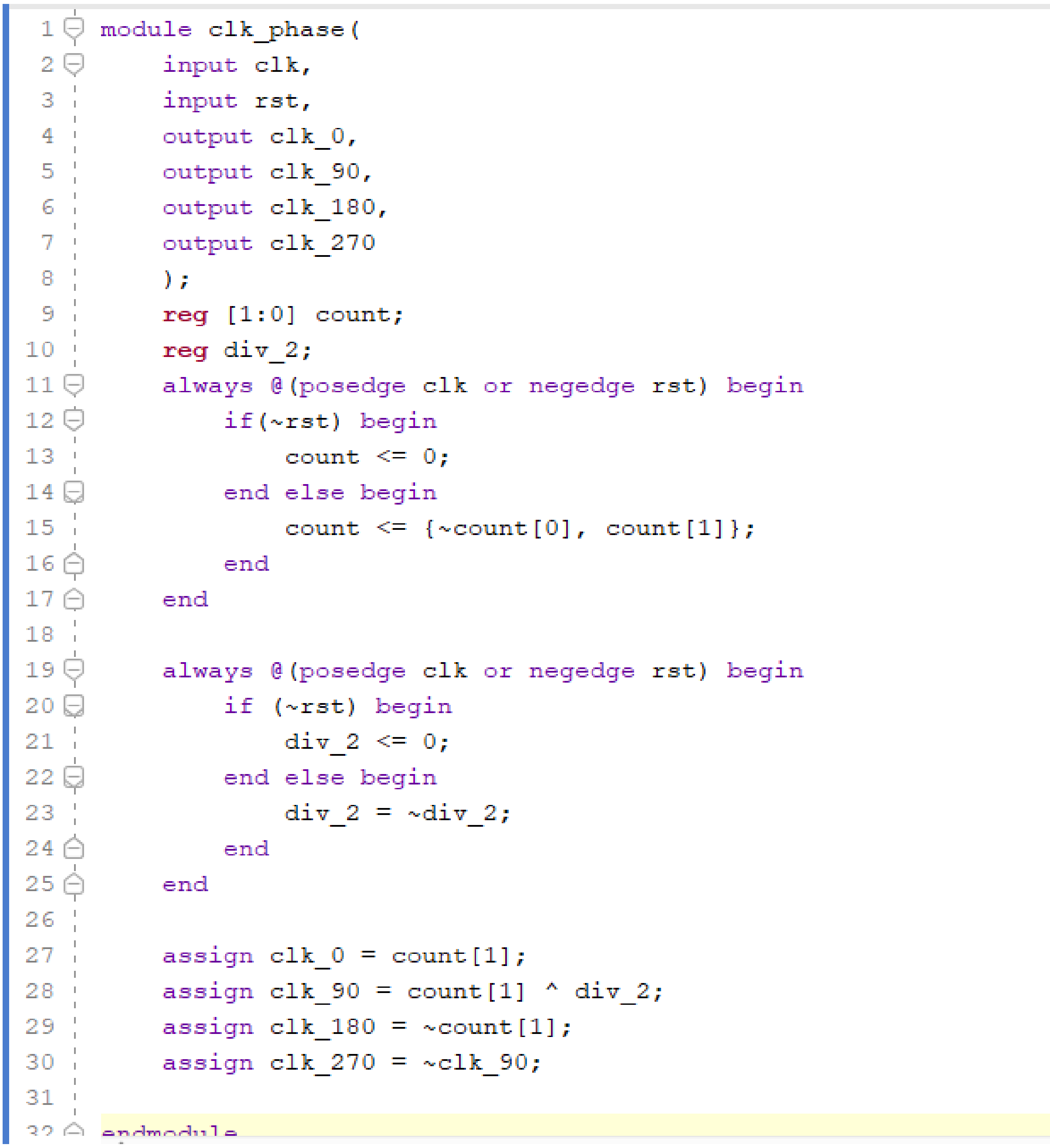
DAY-89
#100DAYSOFRTL

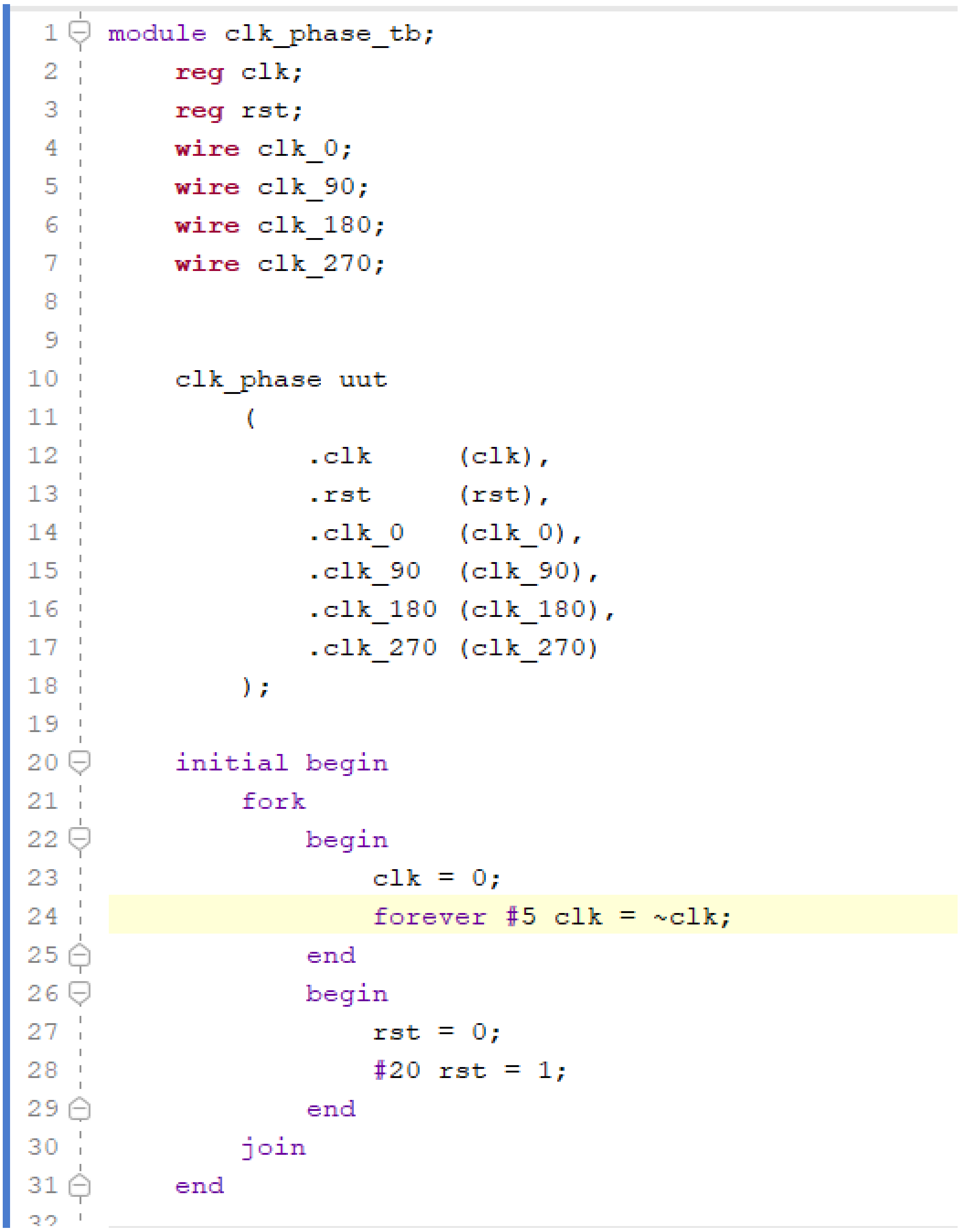
AIM:--- IMPLEMENTATION OF CLOCK PHASING

VERILOG CODE:--



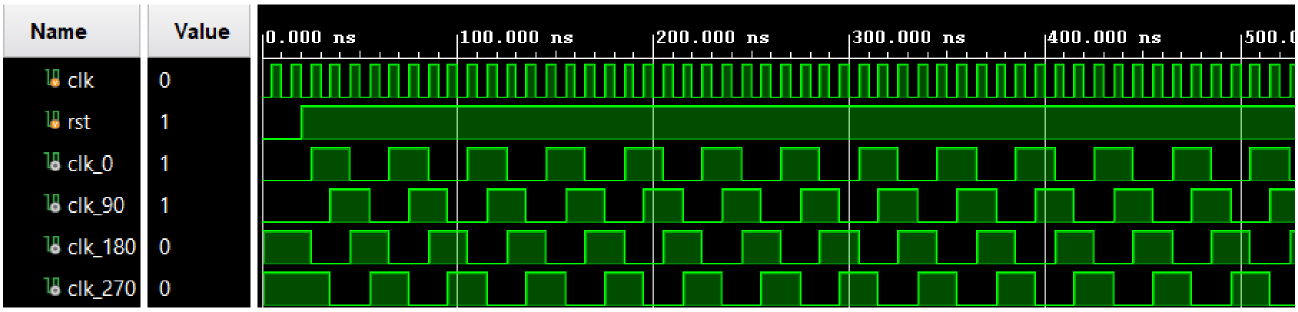
```
module clk_phase(
    input clk,
    input rst,
    output clk_0,
    output clk_90,
    output clk_180,
    output clk_270
    );
    reg [1:0] count;
    reg div_2;
    always @(posedge clk or negedge rst) begin
        if(~rst) begin
            count <= 0;
        end else begin
            count <= {~count[0], count[1]};
        end
    end

    always @(posedge clk or negedge rst) begin
        if (~rst) begin
            div_2 <= 0;
        end else begin
            div_2 = ~div_2;
        end
    end

    assign clk_0 = count[1];
    assign clk_90 = count[1] ^ div_2;
    assign clk_180 = ~count[1];
    assign clk_270 = ~clk_90;

endmodule
```

TESTBENCH CODE:---



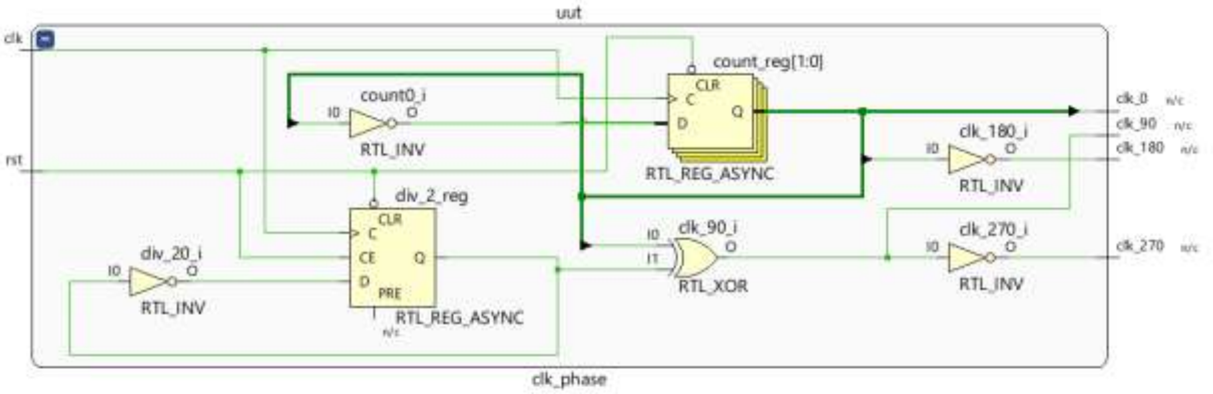
```
module clk_phase_tb;
    reg clk;
    reg rst;
    wire clk_0;
    wire clk_90;
    wire clk_180;
    wire clk_270;


    clk_phase uut
        (
            .clk     (clk),
            .rst     (rst),
            .clk_0   (clk_0),
            .clk_90  (clk_90),
            .clk_180 (clk_180),
            .clk_270 (clk_270)
        );

    initial begin
        fork
            begin
                clk = 0;
                forever #5 clk = ~clk;
            end
            begin
                rst = 0;
                #20 rst = 1;
            end
        join
    end
```

WAVEFORM:-----



Name	Value
clk	0
rst	1
clk_0	1
clk_90	1
clk_180	0
clk_270	0

SCHEMATIC BLOCK :-------



uut
clk
rst
count_reg[1:0]
CLR
C
D
Q
RTL_REG_ASYNC
count0_i
I0
O
RTL_INV
div_2_reg
CLR
C
CE
Q
D
PRE
RTL_REG_ASYNC
n/c
div_20_i
I0
O
RTL_INV
clk_90_i
I0
I1
O
RTL_XOR
clk_180_i
I0
O
RTL_INV
clk_270_i
I0
O
RTL_INV
clk_0
n/c
clk_90
n/c
clk_180
n/c
clk_270
n/c
clk_phase