```
module dual_display (
    input  logic        reset,          // active low reset
    input  logic [3:0]  in0, in1,       // two 4-bit inputs
    output logic [6:0]  seg0, seg1      // two 7-bit decoded outputs
);

    // --- Internal signals ---
    logic clk;
    logic [23:0] counter;
    logic sel;               // which input is currently active
    logic [3:0] mux_in;      // input to the decoder
    logic [6:0] seg_decoded; // output of the single decoder

    // Oscillator (from lab1_jc.sv)
    HSOSC hf_osc (
        .CLKHFPU(1'b1),
        .CLKHFEN(1'b1),
        .CLKHF(clk)
    );

    // Divider: slow down for visible multiplexing
    always_ff @(posedge clk or negedge reset) begin
        if (!reset) begin
            counter <= 0;
            sel     <= 0;
        end else begin
            counter <= counter + 1;
            if (counter == 24'd10_000_00) begin // adjust rate
                counter <= 0;
                sel <= ~sel;
            end
        end
    end

    // Select input
    assign mux_in = sel ? in1 : in0;

    // Single seven-segment decoder instance
    seven_segment u_decoder (
        .s(mux_in),
        .seg(seg_decoded)
    );

    // Latch into seg0 or seg1 depending on which input was active
    always_ff @(posedge clk or negedge reset) begin
        if (!reset) begin
            seg0 <= 7'b1111111;
            seg1 <= 7'b1111111;
        end else begin
            if (sel)
                seg1 <= seg_decoded;
            else
                seg0 <= seg_decoded;
        end
    end

endmodule
```