```
`timescale 1ns / 1ps //ソース3:FPGA回路
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    00:53:02 12/05/2013
// Design Name:
// Module Name:    system
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module system(
    input CLK,
    input RESET,
    input STOP,
    inout PIN,
    input START,
    output [11:0] SEG,
    output [7:0] LED
   );

reg ff;
reg [16:0] temp_count;

reg DATA;

    wire [3:0] digit_0;
    wire [3:0] digit_1;
    wire [3:0] digit_2;
    wire [3:0] digit_3;

    Calculater1 display (
        .CLK(CLK),
        .RESET(RESET),
        .DIGIT_0(digit_0),
        .DIGIT_1(digit_1),
        .DIGIT_2(digit_2),
        .DIGIT_3(digit_3),
        .SEG(SEG)
        );

//state unit

reg [3:0]state;
always @(posedge CLK or posedge RESET)
begin
    if(RESET)
        state[3:0]=4'b0000;
    else if(STOP)
        state[3:0]=4'b0011;
    else
    case (state)
        4'b0000:if (START) state[3:0]=4'b0001;
        4'b0001:if (temp_count==249) state[3:0]=4'b0010;
    endcase
end

//count 5us
always @(posedge CLK)
begin
    if(state[3:0]!=4'b0001)
        temp_count<=0;
    else if (temp_count==249)
        temp_count<=0;
    else
        temp_count<=temp_count+1;
```

```verilog

end

//measure echo

reg [23:0]echo;
reg [23:0]result;

always @(posedge CLK)
begin
    if(state[3:0]!=4'b0010)
    begin
        echo=0;
        result[23:0]=echo[23:0];
    end
    else if(echo>925000||PIN==1'b0)
    begin
        result[23:0]=echo[23:0];
    end
    else if(PIN==1'b1)
        echo=echo+1;
end

always @(state)
begin
    case(state)
        4'b0000:ff<=0;//state1:RESET
        4'b0001:ff<=1;//state2:5us pulse is emitted
        4'b0010:ff<=0;//state3:waiting&output result
        4'b0011:ff<=0;//state4:stop
    endcase
end
    assign PIN = (state[3:0]!=4'b0010)?ff:1'bz;

    assign digit_0=result[3:0];
    assign digit_1=result[7:4];
    assign digit_2=result[11:8];
    assign digit_3=result[15:12];

    assign LED[7:0]=result[23:16];

endmodule
```

```
`timescale 1ns / 1ps//ソース 4:出力用回路(情報工学実験 2 より)
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    13:43:17 05/17/2013
// Design Name:
// Module Name:    Calclater1
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Calculater1(
    input CLK,
    input RESET,
    input [3:0] DIGIT_0,
    input [3:0] DIGIT_1,
    input [3:0] DIGIT_2,
    input [3:0] DIGIT_3,
    output [11:0] SEG
   );

    reg [7:0] div_counter;
    reg [1:0] counter;
    reg [3:0] sel;
    reg [3:0] ff;
    reg [7:0] temp_seg;

    always @(posedge CLK)
    begin
        if(RESET)
            div_counter <= 0;
        else
            div_counter <= div_counter + 1;
    end

    always @(posedge CLK)
    begin
        if(RESET)
            counter <= 0;
        else
            if(div_counter==8'b11111111)
                counter <= counter + 1;
    end


    always @(posedge CLK)
    begin
        if(RESET)
            ff <= 0;
        else
            case(counter)
                2'b00: ff <= DIGIT_3;
                2'b01: ff <= DIGIT_2;
                2'b10: ff <= DIGIT_1;
                2'b11: ff <= DIGIT_0;
            endcase
    end

    always @(posedge CLK)
    begin
        if(RESET)
            sel <= 0;
        else
            case(counter)
                2'b00: sel <= 4'b1110;
                2'b01: sel <= 4'b1101;
                2'b10: sel <= 4'b1011;
                2'b11: sel <= 4'b0111;
            endcase
    end

    always @(posedge CLK)
    begin
        if(RESET)
            temp_seg <= 0;
        else
            case(ff)
                4'b0000:temp_seg <= 8'b1_1000000; // 0
                4'b0001:temp_seg <= 8'b1_1111001; // 1
                4'b0010:temp_seg <= 8'b1_0100100; // 2
                4'b0011:temp_seg <= 8'b1_0110000; // 3
                4'b0100:temp_seg <= 8'b1_0011001; // 4
                4'b0101:temp_seg <= 8'b1_0010010; // 5
                4'b0110:temp_seg <= 8'b1_0000010; // 6
                4'b0111:temp_seg <= 8'b1_1011000; // 7
                4'b1000:temp_seg <= 8'b1_0000000; // 8
                4'b1001:temp_seg <= 8'b1_0010000; // 9
                4'b1010:temp_seg <= 8'b1_0001000; // A
                4'b1011:temp_seg <= 8'b1_0000011; // b
                4'b1100:temp_seg <= 8'b1_1000110; // c
                4'b1101:temp_seg <= 8'b1_0100001; // d
                4'b1110:temp_seg <= 8'b1_0000110; // E
                4'b1111:temp_seg <= 8'b1_0001110; // F
            endcase
    end

    assign SEG = {sel, temp_seg};

endmodule
```