Problem 2 – VHDL Code of the Debouncer Circuit

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity debouncer is
        ……. (
                k : integer := 24;
                DD : integer := 10000000
        );
    …….. (
                        clk     :       in      std_logic;
                        rst     :       in      std_logic;
                        input   :       in      std_logic;
                        output          :       out  std_logic
                );
end debouncer;

architecture rtl of debouncer is

        -- intermediate signals
        signal set : std_logic;
        signal rst_count : std_logic;
        signal count : std_logic;
        signal counter_out : std_logic_vector(……);
        signal xor_out : std_logic;
        signal mux_out : std_logic;
        signal output_sig : std_logic;
        signal prev_input : std_logic;
        signal DD_check : std_logic;

begin
        -- D flip-flop
        D_FF1 : process (……)
        begin
                if rising_edge(clk) then
                        prev_input <= input;
                end if;
        end process;
```

```
        xor_out <= input xor prev_input;
        set <= ……..;


-- Set reset D flip flop
        SR_D_FF : process (……)
        begin
                if(rst_count = '1') then
                        count <= '0';
                elsif rising_edge (clk) then
                        if(set = '1') then
                                count <= ……..….;
                        else
                                count <= count;
                        end if;
                end if;
        end process;

        mux_out <= input when …….. else output_sig;

        -- D flip-flop
        D_FF2 : process (clk)
        begin
                if rising_edge(clk) then
                        output_sig <= mux_out;
                end if;
        end process;
        output <= ……..;

        rst_count <= rst or DD_check;

        -- counter
        count_up : process (……)
        begin
                if …… then
                        counter_out <= ……;
                elsif rising_edge(clk) then
                        if(count = '1') then
                                counter_out <= ……..;
```

```
                    end if;
                end if;
        end process;

        -- check if count = DD-1
        DD_check <= '1' when (conv_integer(……..) =
……) else '0';

end rtl;
```