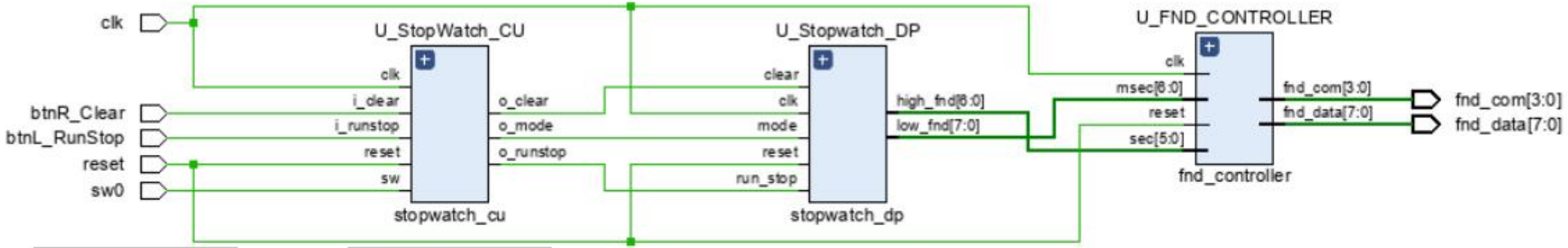
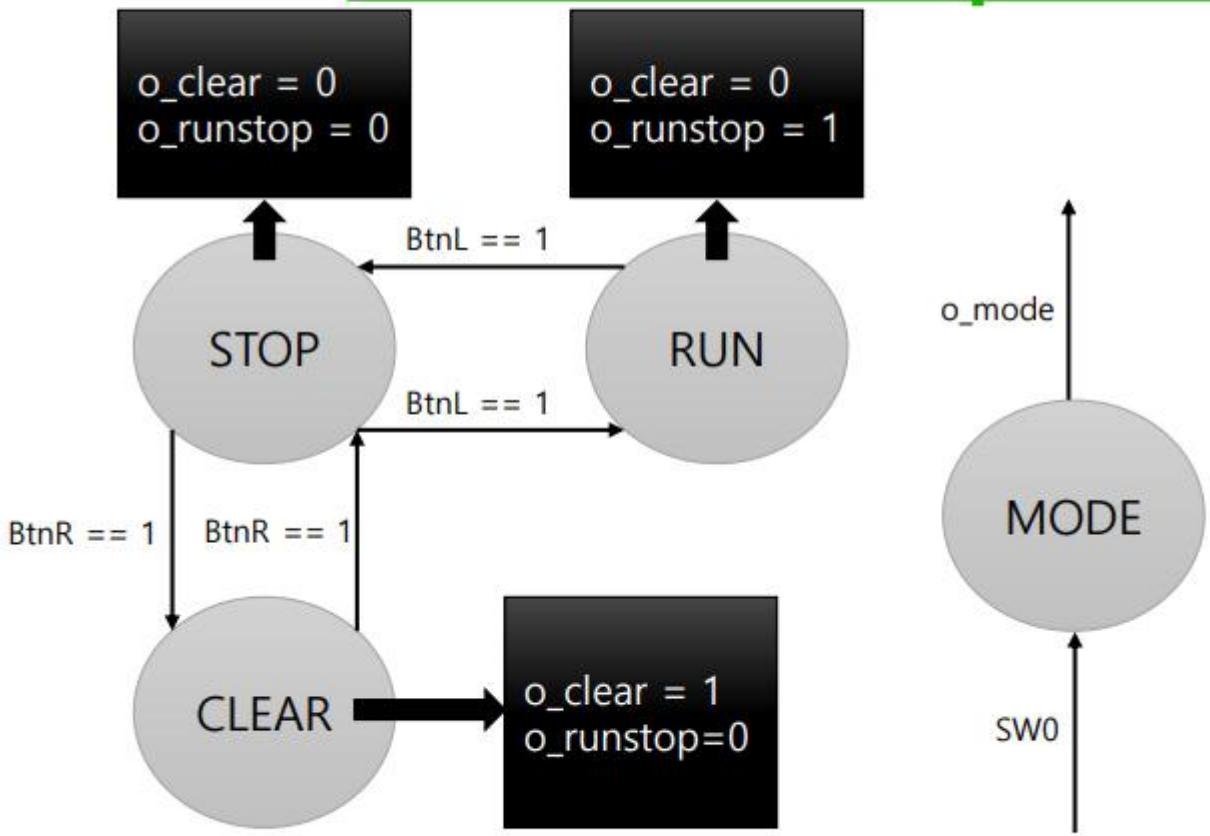
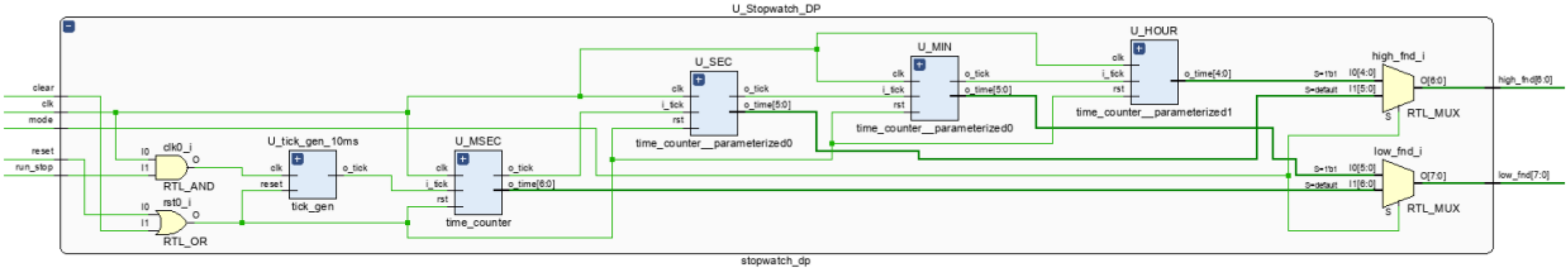




U_StopWatch_CU

```
assign o_clear = (state_reg == CLEAR)?1:0;
assign o_runstop = (state_reg == RUN)?1:0;
assign o_mode = (sw)?1:0;

always @(posedge clk, posedge reset) begin
    if(reset) begin state_reg <= STOP; end
    else begin state_reg <= next_state; end
end

always @(*) begin
    next_state = state_reg;
    case (state_reg)
        STOP:
            if (i_clear) begin
                next_state = CLEAR;
            end else if (i_runstop) begin
                next_state = RUN;
            end else begin
                next_state = state_reg;
            end
        RUN:
            if (i_runstop) begin
                next_state = STOP;
            end else begin
                next_state = state_reg;
            end
        CLEAR:
            if (i_clear) begin
                next_state = STOP;
            end else begin
                next_state = state_reg;
            end
    endcase
end
```

Sw에 따라 o_mode를 출력



U_StopWatch_DP

high_fnd — hour, sec

low_fnd — min, msec

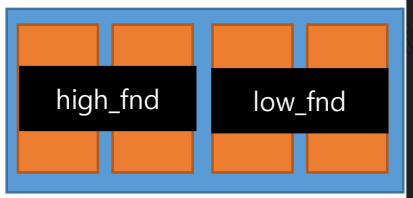
```
assign low_fnd = (mode)?min:msec;
assign high_fnd = (mode)?hour:sec;

tick_gen U_tick_gen_10ms( // 10ms 생성
    .clk(clk & run_stop),
    .reset(reset|clear),
    .o_tick(w_msec_tick)
);

time_counter #(.TICK_COUNT(100)) U_MSEC (
    .clk(clk),
    .rst(reset|clear),
    .i_tick(w_msec_tick),
    .o_time(msec),
    .o_tick(w_sec_tick)
);

time_counter #(.TICK_COUNT(60)) U_SEC (
    .clk(clk),
    .rst(reset|clear),
    .i_tick(w_sec_tick),
    .o_time(sec),
    .o_tick(w_min_tick)
);

time_counter #(.TICK_COUNT(60)) U_MIN (
    .clk(clk),
    .rst(reset|clear),
    .i_tick(w_min_tick),
    .o_time(min),
    .o_tick(w_hour_tick)
);

time_counter #(.TICK_COUNT(24)) U_HOUR (
    .clk(clk),
    .rst(reset|clear),
    .i_tick(w_hour_tick),
    .o_time(hour),
    .o_tick()
);
```

변수 Mode에 따라 fnd 상위 두 자리, fnd 하위 두 자리에 어떤 데이터가 출력될지 경우를 나눔.

분, 시 time_counter 모듈 인스턴스

Parameter: 24

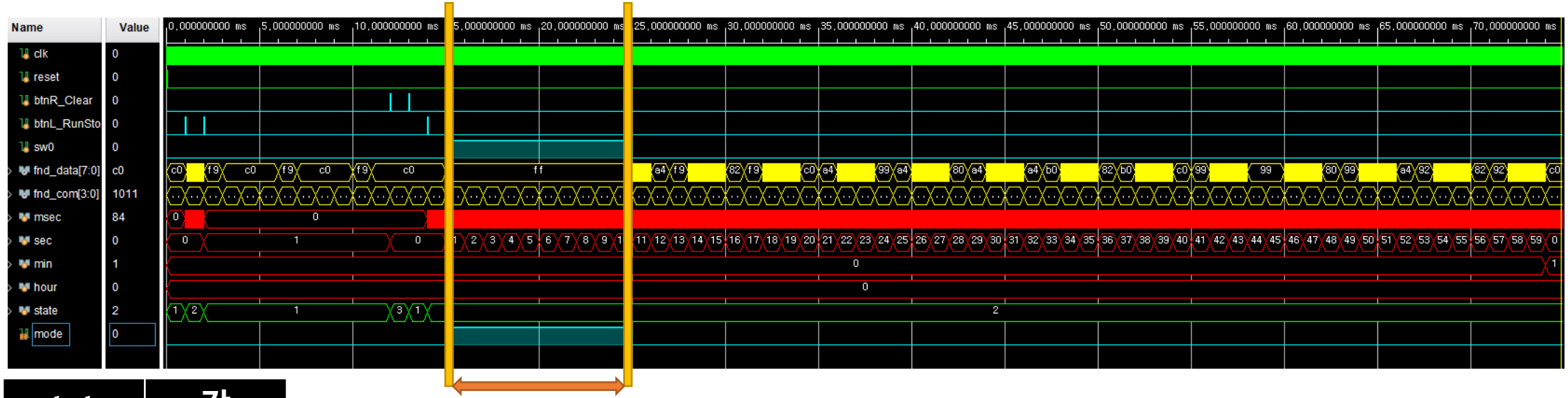
U_StopWatch_DP




```
module stopwatch (//top module
    input        clk,
    input        reset,
    input        btnR_Clear,
    input        btnL_RunStop,
    input        sw0,
    output [7:0] fnd_data,
    output [3:0] fnd_com
);
    wire [$clog2(100)-1:0] w_low_fnd;
    wire [$clog2(60)-1:0] w_high_fnd;
    wire w_clear, w_runstop, w_mode;

    stopwatch_cu U_StopWatch_CU(
        .clk(clk),
        .reset(reset),
        .i_clear(btnR_Clear),
        .i_runstop(btnL_RunStop),
        .sw(sw0),
        .o_clear(w_clear),
        .o_runstop(w_runstop),
        .o_mode(w_mode)
    );

    stopwatch_dp U_Stopwatch_DP(
        .clk(clk),
        .reset(reset),
        .run_stop(w_runstop),
        .clear(w_clear),
        .mode(w_mode),
        .low_fnd(w_low_fnd),
        .high_fnd(w_high_fnd)
    );

    fnd_controller U_FND_CONTROLLER (
        .clk(clk),
        .reset(reset),
        .msec(w_low_fnd),
        .sec(w_high_fnd),
        .fnd_data(fnd_data),
        .fnd_com(fnd_com)
    );
endmodule
```

Fnd출력모드를 바꾸기 위한 입력변수

```
module tb_stopwatch();
    reg clk, reset, btnR_Clear, btnL_RunStop, sw0;
    wire [7:0] fnd_data;
    wire [3:0] fnd_com;

    stopwatch U_StopWatch(//top module
        .clk(clk),
        .reset(reset),
        .btnR_Clear(btnR_Clear),
        .btnL_RunStop(btnL_RunStop),
        .sw0(sw0),
        .fnd_data(fnd_data),
        .fnd_com(fnd_com)
    );

    always #5 clk = ~clk;

    initial begin
        #0 clk = 0; reset = 1; btnR_Clear = 0; sw0 = 0; btnL_RunStop=0;
        #20 reset = 0;
        #1000000 btnL_RunStop=1;
        #10 btnL_RunStop=0;//RUN
        #1000000 btnL_RunStop = 1;
        #10 btnL_RunStop = 0;//STOP
        #10000000 btnR_Clear = 1;
        #10 btnR_Clear = 0;//CLEAR
        #1000000 btnR_Clear = 1;
        #10 btnR_Clear = 0;//STOP
        #1000000 btnL_RunStop = 1;
        #10 btnL_RunStop = 0;//RUN
        #1000000 sw0 = 1;
        #10000000 sw0 = 0;
        #1000000
        $finish;
    end

endmodule
```



FND: 시,분 출력 구간
fnd_data == ff => 0

state	값
STOP	1
RUN	2
CLEAR	3

mode	FND
0	초, msec
1	시, 분