```
module thruout2;// /ch10/10.27/thruout2.sv
       bit clk, sof, eof;
       bit busy_sof;
  default clocking @(posedge clk); endclocking
  function void setval4busy(bit d);
       busy_sof=d;
  endfunction
 // This is not catching multiple sof's and one eof within valid.
 // It should check that for one sof, there should be only one eof.
 ap_sof_eof: assert property
       ((sof && busy_sof==1'b0, setval4busy(1'b1)) |->
              ##[1:$] (eof)) setval4busy(1'b0);   // assertion cannot fail

 ap_neversof_when_busy: assert property(not(sof && busy_sof));
```

## 10.28 Setup and hold checks

**Requirement**: Input signal in1 must be stable for at least 2 ns before the posedge of clk.
How do I write an assertion/checker for this?

**Solution**: SVA is not intended for timing checks. The best solution is to use the SystemVerilog timing checks defined in **1800:31 Timing checks.** These timing checks include:

**$setup $hold $setuphold $recovery $removal $recrem**

The **$setup** and **$hold** checks accept two signals, the *reference event* and the *data event*, and define a time window with respect to one signal while checking the time of transition of the other signal with respect to the window. In general, they all perform the following steps:

a) Define a time window with respect to the reference signal using the specified limit or limits.
b) Check the time of transition of the data signal with respect to the time window.
c) Report a timing violation (e.g., an invert of a declared bit variable) if the data signal transitions within the time window. Below s an example for the use of a **$setup.**

```
module setup1(input logic clk, in1); // /ch10/10.28/setup.sv, setup.png
       event e;
       bit notifier;
       parameter PERIOD=10ns;
   // $setup ( data_event , reference_event , timing_check_limit [ , [ notifier ] ] ) ;
   // $setup( data, posedge clk, 10, notifier ) ;
   specify
      // Define timing check specparam values
      // Specify timing check variable must be a port.
      specparam tSU = 3, tHD = 1, tPW = 25, tWPC = 10, tREC = 5;
     $setup(in1, posedge clk, tSU, notifier);
   endspecify
  // ** Error: setup.sv(20): $setup( in1:73 ns, posedge clk:75 ns, 3 ns );
  // #    Time: 75 ns  Iteration: 1  Instance: /top/setup1_ins
   ap_setup0: assert property(@e !in1 |-> @(posedge clk)  !in1 );
   ap_setup1: assert property(@e in1 |-> @(posedge clk) in1 );

   always @(posedge clk) begin
       # (PERIOD -2ns);
       -> e;
   end
endmodule : setup1
```