

Kazalo:

Kazalo:	1
-- **** STUDENT: 64000225	4
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association). 4	
-- **** STUDENT: 64190088	6
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor); -- ob prvi napaki izhod iz zanke	6
-- **** STUDENT: 64200100	8
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.	8
-- **** STUDENT: 64200112	10
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor); 10	
-- **** STUDENT: 64200163	12
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor); -- ob prvi napaki izhod iz zanke	12
-- **** STUDENT: 64200238	14
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.	14
-- **** STUDENT: 64200288	16
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor); 16	
-- **** STUDENT: 64200296	18
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.	18
-- **** STUDENT: 64200385	20
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor); 20	
-- **** STUDENT: 64210132	22

-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor);22	
-- **** STUDENT: 64210290.....	24
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor);24	
-- **** STUDENT: 64210382.....	26
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert. Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).....	26
-- **** STUDENT: 64210384.....	28
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor);28	
-- **** STUDENT: 64210386.....	30
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert. Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).....	30
-- **** STUDENT: 64210445.....	32
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert. Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).....	32
-- **** STUDENT: 64210455.....	34
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert. Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).....	34
-- **** STUDENT: 64210457.....	36
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert. Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).....	36
-- **** STUDENT: 64240429.....	38
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association). V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial /= O_Xor);	38
-- **** STUDENT: 64240430.....	40

-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente. Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.: exit when (O_Xor_serial != O_Xor);	40
-- **** STUDENT: 64210113		42
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: V zanki ste sicer izvedli zaporedni izračun XOR, vendar ga je bolj smiselno dati v poseben process, ker se v simulaciji odvija vzporedno. V primeru neenakosti imate sicer assert, a je brez pomena ker mu ne sledi neko dejanje (npr. report "failed: " &integer'image(idx) severity error; ali exit when (O_Xor_serial != O_Xor).	42
-- **** PREDLOGA VAJE		44

```

-- *****
-- **** STUDENT: 64000225
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;
    signal fault : STD_LOGIC := '0';

begin

    UUT : Unary_Op_Bin_Tree GENERIC MAP ( I'length ) PORT MAP( I, O_Xor );

    stim_proc: process
        variable check: std_logic;
    begin
        for idx in 0 to 2**N - 1 loop
            I      <= std_logic_vector( to_unsigned( idx, I'length ) );
            wait for 50 ns;
            check := '0';
            for index in I'range loop

```

```
        check := check xor I( index );
    end loop;

    if ( O_Xor /= check ) then
        fault <= '1';
    end if;

end loop;
end process;
end test;
```

```

-- *****
-- **** STUDENT: 64190088
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);           -- ob prvi napaki izhod iz zanke
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin

    uut: Unary_Op_Bin_Tree
generic map ( N => N )    port map ( I => I, O_Xor => O_Xor );

    stim_proc: process
begin
    for idx in 0 to ( 2**N - 1 ) loop
        I    <= std_logic_vector( to_unsigned( idx, I'length ) );
        wait for 50ns;
    end loop
end process
end test;

```

```
        end loop;  
    end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64200100
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin

    UUT: Unary_Op_Bin_Tree
generic map ( N => N )
port map( I=>I, O_Xor=>O_Xor );

    stim_proc: process
variable xor_t : std_logic;
begin
for idx in 0 to ( 2**N-1 ) loop
I    <= std_logic_vector( to_unsigned( idx, I' length ) );
wait for 50 ns;

```



```
xor_t:=I( 0 );  
for j in 1 to N-1 loop  
    xor_t:=xor_t xor I( j );  
end loop;  
assert( xor_t= 0_Xor )  
report "failed: " &integer'image( idx ) severity error;  
end loop;  
wait;  
end process;  
  
end test;
```

```

-- *****
-- **** STUDENT: 64200112
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic (      N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic (N : natural := 64);
port ( I : std_logic_vector(N - 1 downto 0);
      O_Xor : OUT std_logic);
END component;

    constant ZERO : std_logic_vector(N-1 DOWNT0 0) := (others => '0');      -- vsi elementi na 0

    signal I : std_logic_vector (N-1 DOWNT0 0) := ZERO;
    signal O_Xor : STD_LOGIC;

begin

    UUT: Unary_Op_Bin_Tree
        generic map(
            N => N)
        port map(
            I => I,
            O_Xor => O_Xor);

```

```
stim_process:
  process
  begin
    for idx in 0 to 2**N - 1 loop
      I <= std_logic_vector(to_unsigned(idx, I'length));
      wait for 50 ns;
    end loop;
    wait;
  end process stim_process;
```

```
end test;
```

```

-- *****
-- **** STUDENT: 64200163
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);           -- ob prvi napaki izhod iz zanke
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree is
generic( N : natural := 64 );
port( I : std_logic_vector( N - 1 downto 0 );
      O_Xor : out std_logic );
end component;

    constant ZERO : std_logic_vector( N-1 downto 0 ) := ( others => '0' );

    signal I : std_logic_vector ( N-1 downto 0 ) := ZERO;
    signal O_Xor : std_logic;

begin

    Unit_Under_Test: Unary_Op_Bin_Tree
generic map( N => N )
port map( I => I, O_Xor => O_Xor );

stim_proc: process
begin
    for idx in 0 to 2**N - 1 loop
        I      <= std_logic_vector( to_unsigned( idx, I'length ) );

```

```
        wait for 50 ns;  
    end loop;  
    wait;  
end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64200238
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
    generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree is
        generic ( N : natural := 64 );
        port ( I : std_logic_vector( N - 1 downto 0 );
              O_Xor : out std_logic );
    end component;

    constant ZERO : std_logic_vector( N-1 downto 0 ) := ( others => '0' );    -- ALL elements set to 0

    signal I : std_logic_vector( N-1 downto 0 ) := ZERO;
    signal O_Xor : std_logic;

begin

    UUT: Unary_Op_Bin_Tree
        generic map ( N => N )
        port map ( I => I, O_Xor => O_Xor );

    stim_proc: process
        variable test_xor_temp : std_logic;
    begin

        for idx in 0 to ( 2**N - 1 ) loop
            I <= std_logic_vector( to_unsigned( idx, I'length ) );
            wait for 50 ns;
        end loop
    end process
end test;

```

```
test_xor_temp := I( 0 );

for j in 1 to N-1 loop
test_xor_temp := test_xor_temp xor I( j );
end loop;

assert ( test_xor_temp = O_Xor )
report "Napaka pri I = " & integer'image( idx ) & ": O_Xor = " & std_logic'image( O_Xor ) & ", Zakaj ni :" &
std_logic'image( test_xor_temp )
severity error;
end loop;

wait;
end process;

end test;
```

```

-- *****
-- **** STUDENT: 64200288
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);

-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin
    -- KODO VPISUJETE SEM

    UUT: Unary_Op_Bin_Tree
generic map ( N => N )
port map ( I => I, O_Xor => O_Xor );

    stim_proc: process
variable indeks : integer := 0;

```



```
begin
for indeks in 0 to ( 2**N - 1 ) loop
I      <= std_logic_vector( to_unsigned( indeks, I'length ) );
end loop;
wait;
      end process;
end test;
```

```

-- *****
-- **** STUDENT: 64200296
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin
    UUT: Unary_Op_Bin_Tree

        generic map(
            N=>N )
        port map(
            I=>I,          O_Xor=>O_Xor );
    stim_proc: process
        variable idx : integer range 0 to 2**N - 1;
        variable test_xor : std_logic;
    begin
        for idx in 0 to 2**N - 1 loop
I      <= std_logic_vector( to_unsigned( idx, I'length ) );

```

```
wait for 50 ns;

    test_xor:='0';
    for jdx in I' range loop
        test_xor := test_xor xor I( jdx );
    end loop;

    if ( test_xor /= 0_Xor ) then
        assert false
        report "Napaka pri I:" &integer'image( idx ) severity error;
    end if;

end loop;

wait;
end process stim_proc;
end test;
```

```

-- *****
-- **** STUDENT: 64200385
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin
    -- KODO VPISUJETE SEM

    uut : Unary_Op_Bin_Tree
    generic map ( N=>N )
    port map ( I=>I, O_Xor=>O_Xor );
    stim_proc: process
    begin
        for idx in 0 to 2**N-1 loop
            I <= std_logic_vector( to_unsigned( idx, I'length ) );

```

```
        wait for 50 ns;  
    end loop;  
end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64210132
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin
    -- KODO VPISUJETE SEM

    UUT : Unary_Op_Bin_Tree
        generic map ( N => N )
        port map ( O_Xor => O_Xor, I => I );

    stim_proc: process
begin

```

```
    for idx in 0 to ( 2**N )-1 loop
        I      <= std_logic_vector( to_unsigned( idx, I'length ) );
        wait for 50 ns;
    end loop;

    wait;
end process;

end test;
```

```

-- *****
-- **** STUDENT: 64210290
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin

    uut: Unary_Op_Bin_Tree generic map( N => N ) port map( I => I, O_Xor => O_Xor );

    stim_proc: process
variable uut_out : std_logic;
begin
        for idx in 0 to 2**N-1 loop
            I      <= std_logic_vector( to_unsigned( idx, I'length ) );
            wait for 50 ns;

```



```
        end loop;  
        wait;  
    end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64210382
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin

    UUT: Unary_Op_Bin_Tree
generic map ( N )
port map ( I, O_Xor );

    stim_proc: process
variable O_Comp : std_logic;
begin
        for idx in 0 to 2*N - 1 loop
            I      <= std_logic_vector( to_unsigned( idx, I'length ) );
            wait for 50 ns;
            O_Comp := '0';
        end loop
    end process
end test;

```

```
    for jdx in 0 to N-1 loop
      O_Comp := O_Comp xor I( jdx );
    end loop;

    assert( O_Comp = O_Xor )
    report "Test failed: " & integer'image( idx ) severity error;
  end loop;
  wait;
end process;
end test;
```

```

-- *****
-- **** STUDENT: 64210384
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin
    -- Povezovalni stavki
    UUT: unary_op_bin_tree
generic map ( N => N )
port map ( I, O_Xor );

    stim_proc: process
begin
    for idx in 0 to 2**N - 1 loop

```

```
        I      <= std_logic_vector( to_unsigned( idx, I' length ) );    -- pretvorba celostevilskega
indeksa idx v obliko std_logic_vector
        wait for 50 ns;
    end loop;
    wait;
end process;
end test;
```

```

-- *****
-- **** STUDENT: 64210386
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic (      N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (      I : std_logic_vector( N - 1 downto 0 );
          O_Xor : OUT std_logic );
END component;

    constant    ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal      I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal      O_Xor : STD_LOGIC;

begin
    -- KODO VPISUJETE SEM

    UT : Unary_Op_Bin_Tree GENERIC MAP ( I'length ) PORT MAP ( I, O_Xor );

    proc: process
variable test : STD_LOGIC;
begin
        for u in 0 to 2**N - 1 loop
            I      <= STD_LOGIC_VECTOR ( to_unsigned( u, I'length ) );
            wait for 50 ns;
        end loop
    end process
end test;

```

```
test := '0';
for c in I'range loop
test := test xor I( c );
end loop;

if ( 0_Xor /= test ) then
assert false
report "NAPAKA: " &
integer'image( u );
end if;

end loop;
end process;

end test;
```

```

-- *****
-- **** STUDENT: 64210445
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic (      N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 4096 );
port (      I : std_logic_vector( N - 1 downto 0 );
          O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin
    -- KODO VPISUJETE SEM

    UT : Unary_Op_Bin_Tree GENERIC MAP ( I'length ) PORT MAP( I, O_Xor );

    stim_proc: process
        variable test: std_logic;
    begin
        for idx in 0 to 2**N - 1 loop
            I      <= std_logic_vector( to_unsigned( idx, I'length ) );
            wait for 50 ns;
        end loop
    end process
end architecture test;

```



```
test := '0';
for index in I'range loop
test := test xor I( index );
end loop;

if ( 0_Xor /= test ) then
assert false
report "NAPAKA na DEC i: " &
integer'image( idx );
end if;

end loop;
end process;

end test;
```

```

-- *****
-- **** STUDENT: 64210455
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
  generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

  component Unary_Op_Bin_Tree is
    generic ( N : natural := 64 );
    port (
      I : in std_logic_vector( N - 1 downto 0 );
      O_Xor : out std_logic
    );
  end component;

  constant ZERO : std_logic_vector( N - 1 downto 0 ) := ( others => '0' ); -- vsi elementi na 0

  signal I : std_logic_vector( N - 1 downto 0 ) := ZERO;
  signal O_Xor : std_logic;

begin
  -- KODO VPISUJETE SEM

  UUT: Unary_Op_Bin_Tree
  generic map ( N ) port map ( I => I, O_Xor => O_Xor );

  stim_proc: process
    variable O_Comp : std_logic;
  begin
    for idx in 0 to 2**N - 1 loop

```

```

        -- Nastavi vhodni signal          za UUT in pocakaj na obdelavo
I      <= std_logic_vector( to_unsigned( idx, I'length ) );
wait for 50 ns;

        -- Inicializiraj lokalno spremenljivko
O_Comp := '0';

        -- Izracunaj XOR postopoma
for jdx in 0 to N - 1 loop
O_Comp := O_Comp xor I( jdx );
end loop;

        -- Primerjaj rezultate
assert ( O_Comp = O_Xor )
report "Test failed: vrednost = " & integer'image( idx ) severity error;
end loop;
wait;
end process;

end test;

```

```

-- *****
-- **** STUDENT: 64210457
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb. Lepo, da ste uporabili assert.
Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    -- this is the component that we are testing( this is defined in vhd file )
    component Unary_Op_Bin_Tree IS
    generic ( N : natural := 64 );
    port (
        I : std_logic_vector( N - 1 downto 0 );
        O_Xor : OUT std_logic );
    END component;
    --

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    -- we need signals that we can work on testbench - for connections
    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;
    --

begin
    -- initialisation of UUT
    UUT: Unary_Op_Bin_Tree generic map ( N ) port map ( I, O_Xor );

    stim_proc: process
    variable O_Comp : std_logic;
    begin
        for idx in 0 to 2*N - 1 loop

```

```

-- here we input the values to the input signal      of our UUT
    I      <= std_logic_vector( to_unsigned( idx, I'length ) );
    wait for 50 ns;
-- O_Xor should now be the result

-- Set variable to zero
    O_Comp := '0';
-- Compute XOR the sequential way
    for jdx in 0 to N-1 loop
        O_Comp := O_Comp xor I( jdx );
    end loop;

-- we compare if output of our UUT ( LUT 6 Tree XOR ) is the same as sequentially calculating xor
    assert( O_Comp = O_Xor )
    report "Test failed for input " & integer'image( idx ) & " : expected " & std_logic'image( O_Comp
) & ", got " & std_logic'image( O_Xor )
    severity error;

    end loop;
    wait;
end process;

end test;

```

```

-- *****
-- **** STUDENT: 64240429
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Pri generic map/port map stavku uporabljajte imensko povezovanje (ang.named association).
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin

    UUT: Unary_Op_Bin_Tree    -- this entity is defined in unary_op_tree.vhd
generic map ( N )
port map ( I, O_Xor );

    stim_proc: process
begin
        for idx in 0 to 2**N - 1 loop
            I      <= std_logic_vector( to_unsigned( idx, I'length ) );
            wait for 50 ns;
        end loop
    end process
end test;

```

```
        end loop;  
        wait;  
    end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64240430
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki for niste izvedli zaporednega izračuna XOR, s katerim bi primerjali rezultat izračuna komponente.
Ko imate to narejeno, je smiselno dodati kriterij za izhod, če je izračun napačen – npr.:
exit when (O_Xor_serial /= O_Xor);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
generic ( N : natural := 64 );
port (
    I : std_logic_vector( N - 1 downto 0 );
    O_Xor : OUT std_logic );
END component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );    -- vsi elementi na 0

    signal I : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal O_Xor : STD_LOGIC;

begin

    uut: Unary_Op_Bin_Tree
        generic map ( N => N )
        port map (
            I => I,                O_Xor => O_Xor
        );

    stim_proc: process

```



```
begin
    for idx in 0 to 2**N - 1 loop
        I      <= std_logic_vector( to_unsigned( idx, I'length ) );
        wait for 50 ns;
    end loop;
end process;

end test;
```

```

-- *****
-- **** STUDENT: 64210113
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
V zanki ste sicer izvedli zaporedni izračun XOR, vendar ga je bolj smiselno dati v poseben process, ker se v
simulaciji odvija vzporedno. V primeru neenakosti imate sicer assert, a je brez pomena ker mu ne sledi neko dejanje
(npr. report "failed: " &integer'image( idx ) severity error; ali exit when (O_Xor_serial /= O_Xor).

-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Unary_Op_Bin_Tree_tb is
generic ( N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is

    component Unary_Op_Bin_Tree IS
        generic (N : natural := 64);
        port ( I : std_logic_vector(N - 1 downto 0);
              O_Xor : OUT std_logic);
    END component;

    constant ZERO : std_logic_vector(N-1 DOWNT0 0) := (others => '0');    -- vsi elementi na 0

    signal I : std_logic_vector (N-1 DOWNT0 0) := ZERO;
    signal O_Xor : STD_LOGIC;

begin
    UUT: Unary_Op_Bin_Tree
        generic map (N =>N)
        port map (I =>I, O_Xor =>O_Xor);

    stim_proc: process
        variable Sek : std_logic;
        begin
            for x in 0 to 2**N - 1 loop

```

```
Sek := '0';

I <= std_logic_vector(to_unsigned(x, I'length));
wait for 50 ns;

    for m in 0 to N-1 loop
        Sek := Sek xor I(m);
    end loop;

    assert(Sek = 0_Xor);
end loop;
wait;
end process;
end test;
```

```

-- *****
-- **** PREDLOGA VAJE
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;
USE IEEE.STD_LOGIC_TEXTIO.ALL;
USE STD.TEXTIO.ALL;
use work.slv_image_pkg.ALL;

entity Unary_Op_Bin_Tree_tb is
generic (    N: natural := 16 );
end Unary_Op_Bin_Tree_tb;

architecture test of Unary_Op_Bin_Tree_tb is
    FILE RESULTS: TEXT OPEN APPEND_MODE IS "Unary_Op_Bin_Tree.csv";

    component Unary_Op_Bin_Tree IS
    generic (N : natural := 64);
    port ( I : std_logic_vector(N - 1 downto 0);
          O_Xor : OUT std_logic);
    END component;

    constant ZERO : std_logic_vector(N-1 DOWNT0 0) := (others => '0');      -- vsi elementi na 0

    signal I : std_logic_vector (N-1 DOWNT0 0) := ZERO;
    signal O_Xor : STD_LOGIC;
    signal O_Xor_serial : STD_LOGIC;

begin

    U1: Unary_Op_Bin_Tree
    generic map (N => N)      port map (    I => I, O_Xor => O_Xor );

    stim_proc: process
        variable HDR_line : LINE;

        PROCEDURE Log_variables(  O_Xor          : std_logic;
                                   O_Xor_serial : STD_LOGIC;
                                   I              : std_logic_vector (N-1 DownTo 0)

```

```

) IS
VARIABLE RES_LINE : LINE;
BEGIN
    hwrite(RES_LINE, I, right, N-1);
    write(RES_LINE, string'(",");
    write(RES_LINE, O_Xor, right, 1);
    write(RES_LINE, string'(",");
    write(RES_LINE, O_Xor_serial, right, 1);
    writeline(RESULTS, RES_LINE);
END;

begin
    I <= ZERO;
    write(HDR_line, string'("Datoteka, pri katerih nastanejo različne vrednosti xor tree in zaporednim
izracunom XOR"));
    writeline(RESULTS, HDR_line);
    write(HDR_line, string'("I, O_Xor, O_Xor_serial"));
    writeline(RESULTS, HDR_line);
    for idx in 0 to 2**N - 1 loop
        I <= std_logic_vector(to_unsigned(idx, I'length)); -- nastavi vhodni vektor
        wait for 50 ns;
        if (O_Xor_serial /= O_Xor) then
            Log_variables(O_Xor, O_Xor_serial, I); -- zabeleži rezultata xor operacij
        end if;
        exit when (O_Xor_serial /= O_Xor); -- ob prvi napaki izhod iz zanke
    end loop;
    wait;
end process;

xor_reduction: process (I) --zaporedni izračun xor
variable temp_xor: std_logic;
begin
    temp_xor := '0';
    for k in I'range loop
        temp_xor := temp_xor xor I(k);
    end loop;
    O_Xor_serial <= temp_xor;
end process;

end test;

```