Codul pentru DFF(EDFFTR):

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL;

entity EDFFTR is Port ( RN : in STD\_LOGIC;

E : in STD\_LOGIC;

D : in STD\_LOGIC;

CK : in STD\_LOGIC;

Q : out STD\_LOGIC;

QN : out STD\_LOGIC

); end EDFFTR;

architecture Comportament of EDFFTR is

-- Parametrii dinamici (în ps pentru simulare precisă)

constant tpLH\_CK\_Q : time := 498 ps; -- Timp de propagare CK -> Q

constant tpHL\_CK\_Q : time := 660 ps; -- Timp de propagare CK -> Q

constant tpLH\_CK\_QN : time := 808 ps; -- Timp de propagare CK -> QN

constant tpHL\_CK\_QN : time := 644 ps; -- Timp de propagare CK -> QN

constant tSETUP : time := 515 ps; -- Setup time pentru D înainte de CK

constant tHOLD : time := 886 ps; -- Hold time pentru D după CK

begin

process(CK)

begin if rising\_edge(CK) then

-- Verificarea timpului de setup

assert D'stable(tSETUP)

report "Setup time violated" severity error;

-- Verificarea timpului de hold  
 assert D'stable(tHOLD)  
 report "Hold time violated" severity error;  
  
 -- Comportamentul bistabilului  
 if RN = '0' then  
 Q <= '0' after tpHL\_CK\_Q;  
 QN <= '1' after tpLH\_CK\_QN;  
 elsif E = '1' then  
 Q <= D after tpLH\_CK\_Q;  
 QN <= not D after tpHL\_CK\_QN;  
 end if;  
 end if;  
end process;

end Comportament;

Testbench EDFFTR

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL;

entity TB\_EDFFTR is -- Testbench-ul nu are porturi end TB\_EDFFTR;

architecture Comportament of TB\_EDFFTR is

-- Declararea semnalelor de test  
signal RN : STD\_LOGIC := '1'; -- R initial   
signal E : STD\_LOGIC := '0'; -- E initial   
signal D : STD\_LOGIC := '0'; -- D initial  
signal CK : STD\_LOGIC := '0'; -- Ceas ini?ial  
signal Q : STD\_LOGIC;   
signal QN : STD\_LOGIC;

begin

-- Instantierea DFF  
uut: entity work.EDFFTR  
 port map (  
 RN => RN,  
 E => E,  
 D => D,  
 CK => CK,  
 Q => Q,  
 QN => QN  
 );  
  
-- Generarea semnalului de ceas (50% duty cycle)  
clk\_process: process  
begin  
 while true loop  
 CK <= '0';  
 wait for 10 ns;  
 CK <= '1';  
 wait for 10 ns;  
 end loop;  
end process;  
  
-- Proces de testare  
test\_process: process  
begin  
 -- Ini?ializare  
 wait for 20 ns; -- A?teapt? dou? cicluri de ceas pentru stabilizare  
  
 -- Test 1: Reset activ (RN = 0)  
 RN <= '0'; wait for 20 ns; -- Activeaz? resetul  
 RN <= '1'; wait for 20 ns; -- Dezactiveaz? resetul  
  
 -- Test 2: Enable dezactivat (E = 0)  
 D <= '1'; E <= '0'; wait for 40 ns; -- Schimb? valoarea lui D, dar ie?irea nu trebuie s? se modifice  
  
 -- Test 3: Enable activ (E = 1)  
 E <= '1'; wait for 40 ns; -- Ie?irea Q trebuie s? preia valoarea lui D  
  
 -- Test 4: Schimbarea valorii lui D cu enable activ  
 D <= '0'; wait for 20 ns;  
 D <= '1'; wait for 20 ns;  
  
 -- Test 5: Reset activ cu enable activ  
 RN <= '0'; wait for 20 ns; -- Ie?irile trebuie resetate  
 RN <= '1'; wait for 20 ns;  
  
 -- Încheierea simul?rii  
 wait;  
end process;

end Comportament;

