library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity bitshift is

Port ( clk : in STD\_LOGIC;

reset : in STD\_LOGIC;

start\_nappi : in STD\_LOGIC; -- Painonappi (aktiivinen matala)

kytkin : in STD\_LOGIC\_VECTOR(7 downto 0); -- Vipukytkimet (8-bit)

led\_out : out STD\_LOGIC); -- LED-ulostulo (sarjadata)

end bitshift;

architecture Behavioral of bitshift is

signal shift\_reg : STD\_LOGIC\_VECTOR(7 downto 0); -- Siirtorekisteri

signal load : STD\_LOGIC := '0'; -- Siirto päälle/pois

signal counter : natural range 0 to 8 := 0; -- Laskuri lähetystä varten

signal clock\_div : natural range 0 to 49999999 := 0; -- Kellon jakaja 1 Hz taajuuden saavuttamiseksi

begin

process(clk, reset)

begin

if reset = '0' then -- Reset-signaali (aktiivinen matala)

-- Nollataan tilat resetillä

shift\_reg <= (others => '0');

counter <= 0;

load <= '0';

elsif rising\_edge(clk) then

if clock\_div >= 49999999 then -- 1 Hz taajuuden saavuttaminen

clock\_div <= 0;

-- Jos nappia painetaan, ladataan data siirtorekisteriin //HOX nappia tulee pitää painettua 1 sec jotta saadaan arvot ja homma rullaan

if start\_nappi = '0' then -- Painonappi on aktiivinen matala

shift\_reg <= kytkin; -- Ladataan 8-bittinen data kytkimiltä

load <= '1'; -- Aloitetaan siirto

counter <= 0; -- Nollataan laskuri

-- Siirretään bittejä sarjamuotoon

elsif load = '1' then

if counter = 8 then

load <= '0';

else

led\_out <= shift\_reg(7); -- Näytetään ylin bitti LEDissä

shift\_reg <= shift\_reg(6 downto 0) & '0'; -- Siirretään bittijonoa vasemmalle

counter <= counter + 1; -- Päivitetään laskuri

end if;

end if;

else

clock\_div <= clock\_div + 1;

end if;

end if;

end process;

end Behavioral;

Testbench:

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

ENTITY bitshift\_vhd\_tst IS

END bitshift\_vhd\_tst;

ARCHITECTURE bitshift\_arch OF bitshift\_vhd\_tst IS

-- constants

-- signals

SIGNAL clk : STD\_LOGIC;

SIGNAL kytkin : STD\_LOGIC\_VECTOR(7 DOWNTO 0);

SIGNAL led\_out : STD\_LOGIC;

SIGNAL reset : STD\_LOGIC;

SIGNAL start\_nappi : STD\_LOGIC;

COMPONENT bitshift

PORT (

clk : IN STD\_LOGIC;

kytkin : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

led\_out : OUT STD\_LOGIC;

reset : IN STD\_LOGIC;

start\_nappi : IN STD\_LOGIC

);

END COMPONENT;

BEGIN

i1 : bitshift

PORT MAP (

-- list connections between master ports and signals

clk => clk,

kytkin => kytkin,

led\_out => led\_out,

reset => reset,

start\_nappi => start\_nappi

);

init : PROCESS

BEGIN

wait for 10 ns;

kytkin <= "10100110";

start\_nappi <= '0';

wait for 150 ns;

start\_nappi <= '1';

wait for 1000 ns;

kytkin <= "11001010";

start\_nappi <= '0';

wait for 150 ns;

start\_nappi <= '1';

WAIT;

END PROCESS init;

always : PROCESS

BEGIN

LOOP

clk <= '0';

wait for 5 ns;

clk <= '1';

wait for 5 ns;

END LOOP;

WAIT;

END PROCESS always;

END bitshift\_arch;

Simulaatio:

Kuva, joka sisältää kohteen kuvakaappaus, teksti, näyttö, Multimediaohjelmisto

Kuvaus luotu automaattisesti

Demo video:

