Main:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity readWrite is

port(

addr: in std\_logic\_vector (1 downto 0);

data\_in: in std\_logic\_vector (7 downto 0);

data\_out: out std\_logic\_vector (7 downto 0) := (others => '0');

r\_signal: in std\_logic;

w\_signal: in std\_logic;

clock : in std\_logic

);

end entity readWrite;

architecture rtl of readWrite is

type t\_memory is array (255 downto 0) of std\_logic\_vector(7 downto 0);

signal memory: t\_memory;

begin

process (clock) is

begin

if rising\_edge(clock) then

if r\_signal = '0' then --active low

data\_out <= memory(to\_integer(unsigned(addr)));

elsif w\_signal = '0' then -- active low

memory(to\_integer(unsigned(addr))) <= data\_in;

end if;

end if;

end process;

end architecture rtl;

Tb:

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

ENTITY readWrite\_vhd\_tst IS

END readWrite\_vhd\_tst;

ARCHITECTURE readWrite\_arch OF readWrite\_vhd\_tst IS

-- constants

-- signals

SIGNAL addr : STD\_LOGIC\_VECTOR(1 DOWNTO 0);

SIGNAL clock : STD\_LOGIC;

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

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

SIGNAL r\_signal : STD\_LOGIC;

SIGNAL w\_signal : STD\_LOGIC;

COMPONENT readWrite

PORT (

addr : IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

clock : IN STD\_LOGIC;

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

data\_out : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

r\_signal : IN STD\_LOGIC;

w\_signal : IN STD\_LOGIC

);

END COMPONENT;

BEGIN

i1 : readWrite

PORT MAP (

-- list connections between master ports and signals

addr => addr,

clock => clock,

data\_in => data\_in,

data\_out => data\_out,

r\_signal => r\_signal,

w\_signal => w\_signal

);

init : PROCESS

-- variable declarations

BEGIN

data\_in <= (others => '0');

addr <= (others => '0');

r\_signal <= '1';

w\_signal <= '1';

wait for 5 ns;

data\_in <= "01010101";

addr <= "00";

w\_signal <= '0';

wait for 20 ns;

data\_in <= "00011111";

addr <= "01";

w\_signal <= '0';

wait for 20 ns;

data\_in <= "00000000";

addr <= "00";

r\_signal <= '0';

w\_signal <= '1';

wait for 20 ns;

data\_in <= "00000000";

addr <= "01";

r\_signal <= '0';

w\_signal <= '1';

wait for 20 ns;

WAIT;

END PROCESS init;

always : PROCESS

BEGIN

clock <= '0';

wait for 10 ns;

clock <= '1';

wait for 10 ns;

END PROCESS always;

END readWrite\_arch;

Sim:

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

Kuvaus luotu automaattisesti

Demo:

