<top>

```
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
```

<stopwatch_dp>

```
module stopwatch_dp (
    input        clk,
    input        reset,
    input        run_stop,
    input        clear,
    input        mode,
    output [7:0] low_fnd,
    output [6:0] high_fnd
);
    wire w_tick_100hz, w_msec_tick, w_sec_tick, w_min_tick, w_hour_tick;
    wire [7:0] msec, min;
    wire [6:0] sec, hour;

    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()
    );
endmodule
```

<stopwatch_cu>

```
module stopwatch_cu(
    input  clk,
    input  reset,
    input  i_clear,//BtnR
    input  i_runstop,//BtnL
    input  sw,//sw[0]
    output o_clear,
    output o_runstop,
    output o_mode
);
    reg[1:0] state_reg, next_state;

    parameter STOP = 1;
    parameter RUN = 2;
    parameter CLEAR = 3;

    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
endmodule
```

<testbench>

```
`timescale 1ns / 1ps

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
```