

Русенски университет „Ангел Кънчев“ Факултет „Електротехника,
електроника и автоматика“ Дисциплина „Технология на проектирането“

Курсова работа

име: Спас Стефанов Спасов
фак. номер 213137
група 6 Б
курс 2

Проверил:
/..... /

2023.05.30

Задание

Вариант 3

Да се проектира 4-разряден изместващ регистър с паралелни входове.
Изместването да се извършва наляво по задния фронт на тактовия сигнал.
Стойността на регистъра се извежда чрез седемсегментния индикатор на макета.

ТАБЛИЦИ ЗА ИСТИННОСТ

CLK	RESET	SHIFT	DATAIN	DATAOUT
0	1	X	XXXX	00000000
1	0	0	XXXX	DATAOUT
1	0	1	XXXX	Изместване на ляво

Структура на проекта в средата на VIVADO



[illegible]

VHDL описание (модул LeftShiftParallelRegister.vhd)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity LeftShiftParallelRegister is
    Port ( CLK : in STD_LOGIC;
          RESET : in STD_LOGIC;
          SHIFT : in STD_LOGIC;
          DATAIN : in STD_LOGIC_VECTOR (3 downto 0);
          DATAOUT : out STD_LOGIC_VECTOR (7 downto 0));
end LeftShiftParallelRegister;

architecture Behavioral of LeftShiftParallelRegister is
    --сигнал за връзка между вход и изход
    signal A : std_logic_vector(7 downto 0);
begin
    process (CLK, RESET)
    begin
        if RESET = '1' then
            A <= (others => '0');
        elsif rising_edge(CLK) then
            if SHIFT = '1' then
                A <= A(3 downto 0) & DATAIN;
            end if;
        end if;
    end process;

    DATAOUT <= A;
end Behavioral;
```

СИМУЛАЦИЯ

Текстов набор(TestBench)

```
-- задаване на периода за тактовия сигнал
constant clock_period: time := 10 ns;
begin

-- тестване на ресет функция
RESET <= '1';
wait for 5 ns;
RESET <= '0';
wait for 5 ns;

SHIFT <= '1'; -- активиране на изместващия сигнал
DATAIN <= "0101";
wait for 20 ns;

SHIFT <= '1';
DATAIN <= "1110";
wait for 20 ns;

SHIFT <= '1';
DATAIN <= "0001";
wait for 20 ns;

SHIFT <= '1';
DATAIN <= "1001";
wait for 20 ns;

stop_the_clock <= true;
wait;
end process;

clocking: process
begin
while not stop_the_clock loop
CLK <= '0', '1' after clock_period / 2;-- генериране на тактов сигнал
wait for clock_period;
end loop;
wait;
```

Резултат от симулацията

