]), .carry(c[5]), .a(p[6][0]), .b( p[5][1]), .cin( p[4][2]) );
```

پاسخ 1:

```
fulladd fa_16
( .sum(s[6]), .carry(c[6]), .a(p[7][0]), .b( p[6][1]), .cin( p[5][2]) );
fulladd ha_12 ( .sum(s[7]), .carry(c[7]), .a( p[7][1]), .b ( p[6][2]) , .cin(0));
fulladd ha_21 ( .sum( s[8] ), .carry( c[8 ]), .a( p[1][3]), .b( p[0][4]) , .cin(0) );
fulladd fa_21
( .sum( s[9] ), .carry( c[9 ]), .a( p[2][3]), .b( p[1][4]), .cin( p[0][5]) );
fulladd fa_22
( .sum( s[10]), .carry( c[10]), .a( p[3][3]), .b( p[2][4]), .cin( p[1][5]) );
fulladd fa_23
( .sum( s[11]), .carry( c[11]), .a( p[4][3]), .b( p[3][4]), .cin( p[2][5]) );
fulladd fa_24
( .sum( s[12]), .carry( c[12]), .a( p[5][3]), .b( p[4][4]), .cin( p[3][5]) );
fulladd fa_25
( .sum( s[13]), .carry( c[13]), .a( p[6][3]), .b( p[5][4]), .cin( p[4][5]) );
fulladd fa_26
( .sum( s[14]), .carry( c[14]), .a( p[7][3]), .b( p[6][4]), .cin( p[5][5]) );
fulladd ha_22 ( .sum( s[15]), .carry( c[15]), .a( p[7][4]), .b
( p[6][5]) , .cin(0) );
fulladd ha_31 ( .sum( s[16]), .carry( c[16]), .a( c[0]) , .b ( s[1]) , .cin(0) );
fulladd fa_31 ( .sum( s[17]), .carry( c[17]), .a( c[1]) , .b ( s[2]) , .cin
( p[0][3]) );
fulladd fa_32 ( .sum( s[18]), .carry( c[18]), .a( c[2]) , .b ( s[3]) , .cin ( s[8] ) );
fulladd fa_33 ( .sum( s[19]), .carry( c[19]), .a( c[3]) , .b ( s[4]) , .cin ( s[9] ) );
fulladd fa_34 ( .sum( s[20]), .carry( c[20]), .a( c[4]) , .b ( s[5]) , .cin
( s[10]) );
fulladd fa_35 ( .sum( s[21]), .carry( c[21]), .a( c[5]) , .b ( s[6]) , .cin ( s[11]) );
fulladd fa_36 ( .sum( s[22]), .carry( c[22]), .a( c[6]) , .b ( s[7]) , .cin ( s[12]) );
fulladd fa_37 ( .sum( s[23]), .carry( c[23]), .a( c[7]) , .b ( p[7][2]), .cin
( s[13]) );
fulladd ha_41 ( .sum( s[24]), .carry( c[24]), .a( c[9] ) , .b( p[0][6]) , .cin(0) );
fulladd fa_41
( .sum( s[25]), .carry( c[25]), .a( c[10]) , .b( p[1][6]), .cin( p[0][7]) );
fulladd fa_42
( .sum( s[26]), .carry( c[26]), .a( c[11]) , .b( p[2][6]), .cin( p[1][7]) );
fulladd fa_43
( .sum( s[27]), .carry( c[27]), .a( c[12]) , .b( p[3][6]), .cin( p[2][7]) );
fulladd fa_44
( .sum( s[28]), .carry( c[28]), .a( c[13]) , .b( p[4][6]), .cin( p[3][7]) );
fulladd fa_45
( .sum( s[29]), .carry( c[29]), .a( c[14]) , .b( p[5][6]), .cin( p[4][7]) );
```

پاسخ 1:

```
fulladd fa_46
( .sum( s[30]), .carry( c[30]), .a( c[15]) , .b( p[6][6]), .cin( p[5][7]) );
fulladd fa_47 ( .sum( s[31]), .carry( c[31]), .a( p[7][6]), .b
( p[6][7])  , .cin(0));
fulladd ha_51 ( .sum( s[32]), .carry( c[32]), .a( s[17]) , .b( c[16]) , .cin(0) );
fulladd fa_51 ( .sum( s[33]), .carry( c[33]), .a( s[18]) , .b( c[17]) , .cin(0));
fulladd fa_52 ( .sum( s[34]), .carry( c[34]), .a( s[19]) , .b( c[18]), .cin( c[8] ) );
fulladd fa_53
( .sum( s[35]), .carry( c[35]), .a( s[20]) , .b( c[19]), .cin( s[24]) );
fulladd fa_54
( .sum( s[36]), .carry( c[36]), .a( s[21]) , .b( c[20]), .cin( s[25]) );
fulladd fa_55
( .sum( s[37]), .carry( c[37]), .a( s[22]) , .b( c[21]), .cin( s[26]) );
fulladd fa_56
( .sum( s[38]), .carry( c[38]), .a( s[23]) , .b( c[22]), .cin( s[27]) );
fulladd fa_57
( .sum( s[39]), .carry( c[39]), .a( s[14]) , .b( c[23]), .cin( s[28]) );
fulladd ha_52 ( .sum( s[40]), .carry( c[40]), .a( s[15]) , .b ( s[29]) , .cin(0) );
fulladd ha_53 ( .sum( s[41]), .carry( c[41]), .a( p[7][5]), .b ( s[30]) , .cin(0));
fulladd ha_61 ( .sum( s[42]), .carry( c[42]), .a( s[33]) , .b( c[32]) , .cin(0) );
fulladd ha_62 ( .sum( s[43]), .carry( c[43]), .a( s[34]) , .b( c[33]) , .cin(0) );
fulladd ha_63 ( .sum( s[44]), .carry( c[44]), .a( s[35]) , .b( c[34]) , .cin(0) );
fulladd fa_61
( .sum( s[45]), .carry( c[45]), .a( s[36]) , .b( c[35]), .cin( c[24]) );
fulladd fa_62
( .sum( s[46]), .carry( c[46]), .a( s[37]) , .b( c[36]), .cin( c[25]) );
fulladd fa_63
( .sum( s[47]), .carry( c[47]), .a( s[38]) , .b( c[37]), .cin( c[26]) );
fulladd fa_64
( .sum( s[48]), .carry( c[48]), .a( s[39]) , .b( c[38]), .cin( c[27]) );
fulladd fa_65
( .sum( s[49]), .carry( c[49]), .a( s[40]) , .b( c[39]), .cin( c[28]) );
fulladd fa_66
( .sum( s[50]), .carry( c[50]), .a( s[41]) , .b( c[40]), .cin( c[29]) );
fulladd fa_67
( .sum( s[51]), .carry( c[51]), .a( s[31]) , .b( c[41]), .cin( c[30]) );
fulladd ha_64 ( .sum( s[52]), .carry( c[52]), .a( p[7][7]), .b ( c[31]) , .cin(0));
fulladd ha_71 ( .sum( s[53]), .carry( c[53]), .a( s[43]) , .b( c[42]) , .cin(0) );
fulladd fa_71 ( .sum( s[54]), .carry( c[54]), .a( s[44]) , .b( c[43]), .cin
( c[53]) );
```

پاسخ 1:

```
fulladd fa_72 ( .sum( s[55]), .carry( c[55]), .a( s[45]) , .b( c[44]), .cin
( c[54]) );
fulladd fa_73 ( .sum( s[56]), .carry( c[56]), .a( s[46]) , .b( c[45]), .cin
( c[55]) );
fulladd fa_74 ( .sum( s[57]), .carry( c[57]), .a( s[47]) , .b( c[46]), .cin
( c[56]) );
fulladd fa_75 ( .sum( s[58]), .carry( c[58]), .a( s[48]) , .b( c[47]), .cin
( c[57]) );
fulladd fa_76 ( .sum( s[59]), .carry( c[59]), .a( s[49]) , .b( c[48]), .cin
( c[58]) );
fulladd fa_77 ( .sum( s[60]), .carry( c[60]), .a( s[50]) , .b( c[49]), .cin
( c[59]) );
fulladd fa_78 ( .sum( s[61]), .carry( c[61]), .a( s[51]) , .b( c[50]), .cin
( c[60]) );
fulladd fa_79 ( .sum( s[62]), .carry( c[62]), .a( s[52]) , .b( c[51]), .cin
( c[61]) );
fulladd ha_72 ( .sum( s[63]), .carry( c[63]), .a( c[52]), .b ( c[62]) , .cin(0));
always@(posedge clock, negedge clear)
if(!clear) product <= 16'b0;
else product <= {s[63 : 53],s[42],s[32],s[16],s[0],p[0][0]};
endmodule
`timescale 1ns / 1ns
module D_FlipFlop(
    input D,
    input CLK,
    output Q,
    output nQ
    );
reg D1;
D_Latch d_latch (.D(D1), .Q(Q), .nQ(nQ));
always @(posedge CLK) D1 <= D;
endmodule
`timescale 1ns / 1ns
module D_Latch(
    input D,
    output Q,
    output nQ
    );
RS_Latch rs_latch (.S(D), .R(~D), .Q(Q), .nQ(nQ));
endmodule
```

پاسخ 1:

```
`timescale 1ns / 1ns
module RS_Latch(
    input R,
    input S,
    output Q,
    output nQ
    );
assign Q = ~(R | nQ);
assign nQ = ~(S | Q);
endmodule
```

پاسخ2:

```
module Miangin(input wire [14:0] Array, output integer Miangin_Out);
   wire[7:0] Sum; reg [7:0] Array [15]; output[7:0] Miangin_Out;
   integer i;
   always @(*)
      begin
         for(i = 0; i < 15; i = i+1)
            begin   Sum = Sum + Array[i];   end
      end
   assign Miangin_Out = Sum/15;
endmodule

module Miane( input wire [7:0] Array [15], input wire CLOCK, output integer
Miane_Out );
   reg [7:0] sortArray [15];
   integer i, j, tail, temp;
   assign sortArray = Array;
   always @ (posedge CLOCK)
      begin
         // Sorting the Array _BubbleSort
         for (i = 0; i <= 14; i = i + 1)
            begin
            for(j = i + 1; j <= 14; j = j + 1)
               begin
                  if (sortArray[i] < sortArray[j])
                     begin
                        temp = sortArray[i];
                        sortArray[i] = sortArray[j];
                        sortArray[j] = temp;
                     end
               end
            end
      end
   Miane_Out = sortArray[7];
endmodule
[5:52 PM]
module array_calcs(input clk);
   reg [7:0] sum,mean,mid,mod;

   reg [7:0] data[14:0] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
   reg [7:0] occs[14:0] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

پاسخ2:

```
    integer i, j, tail, temp;
    initial begin
        tail = -1;
    end
    always@(posedge clk)
    begin
        // bubble sort array
        for (i = 0; i < = tail; i = i + 1) begin
            for(j = i + 1; j < = tail; j = j + 1) begin
                if (data[i] < data[j]) begin
                    temp    = data[i];
                    data[i] = data[j];
                    data[j] = temp;
                end
            end
        end
        for (i = 0; i <  15; i = i + 1) begin
            for(j = i+1; j <  15; j = j + 1) begin
                if (data[i] == data[j]) begin
                    occs[i] = occs[i]+ 1
                end
            end
        end
        sum = 0;
        mod = 0;
        for(i = 0;i<15;i = i+1)
            if (i<15)
            begin
                sum = sum+data[i];
                if (occs[i]>mod) begin
                    mod = occs[i]
                end
            end
        mean = sum / 15;
        mid  = data[7];
        $display("%b",sum);
    end
endmodule
```

پاسخ3:

الف) در 25ns :

A=1 , B=1

در 35ns :

C=1

در لحظه آخر هم همگی برابر با 1 **هستند.**

ب) در 25ns :

A=1

در 25+epsilon ns :

B=1

در تمامی لحظات :

C=0

در لحظه آخر هم همگی به جز C **برابر با** 1 **هستند.**

پاسخ4:

```
module aclock (
 input rst,
 input clock,
 input [1:0] H_in1,
 input [3:0] H_in0,
 input [3:0] M_in1,
 input [3:0] M_in0,
 input LD_time,
 input LD_alarm,
 input STOP_al,
 input AL_ON,
 output reg Alarm,
 output [1:0]  H_out1,
 output [3:0]  H_out0,
 output [3:0]  M_out1,
 output [3:0]  M_out0,
 output [3:0]  S_out1,
 output [3:0]  S_out0
 );
 reg clock_1s;
 reg [3:0] tmp_1s;
 reg [5:0] tmp_hour, tmp_minute, tmp_second;
 reg [1:0] c_hour1,a_hour1;
 reg [3:0] c_hour0,a_hour0;
 reg [3:0] c_min1,a_min1;
 reg [3:0] c_min0,a_min0;
 reg [3:0] c_sec1,a_sec1;
 reg [3:0] c_sec0,a_sec0;

 function [3:0] mod_10;
 input [5:0] number;
 begin
 mod_10 = (number >=50) ? 5 : ((number >= 40)? 4 :((number
>= 30)? 3 :((number >= 20)? 2 :((number >= 10)? 1 :0))));
 end
 endfunction
```

پاسخ4:

```
always @(posedge clock_1s or posedge rst )
begin
if(rst) begin
a_hour1 <= 2'b00;
a_hour0 <= 4'b0000;
a_min1 <= 4'b0000;
a_min0 <= 4'b0000;
a_sec1 <= 4'b0000;
a_sec0 <= 4'b0000;
tmp_hour <= H_in1*10 + H_in0;
tmp_minute <= M_in1*10 + M_in0;
tmp_second <= 0;
end
else begin
if(LD_alarm) begin
a_hour1 <= H_in1;
a_hour0 <= H_in0;
a_min1 <= M_in1;
a_min0 <= M_in0;
a_sec1 <= 4'b0000;
a_sec0 <= 4'b0000;
end
if(LD_time) begin
tmp_hour <= H_in1*10 + H_in0;
tmp_minute <= M_in1*10 + M_in0;
tmp_second <= 0;
end
else begin
tmp_second <= tmp_second + 1;
if(tmp_second >=59) begin
tmp_minute <= tmp_minute + 1;
tmp_second <= 0;
if(tmp_minute >=59) begin
tmp_minute <= 0;
tmp_hour <= tmp_hour + 1;
```

پاسخ4:

```
if(tmp_hour >= 24) begin
tmp_hour <= 0;
end
end
end
end
end
end
always @(posedge clock or posedge rst)
begin
if(rst)
begin
tmp_1s <= 0;
clock_1s <= 0;
end
else begin
tmp_1s <= tmp_1s + 1;
if(tmp_1s <= 5)
clock_1s <= 0;
else if (tmp_1s >= 10) begin
clock_1s <= 1;
tmp_1s <= 1;
end
else
clock_1s <= 1;
end
end
always @(*) begin
if(tmp_hour>=20) begin
c_hour1 = 2;
end
else begin
if(tmp_hour >=10)
c_hour1  = 1;
```

پاسخ4:

```
 else
 c_hour1 = 0;
 end
 c_hour0 = tmp_hour - c_hour1*10;
 c_min1 = mod_10(tmp_minute);
 c_min0 = tmp_minute - c_min1*10;
 c_sec1 = mod_10(tmp_second);
 c_sec0 = tmp_second - c_sec1*10;
 end
 assign H_out1 = c_hour1;
 assign H_out0 = c_hour0;
 assign M_out1 = c_min1;
 assign M_out0 = c_min0;
 assign S_out1 = c_sec1;
 assign S_out0 = c_sec0;

 always @(posedge clock_1s or posedge rst) begin
 if(rst)
 Alarm <=0;
 else begin

if({a_hour1,a_hour0,a_min1,a_min0,a_sec1,a_sec0}=={c_hour1
,c_hour0,c_min1,c_min0,c_sec1,c_sec0})
 begin
 if(AL_ON) Alarm <= 1;
 end
 if(STOP_al) Alarm <=0;
 end
 end

endmodule
```