«به نام خدا»

نام: فاطمه زهرا بخشنده　　استاد: دکتر میثم عبداللهی

امتحان میان ترم:

سوال اول:

```
module Mux (select, b0, b1, b2, b3, out);

    input [1:0] wire select,
    input [7:0] wire b0, b1, b2, b3,
    output [7:0] reg out

    always @ (*)
    begin
        case(select)
            2'b00 : out = b0;
            2'b01 : out = b1;
            2'b10 : out = b2;
            2'b11 : out = b3;
        endcase
    end

endmodule
```

```
module DFF_E(clock, reset, enable, d, q);

    input wire clock, reset, enable,
    input [7:0] d,
    output [7:0] wire q

    always @ (*)
    begin

        if (clock)
        begin q = d;
        end
        if (reset)
        begin q = 1'b0;
        end
        if (~enable)
        begin q = 1'bz;
        end
    end
endmodule


module DFF(clock, reset, d, q);
    input wire clock, reset,
    input [7:0] wire d,
```

```
    output [7:0] wire q
    always @ (*)
    begin
        if(clock)
        begin q = d;
        end
        if(reset)
        begin q = 1'b0;
        end
    end
endmodule


module Encoder(In, s);

    output s;
    input [3: 0]  In;

    wire b1;
    or(b1,In[0],In[1],In[2],In[3]);

    and(x,b1,s);

endmodule
```

```
module ADD(a, b, out);
    input [7:0] wire a, b,
    output [7:0] wire out;

    assign out = a + b;
endmodule


module PR(a, b, out);
    input [7:0] wire a, b,
    output [7:0] wire out;

    assign out = a * b;
endmodule


module MD(
    input c, input s, input [3:0] b,
    input a, input clk, output Out
    );

    wire w1;
    wire [7:0] w2, w3, w3Bar, w4, w5, w6;
```

```
    not (w3Bar, w3);
    Encoder E(b, w1);
    or (w2, w1, a);

    DFF D_1 (c, clk, w3);
    DFF D_2 (w2, clk, w4);
    DFF_E D1 (w4, clk, w3, w5);
    DFF_E D2 (w4, clk, w3Bar, w6);
    or (w7, w5, w6);
    and (Out, a, clk);
```

سوال دوم:

```
module Array_AVG(Avg);

        reg [7:0] Arr [14:0];
    reg[7:0] S = 0;
        output[7:0] Avg;
    integer i;
    always @(*)
        begin
        for(i = 0; i < 15; i = i+1)
        begin
            S = S + Arr[i];
                end
        end
```

```
    assign Avg = S / 15;

endmodule


module Array_Median(M);

        reg [7:0] Arr [14:0];
    reg[7:0] temp;
        output[7:0] M;
    integer i, j;
    always @(*)
        begin
        for (i = 0; i <= 15; i = i + 1)
        begin
            for(j = i + 1; j <= 15; j = j + 1)
            begin
                if (Arr[i] < Arr[j])
                begin
                    temp = Arr[i];
                    Arr[i] = Arr[j];
                    Arr[j] = temp;
                end
            end
```

```
        end
        end

    assign M = Arr[7];

endmodule


module Array_Mod(M);

        reg [7:0] Arr [14:0];
    reg [3:0] count [255:0];
    reg[7:0] max;
        output[7:0] M;
    integer i, j;
    always @(*)
        begin
        for (i = 0; i <= 15; i = i + 1)
        begin
            for(j = i + 1; j <= 15; j = j + 1)
            begin
                count [Arr[i]] = count [Arr[i]] + 1;
            end
        end
```

```
        max = Arr[0];
        for (i = 0; i <= 15; i = i + 1)
        begin
            if (count[Arr[i]] > max)
            begin
                max = count[Arr[i]];
            end
        end
        end

    assign M = max;

endmodule
```

سوال سوم:

الف)

A در زمان 25ns تغییر کرده و مقدار آن 1 می شود.

B در زمان 25ns تغییر کرده و مقدار آن 1 می شود.

C در زمان 35ns تغییر کرده و مقدار آن 1 می شود.

ب)

A در زمان 25ns تغییر کرده و مقدار آن 1 می شود.

B تقریبا در زمان 20ns تغییر کرده و مقدار آن 1 می شود.

C تغییر نمی کند و مقدار آن 0 می ماند.

سوال چهارم:

```
module clock
(
    input wire clk, reset, set_clock, stop_alarm,
    input [7:0] wire in_min,
    input [3:0] wire in_hour,
    output reg alarm_on,
    output [7:0] reg min,
    output [3:0] reg hour
);

    reg [7:0] alarm_min;
    reg [4:0] alarm_hour;
    always @(*)
        begin
        if (reset) begin
            min = 8'h00;
            hour = 4'h0;
        end

        if (set_clock)
        begin
```

```
    min = in_min;
    hour = in_hour;
end

if (set_alarm)
begin
    alarm_min = in_min;
    alarm_hour = in_hour;
end

if (clk)
begin
    min[7:4] = min[7:4] + 4'b0001;
    min[3:0] = min[3:0] + 4'b0001;
    hour[3:2] = hour[3:2] + 2'b01;
    hour[1:0] = hour[1:0] + 2'b01;

    if (alarm_hour == hour && alarm_min == min) begin
        alarm_on = 1;
    end

    if (alarm_on & stop_alarm)
        begin alarm_on = 0;
    end
end
```

```
    end

endmodule
```