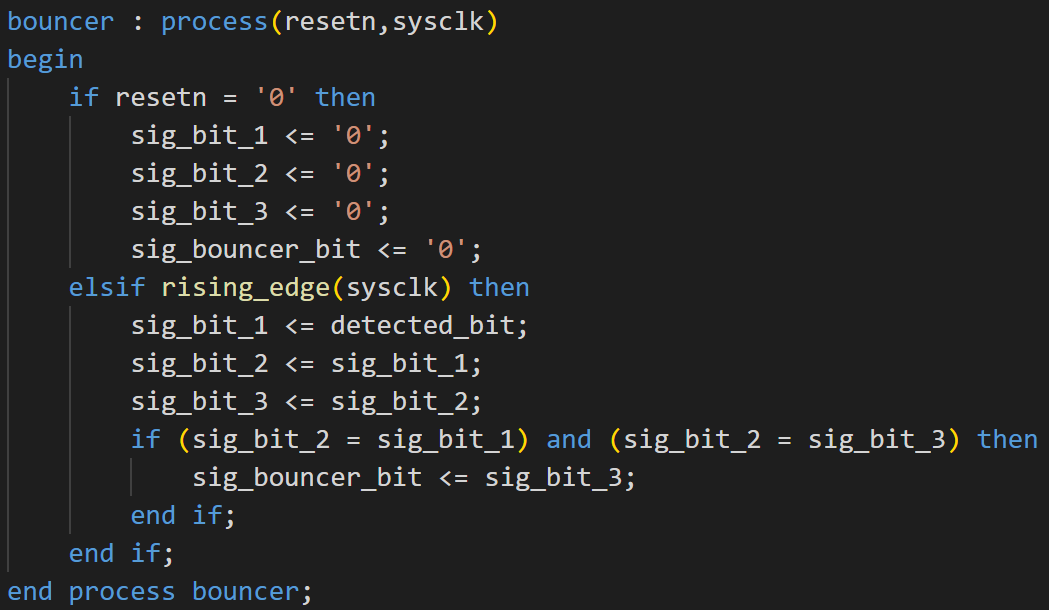
**Debouncer Example – chatGPT**

****

The "**bouncer**" process takes two inputs: "**resetn**" and "sysclk". "**resetn**" is an asynchronous reset signal, and "**sysclk**" is the system clock signal.  
  
When "**resetn**" is low (i.e., the reset signal is active), the process initializes three internal signals ("**sig\_bit\_1**", "**sig\_bit\_2**", and "**sig\_bit\_3**") and the debounced output signal ("**sig\_bouncer\_bit**") to zero.  
  
When "**resetn**" is high (i.e., the reset signal is inactive), the process waits for the rising edge of the "**sysclk**" signal. When a rising edge is detected, the process updates the three internal signals with the value of "**detected\_bit**", which is assumed to be the input signal to be debounced.  
  
The three internal signals ("**sig\_bit\_1**", "**sig\_bit\_2**", and "**sig\_bit\_3**") are used to delay the input signal by three clock cycles. This helps to eliminate any bouncing effect by allowing the input signal to stabilize before it is processed.  
  
After the input signal has been delayed by three clock cycles, the process checks whether the last three bits ("**sig\_bit\_1**", "**sig\_bit\_2**", and "**sig\_bit\_3**") are all equal. If they are, then the input signal is considered stable, and the debounced output signal ("**sig\_bouncer\_bit**") is set to the value of the last three bits.

In other words, if the input signal remains stable for at least three clock cycles, the output signal will reflect the stable state. If the input signal changes again within those three clock cycles, the output signal will not change. This helps to eliminate any bouncing effect and ensures that the system can accurately detect the stable state of the input signal.