```
`timescale 1ns / 1ps //ソース 3:FPGA 回路
 1:
      2:
 3:
      // Company:
 4:
      // Engineer:
      //
5:
      // Create Date:
 6:
                       00:53:02 12/05/2013
 7:
      // Design Name:
8:
      // Module Name:
                       system
9:
      // Project Name:
      // Target Devices:
10:
      // Tool versions:
11:
      // Description:
12:
      //
13:
14:
      // Dependencies:
15:
      //
      // Revision:
16:
      // Revision 0.01 - File Created
17:
18:
      // Additional Comments:
19:
      //
20:
      21:
      module system(
         input CLK,
22:
23:
          input RESET,
24:
         input STOP,
25:
          inout PIN,
26:
          input START,
27:
          output [11:0] SEG,
28:
         output [7:0] LED
29:
         );
30:
31:
      reg ff;
32:
      reg [16:0] temp count;
33:
34:
     reg DATA;
35:
36:
          wire [3:0] digit_0;
         wire [3:0] digit_1;
37:
         wire [3:0] digit_2;
38:
39:
         wire [3:0] digit_3;
40:
41:
         Calculater1 display (
42:
             .CLK (CLK),
43:
              .RESET (RESET),
44:
             .DIGIT 0(digit 0),
45:
             .DIGIT 1 (digit 1),
46:
             .DIGIT_2(digit_2),
47:
              .DIGIT_3(digit_3),
48:
              .SEG (SEG)
49:
             );
50:
51:
     //state unit
52:
53:
      reg [3:0]state;
      always @(posedge CLK or posedge RESET)
54:
55:
      begin
56:
         if(RESET)
             state[3:0]=4'b0000;
57:
58:
          else if (STOP)
59:
             state[3:0]=4'b0011;
60:
          else
61:
          case (state)
62:
             4'b0000:if (START) state[3:0]=4'b0001;
             4'b0001:if (temp_count==249) state[3:0]=4'b0010;
63:
64:
          endcase
      end
65:
66:
67:
      //count 5us
68:
      always @(posedge CLK)
69:
      begin
70:
          if(state[3:0]!=4'b0001)
71:
             temp_count<=0;
          else if (temp_count==249)
72:
73:
             temp_count<=0;
74:
          else
75:
             temp count <= temp count +1;
```

```
76:
 77:
       end
 78:
 79:
       //measure echo
 80:
      reg [23:0]echo;
 81:
 82:
       reg [23:0]result;
 83:
 84:
        always @(posedge CLK)
 85:
      begin
 86:
           if(state[3:0]!=4'b0010)
 87:
           begin
 88:
                echo=0;
                result[23:0]=echo[23:0];
 89:
 90:
           end
 91:
            else if(echo>925000||PIN==1'b0)
 92:
           begin
 93:
            result[23:0]=echo[23:0];
 94:
            end
            else if(PIN==1'b1)
 95:
 96:
               echo=echo+1;
 97:
     end
 98:
 99:
       always @(state)
100:
       begin
101:
           case(state)
102:
                4'b0000:ff<=0;//state1:RESET
103:
                4'b0001:ff<=1;//state2:5us pulse is emitted
104:
                4'b0010:ff<=0;//state3:waiting&output result
105:
                4'b0011:ff<=0;//state4:stop
106:
           endcase
107:
      end
108:
            assign PIN = (state[3:0]!=4'b0010)?ff:1'bz;
109:
           assign digit_0=result[3:0];
assign digit_1=result[7:4];
110:
111:
           assign digit_2=result[11:8];
112:
113:
           assign digit_3=result[15:12];
114:
115:
            assign LED[7:0]=result[23:16];
116:
117:
       endmodule
```

```
`timescale 1ns / 1ps//ソース 4:出力用回路(情報工学実験 2 より)
 1:
      2:
 3:
      // Company:
 4:
      // Engineer:
5:
      //
      // Create Date:
 6:
                       13:43:17 05/17/2013
 7:
      // Design Name:
8:
      // Module Name:
                       Calclater1
9:
      // Project Name:
      // Target Devices:
10:
11:
      // Tool versions:
      // Description:
12:
      //
13:
14:
      // Dependencies:
15:
      //
      // Revision:
16:
      // Revision 0.01 - File Created
17:
18:
      // Additional Comments:
19:
      //
20:
      21:
      module Calculater1(
22:
         input CLK,
23:
          input RESET,
24:
         input [3:0] DIGIT_0,
25:
          input [3:0] DIGIT_1,
          input [3:0] DIGIT_
input [3:0] DIGIT_
26:
27:
28:
         output [11:0] SEG
29:
         );
30:
31:
          reg [7:0] div_counter;
32:
          reg [1:0] counter;
          reg [3:0] sel;
33:
34:
          reg [3:0] ff;
35:
          reg [7:0] temp seg;
36:
37:
          always @(posedge CLK)
38:
          begin
39:
             if (RESET)
40:
                 div counter <= 0;
41:
              else
42:
                 div counter <= div counter + 1;
43:
          end
44:
45:
          always @ (posedge CLK)
46:
          begin
47:
              if (RESET)
48:
                 counter <= 0;
49:
              else
                 if(div counter==8'b11111111)
50:
51:
                     counter <= counter + 1;</pre>
52:
          end
53:
54:
55:
          always @(posedge CLK)
56:
          begin
57:
             if (RESET)
58:
                 ff <= 0;
59:
              else
60:
                 case (counter)
                     2'b00: ff <= DIGIT 3;
61:
                     2'b01: ff <= DIGIT_2;
62:
63:
                     2'b10: ff <= DIGIT 1;
64:
                     2'b11: ff <= DIGIT 0;
65:
                 endcase
66:
          end
67:
68:
          always @(posedge CLK)
69:
          begin
70:
              if (RESET)
71:
                 sel <= 0;
72:
73:
                 case(counter)
74:
                     2'b00: sel <= 4'b1110;
75:
                     2'b01: sel <= 4'b1101;
```

```
76:
                            2'b10: sel <= 4'b1011;
 77:
                            2'b11: sel <= 4'b0111;
 78:
                        endcase
 79:
              end
 80:
              always @(posedge CLK)
 81:
 82:
              begin
 83:
                  if(RESET)
 84:
                       temp_seg <= 0;</pre>
 85:
                   else
 86:
                        case(ff)
                            4'b0000:temp_seg <= 8'b1_1000000; // 0
 87:
                            4'b0001:temp_seg <= 8'b1_1111001; // 1
4'b0010:temp_seg <= 8'b1_0100100; // 2
 88:
 89:
                            4'b0011:temp_seg <= 8'b1_0110000; // 3
 90:
                            4'b0100:temp_seg <= 8'b1_0011001; // 4
 91:
                            4'b0101:temp seg <= 8'b1 0010010; // 5
 92:
 93:
                            4'b0110:temp_seg <= 8'b1_0000010; // 6
                            4'b0111:temp_seg <= 8'b1_1011000; // 7
4'b1000:temp_seg <= 8'b1_0000000; // 8
 94:
 95:
 96:
                            4'b1001:temp_seg <= 8'b1_0010000; // 9
                            4'b1010:temp_seg <= 8'b1_0001000; // A
 97:
 98:
                            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
 99:
100:
101:
                            4'b1111:temp_seg <= 8'b1_0001110; // F
102:
                        endcase
103:
104:
              end
105:
106:
              assign SEG = {sel, temp seg};
107:
108:
      endmodule
```