```
module trap_edge(
input logic clk,
input logic async_sig,
input logic reset,
output logic trapped_edge
);


logic high_edge;

sync sync1(
.clk(clk),
.async_sig(async_sig),
.high_edge(high_edge)
);

mux mux1(
.signal_in(high_edge),
.reset(reset),
.signal_out(trapped_edge)
);

endmodule

////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////////////////

module sync(
input logic clk,
input logic async_sig,
output logic high_edge
);

logic n1;

always_ff@(posedge clk) begin

    n1<= async_sig;
    high_edge<= n1;

end

endmodule

////////////////////////////////////////////////////////////////////////////////
```

```
module mux(
input logic signal_in,
input logic reset,
output logic signal_out
);

reg flag;

always@(*) begin

    if ( reset) begin
        flag <= 0;
        signal_out<= 0;
    end

    else begin
        if ( flag == 1 | signal_in == 1) begin

            signal_out <= 1;
            lag <= 1;
        end

        else begin

            signal_out <= 0;
            flag <= 0;
        end

    end

end

endmodule
```