```
-- COUNT.VHD --
library ieee;
use ieee.std_logic_1164.all;


entity Count is generic(threshold: natural := 10);
 port(reset, clk, start: in std_logic; aboveth: out std_logic);
end Count;

architecture Behav of Count is
    type States is (IDLE, COUNTING);
    Signal state, nextstate : States := IDLE;
    Signal c : natural := 0;
begin
    -- Calcul de l'état suivant
    -- Comme on est en std_logic,"elsif ='0'" et non "else", car le signal peux avoir
    d'autre valeur
    process (state, start, c)
    begin
        case state is
        when IDLE =>
            if start = '1' then
                nextstate <= COUNTING;
            elsif start = '0' then
                nextstate <= IDLE;
            end if;
        when COUNTING =>
            if c < threshold then
                nextstate <= COUNTING;
            else
                nextstate <= IDLE;
            end if;
        end case;
    end process;

    -- MISE A JOUR DU REGISTRE D'ETAT
    process(reset, clk, start)
    begin
        -- RESET : asynchrone haut
        if reset = '1' then
            state <= IDLE;
        -- HORLOGE : front montant
        elsif ( (start = '1') and (state = IDLE) )then
                state <= COUNTING;

        elsif (clk'event and clk = '1') then
            state <= nextstate;
        -- Detecter un pic sur start

        end if;


    end process;

        -- MISE A JOUR A CHAQUE FRONT MONTANT DE LA CLOCK POUR C ou sur un reset
    process(start, clk, reset)
    begin
        if(reset = '1') then
            c <= 0;

        else
            if (clk'event and clk = '1') then
                if (state = IDLE and start = '0') then
                    C <= 0;
                elsif ( state = IDLE and start = '1') then
                    c <= c + 1;
                elsif (state = COUNTING and c < threshold) then
                    c <= c + 1;
                elsif(state = COUNTING and c >= threshold) then
                    c <= 0;
                end if;
```



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

    -- Process d'Output

    process(c)
    begin
        if  (c >= threshold) then
            aboveth <= '1';
        else aboveth <= '0';
        end if;
    end process;

end Behav;
```