

-- **** STUDENT: 64000225.....	2
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	2
-- **** STUDENT: 64200100.....	6
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	6
-- **** STUDENT: 64200163.....	9
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	9
-- **** STUDENT: 64200238.....	12
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	12
-- **** STUDENT: 64200288.....	16
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	16
-- **** STUDENT: 64200296.....	19
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	19
-- **** STUDENT: 64200385.....	23
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	23
-- **** STUDENT: 64210113.....	27
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	27
-- **** STUDENT: 64210132.....	32
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	32
-- **** STUDENT: 64210290.....	35
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	35
-- **** STUDENT: 64210382.....	38
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	38
-- **** STUDENT: 64210384.....	41
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	41
-- **** STUDENT: 64210386.....	45
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	45
-- **** STUDENT: 64210455.....	53
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	53
-- **** STUDENT: 64210457.....	57
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	57
-- **** STUDENT: 64240430.....	61
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	61
-- **** PREDLOGA VAJE.....	65
-- KOMENTARJI K OCENI NALOGE -- Matej Možek: Ni pripomb.....	65

```

-- *****
-- **** 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 lifo_tb IS

```

```

    generic(
        lifo_width: natural := 4; -- velikost podatka sklada
        lifo_size: natural := 8 ); -- velikost sklada

```

```

END lifo_tb;

```

```

ARCHITECTURE behavior OF lifo_tb IS

```

```

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

```

```

    COMPONENT lifo
        generic(
            lifo_width: natural := 4; -- velikost podatka sklada
            lifo_size: natural := 8 ); -- velikost sklada

```

```

    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
END COMPONENT;

```

```

-- Inputs
signal      clk : std_logic := '0';
signal      nCLR : std_logic := '1';
signal      nEnable : std_logic := '1';
signal      PUSH : std_logic := '0';
signal      POP : std_logic := '0';

-- BiDirs
signal      data : std_logic_vector( 3 downto 0 ):= ( others => '0' );

-- Outputs
signal      FULL : std_logic;
signal      EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo
        GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            nEnable => nEnable,
            PUSH => PUSH,
            POP => POP,
            data => data,
            FULL => FULL,
            EMPTY => EMPTY
        );

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

```

```

        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;

nEnable    <= '0';
    wait for clk_period*3;
    PUSH    <= '1';
    data    <= "0001";
    wait for clk_period;
    data    <= "0010";
    wait for clk_period;
    data    <= "0011";
    wait for clk_period;
    data    <= "0100";
    wait for clk_period;
    data    <= "0101";
    wait for clk_period;
    data    <= "0110";
    wait for clk_period;
    data    <= "0111";
    wait for clk_period;
    data    <= "1000";
    wait for clk_period;
    data    <= "1001";
    wait for clk_period;
    data    <= "1010";
    wait for clk_period;
    PUSH    <= '0';
    data    <= "ZZZZ";
    wait for clk_period*3;
    POP     <= '1';
    wait for clk_period*10;
    POP     <= '0';

```

```
PUSH  <= '1';  
data  <= "1111";  
wait for clk_period*3;  
PUSH  <= '0';  
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 lifo_tb IS
generic(      lifo_width: natural := 4;
              lifo_size: natural := 8 );
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

COMPONENT lifo
generic( lifo_width: natural := 4;
lifo_size: natural := 8 );
PORT(
clk : IN std_logic;
nCLR : IN std_logic;
nEnable : IN std_logic;
PUSH : IN std_logic;
POP : IN std_logic;
data : INOUT std_logic_vector( 3 downto 0 );
FULL : OUT std_logic;
EMPTY : OUT std_logic );
END COMPONENT;

signal      clk: std_logic := '0';
signal      nCLR: std_logic := '1';
signal      nEnable: std_logic := '1';
signal      PUSH: std_logic := '0';
signal      POP: std_logic := '0';
signal      data: std_logic_vector( 3 downto 0 ) := ( others => '0' );
signal      FULL: std_logic;
signal      EMPTY: std_logic;
constant    clkP: time := 10 ns;

```

```
BEGIN
```

```
uut: lifo
GENERIC MAP ( lifo_width => lifo_width, lifo_size => lifo_size )
PORT MAP (
  clk => clk,
  nCLR => nCLR,
  nEnable => nEnable,
  PUSH => PUSH,
  POP => POP,
  data => data,
  FULL => FULL,
  EMPTY => EMPTY );
```

```
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;
```

```
nEnable    <= '0';
wait for clkP*3;
PUSH    <= '1';
data    <= "0001";
wait for clkP;
data    <= "0010";
wait for clkP;
data    <= "0011";
wait for clkP;
data    <= "0100";
wait for clkP;
```

```
data    <= "0101";
wait for clkP;
data    <= "0110";
wait for clkP;
data    <= "0111";
wait for clkP;
data    <= "1000";
wait for clkP;
data    <= "1001";
wait for clkP;
data    <= "1010";
wait for clkP;
PUSH    <= '0';
data    <= "ZZZZ";
wait for clkP*3;
POP      <= '1';
wait for clkP*10;
POP      <= '0';
PUSH     <= '1';
data     <= "1111";
wait for clkP*3;
PUSH     <= '0';
nCLR     <= '0';
wait for clkP/5;
nCLR     <= '1';

wait;
end process;
END;
```



```

-- *****
-- **** STUDENT: 64200163
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Izhod ni dvojno shranjen (ang. Double buffering) – Ko pride prva POP operacija, je na izhodu nič, ne
-- pa zadnji vpisan podatek v sklad.
-- *****

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity lifo_tb is
    generic( lifo_width: natural := 4;
             lifo_size: natural := 8 );
end lifo_tb;

architecture behavior of lifo_tb is
    component lifo
        generic( lifo_width: natural := 4;
                 lifo_size : natural := 8 );
        port(
            clk : in std_logic;
nCLR : in std_logic;
nEnable : in std_logic;
PUSH : in std_logic;
POP : in std_logic;
data : inout std_logic_vector( lifo_width - 1 downto 0 );
FULL : out std_logic;
EMPTY : out std_logic
        );
    end component;
    signal
        clk : std_logic := '0';
    signal
        nCLR : std_logic := '1';
    signal
        nEnable : std_logic := '1';
    signal
        PUSH : std_logic := '0';
    signal
        POP : std_logic := '0';
    signal
        data : std_logic_vector( lifo_width - 1 downto 0 ) := ( others => '0' );
    signal
        FULL : std_logic;
    signal
        EMPTY : std_logic;
    constant
        clk_period : time := 10 ns;

```

```

begin
  uut : lifo
    generic map( lifo_width => lifo_width,          lifo_size => lifo_size )
    port map(
      clk => clk,          nCLR => nCLR,          nEnable => nEnable,          PUSH => PUSH,
      POP => POP,          data => data,          FULL => FULL,          EMPTY => EMPTY
    );
  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*5;
    nEnable    <= '0';
    PUSH    <= '1';
    data    <= "0001";
    wait for clk_period;
    data    <= "0010";
    wait for clk_period;
    data    <= "0011";
    wait for clk_period;
    data    <= "0100";
    wait for clk_period;
    data    <= "0101";
    wait for clk_period;
    data    <= "0110";
    wait for clk_period;
    data    <= "0111";
    wait for clk_period;
    data    <= "1000";
    wait for clk_period;
    data    <= "1001";
    wait for clk_period;
    PUSH    <= '0';
    data    <= "ZZZZ";
    wait for clk_period;
  end process;
end

```

```
    POP    <= '1';  
    wait for clk_period*5;  
    POP    <= '0';  
    PUSH   <= '1';  
    data   <= "1111";  
    wait for clk_period*5;  
    PUSH   <= '0';  
    wait for clk_period*2.5;  
    nCLR    <= '0';  
    wait for clk_period*5;  
    nCLR    <= '1';  
wait;  
end process;  
end;
```

```

-- *****
-- **** STUDENT: 64200238
-- *****
-- 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 lifo_tb IS
  generic(
    lifo_width: natural := 4; -- velikost podatka sklada
    lifo_size: natural := 8  -- velikost sklada
  );
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

  -- Component Declaration for the Unit Under Test ( UUT )
  COMPONENT lifo
    generic(
      lifo_width: natural := 4; -- velikost podatka sklada
      lifo_size: natural := 8  -- velikost sklada
    );
    PORT(
      clk : IN std_logic;
      nCLR : IN std_logic;
      nEnable : IN std_logic;
      PUSH : IN std_logic;
      POP : IN std_logic;
      data : INOUT std_logic_vector( 3 downto 0 );
      FULL : OUT std_logic;
      EMPTY : OUT std_logic
    );
  END COMPONENT;

```

```

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      nEnable : std_logic := '1';
    signal      PUSH : std_logic := '0';
    signal      POP : std_logic := '0';

    -- BiDirs
    signal      data : std_logic_vector( 3 downto 0 ) := ( others => '0' );

    -- Outputs
    signal      FULL : std_logic;
    signal      EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo
    GENERIC MAP ( lifo_width => lifo_width,
    lifo_size => lifo_size )
    PORT MAP (
    clk => clk,
    nCLR => nCLR,
    nEnable => nEnable,
    PUSH => PUSH,
    POP => POP,
    data => data,
    FULL => FULL,
    EMPTY => EMPTY
    );

    -- 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;

    -- First sequence: Fill the stack completely
wait for clk_period*10;
nEnable    <= '0';
PUSH <= '1';
data <= "0001"; wait for clk_period;
data <= "0010"; wait for clk_period;
data <= "0011"; wait for clk_period;
data <= "0100"; wait for clk_period;
data <= "0101"; wait for clk_period;
data <= "0110"; wait for clk_period;
data <= "0111"; wait for clk_period;
data <= "1000"; wait for clk_period;
PUSH <= '0'; data <= "ZZZZ"; wait for clk_period;

    -- Check if FULL is set
wait for clk_period*3;

    -- Second sequence: Pop all elements from the stack
POP <= '1'; wait for clk_period*5;
POP <= '0'; wait for clk_period*2;

    -- Third sequence: Push some values after stack is empty
PUSH <= '1';
data <= "1111"; wait for clk_period;
data <= "1010"; wait for clk_period;
PUSH <= '0'; data <= "ZZZZ"; wait for clk_period;

    -- Fourth sequence: Test overflow ( push beyond the size of the stack )
wait for clk_period*3;
PUSH <= '1';
data <= "0001"; wait for clk_period;

```

```

data <= "0010"; wait for clk_period;
data <= "0011"; wait for clk_period;
data <= "0100"; wait for clk_period;
data <= "0101"; wait for clk_period;
data <= "0110"; wait for clk_period;
data <= "0111"; wait for clk_period;
data <= "1000"; wait for clk_period;
data <= "1001"; wait for clk_period;
PUSH <= '0'; data <= "ZZZZ"; wait for clk_period;

    -- Fifth sequence: Pop after overflow ( ensure full and empty conditions )
POP <= '1'; wait for clk_period*5;
POP <= '0'; wait for clk_period*2;

    -- Sixth sequence: Test with mixed push and pop
PUSH <= '1';
data <= "0101"; wait for clk_period;
POP <= '1'; wait for clk_period;
PUSH <= '1';
data <= "1011"; wait for clk_period;
POP <= '1'; wait for clk_period;

    -- Seventh sequence: Reset the stack ( nCLR )
nCLR <= '0'; wait for clk_period/5;
nCLR <= '1'; wait for clk_period*2;

    -- Eighth sequence: Push and pop after reset
PUSH <= '1';
data <= "1101"; wait for clk_period;
POP <= '1'; wait for clk_period;

wait;
end process;

END;

```

```

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

ENTITY lifo_tb IS
    generic(
        lifo_width: natural := 4;
        lifo_size: natural := 8
    );
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

    COMPONENT lifo
    generic(
        lifo_width: natural := 4;
        lifo_size: natural := 8
    );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
END COMPONENT;

    signal          clk : std_logic := '0';
    signal          nCLR : std_logic := '1';
    signal          nEnable : std_logic := '1';
    signal          PUSH : std_logic := '0';
    signal          POP : std_logic := '0';

```



```
signal      data : std_logic_vector( 3 downto 0 ) := ( others => '0' );
signal      FULL : std_logic;
signal      EMPTY : std_logic;
constant    ClockPeriode : time := 10 ns;
```

BEGIN

```
    uut: lifo
    GENERIC MAP (
        lifo_width => lifo_width,
        lifo_size => lifo_size
    )
    PORT MAP (
        clk => clk,
        nCLR => nCLR,
        nEnable => nEnable,
        PUSH => PUSH,
        POP => POP,
        data => data,
        FULL => FULL,
        EMPTY => EMPTY
    );
```

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

```
    test_proc: process
    begin
        wait for 100 ns;
```

```
        nEnable <= '0';
        PUSH <= '1';
        data <= "1010"; wait for ClockPeriode;
        data <= "0101"; wait for ClockPeriode;
        data <= "1100"; wait for ClockPeriode;
```

```

data <= "0011"; wait for ClockPeriode;
PUSH <= '0'; data <= "ZZZZ"; wait for ClockPeriode;

wait for ClockPeriode * 5;
POP <= '1'; wait for ClockPeriode * 5;
POP <= '0'; wait for ClockPeriode * 5;

nCLR <= '0'; wait for ClockPeriode / 5;
nCLR <= '1'; wait for ClockPeriode * 2;

PUSH <= '1';
data <= "0110"; wait for ClockPeriode;
data <= "1001"; wait for ClockPeriode;
PUSH <= '0'; data <= "ZZZZ"; wait for ClockPeriode;

PUSH <= '1';
data <= "1111"; wait for ClockPeriode;
POP <= '1'; wait for ClockPeriode;
PUSH <= '0'; wait for ClockPeriode;
POP <= '0'; wait for ClockPeriode;

PUSH <= '1';
for i in 0 to lifo_size + 2 loop
data <= std_logic_vector( to_unsigned( i mod 16, 4 ) );
wait for ClockPeriode;
end loop;
PUSH <= '0'; data <= "ZZZZ"; wait for ClockPeriode;

POP <= '1'; wait for ClockPeriode * 10;
POP <= '0'; wait for ClockPeriode * 10;

wait;
end process;

END behavior;

```

```

-- *****
-- **** STUDENT: 64200296
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 22:07:22 12/22/2024
-- Design Name:
-- Module Name: /home/ise/Shared_folder/LIFO/LIFO/Lifo_tb.vhd
-- Project Name: LIFO
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: Lifo
--
-- 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 lifo_tb IS
END lifo_tb;
```

```
ARCHITECTURE behavior OF lifo_tb IS
```

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

```
    COMPONENT lifo
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;
```

```
    -- Inputs
```

```
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      nEnable : std_logic := '1';
    signal      PUSH : std_logic := '0';
    signal      POP : std_logic := '0';
```

```
    -- BiDirs
```

```
    signal      data : std_logic_vector( 3 downto 0 ):= ( others => '0' );
```

```
    -- Outputs
```

```
    signal      FULL : std_logic;
    signal      EMPTY : std_logic;
```

```
    -- Clock period definitions
```

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

```
BEGIN
```

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

```
    uut: lifo PORT MAP (  
        clk => clk,  
        nCLR => nCLR,  
        nEnable => nEnable,  
        PUSH => PUSH,  
        POP => POP,  
        data => data,  
        FULL => FULL,  
        EMPTY => EMPTY  
    );
```

```
    -- 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
```

```
        nEnable    <= '0';  
        wait for clk_period*3;  
        PUSH    <= '1';  
        data    <= "0001";  
        wait for clk_period;  
        data    <= "0010";  
        wait for clk_period;
```

```
data    <= "0011";
wait for clk_period;
data    <= "0100";
wait for clk_period;
data    <= "0101";
wait for clk_period;
data    <= "0110";
wait for clk_period;
data    <= "0111";
wait for clk_period;
data    <= "1000";
wait for clk_period;
data    <= "1001";
wait for clk_period;
data    <= "1010";
wait for clk_period;
PUSH    <= '0';
data    <= "ZZZZ";
wait for clk_period*3;
POP     <= '1';
wait for clk_period*10;
POP     <= '0';
PUSH    <= '1';
data    <= "1111";
wait for clk_period*3;
PUSH    <= '0';
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 lifo_tb IS
generic(      lifo_width: natural := 4; -- velikost podatka sklada
             lifo_size:  natural := 8 );-- velikost sklada
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

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

    COMPONENT lifo
        generic(      lifo_width: natural := 4; -- velikost podatka sklada
                     lifo_size:  natural := 8 );-- velikost sklada
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;

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

```

```

signal          nEnable : std_logic := '0';
signal          PUSH : std_logic := '0';
signal          POP : std_logic := '0';

    -- BiDirs
signal          data : std_logic_vector( 3 downto 0 ) := ( others => '0' );

    -- Outputs
signal          FULL : std_logic;
signal          EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo
        GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            nEnable => nEnable,
            PUSH => PUSH,
            POP => POP,
            data => data,
            FULL => FULL,
            EMPTY => EMPTY
        );

    -- 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
    -- wait for clk_period;
    data <= "ZZZZ";
    wait for clk_period;
    nEnable <= '1';
    wait for clk_period;
    nCLR <= '1';
    wait for clk_period;
    nEnable <= '0';
    PUSH <= '1';
    data <= "0001";
    wait for clk_period;

    data <= "0010";
    wait for clk_period;

    data <= "0011";
    wait for clk_period;

    data <= "0100";
    wait for clk_period;

    data <= "0101";
    wait for clk_period;

    data <= "0110";
    wait for clk_period;

    data <= "0111";
    wait for clk_period;

    data <= "1000";
    wait for clk_period;

```

```
data    <= "1001";  
wait for clk_period;  
  
data    <= "1010";  
wait for clk_period;  
  
PUSH    <= '0';  
data    <= "ZZZZ";  
wait for clk_period*3;  
  
POP     <= '1';  
wait for clk_period*12;
```

```
-- POP <= '0';
```

```
wait;  
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210113
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 16:09:09 12/19/2024
-- Design Name:
-- Module Name: C:/Users/Mihaheh/Downloads/Trash/LIFO/LIFO/lifo_tb.vhd
-- Project Name: LIFO
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: Lifo
--
-- 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 lifo_tb IS
generic(
    lifo_width: natural := 4; -- velikost podataka sklada
    lifo_size: natural := 8 );-- velikost sklada
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

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

    COMPONENT lifo
        generic(
            lifo_width: natural := 4; -- velikost podataka sklada
            lifo_size: natural := 8 );-- velikost sklada
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      nEnable : std_logic := '1';
    signal      PUSH : std_logic := '0';
    signal      POP : std_logic := '0';

    -- BiDirs
    signal      data : std_logic_vector( 3 downto 0 ):= ( others => '0' );

    -- Outputs
    signal      FULL : std_logic;

```

```

signal          EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo
        GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            nEnable => nEnable,
            PUSH => PUSH,
            POP => POP,
            data => data,
            FULL => FULL,
            EMPTY => EMPTY
        );

    -- 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
        nEnable    <= '0';

```

```
wait for clk_period*3;

PUSH  <= '1';
data  <= "0001";
wait for clk_period;

data  <= "0010";
wait for clk_period;

data  <= "0011";
wait for clk_period;

data  <= "0100";
wait for clk_period;

data  <= "0101";
wait for clk_period;

data  <= "0110";
wait for clk_period;

data  <= "0111";
wait for clk_period;

data  <= "1000";
wait for clk_period;

data  <= "1001";
wait for clk_period;

data  <= "1010";
wait for clk_period;

PUSH  <= '0';
data  <= "ZZZZ";
wait for clk_period*3;

POP   <= '1';
wait for clk_period*10;
```

```
POP    <= '0';  
PUSH   <= '1';  
data   <= "1111";  
wait for clk_period*3;
```

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

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

```
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 lifo_tb IS
generic(
    lifo_width: natural := 4;
    lifo_size: natural := 8 );
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

    COMPONENT lifo
        generic(
            lifo_width: natural := 4;
            lifo_size: natural := 8 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;

    -- Inputs
    signal
        clk : std_logic := '0';
        nCLR : std_logic := '1';
        nEnable : std_logic := '1';
        PUSH : std_logic := '0';
        POP : std_logic := '0';

    signal
        data : std_logic_vector( 3 downto 0 ) := ( others => '0' );

```



```

signal          FULL : std_logic;
signal          EMPTY : std_logic;

constant        clk_period : time := 10 ns;

BEGIN

    uut: lifo
        GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            nEnable => nEnable,
            PUSH => PUSH,
            POP => POP,
            data => data,
            FULL => FULL,
            EMPTY => EMPTY
        );

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

```

```
PUSH  <= '1';
data  <= "0001";
wait for clk_period;
data  <= "0010";
wait for clk_period;
data  <= "0011";
wait for clk_period;
data  <= "0100";
wait for clk_period;
data  <= "0101";
wait for clk_period;
data  <= "0110";
wait for clk_period;
data  <= "0111";
wait for clk_period;
data  <= "1000";
wait for clk_period;
data  <= "1001";
wait for clk_period;
data  <= "1010";
wait for clk_period;
PUSH  <= '0';
data  <= "ZZZZ";
wait for clk_period*3;
POP   <= '1';
wait for clk_period*10;
POP   <= '0';
PUSH  <= '1';
data  <= "1111";
wait for clk_period*3;
PUSH  <= '0';
nCLR  <= '0';
wait for clk_period/5;
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 lifo_tb IS
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

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

    COMPONENT lifo
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '0';
    signal      nEnable : std_logic := '0';
    signal      PUSH : std_logic := '0';
    signal      POP : std_logic := '0';

    -- BiDirs
    signal      data : std_logic_vector( 3 downto 0 ) := ( others=>'Z' );

```

```

    -- Outputs
    signal FULL : std_logic;
    signal EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo PORT MAP (
        clk => clk,
        nCLR => nCLR,
        nEnable => nEnable,
        PUSH => PUSH,
        POP => POP,
        data => data,
        FULL => FULL,
        EMPTY => EMPTY
    );

    -- 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

        nCLR    <= '0';
        nENABLE  <= '1';
        PUSH    <= '0';
        POP     <= '0';
        data    <= ( others=>'Z' );
    end process;

```

```
wait for clk_period*0.25;
nCLR  <= '1';

wait for clk_period*1.25;
nENABLE  <= '0';
PUSH  <= '1';

wait for clk_period*0.9;
for idi in 1 to 10 loop
    data <= std_logic_vector( to_unsigned( idi, data'length ) );
    wait for clk_period;
end loop;

wait for clk_period*5;
PUSH  <= '0';
data  <= ( others=>'Z' );
wait for clk_period*2;

POP  <= '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 lifo_tb IS
generic(
    lifo_width: natural := 4;
    lifo_size: natural := 8 );
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

    COMPONENT lifo
        generic(
            lifo_width: natural := 4;
            lifo_size: natural := 8 );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;

    -- Inputs
    signal
        clk : std_logic := '0';
        nCLR : std_logic := '1';
        nEnable : std_logic := '1';
        PUSH : std_logic := '0';
        POP : std_logic := '0';

    signal
        data : std_logic_vector( 3 downto 0 ) := ( others => '0' );

```

```

signal          FULL : std_logic;
signal          EMPTY : std_logic;

constant        clk_period : time := 10 ns;

BEGIN

    uut: lifo
        GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            nEnable => nEnable,
            PUSH => PUSH,
            POP => POP,
            data => data,
            FULL => FULL,
            EMPTY => EMPTY
        );

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

```

```

PUSH  <= '1';
data  <= "0001";
wait for clk_period;
data  <= "0010";
wait for clk_period;
data  <= "0011";
wait for clk_period;
data  <= "0100";
wait for clk_period;
data  <= "0101";
wait for clk_period;
data  <= "0110";
wait for clk_period;
data  <= "0111";
wait for clk_period;
data  <= "1000";
wait for clk_period;
data  <= "1001";
wait for clk_period;
data  <= "1010";
wait for clk_period;
PUSH  <= '0';
data  <= "ZZZZ";
wait for clk_period*3;
POP   <= '1';
wait for clk_period*10;
POP   <= '0';
PUSH  <= '1';
data  <= "1111";
wait for clk_period*3;
PUSH  <= '0';
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: 17:29:25 12/24/2024
-- Design Name:
-- Module Name: C:/NDV/DN5/Lifo_tb.vhd
-- Project Name: DN5
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: Lifo
--
-- 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 lifo_tb IS
    generic(
        lifo_width: natural := 4; -- velikost podatka sklada
        lifo_size: natural := 8  -- velikost sklada
    );
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

    -- Component Declaration for the Unit Under Test ( UUT )
    COMPONENT lifo
        generic(
            lifo_width: natural := 4; -- velikost podatka sklada
            lifo_size: natural := 8  -- velikost sklada
        );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      nEnable : std_logic := '1';
    signal      PUSH : std_logic := '0';
    signal      POP : std_logic := '0';

    -- BiDirs
    signal      data : std_logic_vector( 3 downto 0 ) := ( others => '0' );

```

```

-- Outputs
signal      FULL : std_logic;
signal      EMPTY : std_logic;

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

BEGIN

-- Instantiate the Unit Under Test ( UUT )
 uut: lifo
   GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
   PORT MAP (
     clk => clk,          nCLR => nCLR,          nEnable => nEnable,          PUSH => PUSH,
     POP => POP,          data => data,          FULL => FULL,          EMPTY => EMPTY
   );

-- 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
  nEnable    <= '0';
  wait for clk_period*3;
  PUSH    <= '1';
  data    <= "0001";
  wait for clk_period;

```

```
data    <= "0010";
wait for clk_period;
data    <= "0011";
wait for clk_period;
data    <= "0100";
wait for clk_period;
data    <= "0101";
wait for clk_period;
data    <= "0110";
wait for clk_period;
data    <= "0111";
wait for clk_period;
data    <= "1000";
wait for clk_period;
data    <= "1001";
wait for clk_period;
data    <= "1010";
wait for clk_period;
PUSH    <= '0';
data    <= "ZZZZ";
wait for clk_period*3;
POP     <= '1';
wait for clk_period*10;
POP     <= '0';
PUSH    <= '1';
data    <= "1111";
wait for clk_period*3;
PUSH    <= '0';
nCLR    <= '0';
wait for clk_period/5;
nCLR    <= '1';
```

```
wait;
end process;
```

```
END;
```

```

-- *****
-- **** STUDENT: 64210386
-- *****
-- KOMENTARJI K OCENI NALOGE
-- Matej Možek: Ni pripomb
-- *****
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --      --      --      --      --      --      --      --      --      --
--      --      --      --      --      --
-- Company:
-- Engineer:
--
-- Create Date: 03:36:01 12/21/2024
-- Design Name:
-- Module Name: /home/ise/Xilinx_shared/domaca_naloga_5/lifo_tb.vhd
-- Project Name: domaca_naloga_5
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: Lifo
--
-- 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 lifo_tb IS
generic(
    lifo_width: natural := 4; -- velikost podatka sklada
    lifo_size: natural := 8 );-- velikost sklada
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

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

    COMPONENT lifo
        generic(
            lifo_width: natural := 4; -- velikost podatka sklada
            lifo_size: natural := 8 );-- velikost sklada
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( lifo_width - 1 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      nEnable : std_logic := '1';
    signal      PUSH : std_logic := '0';
    signal      POP : std_logic := '0';

    -- BiDirs
    signal      data : std_logic_vector( lifo_width - 1 downto 0 ) := ( others => '0' );

    -- Outputs
    signal      FULL : std_logic;

```

```

signal          EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo PORT MAP (
    clk => clk,
    nCLR => nCLR,
    nEnable => nEnable,
    PUSH => PUSH,
    POP => POP,
    data => data,
    FULL => FULL,
    EMPTY => EMPTY
    );

    -- 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

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

```

```
PUSH  <= '1';
data  <= "0001";
wait for clk_period;
data  <= "0010";
wait for clk_period;
data  <= "0011";
wait for clk_period;
data  <= "0100";
wait for clk_period;
data  <= "0101";
wait for clk_period;
data  <= "0110";
wait for clk_period;
data  <= "0111";
wait for clk_period;
data  <= "1000";
wait for clk_period;
data  <= "1001";
wait for clk_period;
data  <= "1010";
wait for clk_period;
PUSH  <= '0';
data  <= "ZZZZ";
wait for clk_period*3;
POP   <= '1';
wait for clk_period*10;
POP   <= '0';
PUSH  <= '1';
data  <= "1111";
wait for clk_period*3;
PUSH  <= '0';
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 lifo_tb IS
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

    COMPONENT lifo
    PORT(
        clk : IN  std_logic;
        nCLR : IN  std_logic;
        nEnable : IN  std_logic;
        PUSH : IN  std_logic;
        POP : IN  std_logic;
        data : INOUT std_logic_vector(3 downto 0);
        FULL : OUT  std_logic;
        EMPTY : OUT  std_logic
    );
    END COMPONENT;

--Inputs
signal clk : std_logic := '0';
signal nCLR : std_logic := '0';
signal nEnable : std_logic := '0';
signal PUSH : std_logic := '0';
signal POP : std_logic := '0';

--BiDirs
signal data : std_logic_vector(3 downto 0);

--Outputs
signal FULL : std_logic;

```

```

signal EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: lifo PORT MAP (
        clk => clk,
        nCLR => nCLR,
        nEnable => nEnable,
        PUSH => PUSH,
        POP => POP,
        data => data,
        FULL => FULL,
        EMPTY => EMPTY
    );

    -- 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

        nEnable <= '0';
        wait for clk_period;

        PUSH <= '1';

```

```
data <= "0001";
wait for clk_period;

data <= "0010";
wait for clk_period;

data <= "0011";
wait for clk_period;

data <= "0100";
wait for clk_period;

data <= "0101";
wait for clk_period;

data <= "0110";
wait for clk_period;

data <= "0111";
wait for clk_period;

data <= "1000";
wait for clk_period;

data <= "1001";
wait for clk_period;

data <= "1010";
wait for clk_period;

PUSH <= '0';
data <= "ZZZZ";
wait for clk_period;

POP <= '1';
wait for clk_period;

POP <= '0';
PUSH <= '1';
data <= "1111";
```

```
wait for clk_period;
```

```
PUSH <= '0';
```

```
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 lifo_tb IS
generic(      lifo_width: natural := 4; -- velikost podatka sklada
              lifo_size: natural := 8 );-- velikost sklada
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

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

    COMPONENT lifo
        generic(      lifo_width: natural := 4; -- velikost podatka sklada
                      lifo_size: natural := 8 );-- velikost sklada
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
    END COMPONENT;

    -- Inputs
    signal      clk : std_logic := '0';
    signal      nCLR : std_logic := '1';
    signal      nEnable : std_logic := '1';
    signal      PUSH : std_logic := '0';
    signal      POP : std_logic := '0';

```

```

    -- BiDirs
signal      data : std_logic_vector( 3 downto 0 ) := ( others => '0' );

    -- Outputs
signal      FULL : std_logic;
signal      EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo
        GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            nEnable => nEnable,
            PUSH => PUSH,
            POP => POP,
            data => data,
            FULL => FULL,
            EMPTY => EMPTY
        );

    -- 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
nEnable    <= '0';
wait for clk_period*3;
PUSH    <= '1';
data    <= "0001";
wait for clk_period;
data    <= "0010";
wait for clk_period;
data    <= "0011";
wait for clk_period;
data    <= "0100";
wait for clk_period;
data    <= "0101";
wait for clk_period;
data    <= "0110";
wait for clk_period;
data    <= "0111";
wait for clk_period;
data    <= "1000";
wait for clk_period;
data    <= "1001";
wait for clk_period;
data    <= "1010";
wait for clk_period;
PUSH    <= '0';
data    <= "ZZZZ";
wait for clk_period*3;
POP     <= '1';
wait for clk_period*10;
POP     <= '0';
PUSH    <= '1';
data    <= "1111";
wait for clk_period*3;
PUSH    <= '0';
nCLR    <= '0';
wait for clk_period/5;
nCLR    <= '1';

```

```
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 lifo_tb IS
generic(
    lifo_width: natural := 4; -- velikost podatka sklada
    lifo_size: natural := 8 );-- velikost sklada
END lifo_tb;

```

```

ARCHITECTURE behavior OF lifo_tb IS

```

```

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

```

```

    COMPONENT lifo
    generic(
        lifo_width: natural := 4; -- velikost podatka sklada
        lifo_size: natural := 8 );-- velikost sklada

```

```

    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
END COMPONENT;

```

```

    -- Inputs

```

```

    signal      clk : std_logic := '0';

```

```

signal          nCLR : std_logic := '1';
signal          nEnable : std_logic := '1';
signal          PUSH : std_logic := '0';
signal          POP : std_logic := '0';

    -- BiDirs
signal          data : std_logic_vector( 3 downto 0 ) := ( others => '0' );

    -- Outputs
signal          FULL : std_logic;
signal          EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo
        GENERIC MAP ( lifo_width => lifo_width,          lifo_size => lifo_size )
        PORT MAP (
            clk => clk,
            nCLR => nCLR,
            nEnable => nEnable,
            PUSH => PUSH,
            POP => POP,
            data => data,
            FULL => FULL,
            EMPTY => EMPTY
        );

    -- 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
        nEnable    <= '0';
        wait for clk_period*3;
        PUSH    <= '1';
        data    <= "0001";
        wait for clk_period;
        data    <= "0010";
        wait for clk_period;
        data    <= "0011";
        wait for clk_period;
        data    <= "0100";
        wait for clk_period;
        data    <= "0101";
        wait for clk_period;
        data    <= "0110";
        wait for clk_period;
        data    <= "0111";
        wait for clk_period;
        data    <= "1000";
        wait for clk_period;
        data    <= "1001";
        wait for clk_period;
        data    <= "1010";
        wait for clk_period;
        PUSH    <= '0';
        data    <= "ZZZZ";
        wait for clk_period*3;
        POP     <= '1';
        wait for clk_period*10;
        POP     <= '0';
        PUSH    <= '1';
        data    <= "1111";

```

```
wait for clk_period*3;  
PUSH  <= '0';  
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;
use ieee.numeric_std.all;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- USE ieee.numeric_std.ALL;

ENTITY lifo_tb IS
    generic(
        lifo_width: natural := 4;
        lifo_size: natural := 8
    );
END lifo_tb;

ARCHITECTURE behavior OF lifo_tb IS

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

    COMPONENT lifo
        generic(
            lifo_width: natural := 4;
            lifo_size: natural := 8
        );
    PORT(
        clk : IN std_logic;
        nCLR : IN std_logic;
        nEnable : IN std_logic;
        PUSH : IN std_logic;
        POP : IN std_logic;
        data : INOUT std_logic_vector( 3 downto 0 );
        FULL : OUT std_logic;
        EMPTY : OUT std_logic
    );
END COMPONENT;

```

```

-- Inputs
signal      nCLR : std_logic := '1';
signal      nEnable : std_logic := '1';
signal      PUSH : std_logic := '0';
signal      POP : std_logic := '0';

-- BiDirs
signal      data : std_logic_vector( 3 downto 0 );

-- Outputs
signal      FULL : std_logic;
signal      EMPTY : std_logic;

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

BEGIN

    -- Instantiate the Unit Under Test ( UUT )
    uut: lifo PORT MAP (
        clk => clock,
        nCLR => nCLR,
        nEnable => nEnable,
        PUSH => PUSH,
        POP => POP,
        data => data,
        FULL => FULL,
        EMPTY => EMPTY
    );

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

```

```

-- Stimulus process
stimulus_process: process
begin

    nCLR    <= '0';
    nEnable    <= '0';
--    PUSH    <= '1';
    data    <= "0101";
    wait for 2*clock_period;

    nCLR    <= '1';
    nEnable    <= '1';
--    PUSH    <= '1';
    wait for clock_period;

--    wait for clock_period;
    nEnable    <= '0';
    PUSH    <= '1';
    POP    <= '0';
    for i in 1 to 10 loop
        data    <= std_logic_vector( to_unsigned( i, data'length ) );
        wait for clock_period;
    end loop;

    PUSH    <= '0';
    data    <= "ZZZZ";
    wait for 2*clock_period;
    nCLR    <= '0';
    wait for clock_period;
    nCLR    <= '1';
--    nEnable    <= '0';
    PUSH    <= '1';
--    POP    <= '0';
    for i in 1 to 6 loop
        data    <= std_logic_vector( to_unsigned( i, data'length ) );
        wait for clock_period;
    end loop;

    PUSH    <= '0';

```

```
data  <= "ZZZZ";  
wait for 3*clock_period;  
  
POP   <= '1';  
      wait for clock_period * 10;
```

```
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 lifo_tb IS
    generic(
        lifo_width: natural := 8; -- dolžina vhodnega podatka
        lifo_size: natural := 11 -- število hranjenih podatkov
    );
END lifo_tb;

ARCHITECTURE lifo_tb_ideal OF lifo_tb IS
    FILE RESULTS: TEXT OPEN WRITE_MODE IS "lifo.csv";

    COMPONENT lifo
    generic(
        lifo_width: natural := 4; -- velikost podatka sklada
        lifo_size: natural := 8 ); -- velikost sklada
    PORT (
        clk, -- signal ure
        nCLR, -- signal asinhronnega brisanja vsebine sklada ( aktiven '0' )
        nEnable, -- signal omogočanja sklada ( aktiven '0', sicer drži vsebino )
        PUSH, -- operacija vpisa na sklad ( aktiven '1' )
        POP : IN std_logic; -- operacija branja s sklada ( aktiven '1' )
        data : inout std_logic_vector( lifo_width - 1 downto 0 ); -- tristanski izhod sklada
        FULL, -- izhod, ki postane '1', ko je sklad poln
        EMPTY : OUT std_logic -- izhod, ki postane '0', ko je sklad prazen
    );
END COMPONENT;

SIGNAL clk : std_logic := '0';
SIGNAL nCLR : std_logic := '0';
SIGNAL nEnable : std_logic := '0';
SIGNAL PUSH : std_logic := '0';
SIGNAL POP : std_logic := '0';

```

```

SIGNAL      data : std_logic_vector ( lifo_width-1 DownTo 0 ) := ( others => 'Z' );
SIGNAL      FULL : std_logic := '0';
SIGNAL      EMPTY : std_logic := '0';

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

```

```
BEGIN
```

```

    UUT : lifo
    generic MAP (
        lifo_width => lifo_width, -- dolžina vhodnega podatka
        lifo_size => lifo_size    -- število hranjenih podatkov
    )
    PORT MAP (
        clk => clk,          nCLR => nCLR,          nEnable => nEnable,          PUSH => PUSH,
        POP => POP,          data => data,          FULL => FULL,          EMPTY => EMPTY
    );

```

```
PROCESS    -- clock process for clk
```

```

    PROCEDURE Log_variables(
        nCLR : std_logic;
        nEnable : std_logic;
        PUSH : std_logic;
        POP : std_logic;
        data : std_logic_vector ( lifo_width-1 DownTo 0 );
        FULL : std_logic;
        EMPTY : std_logic
    ) IS
        VARIABLE RES_LINE : LINE;
    BEGIN
        write( RES_LINE, nCLR, right, 1 );
        write( RES_LINE, string'( "," ) );
        write( RES_LINE, nEnable, right, 1 );
        write( RES_LINE, string'( "," ) );
        write( RES_LINE, PUSH , right, 1 );
        write( RES_LINE, string'( "," ) );
        write( RES_LINE, POP , right, 1 );
        write( RES_LINE, string'( "," ) );
        hwrite( RES_LINE, data, right, 4 );
    END LOG_VARIABLES;

```

```

        write( RES_LINE, string'( "," ) );
        write( RES_LINE, FULL , right, 1 );
        write( RES_LINE, string'( "," ) );
        write( RES_LINE, EMPTY , right, 1 );
        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 , nEnable, PUSH, POP, data, FULL, EMPTY );
    END LOOP CLOCK_LOOP;
END PROCESS;

PROCESS
    variable HDR_line : LINE;

    BEGIN
        write( HDR_line, string'( " nCLR , nEnable, PUSH, POP, data, FULL, EMPTY" ) );
        writeline( RESULTS, HDR_line );

        WAIT FOR PERIOD;
        nEnable      <= '1';          -- disable stack

        WAIT FOR PERIOD;
        nCLR  <= '1';          -- activate clear

        WAIT FOR PERIOD;
        PUSH  <= '1';          -- set push
        nEnable      <= '0';          -- enable stack

        for i in 0 to lifo_size loop
            -- Loop over Lifo capactiy ( push one more )
            data  <= std_logic_vector( to_unsigned( i, data'length ) );
            WAIT FOR PERIOD;
        end loop;
    
```

```
PUSH  <= '0';      -- stop pushing
data  <= ( others => 'Z' );    -- release bus

WAIT FOR PERIOD;
POP   <= '1';      -- start popping the contents

WAIT; -- sit here

END PROCESS;

END lifo_tb_ideal;
```

