

23 April 2007

Create a clock-synchronous VHDL entity with a 4-bits control input (**CTL4**) and a parallel 12-bits output (**OUT12**).

Starting from a reset, the entity must sample the input **CTL4** and receive data from a serial input (**DATAIN**). If the first 4 bits from the serial input stream are equal to the control word sampled from **CTL4**, the entity must transfer these 4 bits to the output followed by the subsequent 8 bits from **DATAIN**. The entity must set the output to high impedance in any other situation and when the enable signal is set to 0.

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY key_recognizer IS
PORT (
        CLK, ENABLE, RESET: IN std_logic;           -- INGRESSI DI CONTROLLO
        DATAIN: IN std_logic;                       -- DATI IN INGRESSO
        CTL4 : IN std_logic_vector(3 downto 0);      -- PAROLA CHIAVE
        OUT12 : OUT std_logic_vector(11 downto 0)    -- DATI IN USCITA
    );
END key_recognizer;

architecture Behavioral of key_recognizer is
    signal CONT: integer range 0 to 15:=15;          -- segnale di conteggio
    signal data_reg:std_logic_vector(11 downto 0);  -- registro per appoggiare i dati
    signal key_reg:std_logic_vector(3 downto 0);    -- registro per la parola chiave

begin
    process(CLK,RESET) -- il processo è attivato dal clock e dal reset asincrono
    begin
        if(RESET='1') then
            key_reg <= (others=>'0');                -- azzeriamo il contenuto dei registri
            data_reg<=(others=>'0');
            CONT <=0;                                -- azzeriamo il contatore
        elsif(CLK'event and CLK='1') then
            data_reg<=data_reg(10 downto 0) & DATAIN; -- i dati entrano nel registro
            -- il conteggio si incrementa fino a 13 e poi si ferma
            -- inoltre quando è 0 campioniamo pure l'ingresso CTL4
            case CONT is
                when 0      =>    CONT <= CONT + 1;
                                key_reg <= CTL4;
                when 1 to 12 =>    CONT <= CONT + 1;
                when others =>    null;
            end case;
        end if;
    end process;

    -- a questo punto non resta che copiare il data_reg all'uscita OUT12
    -- nelle condizioni di conteggio a 12, di abilitazione attiva
    -- e di soddisfacimento della parola chiave
    OUT12 <= data_reg
    when ((data_reg(11 downto 8)=key_reg) and ENABLE='1' and CONT=12)
    else "ZZZZZZZZZZZZ";
end Behavioral;

```