```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2024
// Design Name:
// Module Name: ClkGen
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module ClkGen(
    input DATA,
    input SCLK,
    input SET,
    input RESET,
    input CLK,
    output OUT,
    output led2
    );

//---DDS
    reg [23:0] SRO;
    reg [23:0] LAO;
    wire [23:0] SUMO;
    reg [23:0] SUMLO;
    wire [7:0] LUTO;
    reg [7:0] LUTLO;

    reg     P_SCLK;     // previous
    reg     O_SCLK;     // OnePulse
    reg     P_SET;
    reg     O_SET;

   // set  DATA to shift register
   always @(posedge CLK) begin
      if(SCLK && !P_SCLK)
         O_SCLK <= 1'b1;
      else
         O_SCLK <= 1'b0;
      P_SCLK <= SCLK;
   end

   always @(posedge CLK or negedge RESET) begin
     if(RESET == 1'b0)
       SRO     <= 24'h0;
     else if(O_SCLK) begin
       SRO     <= SRO >> 1;
       SRO[23] <= DATA;
     end
   end

   // set shift register output to latch
   always @(posedge CLK) begin
     if(SET && !P_SET)
       O_SET <= 1'b1;
     else
       O_SET <= 1'b0;
     P_SET <= SET;
   end

   always @(posedge CLK or negedge RESET) begin
     if(RESET == 1'b0)
       LAO <= 24'h0;
     else if(O_SET)
```

```
    LA0 <= SR0;
end

// sum
assign SUM0 = SUML0 + LA0;

// clock:12MHz
// 24bits latch
always @(posedge CLK or negedge RESET) begin
  if (RESET == 1'b0)
    SUML0 <= 24'h0;
  else
    SUML0 <= SUM0;
end

// upper 8bits latch
always @(posedge CLK or negedge RESET) begin
  if(RESET == 1'b0)
    LUTL0 <= 8'b0;
  else
    LUTL0 <= LUT0;          // --- rom output
end

assign OUT = LUTL0[7];     // --- rom output only upper bit
assign led2 = LUTL0[7];     // --- rom output only upper bit


// function name : rom8
// output bits width : 8
// formal argument : addr8 : 8bits
function     [7:0]   rom8;
input        [7:0]   addr8;
  case(addr8)
    8'd0  : rom8 = 8'b10000000;  8'd1  : rom8 = 8'b10000011;  8'd2  : rom8 = 8'b10000110;  8'd3  : rom8 = 8'b10001001;
    8'd4  : rom8 = 8'b10001101;  8'd5  : rom8 = 8'b10010000;  8'd6  : rom8 = 8'b10010011;  8'd7  : rom8 = 8'b10010110;
    8'd8  : rom8 = 8'b10011001;  8'd9  : rom8 = 8'b10011100;  8'd10 : rom8 = 8'b10011111;  8'd11 : rom8 = 8'b10100010;
    8'd12 : rom8 = 8'b10100101;  8'd13 : rom8 = 8'b10101000;  8'd14 : rom8 = 8'b10101011;  8'd15 : rom8 = 8'b10101110;
    8'd16 : rom8 = 8'b10110001;  8'd17 : rom8 = 8'b10110100;  8'd18 : rom8 = 8'b10110111;  8'd19 : rom8 = 8'b10111010;
    8'd20 : rom8 = 8'b10111100;  8'd21 : rom8 = 8'b10111111;  8'd22 : rom8 = 8'b11000010;  8'd23 : rom8 = 8'b11000100;
    8'd24 : rom8 = 8'b11000111;  8'd25 : rom8 = 8'b11001010;  8'd26 : rom8 = 8'b11001100;  8'd27 : rom8 = 8'b11001111;
    8'd28 : rom8 = 8'b11010001;  8'd29 : rom8 = 8'b11010100;  8'd30 : rom8 = 8'b11010110;  8'd31 : rom8 = 8'b11011000;

    8'd32 : rom8 = 8'b11011011;  8'd33 : rom8 = 8'b11011101;  8'd34 : rom8 = 8'b11011111;  8'd35 : rom8 = 8'b11100001;
    8'd36 : rom8 = 8'b11100011;  8'd37 : rom8 = 8'b11100101;  8'd38 : rom8 = 8'b11100111;  8'd39 : rom8 = 8'b11101001;
    8'd40 : rom8 = 8'b11101010;  8'd41 : rom8 = 8'b11101100;  8'd42 : rom8 = 8'b11101110;  8'd43 : rom8 = 8'b11101111;
    8'd44 : rom8 = 8'b11110001;  8'd45 : rom8 = 8'b11110010;  8'd46 : rom8 = 8'b11110100;  8'd47 : rom8 = 8'b11110101;
    8'd48 : rom8 = 8'b11110110;  8'd49 : rom8 = 8'b11110111;  8'd50 : rom8 = 8'b11111001;  8'd51 : rom8 = 8'b11111010;
    8'd52 : rom8 = 8'b11111010;  8'd53 : rom8 = 8'b11111011;  8'd54 : rom8 = 8'b11111100;  8'd55 : rom8 = 8'b11111101;
    8'd56 : rom8 = 8'b11111110;  8'd57 : rom8 = 8'b11111110;  8'd58 : rom8 = 8'b11111111;  8'd59 : rom8 = 8'b11111111;
    8'd60 : rom8 = 8'b11111111;  8'd61 : rom8 = 8'b11111111;  8'd62 : rom8 = 8'b11111111;  8'd63 : rom8 = 8'b11111111;

    8'd64 : rom8 = 8'b11111111;  8'd65 : rom8 = 8'b11111111;  8'd66 : rom8 = 8'b11111111;  8'd67 : rom8 = 8'b11111111;
    8'd68 : rom8 = 8'b11111111;  8'd69 : rom8 = 8'b11111111;  8'd70 : rom8 = 8'b11111111;  8'd71 : rom8 = 8'b11111110;
    8'd72 : rom8 = 8'b11111110;  8'd73 : rom8 = 8'b11111101;  8'd74 : rom8 = 8'b11111100;  8'd75 : rom8 = 8'b11111011;
    8'd76 : rom8 = 8'b11111010;  8'd77 : rom8 = 8'b11111010;  8'd78 : rom8 = 8'b11111001;  8'd79 : rom8 = 8'b11110111;
    8'd80 : rom8 = 8'b11110110;  8'd81 : rom8 = 8'b11110101;  8'd82 : rom8 = 8'b11110100;  8'd83 : rom8 = 8'b11110010;
    8'd84 : rom8 = 8'b11110001;  8'd85 : rom8 = 8'b11101111;  8'd86 : rom8 = 8'b11101110;  8'd87 : rom8 = 8'b11101100;
    8'd88 : rom8 = 8'b11101010;  8'd89 : rom8 = 8'b11101001;  8'd90 : rom8 = 8'b11100111;  8'd91 : rom8 = 8'b11100101;
    8'd92 : rom8 = 8'b11100011;  8'd93 : rom8 = 8'b11100001;  8'd94 : rom8 = 8'b11011111;  8'd95 : rom8 = 8'b11011101;

    8'd96 : rom8 = 8'b11011011;  8'd97 : rom8 = 8'b11011000;  8'd98 : rom8 = 8'b11010110;  8'd99 : rom8 = 8'b11010100;
    8'd100: rom8 = 8'b11010001;  8'd101: rom8 = 8'b11001111;  8'd102: rom8 = 8'b11001100;  8'd103: rom8 = 8'b11001010;
    8'd104: rom8 = 8'b11000111;  8'd105: rom8 = 8'b11000100;  8'd106: rom8 = 8'b11000010;  8'd107: rom8 = 8'b10111111;
    8'd108: rom8 = 8'b10111100;  8'd109: rom8 = 8'b10111010;  8'd110: rom8 = 8'b10110111;  8'd111: rom8 = 8'b10110100;
    8'd112: rom8 = 8'b10110001;  8'd113: rom8 = 8'b10101110;  8'd114: rom8 = 8'b10101011;  8'd115: rom8 = 8'b10101000;
    8'd116: rom8 = 8'b10100101;  8'd117: rom8 = 8'b10100010;  8'd118: rom8 = 8'b10011111;  8'd119: rom8 = 8'b10011100;
    8'd120: rom8 = 8'b10011001;  8'd121: rom8 = 8'b10010110;  8'd122: rom8 = 8'b10010011;  8'd123: rom8 = 8'b10010000;
    8'd124: rom8 = 8'b10001101;  8'd125: rom8 = 8'b10001001;  8'd126: rom8 = 8'b10000110;  8'd127: rom8 = 8'b10000011;

    8'd128: rom8 = 8'b10000000;  8'd129: rom8 = 8'b01111101;  8'd130: rom8 = 8'b01111010;  8'd131: rom8 = 8'b01110111;
    8'd132: rom8 = 8'b01110011;  8'd133: rom8 = 8'b01110000;  8'd134: rom8 = 8'b01101101;  8'd135: rom8 = 8'b01101010;
    8'd136: rom8 = 8'b01100111;  8'd137: rom8 = 8'b01100100;  8'd138: rom8 = 8'b01100001;  8'd139: rom8 = 8'b01011110;
    8'd140: rom8 = 8'b01011011;  8'd141: rom8 = 8'b01011000;  8'd142: rom8 = 8'b01010101;  8'd143: rom8 = 8'b01010010;
    8'd144: rom8 = 8'b01001111;  8'd145: rom8 = 8'b01001100;  8'd146: rom8 = 8'b01001001;  8'd147: rom8 = 8'b01000110;
    8'd148: rom8 = 8'b01000100;  8'd149: rom8 = 8'b01000001;  8'd150: rom8 = 8'b00111110;  8'd151: rom8 = 8'b00111100;
    8'd152: rom8 = 8'b00111001;  8'd153: rom8 = 8'b00110110;  8'd154: rom8 = 8'b00110100;  8'd155: rom8 = 8'b00110001;
```

```
        8'd156: rom8 = 8'b00101111;  8'd157: rom8 = 8'b00101100;  8'd158: rom8 = 8'b00101010;  8'd159: rom8 = 8'b00101000;

        8'd160: rom8 = 8'b00100101;  8'd161: rom8 = 8'b00100011;  8'd162: rom8 = 8'b00100001;  8'd163: rom8 = 8'b00011111;
        8'd164: rom8 = 8'b00011101;  8'd165: rom8 = 8'b00011011;  8'd166: rom8 = 8'b00011001;  8'd167: rom8 = 8'b00010111;
        8'd168: rom8 = 8'b00010110;  8'd169: rom8 = 8'b00010100;  8'd170: rom8 = 8'b00010010;  8'd171: rom8 = 8'b00010001;
        8'd172: rom8 = 8'b00001111;  8'd173: rom8 = 8'b00001110;  8'd174: rom8 = 8'b00001100;  8'd175: rom8 = 8'b00001011;
        8'd176: rom8 = 8'b00001010;  8'd177: rom8 = 8'b00001001;  8'd178: rom8 = 8'b00000111;  8'd179: rom8 = 8'b00000110;
        8'd180: rom8 = 8'b00000110;  8'd181: rom8 = 8'b00000101;  8'd182: rom8 = 8'b00000100;  8'd183: rom8 = 8'b00000011;
        8'd184: rom8 = 8'b00000010;  8'd185: rom8 = 8'b00000010;  8'd186: rom8 = 8'b00000001;  8'd187: rom8 = 8'b00000001;
        8'd188: rom8 = 8'b00000001;  8'd189: rom8 = 8'b00000000;  8'd190: rom8 = 8'b00000000;  8'd191: rom8 = 8'b00000000;

        8'd192: rom8 = 8'b00000000;  8'd193: rom8 = 8'b00000000;  8'd194: rom8 = 8'b00000000;  8'd195: rom8 = 8'b00000000;
        8'd196: rom8 = 8'b00000001;  8'd197: rom8 = 8'b00000001;  8'd198: rom8 = 8'b00000001;  8'd199: rom8 = 8'b00000010;
        8'd200: rom8 = 8'b00000010;  8'd201: rom8 = 8'b00000011;  8'd202: rom8 = 8'b00000100;  8'd203: rom8 = 8'b00000101;
        8'd204: rom8 = 8'b00000110;  8'd205: rom8 = 8'b00000110;  8'd206: rom8 = 8'b00000111;  8'd207: rom8 = 8'b00001001;
        8'd208: rom8 = 8'b00001010;  8'd209: rom8 = 8'b00001011;  8'd210: rom8 = 8'b00001100;  8'd211: rom8 = 8'b00001110;
        8'd212: rom8 = 8'b00001111;  8'd213: rom8 = 8'b00010001;  8'd214: rom8 = 8'b00010010;  8'd215: rom8 = 8'b00010100;
        8'd216: rom8 = 8'b00010110;  8'd217: rom8 = 8'b00010111;  8'd218: rom8 = 8'b00011001;  8'd219: rom8 = 8'b00011011;
        8'd220: rom8 = 8'b00011101;  8'd221: rom8 = 8'b00011111;  8'd222: rom8 = 8'b00100001;  8'd223: rom8 = 8'b00100011;

        8'd224: rom8 = 8'b00100101;  8'd225: rom8 = 8'b00101000;  8'd226: rom8 = 8'b00101010;  8'd227: rom8 = 8'b00101100;
        8'd228: rom8 = 8'b00101111;  8'd229: rom8 = 8'b00110001;  8'd230: rom8 = 8'b00110100;  8'd231: rom8 = 8'b00110110;
        8'd232: rom8 = 8'b00111001;  8'd233: rom8 = 8'b00111100;  8'd234: rom8 = 8'b00111110;  8'd235: rom8 = 8'b01000001;
        8'd236: rom8 = 8'b01000100;  8'd237: rom8 = 8'b01000110;  8'd238: rom8 = 8'b01001001;  8'd239: rom8 = 8'b01001100;
        8'd240: rom8 = 8'b01001111;  8'd241: rom8 = 8'b01010010;  8'd242: rom8 = 8'b01010101;  8'd243: rom8 = 8'b01011000;
        8'd244: rom8 = 8'b01011011;  8'd245: rom8 = 8'b01011110;  8'd246: rom8 = 8'b01100001;  8'd247: rom8 = 8'b01100100;
        8'd248: rom8 = 8'b01100111;  8'd249: rom8 = 8'b01101010;  8'd250: rom8 = 8'b01101101;  8'd251: rom8 = 8'b01110000;
        8'd252: rom8 = 8'b01110011;  8'd253: rom8 = 8'b01110111;  8'd254: rom8 = 8'b01111010;  8'd255: rom8 = 8'b01111101;

      endcase
    endfunction

    // formal argument:upper 8bits of SUMLO
    assign LUTO = rom8(SUMLO[23:16]);


endmodule
```