```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;


--WRITTEN BY DAVE
ENTITY debouncer IS

  PORT(
    clk     : IN  STD_LOGIC;  -- constants adjusted for 25 MHZ clk
    button  : IN  STD_LOGIC;  -- input signal to be debounced
    result  : OUT STD_LOGIC); -- debounced signal out
END debouncer;

ARCHITECTURE logic OF debouncer IS
  SIGNAL flipflops   : STD_LOGIC_VECTOR(1 DOWNTO 0); --input flip flops
  SIGNAL counter_set : STD_LOGIC;                    --sync reset to zero
  SIGNAL counter_out : UNSIGNED(10 DOWNTO 0) := (OTHERS => '0');
     --counter size (20 bits gives 10.5ms with 100MHz clock)
BEGIN

  counter_set <= flipflops(0) xor flipflops(1);   --determine when to start/reset counter

  PROCESS(clk)
  BEGIN
    IF(clk'EVENT and clk = '1') THEN
      flipflops(0) <= button;
      flipflops(1) <= flipflops(0);
      If(counter_set = '1') THEN      -- reset counter because input is still changing
        counter_out <= (OTHERS => '0');
--      ELSIF(counter_out(5) = '0') THEN         -- use for simulation
      ELSIF(counter_out(10) = '0') THEN      -- stable input time is not yet met
        counter_out <= counter_out + 1;
      ELSE                                        --stable input time is met
        result <= flipflops(1);
      END IF;
    END IF;
  END PROCESS;
END logic;
```