```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;
use work.meu_pacote_led.all;

entity top is

    GENERIC (N : integer := 258);
    port(mclk, reset: in std_logic;
        anodos: out std_logic_vector(3 downto 0);
        catodos: out std_logic_vector(0 to 7);
        aux4 : INOUT std_logic_vector(3 downto 0)       );
end top;

architecture Behavioral of top is

    TYPE array_integer IS ARRAY (3 DOWNTO 0) of integer range 0 to 15 ;
    TYPE array_integer2 IS ARRAY (1 DOWNTO 0) of integer range 0 to 15 ;
    signal entradas: array_integer := ( 0, 0, 0, 0);
    signal s: array_integer2;
    signal clk1, clk2, clk3 :  std_logic;
    signal aux3 : std_logic_vector(1 downto 0);
    signal binario : std_logic_vector(3 downto 0);
    signal ymux: integer range 0 to 15;
    signal soma: std_logic_vector(11 downto 0);
    signal numero: integer range 0 to 255;
    signal chaves: std_logic_vector(7 downto 0);

    begin

        map0 : entity work.ccto0(arq)
        port map( mclk, reset,  clk3 ); --in,in,out

        map01 : entity work.contador255(arq)
        port map( clk3,  reset, numero );   --in,out

        chaves <= std_logic_vector(to_unsigned( numero, 8 ));

        map02 : entity work.somachaves(arq)
        port map( soma, chaves, entradas(2), entradas(1), entradas(0) );    --in,in,out


        map1 : entity work.ccto(arq)
        generic map (N => N)
        port map( mclk, reset,  clk1 ); --in,in,out

        map2 : entity work.ccto2(arq)
        generic map (N => N)
        port map( mclk, reset,  clk2 ); --in,in,out

        aux3 <= clk2 & clk1 ;

            map_3: entity work.mux2x1(arq)      --mux2X1
            port map( clk1, entradas(0), entradas(1), s(0));

            map_4: entity work.mux2x1(arq)      --mux2X1
            port map( clk1, entradas(2), entradas(3), s(1));

            map_5: entity work.mux2x1(arq)      --mux2X1
            port map( clk2, s(0), s(1), ymux);

            map_6: entity work.dem2x4(arq)
            port map(aux3, aux4);

        binario <= std_logic_vector(to_unsigned( ymux, 4 ));

        map_7 : entity work.decod_7_seg(arq)
        port map(binario, aux4, anodos, catodos);   --entrada,entrada,saida,saida
```

```
end Behavioral;
```