

```
1  module trap_edge(
2  input logic clk,
3  input logic async_sig,
4  input logic reset,
5  output logic trapped_edge
6  );
7
8
9  logic high_edge;
10
11  sync sync1(
12  .clk(clk),
13  .async_sig(async_sig),
14  .high_edge(high_edge)
15  );
16
17  mux mux1(
18  .signal_in(high_edge),
19  .reset(reset),
20  .signal_out(trapped_edge)
21  );
22
23  endmodule
24
25  //////////////////////////////////////
26  //////////////////////////////////////
27
28  module sync(
29  input logic clk,
30  input logic async_sig,
31  output logic high_edge
32  );
33
34  logic n1;
35
36  always_ff@(posedge clk) begin
37
38      n1<= async_sig;
39      high_edge<= n1;
40
41  end
42
43  endmodule
44
45  //////////////////////////////////////
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
```

```
70
71
72 module mux(
73     input logic signal_in,
74     input logic reset,
75     output logic signal_out
76 );
77
78 reg flag;
79
80 always@(*) begin
81
82     if ( reset) begin
83         flag <= 0;
84         signal_out<= 0;
85     end
86
87     else begin
88         if ( flag == 1 | signal_in == 1) begin
89
90             signal_out <= 1;
91             lag <= 1;
92         end
93
94         else begin
95
96             signal_out <= 0;
97             flag <= 0;
98         end
99     end
100 end
101
102 end
103
104 endmodule
105
106
107
```