Ministerul Educaţiei, al Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Informatică și Ingineria Sistemelor

**RAPORT**

Lucrarea de laborator nr.3

PDP

A efectuat:

st. gr. C-171 D. Melniciuc

A verificat:

dr., conf.univ. V. Cărbune

Chişinău 2020

Varianta:14 (2)

1. Se vor modela circuitele pentru exemplele Nr. 3, 5, 7 și 9. Pentru fiecare circuit se va crea un proiect nou.

3)

library ieee;

use ieee.std\_logic\_1164.all;

entity dff is

port (d: in std\_logic;

clk: in std\_logic;

q: out std\_logic;

reset: in std\_logic);

end dff;

architecture example\_3 of dff is

begin

process (clk, reset)

begin

if (reset = '1') then

q <= '0';

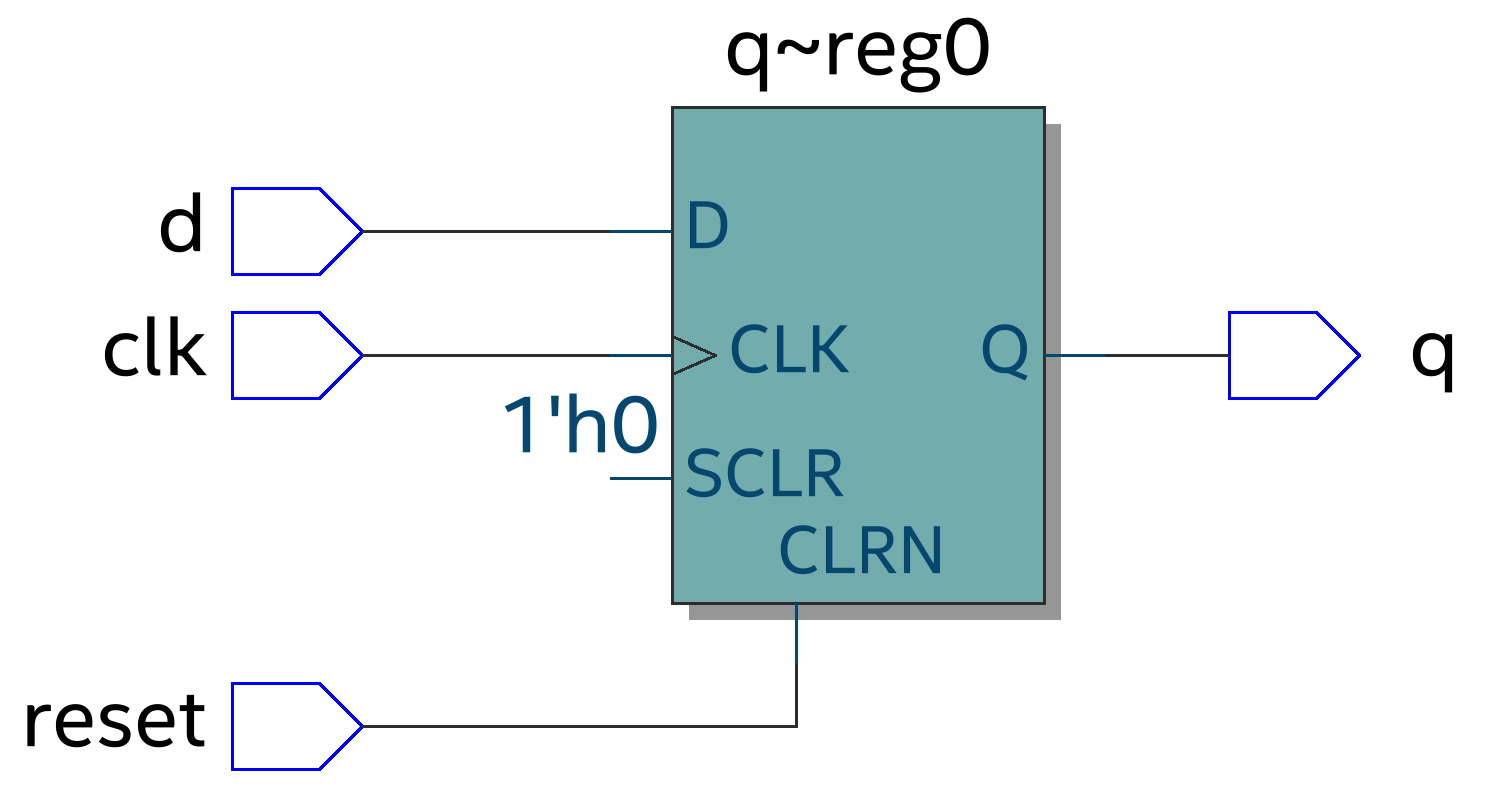
elsif rising\_edge (clk) then

q <= d;

end if;

end process;

end example\_3;



5)

library ieee;

use ieee.std\_logic\_1164.all;

entity reg8 is

port (d: in std\_logic\_vector (7 downto 0);

clk: in std\_logic;

q: out std\_logic\_vector (7 downto 0));

end reg8;

architecture ex\_reg of reg8 is

begin

process (clk)

begin

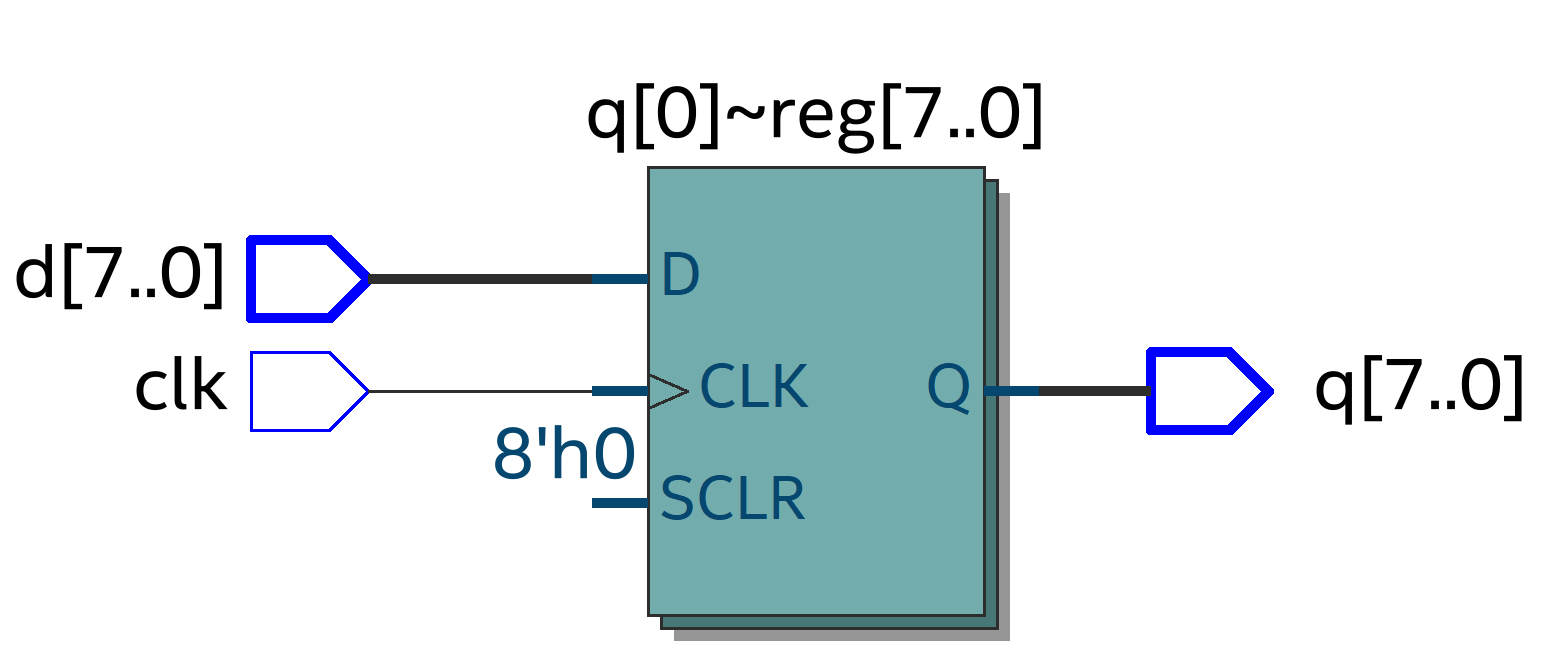
if (clk'event and clk = '1') then

q <= d;

end if;

end process;

end ex\_reg;



7)

library ieee;

use ieee.std\_logic\_1164.all;

entity reg8\_depl is

port (clk, ce, si: in std\_logic;

so: out std\_logic);

end reg8\_depl;

architecture reg\_depl of reg8\_depl is

signal tmp: std\_logic\_vector (7 downto 0);

begin

process (clk)

begin

if (clk'event and clk = '1') then

if (ce = '1') then

for i in 0 to 6 loop

tmp(i+1) <= tmp(i);

end loop;

tmp(0) <= si; -- încărcare serială

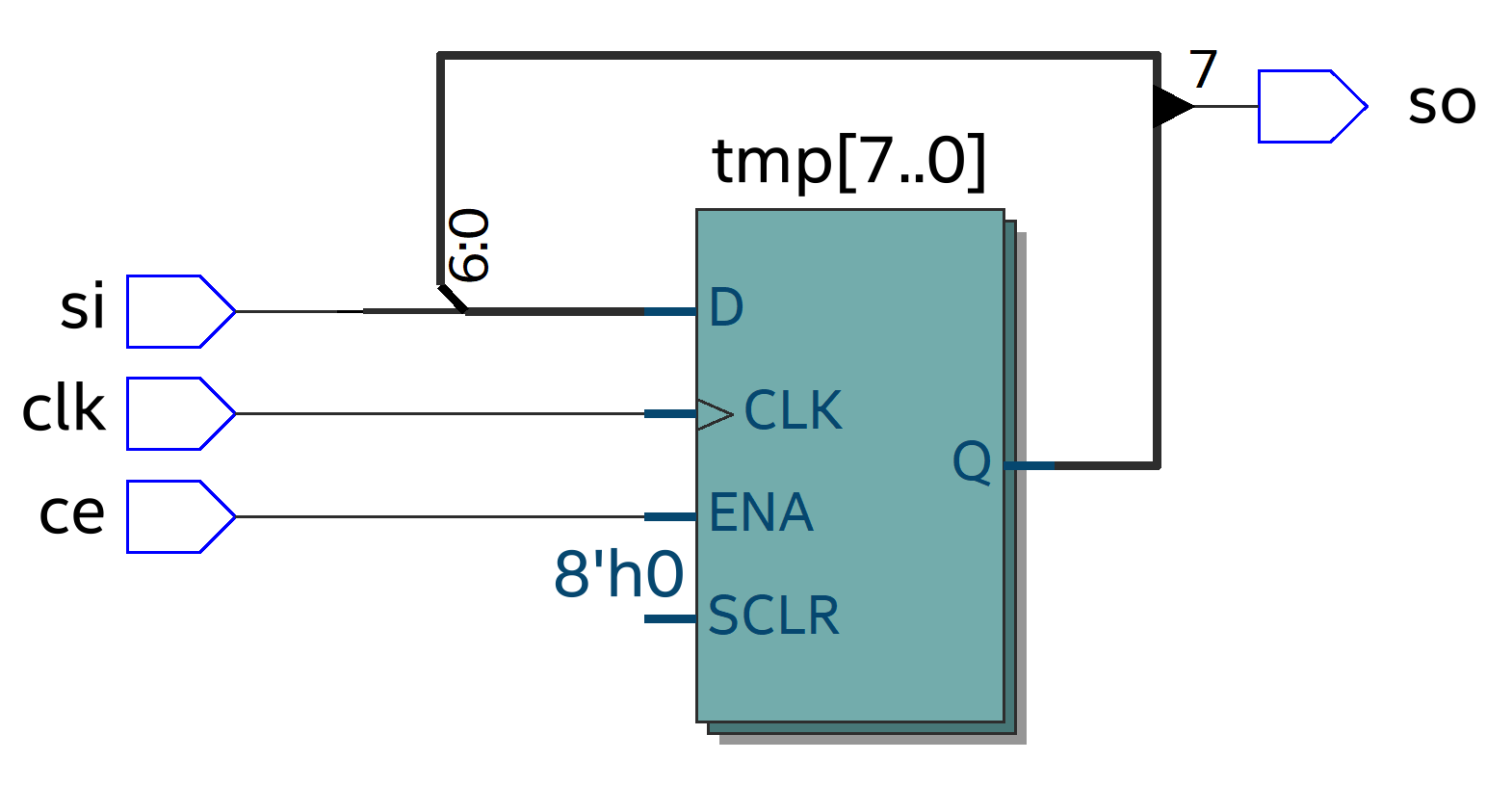
end if;

end if;

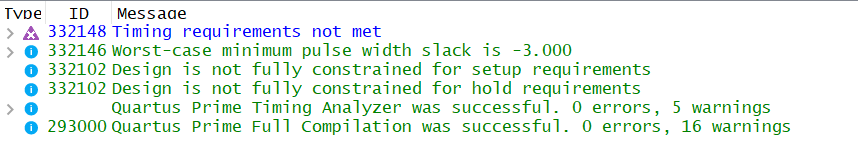
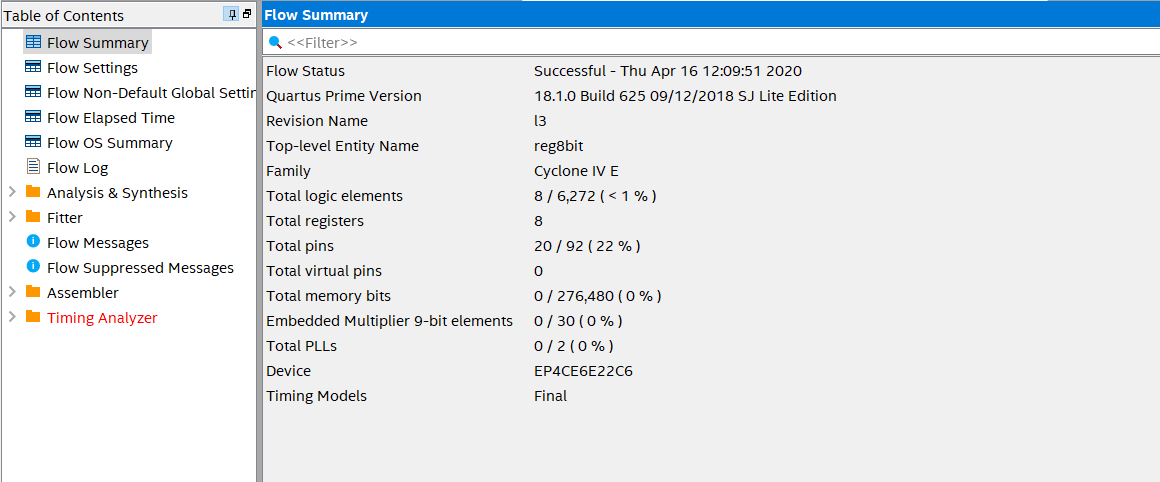
end process;

so <= tmp(7); -- ieșire serială

end reg\_depl;



1. Se va proiecta un registru conform variantei din tabelul 1.



-- Num. de biți 8

-- CLK Front descrescător

-- Resetare asincronă

-- Setare nu

-- Direcție deplasare stânga

-- Încărcare paralelă

-- Iesire paralelă

library ieee;

use ieee.std\_logic\_1164.all;

entity reg8bit is

port(clk, SI, SLOAD,reset : in std\_logic;

D : in std\_logic\_vector(7 downto 0);

SO : out std\_logic\_vector(7 downto 0));

end reg8bit;

architecture archi of reg8bit is

signal tmp : std\_logic\_vector(7 downto 0);

signal p\_state : std\_logic\_vector(7 downto 0);

begin

process (clk, reset)

begin

if (reset = '0') then

p\_state <= (others => '0');

elsif (clk'event and clk='0') then

if (SLOAD='1') then

tmp <= D; --încărcare paralelă

else

tmp <= tmp(6 downto 0) & SI;

end if;

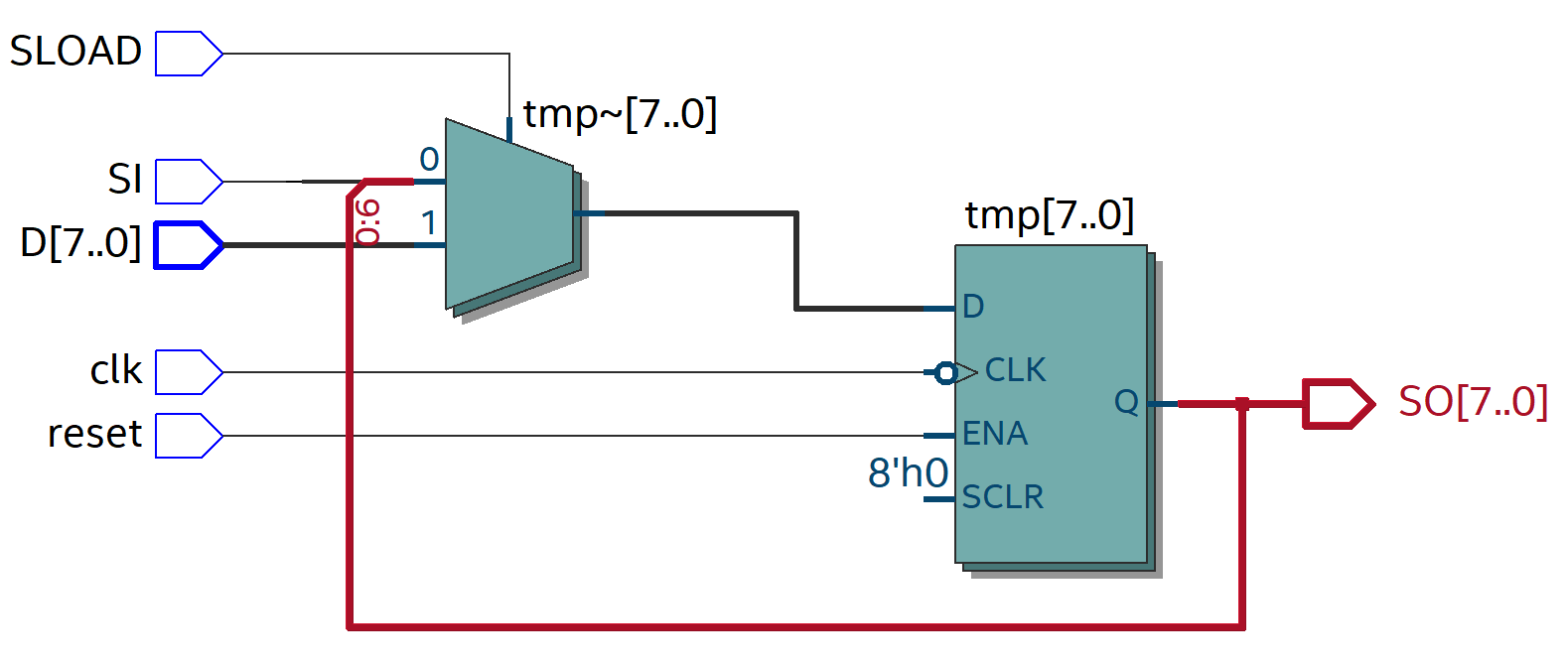
p\_state <= tmp;

end if;

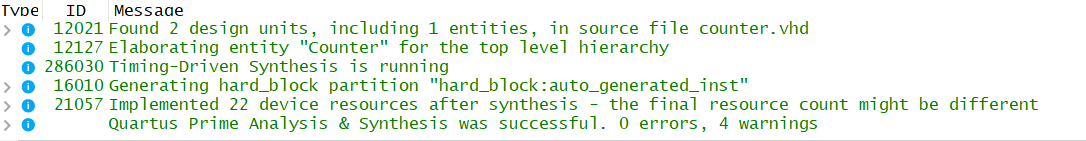
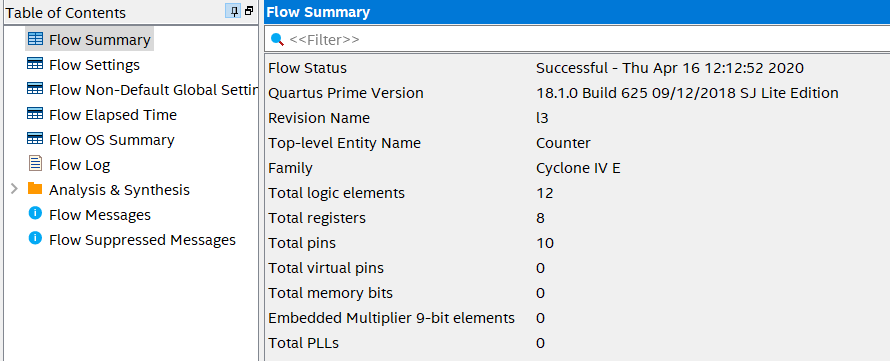
end process;

SO <= tmp;

end archi;



1. Se va proiecta un numărător conform variantei din tabelul 2.



-- mod 9

-- CLK Front crescător

-- Resetare asincronă

-- Tip numărător Invers

-- Secvența de numărare 14-6

-- Încărcare paralelă

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Counter is

Port(clk : in STD\_LOGIC;

reset : in STD\_LOGIC;

count : out STD\_LOGIC\_VECTOR (7 downto 0));

end Counter;

architecture Behavioral of Counter is

signal count\_int : std\_logic\_vector(7 downto 0);

begin

process (reset, clk)

begin

if reset = '1' then

count\_int <= "00001110";

elsif clk'event and clk = '1' then

if count\_int <= "00001110" and count\_int > "0000110" then

count\_int <= count\_int - "1";

else

count\_int <= "00001110";

end if;

end if;

end process;

count <= count\_int;

end Behavioral;

