

-- **** STUDENT: 64000225.....	4
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	4
-- **** 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: RCO ne deluje pravilno pri štetju navzdol. RCO se postavi vedno, ko je štetje navzgor (štetje so same enke) ali ko je štetje navzdol in je štetje vrednost nič, oz. zapis povedanega z when-else (glej komentar spodaj). 14	
-- **** STUDENT: 64200288.....	16
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	16
-- **** STUDENT: 64200296.....	18
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: RCO deluje narobe, zato tudi šteje -3,-4,-13,-14 ... (moral bi nadaljevati -5,-6...) 18 Namesto: rco<= t(ctr_size-1); bi morali imeti: rco<= t(ctr_size); 18	
-- **** STUDENT: 64200385.....	20
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	20
-- **** STUDENT: 64210113.....	22
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Če sestavimo verigo dveh števecv (LO – šteje sp. 4 bite, HI - šteje zg. 4 bite), potem RCO HI števca ne deluje pravilno. Pri štetju 0, v smeri navzdol bi se moral tudi izhodni RCO_HI postaviti na '1', pa se postavi pri štetju 1, ne 0. Pri štetju navzgor bi se moral postaviti nazaj na '0', ko gre štetje iz -7 na -8. RCO ni preliv – služi zgolj krmiljenju nadrejenih števecv. 22 Manjka krmiljenje T(0) – podobno za prvi element zg. In sp. verige AND vrat: Nista '1', ampak sta pogojena s smerjo: 22 Veriga T se nadaljuje do ctr_size, ne do ctr_size-1. Izhodni RCO je potem zadnji element verige T (RCO <= T(ctr_size);) 22	
-- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let. 22	
-- **** STUDENT: 64210132.....	24
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let. 24 Rezultatov simulacije ni, ker so napake sintetizatorja: 24	

ERROR:HDLCompiler:40 - "ud_counter.vhd" Line 31: tff is not a component.....	24
ERROR:HDLCompiler:854 - "ud_counter.vhd" Line 17: Unit <ndv> ignored due to previous errors.	Error! Bookmark not defined.
Pri poimenovanju neke komponente v povezovalnem stavku lahko uporabite katerokoli ime **RAZEN** imena te komponente:	24
Namesto: tff: tff port map(... naredite t_ff: tff port map(.....	24
Po tej spremembi simulacija deluje pravilno.	24
-- *****	24
-- **** STUDENT: 64210290.....	26
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	26
-- **** STUDENT: 64210382.....	28
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let. 28	
-- *****	28
-- **** STUDENT: 64210384.....	30
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	30
-- **** STUDENT: 64210386.....	32
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let. 32	
-- *****	32
-- **** STUDENT: 64210455.....	36
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let. 36	
Nalogo ste naredili z drugimi signali v entiteti, zato je ne morem oceniti.....	36
Rezultatov simulacije ni, ker so napake sintetizatorja:	36
ERROR:HDLCompiler:244 - "ud_counter_IDEAL_tb.vhd" Line 15: Binding entity ud_counter does not have generic ctr_size	36
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 18: Formal port <nCLR> does not exist in entity <ud_counter>. Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the mismatch.	36
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 19: Formal port <D_nU> does not exist in entity <ud_counter>. Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the mismatch.	36

ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 20: Formal port <EN> does not exist in entity <ud_counter>. Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the mismatch. ...	36
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 21: Formal port <RCO> does not exist in entity <ud_counter>. Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the mismatch.	36
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 22: Formal port <Q> does not exist in entity <ud_counter>. Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the mismatch. ...	36
ERROR:Simulator:777 - Static elaboration of top level VHDL design unit ud_counter_ideal_tb in library work failed	36
'ud_counter_tb_isim_beh.exe' is not recognized as an internal or external command,	36
operable program or batch file.	36
-- *****	36
-- **** STUDENT: 64210457	39
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let.	39
-- *****	39
-- **** STUDENT: 64240430	41
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	41
-- **** PREDLOGA VAJE	43
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	43

```

-- *****
-- **** STUDENT: 64000225
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

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

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje števca ( aktiven '0' )
        D_nU,   -- signal      za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;  -- signal      za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal
        T,      -- vhodni signali v T flip flope
        zgornja_veriga_and, -- zgornja veriga AND vrat
        spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );
    signal
        Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal      števnega izhoda
        -- spodnja veriga AND vrat

    COMPONENT tff IS
    PORT (
        T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
        Q : OUT STD_LOGIC );
    END COMPONENT;

begin

    ctr: for i in 0 to ctr_size-1 generate
        tf: tff port map(
            T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,          Q =>
            Q_sig( i )
        );
    end generate;
end ideal;

```

```
    zgornja_veriga_and( i+1 ) <= zgornja_veriga_and( i ) and EN and Q_sig( i );  
    spodnja_veriga_and( i+1 ) <= spodnja_veriga_and( i ) and EN and not Q_sig( i );  
    T( i+1 ) <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );  
end generate;
```

```
zgornja_veriga_and( 0 ) <= not D_nU;  
spodnja_veriga_and( 0 ) <= D_nU;  
T( 0 ) <= EN;  
RCO <= T( ctr_size );
```

```
Q <= Q_sig;
```

```
end ideal;
```

```

-- *****
-- **** STUDENT: 64190088
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje števca ( aktiven '0' )
        D_nU,   -- signal      za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;  -- signal      za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal
        T,      -- vhodni signali v T flip flope
        zgornja_veriga_and, -- zgornja veriga AND vrat
        spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );
    signal
        Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal      -- spodnja veriga AND vrat
                                                števnega izhoda

    COMPONENT tff IS
    PORT (
        T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
        Q : OUT STD_LOGIC );
    END COMPONENT;

begin

    U0: tff port map ( T => EN, clk => clk, nPRESET => '1', nCLEAR => nCLR, Q => Q_sig( 0 ) );

    zgornja_veriga_and( 0 )  <= ( ( not D_nU ) and EN and Q_sig( 0 ) );
    spodnja_veriga_and( 0 )  <= ( D_nU and EN and ( not Q_sig( 0 ) ) );

    zanka_for_gen: for i in 1 to ( ctr_size - 1 ) generate

```

```
U1: tff port map ( T => T( i ), clk => clk, nPRESET => '1', nCLEAR => nCLR, Q => Q_sig( i ) );
```

```
T( i ) <= zgornja_veriga_and( i-1 ) or spodnja_veriga_and( i-1 );
```

```
zgornja_veriga_and( i ) <= ( zgornja_veriga_and( i-1 ) and EN and Q_sig( i ) );
```

```
spodnja_veriga_and( i ) <= ( spodnja_veriga_and( i-1 ) and EN and ( not q_sig( i ) ) );
```

```
end generate;
```

```
RCO <= zgornja_veriga_and( ctr_size - 1 ) or spodnja_veriga_and( ctr_size - 1 );
```

```
Q <= Q_sig;
```

```
end ideal;
```

```

-- *****
-- **** STUDENT: 64200100
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje tevca ( aktiven '0' )
        D_nU,   -- signal      za smer tetja tevca ( naraajoe -> '0' )
        EN      : IN std_logic;  -- signal      za omogoanje tetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno tetje tevca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal      T,    -- vhodni signali v T flip flope
        -- zgornja_veriga_and,  -- zgornja veriga AND vrat
        spodnja_veriga_and ,zgornja_veriga_and: std_logic_vector( ctr_size downto 0 ); -- spodnja veriga AND vrat
    signal      Q_sig : std_logic_vector( ctr_size - 1 downto 0 );  -- signal      tevnega izhoda

    COMPONENT tff IS
    PORT( T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
          Q : OUT STD_LOGIC );
    END COMPONENT;

begin
    ctr: for i in 0 to ctr_size-1 generate
    tf1: tff port map(
        T => T( i ),
        clk => clk,
        nPRESET => '1',
        nCLEAR => nCLR,
        Q => Q_sig( i ) );
        zgornja_veriga_and( i+1 ) <= zgornja_veriga_and( i ) and EN and Q_sig( i );
    end ctr;
end ideal;

```



```
spodnja_veriga_and( i+1 ) <= spodnja_veriga_and( i ) and EN and not Q_sig( i );
T( i+1 ) <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );
end generate;

zgornja_veriga_and( 0 ) <= not D_nU;
spodnja_veriga_and( 0 ) <= D_nU;
T( 0 ) <= EN;
RCO <= T( ctr_size );

Q <= Q_sig;
end ideal;
```

```

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

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_MISC.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje tevca ( aktiven '0' )
        D_nU,   -- signal      za smer tetja tevca ( naraajoe -> '0' )
        EN      : IN std_logic;  -- signal      za omogoanje tetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno tetje tevca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal
        T,      -- vhodni signali v T flip flope
        zgornja_veriga_and, -- zgornja veriga AND vrat
        spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );  -- spodnja veriga AND vrat
    signal
        Q_sig : std_logic_vector( ctr_size - 1 downto 0 );  -- signal      tevnega izhoda

    COMPONENT tff IS
    PORT (
        T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
        Q : OUT STD_LOGIC );
    END COMPONENT;

begin

    U0 : for i in 0 to ctr_size - 1 generate
        U1 : tff port map( T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,
        Q => Q_sig( i )
        );

        zgornja_veriga_and( i + 1 )      <= EN and Q_sig( i ) and zgornja_veriga_and( i );

```

```
    spodnja_veriga_and( i + 1 )      <= EN and ( not Q_sig( i ) ) and spodnja_veriga_and( i );  
    T( i + 1 )      <= zgornja_veriga_and( i + 1 ) or spodnja_veriga_and( i + 1 );  
end generate;
```

```
T( 0 ) <= EN;  
zgornja_veriga_and( 0 )      <= not D_nU;  
spodnja_veriga_and( 0 )      <= D_nU;
```

```
RCO      <= T( ctr_size );  
Q        <= Q_sig;
```

```
end ideal;
```

```

-- *****
-- **** STUDENT: 64200163
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library ieee;
use ieee.std_logic_1164.all;

entity ud_counter is
    generic( ctr_size : natural := 6 );
    port( clk,          nCLR,          D_nU,
          EN : in std_logic;
          RCO : out std_logic;
          Q : out std_logic_vector( ctr_size - 1 downto 0 )
    );
end ud_counter;

architecture ideal of ud_counter is
    signal T,          zgornja_veriga_and,          spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );
    signal Q_sig : std_logic_vector( ctr_size - 1 downto 0 );

    component tff is
        port( T, clk, nPRESET, nCLEAR : in std_logic;
              Q : out std_logic
        );
    end component;
begin
    ctr: for i in 0 to ctr_size - 1 generate
        u1: tff
            port map( T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,
            Q => Q_sig( i )
        );
        zgornja_veriga_and( i+1 ) <= zgornja_veriga_and( i ) and EN and Q_sig( i );
        spodnja_veriga_and( i+1 ) <= spodnja_veriga_and( i ) and EN and not Q_sig( i );
        T( i+1 ) <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );
    end generate;
    zgornja_veriga_and( 0 ) <= not D_nU;
    spodnja_veriga_and( 0 ) <= D_nU;
    T( 0 ) <= EN;

```

```
        RCO    <= T( ctr_size );  
        Q      <= Q_sig;  
end ideal;
```

```

-- *****
-- **** STUDENT: 64200238
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: RCO ne deluje pravilno pri štetju navzdol. RCO se postavi vedno, ko je štetje navzgor (štetje so same
enke) ali ko je štetje navzdol in je štetje vrednost nič, oz. zapis povedanega z when-else (glej komentar spodaj).
RCO <= '1' WHEN ( (Q_sig = zeroes) and (D_nU = '1') ) or ( (Q_sig = ones) and (D_nU = '0') ) else '0';
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,      -- signal      ure
        nCLR,     -- signal      za brisanje števca ( aktiven '0' )
        D_nU,     -- signal      za smer štetja števca ( naraščajoče -> '0' )
        EN       : IN std_logic;  -- signal      za omogočanje štetja ( aktiven '1' )
        RCO      : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q        : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
    );
end ud_counter;

architecture ideal of ud_counter is

    COMPONENT tff IS
    PORT (
        T,      -- vhod flip-flopa
        clk,    -- signal      ure
        nPRESET, -- asinhrono postavljanje izhoda ( aktiven '0' )
        nCLEAR  : IN STD_LOGIC; -- asinhrono brisanje izhoda ( aktiven '0' )
        Q       : OUT STD_LOGIC ); -- izhod flip-flopa
    END COMPONENT;

    signal
        T,      -- vhodni signali v T flip flope
        zgornja_veriga_and, -- zgornja veriga AND vrat
        spodnja_veriga_and : std_logic_vector( ctr_size downto 0 ); -- spodnja veriga AND vrat
    signal
        Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal      števnege izhoda
    CONSTANT
        zeroes : std_logic_vector( ctr_size - 1 downto 0 ) := ( others => '0' );
    CONSTANT
        ones : std_logic_vector( ctr_size - 1 downto 0 ) := ( others => '1' );

```

```
begin
```

```
    kapitalizem: for i in 0 to ctr_size -1 generate
```

```
        tflipflip: tff
```

```
        PORT map (
```

```
            T => T( i ), clk => clk,  nPRESET => '1',      nCLEAR => nCLR,      Q => Q_sig( i )
        );
```

```
    zgornja_veriga_and( i+1 ) <= zgornja_veriga_and( i ) and Q_sig( i ) and EN;
```

```
    spodnja_veriga_and( i+1 ) <= spodnja_veriga_and( i ) and not Q_sig( i ) and EN;
```

```
    T( i+1 )      <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );
```

```
        end generate;
```

```
    -- zgornja_veriga_and( 0 ) <= not D_nU and Q_sig( 0 ) and EN;
```

```
    -- spodnja_veriga_and( 0 ) <= D_nU and not Q_sig( 0 ) and EN;
```

```
    zgornja_veriga_and( 0 )      <= not D_nU;
```

```
    spodnja_veriga_and( 0 )      <= D_nU;
```

```
    T( 0 ) <=      EN;
```

```
    -- RCO <= T( ctr_size );
```

```
    WITH Q_sig SELECT
```

```
        RCO      <= '1' WHEN ( zeroes OR ones ),
```

```
        '0' WHEN OTHERS;
```

```
    -- Q_sig      <= zeroes when nCLR = '0';
```

```
        Q      <= Q_sig;
```

```
end ideal;
```

```

-- *****
-- **** STUDENT: 64200288
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

entity ud_counter is
  generic( ctr_size: natural := 6 );
  PORT (
    clk : IN std_logic;
    nCLR : IN std_logic;
    D_nU : IN std_logic;
    EN : IN std_logic;
    RCO : OUT std_logic;
    Q : OUT std_logic_vector( ctr_size - 1 downto 0 )
  );
end ud_counter;

architecture ideal of ud_counter is
  signal Q_sig : std_logic_vector( ctr_size - 1 downto 0 );
  signal T : std_logic_vector( ctr_size downto 0 );
  signal zgornja_veriga_and, spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );

  COMPONENT tff IS
    PORT( T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
          Q : OUT STD_LOGIC );
  END COMPONENT;

begin
  zgornja_veriga_and( 0 ) <= not D_nU;
  spodnja_veriga_and( 0 ) <= D_nU;
  T( 0 ) <= EN;

  ctr: for i in 0 to ctr_size-1 generate
    tf1: tff port map(
      T => T( i ),

```



```
clk => clk,  
nPRESET => '1',  
nCLEAR => nCLR,  
Q => Q_sig( i )  
);  
zgornja_veriga_and( i+1 )<= zgornja_veriga_and( i ) and EN and Q_sig( i );  
spodnja_veriga_and( i+1 )<= spodnja_veriga_and( i ) and EN and not Q_sig( i );  
T( i+1 )    <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );  
end generate;  
  
RCO    <= T( ctr_size );  
Q      <= Q_sig;  
end ideal;
```

```

-- *****
-- **** STUDENT: 64200296
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: RCO deluje narobe, zato tudi šteje -3,-4,-13,-14 ... (moral bi nadaljevati -5,-6...)
Namesto: rco<= t(ctr_size-1); bi morali imeti: rco<= t(ctr_size);
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje tevca ( aktiven '0' )
        D_nU,   -- signal      za smer tetja tevca ( naraajoe -> '0' )
        EN      : IN std_logic;  -- signal      za omogoanje tetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno tetje tevca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal
        T,    -- vhodni signali v T flip flope
        zgornja_veriga_and, -- zgornja veriga AND vrat
        spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );  -- spodnja veriga AND vrat
    signal
        Q_sig : std_logic_vector( ctr_size - 1 downto 0 );  -- signal      tevnega izhoda

    COMPONENT tff IS
    PORT (
        T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
        Q : OUT STD_LOGIC );
    END COMPONENT;

begin
    count: for i in 0 to ctr_size-1 generate
        tff1: tff port map (
            t=>t( i ),          clk=>clk,          nPRESET=>'1',          nCLEAR=>nCLR,
            q=>q_sig( i )
        );
        zgornja_veriga_and( i+1 ) <=zgornja_veriga_and( i )and en and q_sig( i );
        spodnja_veriga_and( i+1 ) <=spodnja_veriga_and( i )and en and not q_sig( i );
    end generate
end ideal;

```

```
        t( i+1 )      <= spodnja_veriga_and( i+1 ) or zgornja_veriga_and( i+1 );
end generate;
zgornja_veriga_and( 0 )    <= not D_nU;
spodnja_veriga_and( 0 )    <= D_nU;
t( 0 ) <= en;
rco    <= t( ctr_size-1 );

q      <= q_sig;
end ideal;
```

```

-- *****
-- **** STUDENT: 64200385
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje števca ( aktiven '0' )
        D_nU,   -- signal      za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;  -- signal      za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal
        T,      -- vhodni signali v T flip flope
        zgornja_veriga_and, -- zgornja veriga AND vrat
        spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );
    signal
        Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal      -- spodnja veriga AND vrat
                                                -- števnega izhoda

    COMPONENT tff IS
    PORT (
        T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
        Q : OUT STD_LOGIC );
    END COMPONENT;

begin

    ud: for i in 0 to ctr_size - 1 generate
    U0: tff port map(
        T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,          Q => Q_sig( i
    ) );

    zgornja_veriga_and( i+1 )<= zgornja_veriga_and( i ) and EN and Q_sig( i );
    spodnja_veriga_and( i+1 )<= spodnja_veriga_and( i ) and EN and not Q_sig( i );

```

```

        T( i+1 )    <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );

end generate;

T( 0 )            <= EN;
zgornja_veriga_and( 0 )  <= not D_nU and EN and Q_sig( 0 );
spodnja_veriga_and( 0 )  <= D_nU and EN and not Q_sig( 0 );
RCO    <= zgornja_veriga_and( ctr_size ) or spodnja_veriga_and( ctr_size );
Q      <= Q_sig;

end ideal;

```

```

-- *****
-- **** STUDENT: 64210113
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Če sestavimo verigo dveh števcov (LO – šteje sp. 4 bite, HI - šteje zg. 4 bite), potem RCO HI števca ne
deluje pravilno. Pri štetju 0, v smeri navzdol bi se moral tudi izhodni RCO_HI postaviti na '1', pa se postavi pri
štetju 1, ne 0. Pri štetju navzgor bi se moral postaviti nazaj na '0', ko gre štetje iz -7 na -8. RCO ni preliv – služi
zgolj krmiljenju nadrejenih števcov.
Manjka krmiljenje T(0) – podobno za prvi element zg. In sp. verige AND vrat: Nista '1', ampak sta pogojena s smerjo:
    up(0) <= not D_nU;    down(0) <= D_nU;
Veriga T se nadaljuje do ctr_size, ne do ctr_size-1. Izhodni RCO je potem zadnji element verige T (RCO <=
T(ctr_size));
-- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let.
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk : IN std_logic;      -- signal          ure
        nCLR : IN std_logic;      -- signal          za brisanje števca ( aktiven '0' )
        D_nU : IN std_logic;      -- signal          za smer štetja števca ( naraščajoce -> '0' )
        EN : IN std_logic; -- signal          za omogocanje štetja ( aktiven '1' )
        RCO : OUT std_logic;      -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q : OUT std_logic_vector( ctr_size - 1 downto 0 ) -- izhodno štetje števca
    );
end ud_counter;

architecture ndv of ud_counter is
    COMPONENT tff IS
    PORT (
        T : IN std_logic; -- vhod flip-flopa
        clk : IN std_logic;      -- signal          ure
        nPRESET : IN std_logic; -- asinhrono postavljanje izhoda ( aktiven '0' )
        nCLEAR : IN std_logic;  -- asinhrono brisanje izhoda ( aktiven '0' )
        Q : OUT std_logic -- izhod flip-flopa
    );

```

```

);
END COMPONENT;

signal      T : std_logic_vector( ctr_size - 1 downto 0 );
signal      Qnov : std_logic_vector( ctr_size - 1 downto 0 );
signal      up : std_logic_vector( ctr_size downto 0 );
signal      down : std_logic_vector( ctr_size downto 0 );

begin

  up( 0 )    <= '1';
  down( 0 )  <= '1';

  gen: for i in 0 to ctr_size - 1 generate
  gene: tff
    port map(
      T => T( i ),
      clk => clk,
      nPRESET => '1',
      nCLEAR => nCLR,
      Q => Qnov( i )
    );

    T( i )    <= EN and ( ( not D_nU and up( i ) ) or ( D_nU and down( i ) ) );
    up( i + 1 ) <= Qnov( i ) and up( i );
    down( i + 1 ) <= ( not Qnov( i ) ) and down( i );

  end generate;

  Q    <= Qnov;
  RCO  <= ( up( ctr_size ) and not D_nU ) or ( down( ctr_size ) and D_nU );

end ndv;

```

```

-- *****
-- **** STUDENT: 64210132
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Uporabljate stare Synopsys knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let.
Rezultatov simulacije ni, ker so napake sintetizatorja:
ERROR:HDLCompiler:40 - "ud_counter.vhd" Line 31: tff is not a component
Pri poimenovanju neke komponente v povezovalnem stavku lahko uporabite katerokoli ime **RAZEN** imena te komponente:
Namesto: tff: tff port map(      ... naredite t_ff: tff port map(
Po tej spremembi simulacija deluje pravilno.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,      -- signal      ure
        nCLR,     -- signal      za brisanje števca ( aktiven '0' )
        D_nU,     -- signal      za smer štetja števca ( naraščajoče -> '0' )
        EN       : IN std_logic;  -- signal      za omogočanje štetja ( aktiven '1' )
        RCO      : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q        : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
    );
end ud_counter;

architecture ndv of ud_counter is
    COMPONENT tff IS
    PORT (
        T,      -- vhod flip-flopa
        clk,    -- signal      ure
        nPRESET, -- asinhrono postavljanje izhoda ( aktiven '0' )
        nCLEAR  : IN STD_LOGIC;  -- asinhrono brisanje izhoda ( aktiven '0' )
        Q       : OUT STD_LOGIC ); -- izhod flip-flopa
    END COMPONENT;

    signal Q_s : STD_LOGIC_VECTOR( ctr_size - 1 downto 0 );
    signal T, and1, and2 : STD_LOGIC_VECTOR( ctr_size downto 0 );

begin

```



```

count: for i in 0 to ctr_size-1 generate
    tff: tff port map(
        T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,          Q =>
Q_s( i )
    );
    and1( i+1 ) <= and1( i ) and EN and Q_s( i );
    and2( i+1 ) <= and2( i ) and EN and not Q_s( i );
    T( i+1 ) <= and1( i+1 ) or and2( i+1 );
end generate;

and1( 0 ) <= not D_nU;
and2( 0 ) <= D_nU;
T( 0 ) <= EN;
RCO <= T( ctr_size );

Q <= Q_s;
end ndv;

```

```

-- *****
-- **** STUDENT: 64210290
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje števca ( aktiven '0' )
        D_nU,   -- signal      za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;  -- signal      za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
    );
end ud_counter;

architecture ideal of ud_counter is

    signal      T,    -- vhodni signali v T flip flope
                zgornja_veriga_and, -- zgornja veriga AND vrat
                spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );
    signal      Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal      -- spodnja veriga AND vrat
                                                števnega izhoda

    component tff is
        port(
            T, clk, nPRESET, nCLEAR : in std_logic;
            Q : out std_logic
        );
    end component;

begin

    T( 0 ) <= EN;
    RCO    <= T( T'high );
    zgornja_veriga_and( 0 )    <= not D_nU;
    spodnja_veriga_and( 0 )    <= D_nU;

```

```

stages : for idi in 0 to ctr_size-1 generate
    tffx : tff
        port map(
            clk => clk,          nCLEAR => nCLR,          nPRESET => '1',          T => T( idi ),
Q => Q_sig( idi )
        );
    T( idi+1 )    <= zgornja_veriga_and( idi+1 ) xor spodnja_veriga_and( idi+1 );
    zgornja_veriga_and( idi+1 )    <= zgornja_veriga_and( idi ) and EN and Q_sig( idi );
    spodnja_veriga_and( idi+1 )    <= spodnja_veriga_and( idi ) and ( not Q_sig( idi ) ) and EN;
end generate stages;

Q    <= Q_sig;

end ideal;

```

```

-- *****
-- **** STUDENT: 64210382
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal           ure
        nCLR,   -- signal           za brisanje števca ( aktiven '0' )
        D_nU,   -- signal           za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;    -- signal           za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;    -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 ) -- izhodno štetje števca
    );
end ud_counter;

architecture ndv of ud_counter is
    COMPONENT tff IS
        PORT (
            T,    -- vhod flip-flopa
            clk,  -- signal           ure
            nPRESET, -- asinhrono postavljanje izhoda ( aktiven '0' )
            nCLEAR : IN STD_LOGIC;  -- asinhrono brisanje izhoda ( aktiven '0' )
            Q      : OUT STD_LOGIC );
        END COMPONENT;

    signal Q_sig : STD_LOGIC_VECTOR( ctr_size - 1 downto 0 );
    signal T, andp, andn : STD_LOGIC_VECTOR( ctr_size downto 0 );

begin
    ctr: for i in 0 to ctr_size-1 generate
        tf: tff port map(
            T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,          Q =>
            Q_sig( i )
        );
    end generate;
end architecture;

```

```

        andp( i+1 )  <= andp( i ) and EN and Q_sig( i );
        andn( i+1 )  <= andn( i ) and EN and not Q_sig( i );
        T( i+1 )     <= andp( i+1 ) or andn( i+1 );
    end generate;

    andp( 0 )        <= not D_nU;
    andn( 0 )        <= D_nU;
    T( 0 ) <= EN;
    RCO              <= T( ctr_size );

    Q                <= Q_sig;
end ndv;

```

```

-- *****
-- **** STUDENT: 64210384
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT ( clk,    -- signal        ure
           nCLR,   -- signal        za brisanje števca ( aktiven '0' )
           D_nU,   -- signal        za smer štetja števca ( naraščajoče -> '0' )
           EN      : IN std_logic;  -- signal        za omogočanje štetja ( aktiven '1' )
           RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
           Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
           );
end ud_counter;

architecture ideal of ud_counter is
    COMPONENT tff IS
        PORT ( T,      -- vhod flip-flopa
               clk,    -- signal        ure
               nPRESET, -- asinhrono postavljanje izhoda ( aktiven '0' )
               nCLEAR  : IN STD_LOGIC;  -- asinhrono brisanje izhoda ( aktiven '0' )
               Q       : OUT STD_LOGIC ); -- izhod flip-flopa
    END COMPONENT;

    signal T,      -- vhodni signali v T flip flope
           zgornja_veriga_and, -- zgornja veriga AND vrat
           spodnja_veriga_and : std_logic_vector( ctr_size downto 0 ) ;    -- spodnja veriga AND vrat

    signal Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal        števnega izhoda

begin

ctr: for i in 0 to ctr_size-1 generate
    tf: tff port map(

```

```

        T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,          Q =>
Q_sig( i )
    );

    zgornja_veriga_and( i+1 ) <= zgornja_veriga_and( i ) and EN and Q_sig( i );
    spodnja_veriga_and( i+1 ) <= spodnja_veriga_and( i ) and EN and not Q_sig( i );
    T( i+1 ) <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );

end generate;

zgornja_veriga_and( 0 ) <= not D_nU;
spodnja_veriga_and( 0 ) <= D_nU;

T( 0 ) <= EN;
RCO <= T( ctr_size );
Q <= Q_sig;

end ideal;

```

```

-- *****
-- **** STUDENT: 64210386
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje tevca ( aktiven '0' )
        D_nU,   -- signal      za smer tetja tevca ( naraajoe -> '0' )
        EN      : IN std_logic;    -- signal      za omogoanje tetja ( aktiven '1' )
        RCO     : out std_logic;    -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )    -- izhodno tetje tevca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal      T,    -- vhodni signali v T flip flope
                zgornja_veriga_and, -- zgornja veriga AND vrat
                spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );    -- spodnja veriga AND vrat
    signal      Q_sig : std_logic_vector( ctr_size - 1 downto 0 );    -- signal      tevnega izhoda

    COMPONENT tff IS
    PORT (
        T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
        Q : OUT STD_LOGIC );
    END COMPONENT;

    begin

        st: for i in 0 to ctr_size-1 generate
            t_ff: tff port map(
                T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,          Q =>
Q_sig( i )
            );

```



```

        zgornja_veriga_and( i+1 ) <= zgornja_veriga_and( i ) and EN and Q_sig( i );
        spodnja_veriga_and( i+1 ) <= spodnja_veriga_and( i ) and EN and not Q_sig( i );
        T( i+1 ) <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );
    end generate;

    zgornja_veriga_and( 0 ) <= not D_nU;
    spodnja_veriga_and( 0 ) <= D_nU;
    T( 0 ) <= EN;
    RCO <= T( ctr_size );

    Q <= Q_sig;
end ideal;

```

```

-- *****
-- **** STUDENT: 64210445
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6);
    PORT ( clk,
           nCLR,
           D_nU,
           EN      : IN std_logic;
           RCO     : out std_logic;
           Q       : out std_logic_vector(ctr_size - 1 downto 0)
    );
end ud_counter;

architecture ideal of ud_counter is
    signal T, --vhodni signali v T flip flope
           zgornja_veriga_and, --zgornja veriga AND vrat
           spodnja_veriga_and : std_logic_vector(ctr_size downto 0); --spodnja veriga AND vrat
    signal Q_sig : std_logic_vector(ctr_size - 1 downto 0); -- signal števnege izhoda

    COMPONENT tff IS
    PORT (
        T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
        Q : OUT STD_LOGIC);
    END COMPONENT;

begin

    ctr: for i in 0 to ctr_size-1 generate
        tf: tff port map(
            T => T(i),
            clk => clk,
            nPRESET => '1',
            nCLEAR => nCLR,
            Q => Q_sig(i)
        );
    end generate;
end ideal;

```

```
);  
zgornja_veriga_and(i+1) <= zgornja_veriga_and(i) and EN and Q_sig(i);  
spodnja_veriga_and(i+1) <= spodnja_veriga_and(i) and EN and not Q_sig(i);  
T(i+1) <= zgornja_veriga_and(i+1) or spodnja_veriga_and(i+1);  
end generate;
```

```
zgornja_veriga_and(0) <= not D_nU;  
spodnja_veriga_and(0) <= D_nU;  
T(0) <= EN;  
RCO <= T(ctr_size);
```

```
Q <= Q_sig;
```

```
end ideal;
```

```

-- *****
-- **** STUDENT: 64210455
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let.
Nalogo ste naredili z drugimi signali v entiteti, zato je ne morem oceniti.
Rezultatov simulacije ni, ker so napake sintetizatorja:
ERROR:HDLCompiler:244 - "ud_counter_IDEAL_tb.vhd" Line 15: Binding entity ud_counter does not have generic ctr_size
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 18: Formal port <nCLR> does not exist in entity <ud_counter>.
Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the
mismatch.
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 19: Formal port <D_nU> does not exist in entity <ud_counter>.
Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the
mismatch.
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 20: Formal port <EN> does not exist in entity <ud_counter>.
Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the
mismatch.
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 21: Formal port <RCO> does not exist in entity <ud_counter>.
Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the
mismatch.
ERROR:HDLCompiler:1156 - "ud_counter_IDEAL_tb.vhd" Line 22: Formal port <Q> does not exist in entity <ud_counter>.
Please compare the definition of block <ud_counter> to its component declaration and its instantiation to detect the
mismatch.
ERROR:Simulator:777 - Static elaboration of top level VHDL design unit ud_counter_ideal_tb in library work failed
'ud_counter_tb_isim_beh.exe' is not recognized as an internal or external command,
operable program or batch file.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ud_counter is
    generic ( counter_size : natural := 6 );          -- velikost števca
    port (
        clk : in std_logic; -- ura
        reset_n : in std_logic; -- ponastavitev števca ( aktiven '0' )
        count_dir: in std_logic; -- smer štetja ( naraščajoče -> '0' )
        enable : in std_logic; -- omogocanje štetja ( aktiven '1' )
        carry_out: out std_logic; -- prenos na naslednjo stopnjo
    );
end entity ud_counter;

```

```

        count : out std_logic_vector( counter_size-1 downto 0 )    -- trenutna vrednost števca
    );
end ud_counter;

architecture behavior of ud_counter is

    component tff is
        port (
            T : in std_logic;    -- vhod za flip-flop
            clk : in std_logic; -- ura
            nPRESET : in std_logic;    -- asinhrono nastavljanje izhoda ( aktiven '0' )
            nCLEAR : in std_logic;    -- asinhrono brisanje izhoda ( aktiven '0' )
            Q : out std_logic    -- izhod flip-flopa
        );
    end component;

    signal count_internal : std_logic_vector( counter_size-1 downto 0 );    -- vmesna vrednost
    signal T_signals : std_logic_vector( counter_size downto 0 );    -- signali za T flip-flop
    signal pos_chain : std_logic_vector( counter_size downto 0 );    -- pozitivna veriga
    signal neg_chain : std_logic_vector( counter_size downto 0 );    -- negativna veriga

begin

    process_chain: for idx in 0 to counter_size-1 generate
        flip_flop_inst: tff
            port map (
                T => T_signals( idx ),            clk => clk,            nPRESET => '1',            nCLEAR => reset_n,
                -- Popravljen ime porta
                Q => count_internal( idx )
            );

        -- Logične verige za nastavitve vhodov flip-flopov
        pos_chain( idx+1 ) <= pos_chain( idx ) and enable and count_internal( idx );
        neg_chain( idx+1 ) <= neg_chain( idx ) and enable and not count_internal( idx );
        T_signals( idx+1 ) <= pos_chain( idx+1 ) or neg_chain( idx+1 );
    end generate;

    -- Zacetne vrednosti logicnih verig
    pos_chain( 0 ) <= not count_dir;
    neg_chain( 0 ) <= count_dir;

```

```
T_signals( 0 )      <= enable;
```

```
-- Nastavitev prenosnega izhoda in koncnega števc
```

```
carry_out    <= T_signals( counter_size );
```

```
count <= count_internal;
```

```
end behavior;
```

```

-- *****
-- **** STUDENT: 64210457
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Uporabljate stare Synopsis knjižnice (STD_LOGIC_ARITH), česar v predlogah že ni kar nekaj let.
-- *****

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal           ure
        nCLR,   -- signal           za brisanje števca ( aktiven '0' )
        D_nU,   -- signal           za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;    -- signal za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;    -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 ) -- izhodno štetje števca
    );
end ud_counter;

architecture ndv of ud_counter is
    COMPONENT tff IS
        PORT (
            T,    -- vhod flip-flopa
            clk,  -- signal           ure
            nPRESET, -- asinhrono postavljanje izhoda ( aktiven '0' )
            nCLEAR : IN STD_LOGIC;  -- asinhrono brisanje izhoda ( aktiven '0' )
            Q      : OUT STD_LOGIC ); -- izhod flip-flopa
        END COMPONENT;

    signal Q_sig : STD_LOGIC_VECTOR( ctr_size - 1 downto 0 );
    signal T, andp, andn : STD_LOGIC_VECTOR( ctr_size downto 0 );

begin
    ctr: for i in 0 to ctr_size-1 generate
        tf: tff port map(
            T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,          Q =>
            Q_sig( i )
        );
    end generate;
end;

```

```

        andp( i+1 )  <= andp( i ) and EN and Q_sig( i );
        andn( i+1 )  <= andn( i ) and EN and not Q_sig( i );
        T( i+1 )     <= andp( i+1 ) or andn( i+1 );
    end generate;

    andp( 0 )        <= not D_nU;
    andn( 0 )        <= D_nU;
    T( 0 ) <= EN;
    RCO              <= T( ctr_size );

    Q                <= Q_sig;
end ndv;

```



```

-- *****
-- **** STUDENT: 64240430
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal          ure
        nCLR,   -- signal          za brisanje števca ( aktiven '0' )
        D_nU,   -- signal          za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;    -- signal          za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;    -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 ) -- izhodno štetje števca
    );
end ud_counter;

architecture ideal of ud_counter is

COMPONENT tff IS
PORT ( T,      -- vhod flip-flopa
        clk,   -- signal          ure
        nPRESET, -- asinhrono postavljanje izhoda ( aktiven '0' )
        nCLEAR  : IN STD_LOGIC; -- asinhrono brisanje izhoda ( aktiven '0' )
        Q       : OUT STD_LOGIC ); -- izhod flip-flopa
END COMPONENT;

signal      T,      -- vhodni signali v T flip flope
            zgornja_veriga_and, -- zgornja veriga AND vrat
            spodnja_veriga_and : std_logic_vector( ctr_size downto 0 ); -- spodnja veriga AND vrat
signal      Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal          števnege izhoda

begin
    -- zacetne vrednosti
    zgornja_veriga_and( 0 ) <= not D_nU;
    spodnja_veriga_and( 0 ) <= D_nU;
    T( 0 ) <= EN;

```

```

-- povezovanje
povezovanje:
for i in 0 to ctr_size - 1 generate
    tff_blok: tff
        port map ( T => T( i ),          clk => clk,          nPRESET => '1',          nCLEAR => nCLR,
            Q => Q_sig( i ) );

        zgornja_veriga_and( i + 1 )      <= zgornja_veriga_and( i ) and EN and Q_sig( i );
        spodnja_veriga_and( i + 1 )      <= spodnja_veriga_and( i ) and EN and ( not Q_sig( i ) );
        T( i + 1 )      <= zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );

    end generate;

    RCO      <= T( ctr_size );
    Q        <= Q_sig;
end ideal;

```

```

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

entity ud_counter is
    generic( ctr_size: natural := 6 );
    PORT (
        clk,    -- signal      ure
        nCLR,   -- signal      za brisanje števca ( aktiven '0' )
        D_nU,   -- signal      za smer štetja števca ( naraščajoče -> '0' )
        EN      : IN std_logic;  -- signal      za omogočanje štetja ( aktiven '1' )
        RCO     : out std_logic;  -- izhodni prenos na naslednjo stopnjo ( Ripple carry out )
        Q       : out std_logic_vector( ctr_size - 1 downto 0 )  -- izhodno štetje števca
    );
end ud_counter;

architecture ideal of ud_counter is
    signal
        T,      -- vhodni signali v T flip flope
        zgornja_veriga_and, -- zgornja veriga AND vrat
        spodnja_veriga_and : std_logic_vector( ctr_size downto 0 );
    signal
        Q_sig : std_logic_vector( ctr_size - 1 downto 0 ); -- signal      števnega izhoda
        -- spodnja veriga AND vrat

COMPONENT tff IS
PORT (
    T, clk, nPRESET, nCLEAR : IN STD_LOGIC;
    Q : OUT STD_LOGIC );
END COMPONENT;

for all:tff use entity work.tff( ideal );

begin

    T( 0 ) <= EN; -- vhod za omogočanje štetja
    zgornja_veriga_and( 0 ) <= not D_nU;
    spodnja_veriga_and( 0 ) <= D_nU;

    L0: FOR i IN 0 TO ctr_size - 1 GENERATE

```

```

        Ui: tff port map ( T( i ), clk, '1', nCLR, Q_sig( i ) );  -- povezovanje T flip flopov
        zgornja_veriga_and( i+1 ) <= Q_sig( i ) and zgornja_veriga_and( i ) and EN;
        spodnja_veriga_and( i+1 ) <= not Q_sig( i ) and spodnja_veriga_and( i ) and EN;
        T( i+1 )      <=      zgornja_veriga_and( i+1 ) or spodnja_veriga_and( i+1 );
    END GENERATE;

    RCO    <= T( ctr_size );  -- izhodni prenos za naslednje stopnje

    Q      <= Q_sig;  -- povezovanje izhodnega štetja

end ideal;

```