```
1  `timescale 1ns / 1ps
2  /**********************************************************************************
3  *
4  * Engineer: Justin Maeder; 015906629
5  * Email:    Justin_maeder@hotmail.com
6  * Filename: AISO.v
7  * Date:     February 13, 2019
8  * Version:  14.7
9  * Description: The Asynchronous-in, synchronous out module is meant to prevent all
10 *              flops in our design from reaching a metastable state when reset is
11 *              released. With this module when reset it released, all flops will
12 *              reset at the same time.
13 *
14 **********************************************************************************/
15 module AISO(clk, reset, rst_s);
16
17    // Input & Output Declarations
18    input  clk, reset;
19    output wire rst_s;
20
21    // Hold values of flops
22    reg    QMeta, QOK;
23
24    ////////////////////////////////////////////////////////////////////////////
25    // Sequential Block to reset or hold states on the active edge of the clock //
26    ////////////////////////////////////////////////////////////////////////////
27    always @ (posedge clk, posedge reset)
28       if (reset) {QMeta, QOK} <= 2'b0;
29       else       {QMeta, QOK} <= {1'b1,QMeta};
30
31    // Invert output of second flop
32    assign rst_s = ~QOK;
33
34 endmodule
```