

Kazalo:

Kazalo:	1
-- **** STUDENT: 64000225	3
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Procesna za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje. 3	
-- **** STUDENT: 64190088	6
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	6
-- **** STUDENT: 64200100	8
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	8
-- **** STUDENT: 64200112	10
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	10
-- **** STUDENT: 64200163	12
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	12
-- **** STUDENT: 64200238	14
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	14
-- **** STUDENT: 64200288	16
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	16
-- **** STUDENT: 64200296	18
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	18
-- **** STUDENT: 64200385	20
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	20
-- **** STUDENT: 64210132	23
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	23
-- **** STUDENT: 64210290	25
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Procesna za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje. 25	
-- **** STUDENT: 64210382	28
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	28
-- **** STUDENT: 64210384	30
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	30
-- **** STUDENT: 64210386	32

-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Pri povezovalnem stavku manjka generic map (N => N). Proces	32
za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje.....		
-- **** STUDENT: 64210445.....		34
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Pri povezovalnem stavku manjka generic map (N => N). Proces	34
za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje.....		
-- **** STUDENT: 64210455.....		36
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Upoštevajte navodila naloge in ne spreminjajte podatkov entitete –	
sicer vam naloge ne morem popraviti! 36		
-- **** STUDENT: 64210457.....		38
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Ni pripomb.....	38
-- **** STUDENT: 64240429.....		40
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Ni pripomb.....	40
-- **** STUDENT: 64240430.....		42
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Ni pripomb.....	42
-- **** STUDENT: 64210113.....		45
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Ni pripomb.....	45
-- **** PREDLOGA VAJE		47

```

-- *****
-- **** STUDENT: 64000225
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
Procesa za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic (    N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    component reduction_operators is
generic (    N: Natural := 3 );
port (      A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
          reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
end component;
-- konstante vrednosti A--
constant ZERO : std_logic_vector ( N-1 DOWNT0 0 )
:= ( others => '0' );
constant ONE : std_logic_vector ( N-1 DOWNT0 0 )
:= ( 0 => '1', others => '0' );
constant TWO : std_logic_vector ( N-1 DOWNT0 0 )
:= ( 1 => '1', others => '0' );
-- constant THREE : std_logic_vector ( N-1 DOWNT0 0 )
-- := ( 0 | 1 => '1', others => '0' );
constant FOUR : std_logic_vector ( N-1 DOWNT0 0 )
:= ( 2 => '1', others => '0' );
constant FIVE : std_logic_vector ( N-1 DOWNT0 0 )
:= ( 0 => '1', 2 => '1', others => '0' );
constant SIX : std_logic_vector ( N-1 DOWNT0 0 )
:= ( 1 | 2 => '1', others => '0' );
constant SEVEN : std_logic_vector ( N-1 DOWNT0 0 )
:= ( 0 | 1 | 2 => '1', others => '0' );

-- Vhodi--

```

```

signal          A : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( others => '0' );

-- Izhodi--
signal          reduced_OR : STD_LOGIC := '0';
signal          reduced_AND : STD_LOGIC := '0';
signal          reduced_XOR : STD_LOGIC := '0';

constant        clk_period : time := 50 ns;      -- perioda / frekvenca urinega impulza
signal          clk : STD_LOGIC;

begin

    uut: reduction_operators
    generic map ( N => N )
    port map (
        A => A, reduced_OR => reduced_OR, reduced_AND => reduced_AND, reduced_XOR => reduced_XOR
    );

    clk_process : process
begin
        clk    <= '0';
        wait for clk_period/2;
        clk    <= '1';
        wait for clk_period/2;
    end process;

    stim_proc: process
begin
        A      <= ZERO;      -- 0
        wait for 150 ns;
        A      <= ONE;       -- 1
        wait for 50 ns;
        A      <= TWO;       -- 2
        wait for 100 ns;
        -- A    <= THREE after 100 ns;    -- 3
        -- wait for 50 ns;
        A      <= FOUR;      -- 4
        wait for 50 ns;

```

```
A      <= FIVE;      -- 5
wait for 50 ns;
A      <= SIX;       -- 6
wait for 50 ns;
A      <= SEVEN;    -- 7
wait;

end process;
end test;
```

```

-- *****
-- **** STUDENT: 64190088
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    component reduction_operators is
        generic ( N: natural := 10 );
        port ( A: in STD_LOGIC_VECTOR ( N-1 downto 0 );
              reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
    end component;

    constant ZERO : std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );
    constant ONE  : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', others => '0' );
    constant TWO  : std_logic_vector( N-1 DOWNT0 0 ) := ( 1 => '1', others => '0' );
    constant THREE : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 | 1 => '1', others => '0' );
    constant FOUR  : std_logic_vector( N-1 DOWNT0 0 ) := ( 2 => '1', others => '0' );
    constant FIVE  : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', 2 => '1', others => '0' );
    constant SIX   : std_logic_vector( N-1 DOWNT0 0 ) := ( 1 | 2 => '1', others => '0' );
    constant SEVEN : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 | 1 | 2 => '1', others => '0' );

    signal A : std_logic_vector ( N-1 DOWNT0 0 ) := ZERO;
    signal reduced_OR, reduced_AND, reduced_XOR: STD_LOGIC;

begin

    uut: reduction_operators
    generic map ( N => N )
    port map ( A => A, reduced_OR => reduced_OR, reduced_AND => reduced_AND, reduced_XOR => reduced_XOR );

```

```
stim_proc: process
begin
    A    <= ZERO;
    wait for 150 ns;
    A    <= ONE;
    wait for 50 ns;
    A    <= TWO;
    wait for 100 ns;
    -- A  <= THREE after 100 ns;
    -- wait for 50 ns;
    A    <= FOUR;
    wait for 50 ns;
    A    <= FIVE;
    wait for 50 ns;
    A    <= SIX;
    wait for 50 ns;
    A    <= SEVEN;
    wait;
end process;

end test;
```

```

-- *****
-- **** STUDENT: 64200100
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    component reduction_operators is
generic( N: Natural := 10 );
port( A: in std_logic_vector( N-1 DOWNT0 0 ); reduced_OR, reduced_AND, reduced_XOR: out std_logic );
end component;

    constant bn0: std_logic_vector( N-1 DOWNT0 0 ):= ( others=>'0' );
    constant bn1: std_logic_vector( N-1 DOWNT0 0 ):= ( 0=>'1',others=>'0' );
    constant bn2: std_logic_vector( N-1 DOWNT0 0 ):= ( 1=>'1',others=>'0' );
    constant bn3: std_logic_vector( N-1 DOWNT0 0 ):= ( 0|1=>'1',others=>'0' );
    constant bn4: std_logic_vector( N-1 DOWNT0 0 ):= ( 2=>'1',others=>'0' );
    constant bn5: std_logic_vector( N-1 DOWNT0 0 ):= ( 0=>'1',2=>'1',others=>'0' );
    constant bn6: std_logic_vector( N-1 DOWNT0 0 ):= ( 1|2=>'1',others=>'0' );
    constant bn7: std_logic_vector( N-1 DOWNT0 0 ):= ( 0|1|2=>'1',others=>'0' );
    signal A: std_logic_vector( N-1 DOWNT0 0 ):= bn0;
    signal reduced_OR, reduced_AND,reduced_XOR: std_logic;

begin

    U1:reduction_operators
generic map ( N=>N )
port map( A=>A, reduced_OR=>reduced_OR,reduced_AND=>reduced_AND,reduced_XOR=>reduced_XOR );
stim_proc:process
begin
    A <=bn0;

```



```
wait for 150 ns;  
A    <=bn1;  
wait for 50 ns;  
A    <=bn2;  
wait for 100 ns;  
A    <=bn4;  
wait for 50 ns;  
A    <=bn5;  
wait for 50 ns;  
A    <=bn6;  
wait for 50 ns;  
A    <=bn7;  
wait;  
end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64200112
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

component reduction_operators is
generic ( N: Natural := 10);
port ( A: in STD_LOGIC_VECTOR (N-1 DOWNT0 0);
      reduced_OR, reduced_AND, reduced_XOR : out STD_LOGIC);
end component;

constant ZERO : std_logic_vector(N-1 DOWNT0 0)
:= (others => '0');
constant ONE : std_logic_vector(N-1 DOWNT0 0)
:= (0 => '1', others => '0');
constant TWO : std_logic_vector(N-1 DOWNT0 0)
:= (1 => '1', others => '0');
constant THREE : std_logic_vector(N-1 DOWNT0 0)
:= (0 => '1', 1 => '1', others => '0');
constant FOUR : std_logic_vector(N-1 DOWNT0 0)
:= (2 => '1', others => '0');
constant FIVE : std_logic_vector(N-1 DOWNT0 0)
:= (0 => '1', 2 => '1', others => '0');
constant SIX : std_logic_vector(N-1 DOWNT0 0)
:= (1 => '1', 2 => '1', others => '0');
constant SEVEN : std_logic_vector(N-1 DOWNT0 0)
:= (0 => '1', 1 => '1', 2 => '1', others => '0');

```

```
signal A : std_logic_vector (N-1 DOWNT0 0) := ZERO;
signal reduced_OR : STD_LOGIC;
signal reduced_AND : STD_LOGIC;
signal reduced_XOR : STD_LOGIC;
```

```
begin
```

```
    U1: reduction_operators
        generic map (N => N)
        port map (A => A,
            reduced_OR=> reduced_OR,
            reduced_AND=> reduced_AND,
            reduced_XOR=> reduced_XOR);
```

```
stim_proc: process
```

```
begin
```

```
    A <= ZERO;
    wait for 150 ns;
    A <= ONE;
    wait for 50 ns;
    A <= TWO;
    wait for 100 ns;
    A <= FOUR;
    wait for 50 ns;
    A <= FIVE;
    wait for 50 ns;
    A <= SIX;
    wait for 50 ns;
    A <= SEVEN;
    wait;
```

```
end process;
```

```
end test;
```

```

-- *****
-- **** STUDENT: 64200163
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is
    component reduction_operators is
        generic ( N: Natural := 10 );
        port ( A: in std_logic_vector ( N-1 DOWNT0 0 );
              reduced_OR, reduced_AND, reduced_XOR: out std_logic );
    end component;

    constant ZERO: std_logic_vector( N-1 DOWNT0 0 ) := ( others => '0' );
    constant ONE: std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', others => '0' );
    constant TWO: std_logic_vector( N-1 DOWNT0 0 ) := ( 1 => '1', others => '0' );
    constant THREE: std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', others => '0' );
    constant FOUR: std_logic_vector( N-1 DOWNT0 0 ) := ( 2 => '1', others => '0' );
    constant FIVE: std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', 2 => '1', others => '0' );
    constant SIX: std_logic_vector( N-1 DOWNT0 0 ) := ( 1 => '1', 2 => '1', others => '0' );
    constant SEVEN: std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', 2 => '1', others => '0' );

    signal A: std_logic_vector( N-1 DOWNT0 0 ) := ZERO;
    signal reduced_OR : std_logic;
    signal reduced_AND : std_logic;
    signal reduced_XOR : std_logic;

begin
    Unit_Under_Test: reduction_operators
        generic map( N => N )
        port map( A => A, reduced_OR => reduced_OR, reduced_AND => reduced_AND, reduced_XOR => reduced_XOR );

stim_proc: process

```

```
begin
    A    <= ZERO;
    wait for 150 ns;
    A    <= ONE;
    wait for 50 ns;
    A    <= TWO;
    wait for 100 ns;
    A    <= FOUR;
    wait for 50 ns;
    A    <= FIVE;
    wait for 50 ns;
    A    <= SIX;
    wait for 50 ns;
    A    <= SEVEN;
    wait;
end process;
end test;
```

```
-- *****
-- **** STUDENT: 64200238
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;
```

```
architecture test of reduction_operators_tb is
```

```
    component reduction_operators
        generic ( N: natural := 10 );
        port (
            A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
            reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC
        );
    end component;
```

```
    signal reduced_OR, reduced_AND, reduced_XOR: std_logic ;
    signal A: STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
```

```
    constant bin_0 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( others => '0' );
    constant bin_1 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', others => '0' );
    constant bin_2 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 => '1', others => '0' );
    constant bin_3 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', others => '0' );
    constant bin_4 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 2 => '1', others => '0' );
    constant bin_5 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 2 => '1', others => '0' );
    constant bin_6 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 => '1', 2 => '1', others => '0' );
    constant bin_7 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', 2 => '1', others => '0' );
```

```
begin
```

```
    UUT: reduction_operators
        generic map (
            N=>N
```

```

    )
    port map(
A => A, reduced_OR => reduced_OR, reduced_AND => reduced_AND, reduced_XOR => reduced_XOR
    );

stim_proc: process
begin
    A    <= bin_0;
    wait for 150 ns;

    A    <= bin_1;
    wait for 50 ns;

    A    <= bin_2;
    wait for 100 ns;

--  A    <= bin_3;
--  wait for 50 ns;

    A    <= bin_4;
    wait for 50 ns;

    A    <= bin_5;
    wait for 50 ns;

    A    <= bin_6;
    wait for 50 ns;

    A    <= bin_7;

    wait;

end process;

end test;

```

```
-- *****
-- **** STUDENT: 64200288
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
```

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity reduction_operators_tb is
```

```
generic ( N: natural := 3 );
```

```
end reduction_operators_tb;
```

```
architecture test of reduction_operators_tb is
```

```
    component reduction_operators
```

```
        generic ( N: natural := 10 );
```

```
        port (
```

```
            A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
```

```
            reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC
```

```
        );
```

```
    end component;
```

```
    signal reduced_OR, reduced_AND, reduced_XOR: std_logic ;
```

```
    signal A: STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
```

```
    constant bin_0 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( others => '0' );
```

```
    constant bin_1 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', others => '0' );
```

```
    constant bin_2 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 => '1', others => '0' );
```

```
    constant bin_3 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', others => '0' );
```

```
    constant bin_4 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 2 => '1', others => '0' );
```

```
    constant bin_5 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 2 => '1', others => '0' );
```

```
    constant bin_6 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 => '1', 2 => '1', others => '0' );
```

```
    constant bin_7 : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', 2 => '1', others => '0' );
```

```
begin
```

```
    UUT: reduction_operators
```

```
    generic map (
```



```

        N=>N
    )
    port map(
        A => A,          reduced_OR => reduced_OR, reduced_AND => reduced_AND,          reduced_XOR => reduced_XOR
    );

stim_proc: process
begin
    A    <= bin_0;
    wait for 150 ns;
    A    <= bin_1;
    wait for 50 ns;
    A    <= bin_2;
    wait for 100 ns;
    A    <= bin_4;
    wait for 50 ns;
    A    <= bin_5;
    wait for 50 ns;
    A    <= bin_6;
    wait for 50 ns;
    A    <= bin_7;
    wait;
end process;

end test;

```

```

-- *****
-- **** STUDENT: 64200296
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic (    N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is
    component reduction_operators is
        generic (    N: natural := 10 );
        port (
            A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
            reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
    end component;

    signal
        A: STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( others => '0' );
        reduced_OR, reduced_AND, reduced_XOR : std_logic;

    constant b0 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( others => '0' );      -- 000
    constant b1 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 0=>'1', others => '0' );  -- 001
    constant b2 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 1=>'1', others => '0' );  -- 010
    constant b3 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 1 downto 0 => '1', others => '0' );  -- 011
    constant b4 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 2=>'1', others => '0' );  -- 100
    constant b5 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 2=>'1',0=>'1', others => '0' );  -- 101
    constant b6 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 2 downto 1 => '1', others => '0' );  -- 110
    constant b7 : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 2 downto 0 => '1', others => '0' );  -- 111

begin

    UUT:reduction_operators
    generic map ( N=>N )
    port map
        ( A=>A, reduced_OR=>reduced_OR, reduced_AND=>reduced_AND, reduced_XOR=>reduced_XOR );
    stim_proc : process
    begin
        A    <=b0;

```

```
    wait for 150 ns;  
    A    <=b1;  
    wait for 50 ns;  
    A    <=b2;  
    wait for 100 ns;  
    A    <=b4;  
    wait for 50 ns;  
    A    <=b5;  
    wait for 50 ns;  
    A    <=b6;  
    wait for 50 ns;  
    A    <=b7;  
    wait;  
end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64200385
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- moe
entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    component reduction_operators
    generic ( N : integer := 4096 );
    Port ( A : in STD_LOGIC_VECTOR( N-1 downto 0 );
    reduced_OR : out STD_LOGIC;
    reduced_AND : out STD_LOGIC;
    reduced_XOR : out STD_LOGIC );
    end component;

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

    constant ONE : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', others => '0' );

    constant TWO : std_logic_vector( N-1 DOWNT0 0 ) := ( 1 => '1', others => '0' );

    constant THREE : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 | 1 => '1', others => '0' );

    constant FOUR : std_logic_vector( N-1 DOWNT0 0 ) := ( 2 => '1', others => '0' );

    constant FIVE : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 => '1', 2 => '1', others => '0' );

    constant SIX : std_logic_vector( N-1 DOWNT0 0 ) := ( 1 | 2 => '1', others => '0' );

    constant SEVEN : std_logic_vector( N-1 DOWNT0 0 ) := ( 0 | 1 | 2 => '1', others => '0' );

```

```

        signal      A : std_logic_vector ( N-1 DOWNTO 0 ) := ZERO;
        signal      reduced_OR : STD_LOGIC;
        signal      reduced_AND : STD_LOGIC;
        signal      reduced_XOR : STD_LOGIC;

begin

        UUT: reduction_operators
generic map (
N => N
)

port map(
A => A,
reduced_OR => reduced_OR,
reduced_AND => reduced_AND,
reduced_XOR => reduced_XOR
);

        stim_proc: process
begin
        -- Vrednost A = "000" ( 0 )
A      <= ZERO;
wait for 150 ns;
        -- Vrednost A = "001" ( 1 )
A      <= ONE;
wait for 50 ns;
        -- Vrednost A = "010" ( 2 )
A      <= TWO;
wait for 100 ns;
        -- Vrednost A = "100" ( 4 )
A      <= FOUR;
wait for 50 ns;
        -- Vrednost A = "101" ( 5 )
A      <= FIVE;
wait for 50 ns;
        -- Vrednost A = "110" ( 6 )
A      <= SIX;
wait for 50 ns;
        -- Vrednost A = "111" ( 7 )
A      <= SEVEN;

```

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

```

-- *****
-- **** STUDENT: 64210132
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    -- Deklaracija komponente za simulacijo ( Unit under test - UUT )
    COMPONENT reduction_operators
        generic ( N: natural := 3 );
        port ( A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
              reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
    END COMPONENT;

    -- Konstante za vhode vezja
    constant dec_0: std_logic_vector ( 2 downto 0 ) := ( others => '0' );
    constant dec_1: std_logic_vector ( 2 downto 0 ) := "001";
    constant dec_2: std_logic_vector ( 2 downto 0 ) := "010";
    constant dec_3: std_logic_vector ( 2 downto 0 ) := "011";
    constant dec_4: std_logic_vector ( 2 downto 0 ) := "100";
    constant dec_5: std_logic_vector ( 2 downto 0 ) := "101";
    constant dec_6: std_logic_vector ( 2 downto 0 ) := "110";
    constant dec_7: std_logic_vector ( 2 downto 0 ) := ( others => '1' );

    -- Vhodi
    signal A : std_logic_vector ( N-1 downto 0 ) := ( others => '0' );

    -- Izhodi
    signal reduced_OR, reduced_AND, reduced_XOR : std_logic;

begin

```

```

    UUT: reduction_operators
        generic map ( N => N )
        port map
            (
                A => A,
                reduced_OR => reduced_OR,
                reduced_AND => reduced_AND,
                reduced_XOR => reduced_XOR );

    stim_proc: process
begin
    wait for 150 ns;

    A    <= dec_0;

    wait for 50 ns;
    A    <= dec_1;

    wait for 50 ns;
    A    <= dec_2;

    wait for 50 ns;
    -- A  <= dec_3;

    wait for 50 ns;
    A    <= dec_4;

    wait for 50 ns;
    A    <= dec_5;

    wait for 50 ns;
    A    <= dec_6;

    wait for 50 ns;
    A    <= dec_7;

    wait;
end process;

end test;

```



```

-- *****
-- **** STUDENT: 64210290
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek:
Procesa za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic ( N : natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    component reduction_operators
        generic( N : natural := 10 );
        port( a : in std_logic_vector ( N-1 downto 0 );
              reduced_OR, reduced_AND, reduced_XOR : out std_logic );
    end component;

    signal      reduced_or : std_logic := '0';
    signal      reduced_and : std_logic := '0';
    signal      reduced_xor : std_logic := '0';

    constant    clk_period : time := 10 ns;
    signal      clk : std_logic;

    signal      a : std_logic_vector ( N-1 downto 0 );

    constant    b_0 : std_logic_vector ( N-1 downto 0 ) := ( others=>'0' );
    constant    b_1 : std_logic_vector ( N-1 downto 0 ) := ( 0 => '1', others=>'0' );
    constant    b_2 : std_logic_vector ( N-1 downto 0 ) := ( 1 => '1', others=>'0' );
    constant    b_3 : std_logic_vector ( N-1 downto 0 ) := ( 1|0 => '1', others=>'0' );
    constant    b_4 : std_logic_vector ( N-1 downto 0 ) := ( 2 => '1', others=>'0' );
    constant    b_5 : std_logic_vector ( N-1 downto 0 ) := ( 2|0 => '1', others=>'0' );
    constant    b_6 : std_logic_vector ( N-1 downto 0 ) := ( 2|1 => '1', others=>'0' );
    constant    b_7 : std_logic_vector ( N-1 downto 0 ) := ( 2 downto 0 => '1', others=>'0' );

```

begin

```
uut: reduction_operators
    generic map ( N => N )
    port map (    A => a,
reduced_XOR => reduced_xor );
```

```
clk_process: process
begin
    clk    <= '0';
    wait for clk_period/2;
    clk    <= '1';
    wait for clk_period/2;
end process;
```

```
stim_proc: process
begin
-- 0
    a    <= b_0;
    wait for 150 ns;

-- 1
    a    <= b_1;
    wait for 50 ns;

-- 2
    a    <= b_2;
    wait for 100ns;

-- 4
    a    <= b_4;
    wait for 50 ns;

-- 5
    a    <= b_5;
    wait for 50 ns;

-- 6
    a    <= b_6;
    wait for 50 ns;
```

reduced_OR => reduced_or,

reduced_AND => reduced_and,

```
-- 7 a <= b_7;  
    wait;  
end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64210382
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic (      N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is
    component reduction_operators is
        generic (      N: Natural := 10 );
        port (          A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
                reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
    end component;

    constant ZERO : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( others => '0' );
    constant ONE  : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', others => '0' );
    constant TWO  : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 => '1', others => '0' );
    constant THREE : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 | 1 => '1', others => '0' );
    constant FOUR  : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 2 => '1', others => '0' );
    constant FIVE  : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 | 2 => '1', others => '0' );
    constant SIX   : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 | 2 => '1', others => '0' );
    constant SEVEN : STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 | 1 | 2 => '1', others => '0' );
    signal A: STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
    signal reduced_OR, reduced_AND, reduced_XOR: STD_LOGIC;

begin
    U1: reduction_operators
        generic map ( N => N )
        port map (
            A => A,          reduced_OR => reduced_OR, reduced_AND => reduced_AND,          reduced_XOR => reduced_XOR
        );

    stim_proc: process
    begin
        A      <= ZERO;

```

```
wait for 150 ns;  
A    <= ONE;  
wait for 50 ns;  
A    <= TWO;  
wait for 100 ns;  
A    <= FOUR;  
wait for 50 ns;  
A    <= FIVE;  
wait for 50 ns;  
A    <= SIX;  
wait for 50 ns;  
A    <= SEVEN;  
wait;  
end process;  
end test;
```

```

-- *****
-- **** STUDENT: 64210384
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

```

```

architecture test of reduction_operators_tb is

```

```

--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --

```

```

-- Deklaracija komponente

```

```

component reduction_operators is
generic ( N: natural := 10 );
port ( A: in STD_LOGIC_VECTOR ( N-1 downto 0 );
       reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
end component;

```

```

-- Deklaracija konstant

```

```

constant zero: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( others => '0' );
constant one: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( 0 => '1', others => '0' );
constant two: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( 1 => '1', others => '0' );
constant three: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( 0|1 => '1', others => '0' );
constant four: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( 2 => '1', others => '0' );
constant five: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( 0|2 => '1', others => '0' );
constant six: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( 1|2 => '1', others => '0' );
constant seven: STD_LOGIC_VECTOR( N-1 downto 0 ) := ( 0|1|2 => '1', others => '0' );

```

```

-- Notranji signali

```

```

signal A: STD_LOGIC_VECTOR ( N-1 downto 0 ) := zero;
signal reduced_OR: STD_LOGIC;
signal reduced_AND: STD_LOGIC;
signal reduced_XOR: STD_LOGIC;

```



```

-- *****
-- **** STUDENT: 64210386
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Pri povezovalnem stavku manjka generic map (N => N).
Procesa za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje.
-- *****

```

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity reduction_operators_tb is
    generic ( N: natural := 3 );
end reduction_operators_tb;
```

```
architecture test of reduction_operators_tb is
```

```
    component reduction_operators is generic ( N: Natural := 1024 );
    Port ( A: in STD_LOGIC_VECTOR ( N-1 DOWNTO 0 );
           reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
    end component;
```

```
    signal A : STD_LOGIC_VECTOR ( N-1 DOWNTO 0 ) := ( 0|1 => '1', others => '0' );
```

```
    signal reduced_OR : STD_LOGIC := '0';
    signal reduced_AND : STD_LOGIC := '0';
    signal reduced_XOR : STD_LOGIC := '0';
```

```
    constant clk_period : time := 50 ns;
    signal clk : STD_LOGIC;
```

```
begin
```

```
    uut : reduction_operators PORT MAP ( A => A, reduced_OR => reduced_OR, reduced_AND => reduced_AND, reduced_XOR
=> reduced_XOR );
```

```
    clk_process :process
    begin
```

```
        clk    <='0';
        wait for clk_period/2;
        clk    <='1';
        wait for clk_period/2;
```



```
end process;
```

```
stim_proc: process
```

```
begin
```

```
    A    <= ( '0', '0', '0' );
```

```
    wait for clk_period*3;
```

```
    A    <= ( '0', '0', '1' );
```

```
    wait for clk_period;
```

```
    A    <= ( '0', '1', '0' );
```

```
    wait for clk_period*2;
```

```
    A    <= ( '1', '0', '0' );
```

```
    wait for clk_period;
```

```
    A    <= ( '1', '0', '1' );
```

```
    wait for clk_period;
```

```
    A    <= ( '1', '1', '0' );
```

```
    wait for clk_period;
```

```
    A    <= ( '1', '1', '1' );
```

```
    wait;
```

```
end process;
```

```
end test;
```

```

-- *****
-- **** STUDENT: 64210445
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Pri povezovalnem stavku manjka generic map (N => N).
Procesa za krmiljenje signala ure ne rabite, ker gre za kombinacijsko vezje.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic (    N: natural := 1024 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    component reduction_operators is generic (    N: Natural := 1024 );
    port (    A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
            reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
    end component;

    -- Vhodi
    signal    A : STD_LOGIC_VECTOR ( N-1 DOWNT0 0 ) := ( 0|1 => '1', others => '0' );

    -- Izhodi
    signal    reduced_OR : STD_LOGIC := '0';
    signal    reduced_AND : STD_LOGIC := '0';
    signal    reduced_XOR : STD_LOGIC := '0';

    constant    clk_period : time := 50 ns;    -- perioda / frekvenca urinega impulza
    signal    clk : STD_LOGIC;

begin

    uut: reduction_operators PORT MAP ( A => A, reduced_OR => reduced_OR, reduced_AND => reduced_AND, reduced_XOR
=> reduced_XOR );

    clk_process :process

```

```

begin
    clk    <= '0';
    wait for clk_period/2;
    clk    <= '1';
    wait for clk_period/2;
end process;

stim_proc: process
begin
    A      <= ( '0', '0', '0' );
wait for clk_period*3;
    A      <= ( '0', '0', '1' );
    wait for clk_period;
    A      <= ( '0', '1', '0' );

    wait for clk_period*2;
    A      <= ( '1', '0', '0' );
    wait for clk_period;
    A      <= ( '1', '0', '1' );
    wait for clk_period;
    A      <= ( '1', '1', '0' );
    wait for clk_period;
    A      <= ( '1', '1', '1' );
    wait;

end process;

end test;

```

```

-- *****
-- **** STUDENT: 64210455
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Upoštevajte navodila naloge in ne spreminjajte podatkov entitete – sicer vam naloge ne morem
popraviti!
-- *****

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

entity reduction_operators_tb is
  generic ( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

  component reduction_operators is
    generic ( WIDTH: natural := 4096 );
    port (
      A: in std_logic_vector( WIDTH - 1 downto 0 );
      reduced_OR, reduced_AND, reduced_XOR: out std_logic
    );
  end component;

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

  signal A : std_logic_vector( N - 1 downto 0 ) := ZERO;
  signal reduced_OR, reduced_AND, reduced_XOR : std_logic;

begin

  UUT: reduction_operators
    generic map ( WIDTH => N )
    port map (
      A => A,
      reduced_OR => reduced_OR,
      reduced_AND => reduced_AND,

```

```

reduced_XOR => reduced_XOR
);

stim_proc: process
variable or_result, and_result, xor_result : std_logic;
begin
for idx in 0 to 2**N - 1 loop
    -- Nastavi vhodni signal          za testni modul in čakaj
A    <= std_logic_vector( to_unsigned( idx, A'length ) );
wait for 50 ns;

    -- Inicializiraj rezultate za primerjavo
or_result := '0';
and_result := '1';
xor_result := '0';

    -- Izračunaj operacije OR, AND in XOR
for jdx in 0 to N - 1 loop
or_result := or_result or A( jdx );
and_result := and_result and A( jdx );
xor_result := xor_result xor A( jdx );
end loop;

    -- Preveri rezultate in porocaj o napakah
assert ( or_result = reduced_OR )
report "Test failed for reduced_OR at value = " & integer'image( idx ) severity error;
assert ( and_result = reduced_AND )
report "Test failed for reduced_AND at value = " & integer'image( idx ) severity error;
assert ( xor_result = reduced_XOR )
report "Test failed for reduced_XOR at value = " & integer'image( idx ) severity error;
end loop;
wait;
end process;

end test;

```

```

-- *****
-- **** STUDENT: 64210457
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic (    N: natural := 3 ); -- s tem je definirana velikost vektorja A
end reduction_operators_tb;

architecture test of reduction_operators_tb is

-- Notranji signali za povezavo s komponento UUT
signal      A: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ); -- Vhodni signal
signal      reduced_OR, reduced_AND, reduced_XOR: STD_LOGIC; -- Izhodni signali

component reduction_operators is
generic (    N: natural := 3 );
port (      A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
           reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
end component;

-- za vse te primere bomo stestirali naÅe funkcije torej vektor A bo vsakiÅ drugacen
constant    n0: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( others => '0' ); -- 0
constant    n1: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', others => '0' ); -- 1
constant    n2: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 => '1', others => '0' ); -- 010
constant    n3: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', others => '0' ); -- 011
constant    n4: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 2 => '1', others => '0' ); -- 100
constant    n5: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 2 => '1', others => '0' ); -- 101
constant    n6: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 1 => '1', 2 => '1', others => '0' ); -- 110
constant    n7: STD_LOGIC_VECTOR( N-1 DOWNT0 0 ) := ( 0 => '1', 1 => '1', 2 => '1', others => '0' ); -- 111

begin

 uut: reduction_operators
generic map ( N => N ) -- Povezava parametra N
port map (

```

```

A => A,      -- Povezava vektorjev ASS
reduced_OR => reduced_OR,
reduced_AND => reduced_AND,
reduced_XOR => reduced_XOR
);

      -- Proces za stimulacijo vhodov
stim_proc: process
begin
      -- Testiranje vrednosti A in opazovanje izhodov
A      <= n0;
wait for 50 ns;
A      <= n1;
wait for 50 ns;
A      <= n2;
wait for 50 ns;
A      <= n3;
wait for 50 ns;
A      <= n4;
wait for 50 ns;
A      <= n5;
wait for 50 ns;
A      <= n6;
wait for 50 ns;
A      <= n7;
wait for 50 ns;
wait;
end process;

end test;

```

```

-- *****
-- **** STUDENT: 64240429
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    signal        reduced_OR, reduced_AND, reduced_XOR : STD_LOGIC;

    component reduction_operators is
    -- constants for input number
    generic ( N:Natural );
    port (      A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
              reduced_OR, reduced_AND, reduced_XOR: out STD_LOGIC );
    end component;

    -- constant A : std_logic_vector( N-1 downto 0 ) := ( others => '0' ); -- Looks impossible to implement
    signal      A : std_logic_vector( N-1 downto 0 ); -- Input signal for UUT

    constant    ZERO : std_logic_vector( N-1 downto 0 ) := ( others => '0' ); -- 0 I fail to see another
constant    ONE : std_logic_vector( N-1 downto 0 ) := ( 0 => '1', others => '0' ); -- 1 option to add 'others'
constant    TWO : std_logic_vector( N-1 downto 0 ) := ( 1 => '1', others => '0' ); -- 2 as it sais in the
constant    THREE : std_logic_vector( N-1 downto 0 ) := ( 0 => '1', 1 => '1', others => '0' ); -- 3 task sheet
constant    FOUR : std_logic_vector( N-1 downto 0 ) := ( 2 => '1', others => '0' ); -- 4
constant    FIVE : std_logic_vector( N-1 downto 0 ) := ( 0 => '1', 2 => '1', others => '0' ); -- 5
constant    SIX : std_logic_vector( N-1 downto 0 ) := ( 1 => '1', 2 => '1', others => '0' ); -- 6
constant    SEVEN : std_logic_vector( N-1 downto 0 ) := ( 0 => '1', 1 => '1', 2 => '1', others => '0' ); -- 7

begin

```



```
UUT:entity work.reduction_operators-- UUT-- entity work.reduction_operators
```

```
generic map (  
  N=>N  
)
```

```
port map (  
  reduced_OR => reduced_OR, reduced_AND => reduced_AND, reduced_XOR => reduced_XOR, A => A  
);
```

```
stim_proc: process  
begin
```

```
  A      <= ZERO;  
  wait for 150 ns;  
  A      <= ONE;  
  wait for 50 ns;  
  A      <= THREE;  
  wait for 50 ns;  
  A      <= FOUR;  
  wait for 50 ns;  
  A      <= FIVE;  
  wait for 50 ns;  
  A      <= SIX;  
  wait for 50 ns;  
  A      <= SEVEN;  
  wait;
```

```
  end process;
```

```
end test;
```

```

-- *****
-- **** STUDENT: 64240430
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity reduction_operators_tb is
generic ( N: natural := 3 );
end reduction_operators_tb;

```

```

architecture test of reduction_operators_tb is

```

```

-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- --
-- Deklaracija komponente za simulacijo
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- --

```

```

component reduction_operators
generic ( N: Natural );
port (
A: in STD_LOGIC_VECTOR ( N-1 DOWNT0 0 );
reduced_OR : out STD_LOGIC;
reduced_AND: out STD_LOGIC;
reduced_XOR: out STD_LOGIC
);
end component;

```

```

-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- --
-- Signali
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
-- -- --
-- nic -> 000

```

```

constant nic : std_logic_vector ( N-1 downto 0 ) := ( others => '0' );
-- ena -> 001
constant ena : std_logic_vector ( N-1 downto 0 ) := ( 0 => '1', others => '0' );
-- dve -> 010
constant dve : std_logic_vector ( N-1 downto 0 ) := ( 1 => '1', others => '0' );

```



```
stim_proc: process
begin

    A    <= nic;
    wait for 150 ns;

    A    <= ena;
    wait for 50 ns;

    A    <= dve;
    wait for 100 ns;

--    A    <= tri;
--    wait for 0 ns;

    A    <= stiri;
    wait for 50 ns;

    A    <= pet;
    wait for 50 ns;

    A    <= sest;
    wait for 50 ns;

    A    <= sedem;
    wait;

end process;

end test;
```

```

-- *****
-- **** STUDENT: 64210113
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity reduction_operators_tb is
generic (    N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    signal A: std_logic_vector(N-1 downto 0);
    signal reduced_OR, reduced_AND, reduced_XOR: std_logic;

    component reduction_operators is
        generic(N: Natural:=4096);
        port(A: in std_logic_vector (N-1 downto 0);
              reduced_OR, reduced_AND, reduced_XOR: out std_logic);
    end component;

    constant nic  : std_logic_vector(N-1 downto 0):= (others => '0');
    constant ena  : std_logic_vector(N-1 downto 0):= (0 => '1', others => '0');
    constant dva  : std_logic_vector(N-1 downto 0):= (1 => '1', others => '0');
    constant tri  : std_logic_vector(N-1 downto 0):= (0 => '1', 1 => '1', others => '0');
    constant stiri: std_logic_vector(N-1 downto 0):= (2 => '1', others => '0');
    constant pet  : std_logic_vector(N-1 downto 0):= (2 => '1', 0 => '1', others => '0');
    constant sest : std_logic_vector(N-1 downto 0):= (2 => '1', 1 => '1', others => '0');
    constant sedem: std_logic_vector(N-1 downto 0):= (2 => '1', 1 => '1', 0 => '1', others => '0');

begin

Link: reduction_operators
generic map (N => N)
port map (    A => A,          reduced_OR => reduced_OR,          reduced_AND => reduced_AND,
              reduced_XOR => reduced_XOR          );

```

```
stim_proc: process
begin

    A <= nic;    wait for 150 ns;
    A <= ena;    wait for 50 ns;
    A <= dva;    wait for 100 ns;
    A <= stiri;  wait for 50 ns;
    A <= pet;    wait for 50 ns;
    A <= sest;   wait for 50 ns;
    A <= sedem;  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;

entity reduction_operators_tb is
generic (    N: natural := 3 );
end reduction_operators_tb;

architecture test of reduction_operators_tb is

    FILE RESULTS: TEXT OPEN APPEND_MODE IS "reduction_operators.csv";

    component reduction_operators is
generic (    N: Natural := 10 );
port ( A: in STD_LOGIC_VECTOR (N-1 DOWNT0 0);
        reduced_OR, reduced_AND, reduced_XOR : out STD_LOGIC);
    end component;

    for all: reduction_operators use entity work.reduction_operators(vhdl_93_process);

    constant ZERO : std_logic_vector(N-1 DOWNT0 0) := (others => '0');      -- vsi elementi na 0
    constant ONE  : std_logic_vector(N-1 DOWNT0 0) := (0 => '1', others => '0');  -- enka (LSB = 1, ostali 0)
    constant TWO  : std_logic_vector(N-1 DOWNT0 0) := (1 => '1', others => '0');
    constant THREE : std_logic_vector(N-1 DOWNT0 0) := (0 | 1 => '1', others => '0');  --POZOR: definicija
konstante z or operatorjem /
    constant FOUR : std_logic_vector(N-1 DOWNT0 0) := (2 => '1', others => '0');
    constant FIVE : std_logic_vector(N-1 DOWNT0 0) := (0 => '1', 2 => '1', others => '0');
    constant SIX  : std_logic_vector(N-1 DOWNT0 0) := (1 | 2 => '1', others => '0');
    constant SEVEN : std_logic_vector(N-1 DOWNT0 0) := (0 | 1 | 2 => '1', others => '0');

    signal A : std_logic_vector (N-1 DOWNT0 0) := ZERO;
    signal reduced_OR, reduced_AND, reduced_XOR : STD_LOGIC;

begin

```

```

U1: reduction_operators
generic map (N => N)
port map (    A => A,
            reduced_OR=> reduced_OR,
            reduced_AND=> reduced_AND,
            reduced_XOR=> reduced_XOR
            );

stim_proc: process
    variable HDR_line : LINE;

    PROCEDURE Log_variables(
        A : std_logic_vector (N-1 DownTo 0);
        reduced_OR: std_logic;
        reduced_AND: std_logic;
        reduced_XOR : std_logic
    ) IS
    VARIABLE RES_LINE : LINE;
    BEGIN
        hwrite(RES_LINE, '0' & A, right, 4);
        write(RES_LINE, string'(","));
        write(RES_LINE, reduced_OR, right, 1);
        write(RES_LINE, string'(","));
        write(RES_LINE, reduced_AND, right, 1);
        write(RES_LINE, string'(","));
        write(RES_LINE, reduced_XOR , right, 1);
        writeline(RESULTS, RES_LINE);
    END;

begin
    write(HDR_line, string'("Datoteka testnih vrednosti z xor, and in or unarnimi operatorji z zaporednim
izracunom XOR"));
    writeline(RESULTS, HDR_line);
    write(HDR_line, string'("A, reduced_OR, reduced_AND, reduced_XOR"));
    writeline(RESULTS, HDR_line);
    A <= ZERO;                                --0
    wait for 50 ns;
    Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);
    A <= ONE;                                  --1
    wait for 50 ns;
    Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);

```



```

A <= TWO;          --2
wait for 50 ns;
Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);
A <= THREE;  --3
wait for 50 ns;
Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);
A <= FOUR;    --4
wait for 50 ns;
Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);
A <= FIVE;     --5
wait for 50 ns;
Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);
A <= SIX;      --6
wait for 50 ns;
Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);
A <= SEVEN;  --7
wait for 50 ns;
Log_variables(A, reduced_OR, reduced_AND, reduced_XOR);
wait;
end process;
end test;

```