

| | |
|--|----|
| -- **** STUDENT: 64000225..... | 2 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 2 |
| -- **** STUDENT: 64200100..... | 5 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 5 |
| -- **** STUDENT: 64200163..... | 7 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 7 |
| -- **** STUDENT: 64200288..... | 9 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 9 |
| -- **** STUDENT: 64200296..... | 12 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 12 |
| -- **** STUDENT: 64200385..... | 15 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 15 |
| -- **** STUDENT: 64210113..... | 18 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 18 |
| -- **** STUDENT: 64210132..... | 21 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 21 |
| -- **** STUDENT: 64210290..... | 24 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 24 |
| -- **** STUDENT: 64210382..... | 27 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 27 |
| -- **** STUDENT: 64210384..... | 29 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 29 |
| -- **** STUDENT: 64210386..... | 32 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 32 |
| -- **** STUDENT: 64210455..... | 38 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 38 |
| -- **** STUDENT: 64210457..... | 41 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 41 |
| -- **** STUDENT: 64240430..... | 44 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 44 |
| -- **** STUDENT: 64200238..... | 47 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 47 |
| -- **** PREDLOGA VAJE..... | 50 |
| -- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb..... | 50 |

```

-- *****
-- **** STUDENT: 64000225
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;

ENTITY ud_counter_tb IS
END ud_counter_tb;

ARCHITECTURE behavior OF ud_counter_tb IS

    -- Component Declaration for the Unit Under Test ( UUT )

    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';

    -- Outputs
    signal      RCO : std_logic;

```

```

signal          Q : std_logic_vector( 5 downto 0 );

    -- Clock period definitions
constant      clk_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: ud_counter PORT MAP (
    clk => clk,
    nCLR => nCLR,
    D_nU => D_nU,
    EN => EN,
    RCO => RCO,
    Q => Q
    );

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

    -- Stimulus process
    stim_proc: process
    begin
        -- hold reset state for 100 ns.
        wait for 100 ns;

        wait for clk_period*10;

        EN    <= '1';
        wait for clk_period*4;
        D_nU  <= '1';
        wait for clk_period*6;
        D_nU  <= '0';
        wait for clk_period*4;
    end process;

```

```
nCLR    <= '0';  
wait for clk_period/5;  
nCLR    <= '1';
```

```
wait;  
end process;
```

```
END;
```

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

```
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.ALL;
```

```
ENTITY ud_counter_tb IS
END ud_counter_tb;
```

```
ARCHITECTURE behavior OF ud_counter_tb IS
COMPONENT ud_counter
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( 5 downto 0 ) );
END COMPONENT;
```

```
signal      clk : STD_LOGIC := '0';
signal      nCLR : STD_LOGIC := '1';
signal      D_nU : STD_LOGIC := '0';
signal      EN : STD_LOGIC := '0';
signal      RCO : STD_LOGIC;
signal      Q : STD_LOGIC_vector( 5 downto 0 );
constant    clkP : time := 10 ns;
```

```
BEGIN
uut: ud_counter PORT MAP (
clk => clk,
nCLR => nCLR,
D_nU => D_nU,
EN => EN,
RCO => RCO,
Q => Q );
```

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

  wait for 100 ns;
  wait for clkP*10;
  EN    <= '1';
  wait for clkP*4;
  D_nU  <= '1';
  wait for clkP*6;
  D_nU  <= '0';
  wait for clkP*4;
  nCLR  <= '0';
  wait for clkP/5;
  nCLR  <= '1';
  wait;
end process;
END;
```

```

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

entity ud_counter_tb is
    generic( ctr_size : natural := 6 );
end ud_counter_tb;

architecture behavior of ud_counter_tb is
    component ud_counter
        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 component;
    signal
        clk : std_logic := '0';
    signal
        nCLR : std_logic := '1';
    signal
        D_nU : std_logic := '0';
    signal
        EN : std_logic := '0';
    signal
        RCO : std_logic;
    signal
        Q : std_logic_vector( ctr_size - 1 downto 0 );
    constant
        clk_period : time := 10 ns;
begin
    uut : ud_counter port map(
        clk => clk,
        nCLR => nCLR,
        D_nU => D_nU,
        EN => EN,
        RCO => RCO,
        Q => Q
    );
end behavior;

```

```

    );
clk_process : process
begin
    clk    <= '0';
    wait for clk_period/2;
    clk    <= '1';
    wait for clk_period/2;
end process;
stim_proc : process
begin
wait for clk_period*10;
    EN    <= '1';
    wait for clk_period*10;
    D_nU   <= '1';
    wait for clk_period*10;
    D_nU   <= '0';
    wait for clk_period*10;
    EN     <= '0';
    wait for clk_period*10.5;
    nCLR   <= '0';

wait;
end process;
end;

```



```

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

ENTITY ud_counter_tb IS
generic( ctr_size: natural := 6 );
END ud_counter_tb;

ARCHITECTURE behavior OF ud_counter_tb IS

    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;

    signal          clk : std_logic := '0';
    signal          nCLR : std_logic := '1';
    signal          D_nU : std_logic := '0';
    signal          EN : std_logic := '0';

    signal          RCO : std_logic;
    signal          Q : std_logic_vector( 5 downto 0 );

    constant        ClockPeriode : time := 10 ns;

BEGIN

    uut: ud_counter PORT MAP (

```

```
clk => clk,  
nCLR => nCLR,  
D_nU => D_nU,  
EN => EN,  
RCO => RCO,  
Q => Q  
);
```

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

```
stim_proc: process  
begin  
wait for 100 ns;  
EN    <= '0';  
D_nU  <= '0';  
nCLR  <= '0';  
wait for 10 ns;  
nCLR  <= '1';  
wait for 20 ns;  
EN    <= '1';  
D_nU  <= '0';  
wait for 300 ns;  
D_nU  <= '1';  
wait for 100 ns;  
EN    <= '0';  
wait for 8 ns;  
nCLR  <= '0';  
wait for 4 ns;  
nCLR  <= '1';  
wait for 8 ns;  
  
wait;  
end process;
```

END;

```

-- *****
-- **** STUDENT: 64200296
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 13:15:35 12/22/2024
-- Design Name:
-- Module Name: C:/Users/Tim/Downloads/LIFO/LIFO/ud_counter_tb.vhd
-- Project Name: LIFO
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: ud_counter
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-- Notes:
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test. Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation
-- simulation model.
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

```

```
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;
```

```
ENTITY ud_counter_tb IS
END ud_counter_tb;
```

```
ARCHITECTURE behavior OF ud_counter_tb IS
```

```
    -- Component Declaration for the Unit Under Test ( UUT )
```

```
    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';
```

```
    -- Outputs
```

```
    signal      RCO : std_logic;
    signal      Q : std_logic_vector( 5 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```
    -- Instantiate the Unit Under Test ( UUT )
```

```
    uut: ud_counter PORT MAP (
        clk => clk,
```

```

nCLR => nCLR,
D_nU => D_nU,
EN => EN,
RCO => RCO,
Q => Q
);

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

    -- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    wait for clk_period*10;

    -- insert stimulus here
    EN    <= '1';
    wait for clk_period*10;
    D_nU  <= '1';
    wait for clk_period*10;
    D_nU  <= '0';
    wait for clk_period*5;
    nCLR  <= '0';
    wait for clk_period*5;
    nCLR  <= '1';

    wait;
end process;

END;

```

```

-- *****
-- **** STUDENT: 64200385
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;

ENTITY ud_counter_tb IS
END ud_counter_tb;

ARCHITECTURE behavior OF ud_counter_tb IS

    -- Component Declaration for the Unit Under Test ( UUT )

    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '0';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';

    -- Outputs
    signal      RCO : std_logic;
    signal      Q : std_logic_vector( 5 downto 0 );

```

```

    -- Clock period definitions
    constant    clk_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: ud_counter PORT MAP (
    clk => clk,
    nCLR => nCLR,
    D_nU => D_nU,
    EN => EN,
    RCO => RCO,
    Q => Q
    );

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

    -- Stimulus process
    stim_proc: process
    begin
        -- hold reset state for 100 ns.
        wait for 100 ns;

        wait for clk_period*10;

        -- insert stimulus here
        nCLR    <= '1';
        EN      <= '1';
        wait for clk_period;

        D_nU    <= '0';
        wait for clk_period;
    end process;

```



```
D_nU   <= '1';  
wait for clk_period;
```

```
nCLR   <= '0';  
wait for clk_period;
```

```
nCLR   <= '1';
```

```
wait;  
end process;
```

```
END;
```

```
-- *****
-- **** STUDENT: 64210113
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
-- USE ieee.numeric_std.ALL;
```

```
ENTITY ul_counter_tb IS
END ul_counter_tb;
```

```
ARCHITECTURE behavior OF ul_counter_tb IS
```

```
-- Component Declaration for the Unit Under Test ( UUT )
```

```
COMPONENT ud_counter
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( 5 downto 0 )
);
END COMPONENT;
```

```
-- Inputs
```

```
signal clk : std_logic := '0';
signal nCLR : std_logic := '1';
signal D_nU : std_logic := '0';
signal EN : std_logic := '0';
```

```
-- Outputs
```

```
signal RCO : std_logic;
signal Q : std_logic_vector( 5 downto 0 );
```

```
-- Clock period definitions
```

```

constant    clk_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: ud_counter PORT MAP (
        clk => clk,
        nCLR => nCLR,
        D_nU => D_nU,
        EN => EN,
        RCO => RCO,
        Q => Q
    );

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

    -- Stimulus process
    stim_proc: process
    begin
        -- hold reset state for 100 ns.
        wait for 100 ns;

        wait for clk_period*10;

        -- insert stimulus here

        nCLR <= '0';
        wait for clk_period;
        nCLR <= '1';
        wait for clk_period * 5;

        D_nU <= '1';
        EN    <= '1';
    end process;

```

```
wait for clk_period * 10;

D_nU <= '0';
wait for clk_period * 10;

EN    <= '0';
wait for clk_period * 5;

EN    <= '1';
D_nU  <= '0';
wait for clk_period * 10;

wait;
end process;

END;
```

```
-- *****
-- **** STUDENT: 64210132
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
```

```
ENTITY ud_counter_tb IS
END ud_counter_tb;
```

```
ARCHITECTURE behavior OF ud_counter_tb IS
```

```
    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';
```

```
    signal      RCO : std_logic;
    signal      Q : std_logic_vector( 5 downto 0 );
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```

uut: ud_counter PORT MAP (
  clk => clk,
  nCLR => nCLR,
  D_nU => D_nU,
  EN => EN,
  RCO => RCO,
  Q => Q
);

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

stim_proc: process
begin
  wait for 100 ns;

  wait for clk_period*10;

    EN    <= '1';
    wait for clk_period*5;

    D_nU   <= '1';
    wait for clk_period*7;

    D_nU   <= '0';
    wait for clk_period*3;

    nCLR   <= '0';
    wait for clk_period/3;

    nCLR   <= '1';

  wait;
end process;

```

END;

```

-- *****
-- **** STUDENT: 64210290
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY ud_counter_tb IS
END ud_counter_tb;

ARCHITECTURE behavior OF ud_counter_tb IS

    constant    ctr_size : natural := 3;

    -- Component Declaration for the Unit Under Test ( UUT )
    COMPONENT ud_counter
        GENERIC(
            ctr_size : NATURAL
        );
    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 COMPONENT;

    -- Inputs
    signal        clk : std_logic := '0';
    signal        nCLR : std_logic := '0';
    signal        D_nU : std_logic := '0';
    signal        EN : std_logic := '0';

    -- Outputs
    signal        RCO : std_logic;

```



```

signal          Q : std_logic_vector( ctr_size-1 downto 0 );

    -- Clock period definitions
constant  clk_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: ud_counter
        generic map(
            ctr_size => ctr_size
        )
        port map(
            clk => clk,
            nCLR => nCLR,
            D_nU => D_nU,
            EN => EN,
            RCO => RCO,
            Q => Q
        );

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

    -- Stimulus process
    stim_proc: process
    begin
        -- hold reset state for 100 ns.
        wait for 100 ns;

        wait for clk_period*10;

        nCLR <= '1';

```

```
D_nU    <= '0';  
EN      <= '1';  
wait until Q = "000";  
wait until Q = "111";
```

```
D_nU    <= '1';  
wait until Q = "111";  
wait until Q = "000";
```

```
D_nU    <= '0';  
wait until Q = "101";  
EN      <= '0';  
wait for clk_period;  
nCLR    <= '0';  
wait for clk_period;  
EN      <= '1';
```

```
wait;  
end process;
```

```
END;
```

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

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
```

```
ENTITY ud_counter_tb IS
END ud_counter_tb;
```

```
ARCHITECTURE behavior OF ud_counter_tb IS
```

```
    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;
```

```
    signal          clk : std_logic := '0';
    signal          nCLR : std_logic := '1';
    signal          D_nU : std_logic := '0';
    signal          EN : std_logic := '0';
```

```
    signal          RCO : std_logic;
    signal          Q : std_logic_vector( 5 downto 0 );
```

```
    constant      clk_period : time := 10 ns;
```

```
BEGIN
```

```

uut: ud_counter PORT MAP (
  clk => clk,
  nCLR => nCLR,
  D_nU => D_nU,
  EN => EN,
  RCO => RCO,
  Q => Q
);

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

stim_proc: process
begin
  wait for 100 ns;

  wait for clk_period*10;

    EN    <= '1';
    wait for clk_period*4;
    D_nU  <= '1';
    wait for clk_period*6;
    D_nU  <= '0';
    wait for clk_period*4;
    nCLR  <= '0';
    wait for clk_period/5;
    nCLR  <= '1';

  wait;
end process;

END;

```

```

-- *****
-- **** STUDENT: 64210384
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 15:19:13 12/20/2024
-- Design Name:
-- Module Name: C:/NDV/DN5/ud_counter_tb.vhd
-- Project Name: DN5
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: ud_counter
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-- Notes:
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test. Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation
-- simulation model.
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

```

```
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;
```

```
ENTITY ud_counter_tb IS
END ud_counter_tb;
```

```
ARCHITECTURE behavior OF ud_counter_tb IS
```

```
    -- Component Declaration for the Unit Under Test ( UUT )
```

```
    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';
```

```
    -- Outputs
```

```
    signal      RCO : std_logic;
    signal      Q : std_logic_vector( 5 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```
    -- Instantiate the Unit Under Test ( UUT )
```

```
    uut: ud_counter PORT MAP (
        clk => clk,
```

```

nCLR => nCLR,
D_nU => D_nU,
EN => EN,
RCO => RCO,
Q => Q
);

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

-- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    wait for clk_period*10;

    -- insert stimulus here
    EN    <= '1';
    wait for clk_period*4;
    D_nU  <= '1';
    wait for clk_period*6;
    D_nU  <= '0';
    wait for clk_period*4;
    nCLR  <= '0';
    wait for clk_period/4;
    nCLR  <= '1';

    wait;
end process;

END;

```

```

-- *****
-- **** STUDENT: 64210386
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 17:09:08 12/20/2024
-- Design Name:
-- Module Name: /home/ise/Xilinx_shared/domaca_naloga_5/ud_counter_tb.vhd
-- Project Name: domaca_naloga_5
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: ud_counter
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-- Notes:
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test. Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation
-- simulation model.
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

```



```
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;
```

```
ENTITY ud_counter_tb IS
END ud_counter_tb;
```

```
ARCHITECTURE behavior OF ud_counter_tb IS
```

```
    -- Component Declaration for the Unit Under Test ( UUT )
```

```
    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';
```

```
    -- Outputs
```

```
    signal      RCO : std_logic;
    signal      Q : std_logic_vector( 5 downto 0 );
```

```
    -- Clock period definitions
```

```
    constant    clk_period : time := 10 ns;
```

```
BEGIN
```

```
    -- Instantiate the Unit Under Test ( UUT )
```

```
    uut: ud_counter PORT MAP (
        clk => clk,
```

```

nCLR => nCLR,
D_nU => D_nU,
EN => EN,
RCO => RCO,
Q => Q
);

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

    -- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    wait for clk_period*10;

    -- insert stimulus here

    EN    <= '1';
    wait for clk_period*4;
    D_nU  <= '1';
    wait for clk_period*6;
    D_nU  <= '0';
    wait for clk_period*4;
    nCLR  <= '0';
    wait for clk_period/5;
    nCLR  <= '1';

    wait;
end process;

END;

```

```

-- *****
-- **** STUDENT: 64210445
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY ud_counter_tb IS
END ud_counter_tb;

ARCHITECTURE behavior OF ud_counter_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT ud_counter
    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(5 downto 0)
    );
    END COMPONENT;

    --Inputs
    signal clk : std_logic := '0';
    signal nCLR : std_logic := '0';
    signal D_nU : std_logic := '0';
    signal EN : std_logic := '0';

    --Outputs
    signal RCO : std_logic;
    signal Q : std_logic_vector(5 downto 0);

    -- Clock period definitions
    constant clk_period : time := 10 ns;

```

BEGIN

```
-- Instantiate the Unit Under Test (UUT)
uut: ud_counter PORT MAP (
    clk => clk,
    nCLR => nCLR,
    D_nU => D_nU,
    EN => EN,
    RCO => RCO,
    Q => Q
);

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

-- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;
    -- insert stimulus here

    EN <= '1';
    wait for clk_period;

    D_nU <= '1';
    wait for clk_period;

    D_nU <= '0';
    wait for clk_period;

    nCLR <= '0';
```

```
wait for clk_period;
```

```
nCLR <= '1';
```

```
wait;
```

```
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210455
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY ud_counter_tb IS
END ud_counter_tb;

ARCHITECTURE behavior OF ud_counter_tb IS

    -- Component Declaration for the Unit Under Test ( UUT )
    COMPONENT ud_counter
    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
        count : OUT std_logic_vector( 5 downto 0 ) -- trenutna vrednost števca
    );
    END COMPONENT;

    -- Inputs
    signal clk : std_logic := '0';
    signal reset_n : std_logic := '1';
    signal count_dir: std_logic := '0';
    signal enable : std_logic := '0';

    -- Outputs
    signal carry_out : std_logic;
    signal count : std_logic_vector( 5 downto 0 ); -- Velikost števca

    -- Clock period definitions
    constant clk_period : time := 10 ns;

BEGIN

```

```

    -- Instantiate the Unit Under Test ( UUT )
uut: ud_counter
PORT MAP (
clk => clk,
reset_n => reset_n,
count_dir=> count_dir,
enable => enable,
carry_out=> carry_out,
count => count
);

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

    -- Stimulus process
stim_proc: process
begin
    -- Hold reset state for 100 ns.
wait for 100 ns;

wait for clk_period*10;

    -- Insert stimulus here
enable      <= '1';      -- Omogocimo števec
wait for clk_period*4;
count_dir   <= '1';      -- Spremenimo smer štetja
wait for clk_period*6;
count_dir   <= '0';      -- Ponovno nastavitev smeri
wait for clk_period*4;
reset_n     <= '0';      -- Ponastavimo števec
wait for clk_period/5;
reset_n     <= '1';      -- Nadaljujemo štetje

```

```
wait;  
end process;  
  
END;
```



```
-- *****
-- **** STUDENT: 64210457
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
```

```
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;
```

```
ENTITY ud_counter_tb IS
END ud_counter_tb;
```

```
ARCHITECTURE behavior OF ud_counter_tb IS
```

```
    -- Component Declaration for the Unit Under Test ( UUT )
```

```
    COMPONENT ud_counter
    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( 5 downto 0 )
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';
```

```
    -- Outputs
```

```
    signal      RCO : std_logic;
```

```

signal          Q : std_logic_vector( 5 downto 0 );

    -- Clock period definitions
constant  clk_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: ud_counter PORT MAP (
    clk => clk,
    nCLR => nCLR,
    D_nU => D_nU,
    EN => EN,
    RCO => RCO,
    Q => Q
    );

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

    -- Stimulus process
    stim_proc: process
    begin
        -- hold reset state for 100 ns.
        wait for 100 ns;

        wait for clk_period*10;

        -- insert stimulus here
        EN    <= '1';
        wait for clk_period*4;
        D_nU  <= '1';
        wait for clk_period*6;
        D_nU  <= '0';
    end process;

```

```
wait for clk_period*4;  
nCLR  <= '0';  
wait for clk_period/5;  
nCLR  <= '1';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64240430
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;

ENTITY ud_counter_tb IS
    generic ( cnt_size : natural := 6 );
END ud_counter_tb;

ARCHITECTURE behavior OF ud_counter_tb IS

    -- Component Declaration for the Unit Under Test ( UUT )

    COMPONENT ud_counter
        -- generic ( cnt_size : natural := 6 ); imam tezave
    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( cnt_size - 1 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    -- signal      clk : std_logic := '0';
    signal        nCLR : std_logic := '1';
    signal        D_nU : std_logic := '0';
    signal        EN : std_logic := '0';

    -- Outputs

```

```

signal          RCO : std_logic;
signal          Q : std_logic_vector( cnt_size - 1 downto 0 );

    -- Clock period definitions
signal          clock : std_logic;
constant        clock_period : time := 100 ns;

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: ud_counter PORT MAP (
    clk => clock,
    nCLR => nCLR,
    D_nU => D_nU,
    EN => EN,
    RCO => RCO,
    Q => Q
    );

    -- Clock process definitions
    clk_process :process
    begin
        clock <= '0';
        wait for clock_period/2;
        clock <= '1';
        wait for clock_period/2;
    end process;

    -- Stimulus process
    stimulu_process: process
    -- variable delay_cycles : integer;
    begin

        -- estiranje nCLR
        -- izhod je const Q = 0
        nCLR <= '0';
        EN <= '1';
        wait for 4*clock_period;

        nCLR <= '1';

```

```

        EN      <= '0';
        wait for 4*clock_period;

-- deo koda, ki steje od up in down
        EN      <= '1';
        for i in 1 to 4 loop
            for j in 0 to i*2** ( cnt_size - 1 ) - 1 loop
                wait for clock_period;
            end loop;
            D_nU  <= not D_nU;
        end loop;

wait;
end process;

END;

```

```

-- *****
-- **** STUDENT: 64200238
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY ud_cuonter_tb IS
generic( ctr_size: natural := 6 );
END ud_cuonter_tb;

ARCHITECTURE behavior OF ud_cuonter_tb IS

    COMPONENT ud_counter
        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( 5 downto 0 )
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '0';
    signal      D_nU : std_logic := '0';
    signal      EN : std_logic := '0';

    -- Outputs
    signal      RCO : std_logic;
    signal      Q : std_logic_vector( 5 downto 0 );

    -- Clock period definitions
    constant    clk_period : time := 10 ns;

```

BEGIN

-- Instantiate the Unit Under Test (UUT)

```
uut: ud_counter PORT MAP (  
  clk => clk,  
  nCLR => nCLR,  
  D_nU => D_nU,  
  EN => EN,  
  RCO => RCO,  
  Q => Q  
);
```

-- Clock process definitions

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

-- Stimulus process

```
stim_proc: process  
begin  
    -- hold reset state for 100 ns.  
    wait for 10 ns;  
  
    EN    <= '0';  
    D_nU  <= '0';  
    nCLR  <= '0';    -- Reset the counter  
    wait for 20 ns;  
  
    nCLR  <= '1';    -- Release reset  
    wait for 20 ns;  
  
    -- Test Case 1: Count up  
    EN    <= '1';  
    D_nU  <= '0';    -- Count up  
    -- nCLR    <= '0';  
    wait for 700 ns;
```



```
        -- Test Case 2: Count down
D_nU <= '1';      -- Count down
wait for 200 ns;

        -- Test Case 3: Disable counting
EN <= '0';        -- Disable counting
wait for 50 ns;

        -- Test Case 4: Reset the counter
nCLR <= '0';      -- Reset the counter
wait for 20 ns;
nCLR <= '1';      -- Release reset
wait for 50 ns;

        -- End simulation
wait;
end process;

END;
```

```

-- *****
-- **** PREDLOGA VAJE
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****

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

ENTITY ud_counter_tb IS
    generic( ctr_size: natural := 3 );
END ud_counter_tb;

ARCHITECTURE testbench_arch OF ud_counter_tb IS
    FILE RESULTS: TEXT OPEN WRITE_MODE IS "ud_counter.csv";

    COMPONENT ud_counter
        generic( ctr_size: natural := 3 );
    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 COMPONENT;

    SIGNAL          clk          : std_logic := '0';
    SIGNAL          nCLR         : std_logic := '0';
    SIGNAL          D_nU         : std_logic := '0';
    SIGNAL          EN           : std_logic := '0';
    SIGNAL          EN_HI        : std_logic := '0';
    SIGNAL          RCO          : std_logic := '0';
    SIGNAL          Q            : std_logic_vector ( ctr_size - 1 DownTo 0 ) := ( others => '0' );
    SIGNAL          Q_HI         : std_logic_vector ( ctr_size - 1 DownTo 0 ) := ( others => '0' );
    SIGNAL          RCO_HI       : std_logic := '0';

```

```

constant    PERIOD : time := 200 ns;
constant    DUTY_CYCLE : real := 0.5;
constant    OFFSET : time := 100 ns;

BEGIN
ULO : ud_counter
    GENERIC MAP ( ctr_size => ctr_size )
PORT MAP ( clk => clk, nCLR => nCLR, D_nU => D_nU, EN => EN, RCO => RCO, Q => Q );

    EN_HI <= EN and RCO;      -- form high byte enable signal

UHI : ud_counter
    GENERIC MAP ( ctr_size => ctr_size )
PORT MAP ( clk => clk, nCLR => nCLR, D_nU => D_nU, EN => EN_HI, RCO => RCO_HI, Q => Q_HI );

PROCESS

    function to_unsigned_int_string( sv: Std_Logic_Vector ) return string is
    use Std.TextIO.all;
    variable iv: integer := to_integer( unsigned( sv ) );
    variable lp: line;
    begin
    write( lp, iv );
    return lp.all;
    end;

    function to_hex_string( sv_hi: Std_Logic_Vector; sv_lo: Std_Logic_Vector ) return string is
    use Std.TextIO.all;
    variable len_bits: natural := sv_hi'length + sv_lo'length;
    variable len_nibbles: natural := len_bits / 4;
    variable lp: line;
    begin
    if len_bits rem 4 > 0 then
    len_nibbles := len_nibbles + 1;
    hwrite( lp, ( ( len_nibbles * 4 - 1 ) - len_bits ) downto 0 => sv_hi( sv_hi'left ) ) & sv_hi &
sv_lo, right, len_nibbles );
    else
    hwrite( lp, sv_hi & sv_lo, right, len_nibbles );
    end if;

```

```

return lp.all;
end;

integer is
function to_signed_integer_string( sv_hi: Std_Logic_Vector; sv_lo: Std_Logic_Vector ) return

use Std.TextIO.all;
variable len_bits: natural := sv_hi'length + sv_lo'length;
variable len_nibbles: natural := len_bits / 4;
variable int: integer;
begin
if len_bits rem 4 > 0 then
len_nibbles := len_nibbles + 1;
int := to_integer( signed( ( ( len_nibbles * 4 - 1 ) - len_bits ) downto 0 => sv_hi( sv_hi'left )
) & sv_hi & sv_lo ) );
else
int := to_integer( signed( sv_hi & sv_lo ) );
end if;

return int;
end;

PROCEDURE Log_variables(
nCLR   : std_logic;
D_nU   : std_logic;
EN      : std_logic;
RCO     : std_logic;
Q       : std_logic_vector ( ctr_size - 1 DownTo 0 );
Q_HI    : std_logic_vector ( ctr_size - 1 DownTo 0 )
) IS
VARIABLE RES_LINE : LINE;
BEGIN
write( RES_LINE, nCLR, right, 1 );
write( RES_LINE, string'( "," ) );
write( RES_LINE, D_nU, right, 1 );
write( RES_LINE, string'( "," ) );
write( RES_LINE, EN , right, 1 );
write( RES_LINE, string'( "," ) );
write( RES_LINE, RCO , right, 1 );
write( RES_LINE, string'( "," ) );

```

```

write( RES_LINE, RCO_HI, right, 1 );
write( RES_LINE, string'( "," ) );
write( RES_LINE, integer'image( to_signed_integer_string( Q_HI, Q ) ) );
writeline( RESULTS, RES_LINE );
END;

```

```

BEGIN
WAIT for OFFSET;
CLOCK_LOOP : LOOP
clk  <= '0';
WAIT FOR ( PERIOD - ( PERIOD * DUTY_CYCLE ) );
clk  <= '1';
WAIT FOR ( PERIOD * DUTY_CYCLE );
      Log_variables( nCLR, D_nU, EN, RCO, Q, Q_HI );

END LOOP CLOCK_LOOP;
END PROCESS;

```

```

PROCESS

variable HDR_line : LINE;
BEGIN
write( HDR_line, string'( " nCLR, D_nU, EN, RCO, RCO_HI, Q" ) ); -- write results file header
writeline( RESULTS, HDR_line );

WAIT FOR PERIOD;    EN    <= '1';      -- enable counter
WAIT FOR PERIOD;    nCLR  <= '1';      -- clear counter

D_nU  <= '1';        -- setup down counting
WAIT FOR 2** ( 2*ctr_size ) * PERIOD;
D_nU  <= '0';        -- setup up counting
WAIT FOR 2** ( 2*ctr_size ) * PERIOD;

WAIT;

END PROCESS;

END testbench_arch;

```

