

## Ocenjevanje n-bitnega CLA seštevalnika

Ocenjevanje n-bitnega CLA seštevalnika .....	1
-- **** STUDENT: 64000225 .....	3
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	3
-- **** STUDENT: 64190088 .....	4
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Komponente cla_gp dejansko ne rabite. Manjkajo inicializacije signalov (others => '0').	4
-- **** STUDENT: 64200100 .....	5
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Gout se ne postavi za seštevanje, ampak ostane nedefiniran, zato ker gre vektor za Gint do 1, ne do 0. Komponente cla_gp dejansko ne rabite. Manjkajo inicializacije signalov (others => '0').	5
-- **** STUDENT: 64200112 .....	6
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	6
-- **** STUDENT: 64200163 .....	7
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	7
-- **** STUDENT: 64200238 .....	8
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Komponente cla_gp dejansko ne rabite. Manjkajo inicializacije signalov (others => '0').	8
-- **** STUDENT: 64200288 .....	9
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Komponente cla_gp dejansko ne rabite. Manjkajo inicializacije signalov (others => '0').	9
-- **** STUDENT: 64200296 .....	10
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	10
-- **** STUDENT: 64200385 .....	11
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Interni signal Gint se izračuna kot $\text{and\_reduce}(P(n-1 \text{ downto } i))$ and $G(i-1)$ ; ne $\text{downto } 1!$ Manjkajo inicializacije signalov (others => '0').	11
-- **** STUDENT: 64210113 .....	12
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	12
-- **** STUDENT: 64210290 .....	13
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	13
-- **** STUDENT: 64210382 .....	14
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	14
-- **** STUDENT: 64210384 .....	15
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Manjkajo inicializacije signalov (others => '0').	15
-- **** STUDENT: 64210386 .....	16

-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Gout se ne postavi za seštevanje, ampak ostane nedefiniran, zato ker gre vektor za Gint do 1, ne do 0. Manjkajo inicializacije signalov (others => '0').	16
-- **** STUDENT: 64210445		17
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Manjkajo inicializacije signalov (others => '0'). Gout se ne postavi za seštevanje, ampak ostane nedefiniran, zato ker gre vektor za Gint do 1, ne do 0.	17
-- **** STUDENT: 64210455		18
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Napake sintetizatorja:	18
ERROR:HDLCompiler:806 - "cla_add_n_bit.vhd" Line 59: Syntax error near ".".		18
ERROR:HDLCompiler:854 - "cla_add_n_bit.vhd" Line 18: Unit <arch> ignored due to previous errors.		18
Izraz za Gout ni pravilen:		18
Gout <= G(n-1) OR (P(n-1) AND G(n-2)) OR		18
(P(n-1) AND P(n-2) AND G(n-3)) OR		18
(P(n-1) AND P(n-2) AND P(n-3) AND G(n-4)) OR		18
(P(n-1) AND P(n-2) AND P(n-3) AND ... AND P(0) AND G(0));		18
Operator ... ne obstaja v VHDL (obstaja pa v Cju).		18
Za izračun Pout ne morete uporabiti operatorja others kot ste ga:		18
Pout <= '1' WHEN P = (OTHERS => '1') ELSE '0';		18
ker sintetizator vrne napako:		18
ERROR:HDLCompiler:136 - "cla_add_n_bit.vhd" Line 53: OTHERS choice cannot be used in unconstrained array aggregate. Morali bi tvoriti konstanto SAME_ENKE:		18
constant SAME_ENKE : std_logic_vector(n-1 downto 0) := (others => '1');		18
Nato pa nad vektorjem P lahko delate primerjavo med polji v smislu, kot ste napisali: ...		18
Pout <= '1' WHEN P = SAME_ENKE ELSE '0';		18
-- **** STUDENT: 64210457		20
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Manjkajo inicializacije signalov (others => '0').	20
-- **** STUDENT: 64240429		21
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Manjkajo inicializacije signalov (others => '0').	21
-- **** STUDENT: 64240430		23
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Manjkajo inicializacije signalov (others => '0').	23
-- **** STUDENT: 64210132		24
-- KOMENTARJI K OCENI NALOGE	-- Matej Možek: Manjkajo inicializacije signalov (others => '0'). Izraz za g_int ni pravilen – indeks mora iti do 1. Pravilno je ...	24
g_int(i) <= and_reduce( P( n-1 downto i ) ) and G( i-1 );		24
Izraz za Gout je napačen – pravilno je:		24
Gout <= G( n-1 ) or or_reduce( g_int );		24
NASLEDNJIČ KODO NALOŽITE V USTREZEN RAZDELEK (OSTALO_x), KJER JE X ŠTEVILKA DOMAČE NALOGE		24
-- **** PREDLOGA VAJE		25

```

-- *****
-- **** STUDENT: 64000225
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;
ARCHITECTURE NDV OF cla_add_n_bit IS

    -- vektor vmesnih prenosov in funkcij tvorjenja in sirjenja
    SIGNAL      G, P : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      C : STD_LOGIC_VECTOR( n DOWNT0 0 );
    SIGNAL      S_sig : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      Gint : STD_LOGIC_VECTOR( n-1 DOWNT0 1 );

BEGIN
    C( 0 ) <= Cin;

    -- namesto vmesnih stopenj zapisemo raje for ... generate zanko,

    stages: FOR i IN 0 TO n-1 GENERATE
        G( i ) <= X( i ) and Y( i );      -- funkcija tvorjenja ( generate )
        P( i ) <= X( i ) xor Y( i );      -- funkcija sirjenja ( propagate )
        S_sig( i ) <= X( i ) xor Y( i ) xor C( i ); -- vsota
        C( i+1 ) <= G( i ) or ( P( i ) and C( i ) ); -- izhodni prenos
    stopnje
    END GENERATE;

    g_internal: FOR i IN n-1 downto 1 GENERATE
        Gint( i ) <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    END GENERATE;

    S <= S_sig;      -- povezava vsote na izhodni signal

    Cout <= C( n );  -- tvorba izhodnega prenosa ( carry )

    Pout <= AND_REDUCE( P ); -- Pout becomes true, whenever all internal
propagates are true

    Gout <= G( n-1 ) or OR_REDUCE( Gint );      -- perform or between all
intermediate results

END NDV;

```

```

-- *****
-- **** STUDENT: 64190088
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Komponente cla_gp dejansko ne rabite. Manjkajo inicializacije
signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

COMPONENT cla_gp IS
    PORT ( Cin, x, y : IN STD_LOGIC;
          s, Cout, g, p : OUT STD_LOGIC );
END COMPONENT;

-- vektor vmesnih prenosov in funkcij tvorjenja in sirjenja
SIGNAL      C : STD_LOGIC_VECTOR( n DOWNT0 0 );
SIGNAL      G, P : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
SIGNAL      G_int: STD_LOGIC_VECTOR( n-1 DOWNT0 1 );
SIGNAL      S_sig : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );

BEGIN
-- namesto vmesnih stopenj zapisemo raje for ... generate zanko, -- s katero
realiziramo n port map stavkov
    C( 0 ) <= Cin;
    stages: FOR i IN 0 TO n-1 GENERATE
        G( i ) <= X( i ) and Y( i );      -- generate
        P( i ) <= X( i ) xor Y( i );      -- propagate
        S_sig( i ) <= X( i ) xor Y( i ) xor C( i ); -- sum
        C( i+1 ) <= G( i ) or ( P( i ) and C( i ) ); -- carry
    END GENERATE;

    generate_Gint: FOR i in n-1 downto 1 generate
        G_int( i ) <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    end generate;

    S <= S_sig;      -- povezava vsote na izhodni signal
    Cout <= C( n );  -- tvorba izhodnega prenosa ( carry )

    Pout <= AND_REDUCE( P ); -- Pout becomes true, whenever all internal
propagates are true
    Gout <= G( n-1 ) or OR_REDUCE( G_int ); -- perform or between all
intermediate results
END NDV;

```

```

-- *****
-- **** STUDENT: 64200100
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Gout se ne postavi za seštevanje, ampak ostane nedefiniran, zato ker
gre vektor za Gint do 1, ne do 0. Komponente cla_gp dejansko ne rabite. Manjkajo
inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      : in    std_logic ;
        X, Y      : in    std_logic_vector( n-1 downto 0 );
        S         : out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout : out std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS
    component cla_gp is port(
        Cin,x,y: in std_logic;
        s,Cout,g,p:out std_logic );
    end component;

    signal      S_sig, G, P: std_logic_vector( n-1 downto 0 );
    signal      Gint: std_logic_vector( n-1 downto 0 );
    signal      C: std_logic_vector( n downto 0 );

BEGIN
    C( 0 ) <= Cin;
    stages: for i in 0 to n-1 generate
        G( i )      <= X( i ) and Y( i );
        P( i )      <= X( i ) xor Y( i );
        S_sig( i )  <= X( i ) xor Y( i ) xor C( i );
        C( i+1 )    <= G( i ) or( P( i ) and C( i ) );
    end generate;

    smth :for i in n-1 downto 1 generate
        Gint( i )   <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    end generate;

    S      <= S_sig;
    Cout   <= C( n );
    Gout   <= G( n-1 ) or OR_REDUCE( Gint );
    Pout   <= AND_REDUCE( P );
END NDV;

```

```

-- *****
-- **** STUDENT: 64200112
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

    COMPONENT cla_gp IS
    PORT ( Cin, x, y : IN STD_LOGIC;
          s, Cout, g, p : OUT STD_LOGIC );
    END COMPONENT;

    -- vektor vmesnih prenosov in funkcij tvorjenja in sirjenja
    SIGNAL      C : STD_LOGIC_VECTOR( n DOWNT0 0 );
    SIGNAL      G, P : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      S_sig : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      G_int : STD_LOGIC_VECTOR( n-1 DOWNT0 1 );

    BEGIN

    C( 0 ) <= Cin;

    stages: FOR i IN 0 TO n-1
    GENERATE
        cla_stage: cla_gp PORT MAP ( Cin => C( i ),          x => X( i ),
        y => Y( i ),          s => S_sig( i ),          Cout => C( i+1 ),          g => G( i ),
        p => P( i ) );
    END GENERATE;

    Gsig: FOR i IN n-1 downto 1
    GENERATE
        G_int( i ) <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    END GENERATE Gsig;

    S      <= S_sig;      -- povezava vsote na izhodni signal
    Cout   <= C( n );     -- tvorba izhodnega prenosa ( carry )
    Gout   <= G( n-1 ) or OR_REDUCE( G_int );      -- perform or between all intermediate
    results
    Pout   <= AND_REDUCE( p ); -- Pout becomes true, whenever all internal propagates are
    true

    END NDV;

```

```

-- *****
-- **** STUDENT: 64200163
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
library ieee;
use ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.all;

entity cla_add_n_bit is
    generic( n: natural := 8 );
    port( Cin: in std_logic;
          x, y: in std_logic_vector( n-1 downto 0 );
          S: out std_logic_vector( n-1 downto 0 );
          Gout, Pout, Cout: out std_logic
    );
end cla_add_n_bit;

architecture arch of cla_add_n_bit is
    component cla_gp is
        port( Cin, x, y: in std_logic;
              S, Cout, g, p: out std_logic
        );
    end component;

    signal      C: std_logic_vector( n downto 0 );
    signal      S_sig, g, p: std_logic_vector( n-1 downto 0 );
    signal      Gint : std_logic_vector( n-1 downto 1 );

begin
    C( 0 ) <= Cin;
    stages: for i in 0 to n-1 generate
        g( i ) <= x( i ) and y( i );
        p( i ) <= x( i ) xor y( i );
        S_sig( i ) <= x( i ) xor y( i ) xor C( i );
        C( i+1 ) <= g( i ) or ( p( i ) and C( i ) );
    end generate;
    block_generate_bit: for i in n-1 downto 1 generate
        Gint( i ) <= AND_REDUCE( p( n-1 downto i ) ) and g( i-1 );
    end generate;
    S <= S_sig;
    Gout <= g( n-1 ) or OR_REDUCE( Gint );
    Pout <= AND_REDUCE( p );
    Cout <= C( n );
end arch;

```

```

-- *****
-- **** STUDENT: 64200238
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Komponente cla_gp dejansko ne rabite. Manjkajo inicializacije
signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

    COMPONENT cla_gp IS PORT (
        Cin, x, y : IN STD_LOGIC;
        s, Cout, g, p : OUT STD_LOGIC );
    END COMPONENT;

-- vektor vmesnih prenosov in funkcij tvorjenja in sirjenja
    SIGNAL      C : STD_LOGIC_VECTOR( n DOWNTO 0 );
    SIGNAL      S_sig, G, P : STD_LOGIC_VECTOR( n-1 DOWNTO 0 );
    signal      Gint : std_logic_vector( n-1 downto 1 );

    BEGIN

        C( 0 )      <= Cin;
    stages: for i in 0 to n-1 generate
        G( i ) <= X( i ) and Y( i );      -- funkcija tvorjenja ( generate )
        P( i ) <= X( i ) xor Y( i );      -- funkcija sirjenja ( propagate )
        S_sig( i ) <= P( i ) XOR C( i );
        C( i+1 ) <= G( i ) or ( P( i ) and C( i ) );      -- izhodni prenos
stopnje
    end generate;

    Mjav: FOR i IN n-1 DOWNTO 1 GENERATE
        Gint( i ) <= AND_REDUCE( P( n-1 downto i ) ) AND G( i-1 );
    END GENERATE;

    Gout <= G( n-1 ) or OR_REDUCE( Gint );
    S <= S_sig;
    Cout <= C( n );
    Pout <= AND_REDUCE( P );

END NDV;

```



```

-- *****
-- **** STUDENT: 64200288
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Komponente cla_gp dejansko ne rabite. Manjkajo inicializacije
signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )
    -- Pout      <= AND_REDUCE( P );

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S          :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

    COMPONENT cla_gp IS PORT (
        Cin, x, y : IN STD_LOGIC;
        s, Cout, g, p : OUT STD_LOGIC );
    END COMPONENT;

    signal      C : STD_LOGIC_VECTOR( n DOWNTO 0 );
    signal      S_sig, G, P : STD_LOGIC_VECTOR( n-1 DOWNTO 0 );
    signal      Gint : std_logic_vector( n-1 downto 1 );

    BEGIN

        C( 0 )      <= Cin;
    stages: for i in 0 to n-1 generate
        G( i ) <= X( i ) and Y( i );
        P( i ) <= X( i ) xor Y( i );
        S_sig( i )      <= P( i ) XOR C( i );
        C( i+1 )      <= G( i ) or ( P( i ) and C( i ) );
    end generate;

    signalgeneration: for i in n-1 downto 1 generate
        Gint( i )      <= AND_REDUCE( P( n-1 downto i ) ) AND G( i-1 );
    end generate;

        Gout  <= G( n-1 ) or OR_REDUCE( Gint );
        S      <= S_sig;
        Cout  <= C( n );
        Pout  <= AND_REDUCE( P );

END NDV;

```

```

-- *****
-- **** STUDENT: 64200296
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

    signal      c: std_logic_vector( n downto 0 );
    signal      s_sig, g, p : std_logic_vector( n-1 downto 0 );
    signal      gint : std_logic_vector( n-1 downto 1 );
    begin
        stages: for i in 0 to n-1 generate
            g( i ) <= x( i ) and y( i );      -- funkcija tvorjenja ( generate )
            p( i ) <= x( i ) xor y( i );      -- funkcija sirjenja ( propagate )
            S_sig( i ) <= x( i ) xor y( i ) xor c( i ); -- vsota
            C( i+1 ) <= g( i ) or ( p( i ) and C( i ) ); -- izhodni prenos
        stopnje
        end generate;

        gint_stage: for i in n-1 downto 1 generate
            gint( i ) <= and_reduce( p( n-1 downto i ) ) and g( i-1 );
        end generate;

        s <= s_sig;
        cout <= c( n );
        pout <= and_reduce( P ); -- Pout becomes true, whenever all internal
propagates are true
        gout <= G( n-1 ) or OR_REDUCE( Gint );      -- perform or between all
intermediate results
        c( 0 ) <= cin;
    END NDV;

```

```

-- *****
-- **** STUDENT: 64200385
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Interni signal Gint se izračuna kot and_reduce( P( n-1 downto i ) )
and G( i-1 ); ne downto 1! Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout,Pout,Cout :    out   std_logic );
END cla_add_n_bit;
ARCHITECTURE arch OF cla_add_n_bit IS
    SIGNAL      P, G, S_sig : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      C           : STD_LOGIC_VECTOR( n DOWNT0 0 );
    SIGNAL      Gint        : STD_LOGIC_VECTOR( n-1 DOWNT0 1 );
BEGIN

    C( 0 )      <= Cin;

    stages: FOR i IN 0 TO n-1 GENERATE
        G( i )      <= X( i ) and Y( i );
        P( i ) <= X( i ) xor Y( i );
        S_sig( i )  <= X( i ) xor Y( i ) xor C( i );
        C( i+1 )    <= G( i ) or ( P( i ) and C( i ) );

    END GENERATE;

    int:
    for i in 1 to n-1 generate
        Gint( i )  <= and_reduce( P( n-1 downto 1 ) ) and G( i-1 );
    end generate;

    S      <= S_sig;
    Cout <= C( n );
    Pout <= AND_REDUCE( P );
    Gout <= G( n-1 ) or OR_REDUCE( Gint );

END arch;

```

```

-- *****
-- **** STUDENT: 64210113
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;
ARCHITECTURE NDV OF cla_add_n_bit IS

    signal      S_sig,P,G : std_logic_vector( n-1 downto 0 );
    signal      Gint : std_logic_vector( n-1 downto 1 );
    signal      C : std_logic_vector( n downto 0 );

BEGIN

C( 0 ) <= Cin;

stage0: for i in 0 to n-1 generate
    G( i ) <= X( i ) and Y( i );
    P( i ) <= X( i ) xor Y( i );
    S_sig( i )    <= X( i ) xor Y( i ) xor C( i );
    C( i+1 )      <= G( i ) or ( P( i ) and C( i ) );
end generate;

stageG: for j in n-1 downto 1 generate
    Gint( j )    <= G( j-1 ) and AND_REDUCE( P( n-1 downto j ) );
end generate;

S      <= S_sig;
Cout   <= C( n );
Gout   <= G( n-1 ) or OR_REDUCE( Gint );
Pout   <= AND_REDUCE( P );

END NDV;

```

```

-- *****
-- **** STUDENT: 64210290
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT( Cin      : in      std_logic ;
          X, Y      : in      std_logic_vector( n-1 downto 0 );
          S          : out     std_logic_vector( n-1 downto 0 );
          Gout, Pout, Cout : out  std_logic );
END cla_add_n_bit;
ARCHITECTURE NDV OF cla_add_n_bit IS

    COMPONENT cla_gp IS
        PORT (
            Cin, x, y : IN STD_LOGIC;
            S, Cout, g, p : OUT STD_LOGIC );
    END COMPONENT ;

    SIGNAL      C : STD_LOGIC_VECTOR( n DOWNTO 0 );
    SIGNAL      S_sig, G, P : STD_LOGIC_VECTOR( n-1 DOWNTO 0 );
    SIGNAL      G_sig : STD_LOGIC_VECTOR( N-1 downto 1 );

    BEGIN

    C( 0 ) <= Cin;

    stages:
    FOR i IN 0 TO n-1 GENERATE
        G( i ) <= X( i ) and Y( i );
        P( i ) <= X( i ) xor Y( i );
        S_sig( i ) <= X( i ) xor Y( i ) xor C( i );
        C( i+1 ) <= G( i ) or ( P( i ) and C( i ) );
    END GENERATE;

    Cout <= C( n );      -- tvorba izhodnega prenosa ( carry )
    S <= S_sig;          -- povezava vsote na izhodni signal
    Pout <= AND_REDUCE( P );

    g_stages:
    FOR i IN n-1 downto 1 GENERATE
        G_sig( i ) <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    END GENERATE;

    Gout <= G( n-1 ) or OR_REDUCE( G_sig );

    END NDV;

```

```

-- *****
-- **** STUDENT: 64210382
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE arch OF cla_add_n_bit IS

    SIGNAL      S_sig, G, P : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      C : STD_LOGIC_VECTOR( n DOWNT0 0 );
    SIGNAL      G_int : STD_LOGIC_VECTOR( n-1 DOWNT0 1 );
BEGIN

    C( 0 ) <= Cin;
    stages: FOR i IN 0 TO n-1 GENERATE
        G( i ) <= X( i ) and Y( i );
        P( i ) <= X( i ) xor Y( i );
        S_sig( i ) <= X( i ) xor Y( i ) xor C( i );
        C( i+1 ) <= G( i ) or ( P( i ) and C( i ) );
    END GENERATE;

    g_terms: FOR i IN n-1 downto 1 GENERATE
        G_int( i ) <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    END GENERATE;

    S <= S_sig;      -- povezava vsote na izhodni signal
    Cout <= C( n );  -- tvorba izhodnega prenosa ( carry )
    Gout <= G( n-1 ) or OR_REDUCE( G_int );
    Pout <= AND_REDUCE( P );
END arch;

```

```

-- *****
-- **** STUDENT: 64210384
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out    std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out    std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS    -- vektor vmesnih prenosov in fun. tvorjenja
in sirjenja
    SIGNAL      vsota, G, P: STD_LOGIC_VECTOR( n-1 downto 0 );
    SIGNAL      g_int: STD_LOGIC_VECTOR( n-1 downto 1 );
    SIGNAL      carry: STD_LOGIC_VECTOR( n downto 0 );
BEGIN
    carry( 0 )    <= Cin;
    stages: for i in 0 to n-1 generate
        G( i ) <= X( i ) and Y( i );      -- generate funk.
        P( i ) <= X( i ) xor Y( i );      -- propagate funk.
        vsota( i )    <= X( i ) xor Y( i ) xor carry( i );
        carry( i+1 ) <= G( i ) or ( P( i ) and carry( i ) );
    end generate;

    g_terms: for i in n-1 downto 1 generate
        g_int( i )    <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    end generate;

    S        <= vsota;
    Cout     <= carry( n );
    Gout     <= G( n-1 ) or OR_REDUCE( g_int );
    Pout     <= AND_REDUCE( P );
END NDV;

```

```

-- *****
-- **** STUDENT: 64210386
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Gout se ne postavi za seštevanje, ampak ostane nedefiniran, zato ker
gre vektor za Gint do 1, ne do 0. Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S          :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS
    SIGNAL      g_sig, p_sig, s_sig, g_ind : STD_LOGIC_VECTOR( n-1 downto 0 );
    SIGNAL      c_p: STD_LOGIC_VECTOR( n downto 0 );
    BEGIN
        c_p( 0 )      <= Cin;
        tt: for i in 0 to n-1 generate
            g_sig( i )  <= X( i ) and Y( i );      -- funkcija tvorjenja
            p_sig( i )  <= X( i ) or Y( i );        -- funkcija sirjenja
            s_sig( i )  <= X( i ) xor Y( i ) xor c_p( i );    -- povezava na
            izhodni signal
            c_p( i+1 )  <= ( p_sig( i ) and c_p( i ) ) or g_sig( i ); -- povezava
            na izhodni signal
        end generate;

        g_s: for u in n-1 downto 1 generate
            g_ind( u )  <= and_reduce( p_sig( n-1 downto u ) ) and g_sig( u-1 );
        end generate;
        S      <= s_sig;
        Cout   <= c_p( n );
        Gout   <= g_sig( n-1 ) or or_reduce( g_ind );
        Pout   <= and_reduce( p_sig );

    END NDV;

```



```

-- *****
-- **** STUDENT: 64210445
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
Gout se ne postavi za seštevanje, ampak ostane nedefiniran, zato ker gre vektor za
Gint do 1, ne do 0.
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT ( Cin          : in    std_logic ;
           X, Y          : in    std_logic_vector( n-1 downto 0 );
           S             : out   std_logic_vector( n-1 downto 0 );
           Gout, Pout, Cout : out  std_logic );
END cla_add_n_bit;
ARCHITECTURE NDV OF cla_add_n_bit IS

    SIGNAL      g_sig, p_sig, s_sig, g_i : STD_LOGIC_VECTOR( n-1 downto 0 );
    SIGNAL      c_sig: STD_LOGIC_VECTOR( n downto 0 );

BEGIN

    c_sig( 0 )    <= Cin;
    st: for i in 0 to n-1 generate
        g_sig( i )    <= x( i ) and y( i );
        p_sig( i )    <= x( i ) xor y( i );
        s_sig( i )    <= x( i ) xor y( i ) xor c_sig( i );
        c_sig( i + 1 ) <= ( p_sig( i ) and c_sig( i ) ) or g_sig( i );
    end generate;

    g_t: for i in n-1 downto 1 generate
        g_i( i )      <= and_reduce( p_sig( n-1 downto i ) ) and g_sig( i-1 );
    end generate;

    S      <= s_sig;
    Cout   <= c_sig( n );
    Gout   <= g_sig( n-1 ) or or_reduce( g_i );
    Pout   <= and_reduce( p_sig );

END NDV;

```

```
-- *****
-- **** STUDENT: 64210455
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Napake sintetizatorja:
ERROR:HDLCompiler:806 - "cla_add_n_bit.vhd" Line 59: Syntax error near ".".
ERROR:HDLCompiler:854 - "cla_add_n_bit.vhd" Line 18: Unit <arch> ignored due to
previous errors.
Izraz za Gout ni pravilen:
    Gout <= G(n-1) OR (P(n-1) AND G(n-2)) OR
        (P(n-1) AND P(n-2) AND G(n-3)) OR
        (P(n-1) AND P(n-2) AND P(n-3) AND G(n-4)) OR

        (P(n-1) AND P(n-2) AND P(n-3) AND ... AND P(0) AND G(0));
```

Operator ... ne obstaja v VHDL (obstaja pa v Cju).

Za izračun Pout ne morete uporabiti operatorja others kot ste ga:

```
Pout <= '1' WHEN P = (OTHERS => '1') ELSE '0';
```

ker sintetizator vrne napako:

```
ERROR:HDLCompiler:136 - "cla_add_n_bit.vhd" Line 53: OTHERS choice cannot be used in
unconstrained array aggregate. Morali bi tvoriti konstanto SAME_ENKE:
```

```
constant SAME_ENKE : std_logic_vector(n-1 downto 0) := (others => '1');
```

Nato pa nad vektorjem P lahko delate primerjavo med polji v smislu, kot ste napisali:

```
Pout <= '1' WHEN P = SAME_ENKE ELSE '0';
```

```
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;
```

```
ENTITY cla_add_n_bit IS
    GENERIC ( n : natural := 8 );
    PORT (
        Cin : IN std_logic;
        X : IN std_logic_vector( n-1 downto 0 );
        Y : IN std_logic_vector( n-1 downto 0 );
        S : OUT std_logic_vector( n-1 downto 0 );
        Gout : OUT std_logic;
        Pout : OUT std_logic;
        Cout : OUT std_logic
    );
END cla_add_n_bit;
```

```
ARCHITECTURE arch OF cla_add_n_bit IS
```

```
    COMPONENT cla_gp
    PORT (
        Cin : IN std_logic;
        x : IN std_logic;
        y : IN std_logic;
        s : OUT std_logic;
        Cout : OUT std_logic;
        g : OUT std_logic;
```

```
p : OUT std_logic
);
END COMPONENT;
```

```
SIGNAL          C : std_logic_vector( n downto 0 );
SIGNAL          G : std_logic_vector( n-1 downto 0 );
SIGNAL          P : std_logic_vector( n-1 downto 0 );
SIGNAL          S_sig : std_logic_vector( n-1 downto 0 );
```

```
BEGIN
```

```
C( 0 )      <= Cin;
```

```
stages: FOR i IN 0 TO n-1 GENERATE
```

```
cla_stage: cla_gp
```

```
PORT MAP (
```

```
Cin => C( i ),
```

```
x => X( i ),
```

```
y => Y( i ),
```

```
s => S_sig( i ),
```

```
Cout => C( i+1 ),
```

```
g => G( i ),
```

```
p => P( i )
```

```
);
```

```
END GENERATE;
```

```
Pout <= '1' WHEN P = ( OTHERS => '1' ) ELSE '0';
```

```
Gout <= G( n-1 ) OR ( P( n-1 ) AND G( n-2 ) ) OR
```

```
( P( n-1 ) AND P( n-2 ) AND G( n-3 ) ) OR
```

```
( P( n-1 ) AND P( n-2 ) AND P( n-3 ) AND G( n-4 ) ) OR
```

```
( P( n-1 ) AND P( n-2 ) AND P( n-3 ) AND ... AND P( 0 ) AND G( 0 ) );
```

```
Cout <= C( n );
```

```
S      <= S_sig;
```

```
END arch;
```

```

-- *****
-- **** STUDENT: 64210457
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S         :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

    SIGNAL      G, P : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      S_sig : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      Gint : STD_LOGIC_VECTOR( n-1 DOWNT0 1 );
    SIGNAL      C : STD_LOGIC_VECTOR( n DOWNT0 0 );

    BEGIN
        C( 0 ) <= Cin;

        stages: FOR i IN 0 TO n-1 GENERATE
            G( i ) <= x( i ) and y( i );      -- generate function
            P( i ) <= x( i ) xor y( i );      -- propagate function
            S_sig( i ) <= x( i ) xor y( i ) xor C( i ); -- sum
            C( i+1 ) <= G( i ) or ( P( i ) and C( i ) ); -- output of
carry

        END GENERATE;

        g_stages: FOR i IN n-1 downto 1 GENERATE
            g_stage: Gint( i ) <= AND_REDUCE( P( n-1 downto i ) ) and G ( i-1 );
        END GENERATE;

        S <= S_sig;      -- povezava vsote na izhodni signal
        Pout <= AND_REDUCE( P ); -- true when all propagates are true
        Cout <= C( n );
        Gout <= G( n-1 ) or OR_REDUCE( Gint );      -- output generate is true
if the last or gives you true

    END NDV;

```

```

-- *****
-- **** STUDENT: 64240429
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S          :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

    -- COMPONENT cla_gp IS PORT ( Cin, x, y : IN STD_LOGIC;
    -- s, Cout, g, p : OUT STD_LOGIC );
    -- END COMPONENT;

    -- vektor vmesnih prenosov in funkcij tvorjenja in sirjenja
    SIGNAL      C : STD_LOGIC_VECTOR( n DOWNT0 0 );      -- every adder's output plus the
    0th's input
    SIGNAL      G : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );    -- every adder
    SIGNAL      P : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );    -- same as G
    SIGNAL      S_sig : STD_LOGIC_VECTOR( n-1 DOWNT0 0 ); -- every adder
    signal      Gint : std_logic_vector( n-1 downto 1 ); -- every adder besides the
    0th one and the last one

    BEGIN

        -- stage0: cla_gp PORT MAP ( Cin, X( 0 ), Y( 0 ), S_sig( 0 ), C( 1 ), G( 1 ),
        P( 1 ) );
        -- namesto vmesnih stopenj zapisemo raje for ... generate zanko, -- s katero
        realiziramo 7 port map stavkov

        C( 0 ) <= Cin;      -- we need an ith to determine the i+1th
        stages: FOR i IN 0 TO n-1 GENERATE
            -- cla_stage: cla_gp -- couldn't get this to work, says C is not a port
            -- PORT MAP ( C ( i ) => Cin, -- but I checked the order multiple times and it
            -- -- X( i )=>x, -- was correct,
            -- -- Y( i )=>y,
            -- S_sig ( i ) => S,
            -- C ( i+1 )=>Cout,
            -- G ( i+1 )=>g,
            -- P ( i+1 )=>p );

            -- PORT MAP ( -- tried a different order,
            -- Cin => C( i ),      -- didn't work either
            -- x => X( i ),
            -- y => Y( i ),
            -- S => S_sig( i ) ,

```

```

--      Cout => C( i+1 ),
--      g => G( i+1 ),
--      p => P( i+1 )
--      );
G( i )      <= X( i ) and Y( i );
P( i )      <= X( i ) xor Y( i );
S_sig( i )  <= X( i ) xor Y( i ) xor C( i );
C( i+1 )    <= G( i ) or ( P( i ) and C( i ) );
END GENERATE;

S      <= S_sig;    -- povezava vsote na izhodni signal

-- N      <= S_sig( n-1 );    -- tvorba bita predznaka ( negative )
Cout    <= C( n );    -- tvorba izhodnega prenosa ( carry )
-- V      <= C( n ) XOR C( n-1 );    -- tvorba bita preliva ( overflow )
-- Z      <= '1' when S_sig = 0 else '0';    -- tvorba bita nic ( zero )

-- generate_of_blocks: FOR i in n-1 downto 1 GENERATE
--      Gint    <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
--      END ;

--      Pout = AND_REDUCE( P );
--      Gout    <= G( n-1 ) or OR_REDUCE( Gint );    -- perform or between all
intermediate results

generate_of_blocks: FOR i IN n-1 downto 1 GENERATE
      Gint( i )    <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
END GENERATE;

-- S      <= S_sig;    -- povezava vsote na izhodni signal
-- Cout    <= C( n );    -- tvorba izhodnega prenosa ( carry )
Gout    <= G( n-1 ) or OR_REDUCE( Gint );    -- perform or between all
intermediate results
Pout    <= AND_REDUCE( P );

END NDV;

```

```

-- *****
-- **** STUDENT: 64240430
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;      -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin      :    in    std_logic ;
        X, Y      :    in    std_logic_vector( n-1 downto 0 );
        S          :    out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout :    out   std_logic );
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS

    -- vektor vmesnih prenosov in funkcij tvorjenja in sirjenja
    SIGNAL      G, P, vsota : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    -- SIGNAL      S_sig : STD_LOGIC_VECTOR( n-1 DOWNT0 0 );
    SIGNAL      Carry : std_logic_vector ( n downto 0 );
    SIGNAL      g_int : std_logic_vector ( n-1 downto 1 );

BEGIN
    Carry( 0 ) <= Cin;      -- Lako imamo zacetnu vrednost za Carry( 0 )

    posamezni_deli:
        FOR i IN 0 TO n-1 GENERATE

            G( i ) <= X( i ) and Y( i );
            P( i ) <= X( i ) xor Y( i );

            vsota( i )    <= X( i ) xor Y( i ) xor Carry( i );
            Carry( i+1 ) <= G( i ) or ( P( i ) and Carry( i ) );

        END GENERATE;

    g_int_del :
        for i in 1 to n-1 generate
            g_int( i )    <= and_reduce( P( n-1 downto i ) ) and G( i-1 );
        end generate;

    S      <= vsota;
    Gout    <= G( n-1 ) or or_reduce( g_int );
    Pout    <= and_reduce( P );
    Cout    <= Carry( Carry'left );

END NDV;

```

```

-- *****
-- **** STUDENT: 64210132
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Manjkajo inicializacije signalov (others => '0').
Izraz za g_int ni pravilen - indeks mora iti do 1. Pravilno je
g_int( i )  <= and_reduce( P( n-1 downto i ) ) and G( i-1 );
Izraz za Gout je napačen - pravilno je:
Gout  <= G( n-1 ) or or_reduce( g_int );
NASLEDNJIČ KODO NALOŽITE V USTREZEN RAZDELEK (OSTALO_x), KJER JE X ŠTEVILKA DOMAČE
NALOGE
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;                -- reduction operators (AND_REDUCE, OR_REDUCE)

ENTITY cla_add_n_bit IS
    generic(n: natural := 8);
    PORT ( Cin      :    in    std_logic ;
           X, Y      :    in    std_logic_vector(n-1 downto 0);
           S          :    out   std_logic_vector(n-1 downto 0);
           Gout, Pout, Cout :    out   std_logic);
END cla_add_n_bit;

ARCHITECTURE NDV OF cla_add_n_bit IS
    COMPONENT cla_gp IS
        PORT ( Cin, x, y : IN STD_LOGIC;
              s, Cout, g, p : OUT STD_LOGIC );
    END COMPONENT;

--vektor vmesnih prenosov in funkcij tvorjenja in sirjenja
SIGNAL G, P : STD_LOGIC_VECTOR(n DOWNT0 1);
SIGNAL C : STD_LOGIC_VECTOR(n DOWNT0 0);
SIGNAL S_sig : STD_LOGIC_VECTOR(n-1 DOWNT0 0);
SIGNAL Gint : STD_LOGIC_VECTOR(n-1 DOWNT0 1);

BEGIN

    C(0) <= Cin;
    stages: FOR i IN 0 TO (n-1) GENERATE
        cla_stage: cla_gp PORT MAP ( C(i), X(i), Y(i), S_sig(i), C(i+1), G(i+1),
P(i+1) );
    END GENERATE;

    S <= S_sig; -- povezava vsote na izhodni signal
    G_intermediate: FOR i IN n-1 DOWNT0 1 GENERATE
        Gint(i) <= AND_REDUCE(P(n downto i+1)) and G(i);
    END GENERATE;
    Gout <= G(n) or OR_REDUCE(Gint); -- perform or between all intermediate
results
    Pout <= AND_REDUCE(P);
    Cout <= C(n); --tvorba izhodnega prenosa (carry)
END NDV;

```



```

-- *****
-- **** PREDLOGA VAJE
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_MISC.ALL;    -- reduction operators ( AND_REDUCE, OR_REDUCE )

ENTITY cla_add_n_bit IS
    generic( n: natural := 8 );
    PORT (
        Cin          : in    std_logic ;
        X, Y          : in    std_logic_vector( n-1 downto 0 );
        S             : out   std_logic_vector( n-1 downto 0 );
        Gout, Pout, Cout : out  std_logic );
END cla_add_n_bit;

ARCHITECTURE ideal OF cla_add_n_bit IS
    signal    Cint_xtd : std_logic_vector( n downto 0 ) := ( 0 => Cin, others =>'0' );
    -- internal carry vector, extended to fit Cout on LSB position
    signal    G, P : std_logic_vector( n-1 downto 0 ) := ( others =>'0' );    --
    internal signal    G vector as intermediate result for Gout calculation
    signal    Gint : std_logic_vector( n-1 downto 1 ) := ( others =>'0' );    --
    internal signal    Gint vector as intermediate result for Gout calculation
BEGIN

    Cint_xtd( 0 ) <= Cin;    -- for implementation purposes!

    cla_stages: FOR i IN 0 TO n-1 GENERATE
        G( i ) <= X( i ) and Y( i );    -- bit generate
        P( i ) <= X( i ) xor Y( i );    -- bit propagate
        Cint_xtd( i+1 ) <= G( i ) or ( P( i ) and Cint_xtd( i ) );    --
    internal carry vector
        S( i ) <= X( i ) xor Y( i ) xor Cint_xtd( i ); -- sum
    END GENERATE;

    Cout <= Cint_xtd( n );    -- MSB element is Cout

    -- according to 74HC182 datasheet( https://dtsheet.com/doc/210879/philips-74hc182 )
    -- we get equations for block propagate/generate:

    --  $P = P_3 \cdot P_2 \cdot P_1 \cdot P_0$  ( signal group/block propagate - is true, whenever all
    internal propagates are true )
    Pout <= AND_REDUCE( P );

    -- block generate becomes true when:
    --  $G_3=1$  or if
    --  $G_2=1$  and is being propagated through (  $P_2=1$  ) to output or if
    --  $G_1=1$  and is being propagated through (  $P_2=P_3=1$  ) to output or if
    --  $G_0=1$  and is being propagated through (  $P_1=P_2=P_3=1$  ) to output.

    --  $G = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0$  ( signal block generate )
    -- Note: First term is handled outside for-generate Loop
    f2_gen_loop: FOR i IN n-1 DOWNTO 1 GENERATE
        Gint( i ) <= AND_REDUCE( P( n-1 downto i ) ) and G( i-1 );
    END GENERATE;

```

```
        Gout  <= G( n-1 ) or OR_REDUCE( Gint );      -- perform or between all
intermediate results
END ideal;
```