Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №2

Проверил: Выполнил:

Кобяк И. П. студент гр., 150504

Горбачевский К.В.

Минск 2023

1. **Составление программного кода на языке VHDL, реализующего операцию логического И**

Данный код реализует уравнение: y =x1&x2.

entity and2 is

port(

x1, x2: in BIT;

y:out BIT

);

end and2;

architecture func of and2 is

begin

y<=x1 and x2;

end func;

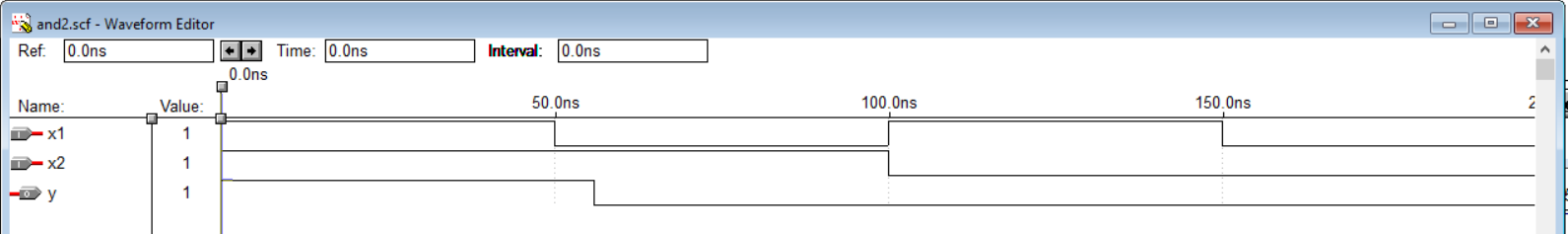


Рисунок 1.2 – Диаграмма состояний логического И

1. **Сумматор**

Данный код реализует сумматор:

S =

P = *ab + bc + ac.*

entity sum is

port(

c, a, b: in BIT;

s, p:out BIT

);

end sum;

architecture func of sum is

begin

s<=(a xor b)xor c;

p<=((a xor b) and c) or (a and b);

end func;

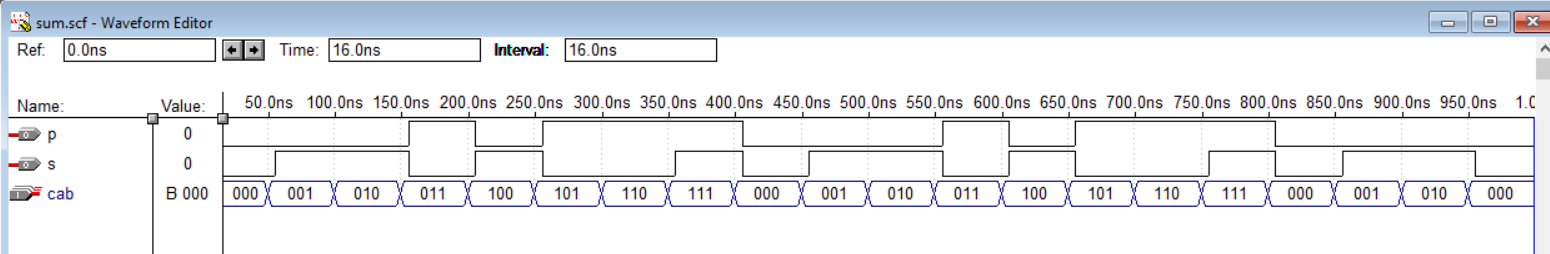


Рисунок 2.1 – Диаграмма состояний сумматора

1. **Умножитель для двухразрядных чисел**

*Получение частичной суммы*

Данный код реализует полусумматор *add1*:

*S = (a ⊕ b)⊕ c,*

*P = (a ⊕ b) & c.*

entity add1 is

port(

a, b: in BIT;

s, p:out BIT

);

end add1;

architecture func of add1 is

begin

s<=a xor b;

p<=a and b;

end func;

Из этого кода был получен символ ADD1.

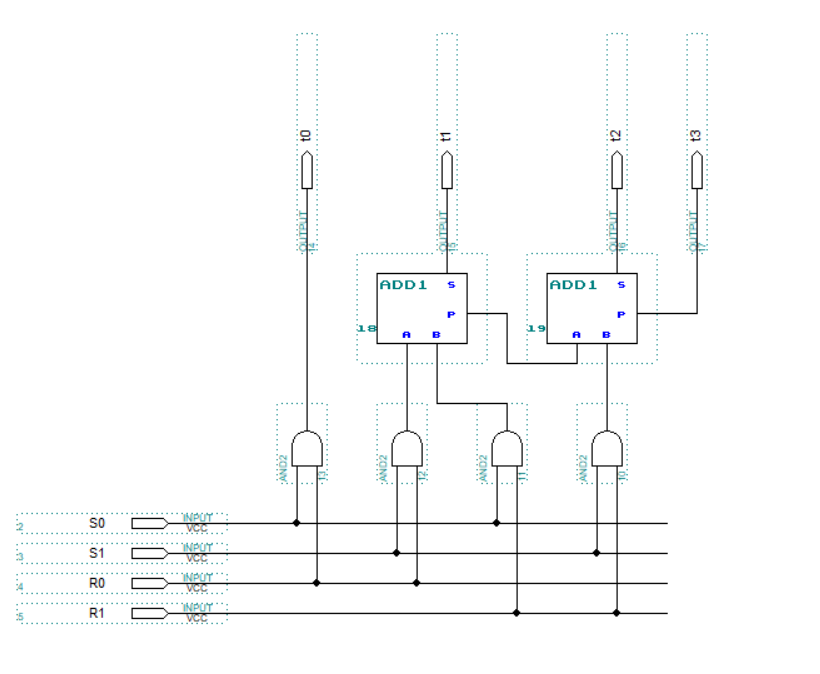
**

Рисунок 3.1 – Схема умножителя двухразрядных чисел

library ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_signed.all;

USE ieee.std\_logic\_arith.all;

entity mult\_2 is

port (s1, s0, r1, r0 : in bit;

t3, t2, t1, t0 : out bit);

end mult\_2;

architecture structure of mult\_2 is

component

add1 port (b1, b2 : in bit;

c1, s1 : out bit);

end component;

signal p1, p2, p3, p4 : bit;

begin

t0 <= r0 and s0;

p2 <= r0 and s1;

p1 <= r1 and s0;

p4 <= r1 and s1;

circ1 : add1 port map (p1, p2, p3, t1);

circ2 : add1 port map (p3, p4, t3, t2);

end structure;

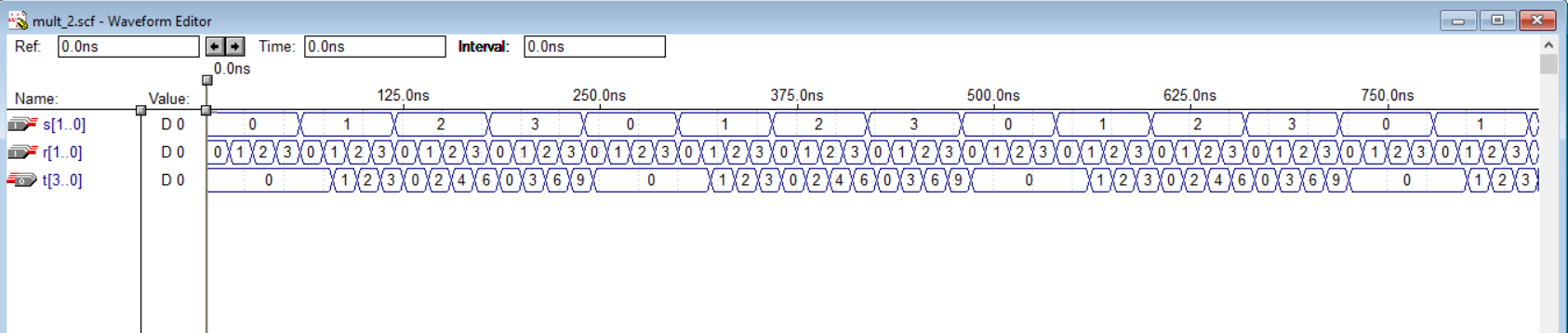


Рисунок 3.2 – Диаграмма состояний умножителя

1. **Арифметическо-логическое устройство.**

Данный код реализует АЛУ:

library ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_signed.all;

entity alu0 is

port (ctrl: in std\_logic\_vector(2 downto 0);

r, s: in std\_logic\_vector (3 downto 0);

c0: in std\_logic;

f\_out: out std\_logic\_vector(3 downto 0);

c4, ovr, z, f3: out std\_logic);

end alu0;

architecture alu of alu0 is

signal tmp: std\_logic\_vector (4 downto 0);

signal ri, si: std\_logic\_vector (4 downto 0);

begin

calc: process

begin

ri(4) <= '0';

si(4) <= '0';

ri(3 downto 0) <= r;

si (3 downto 0) <= s;

if ctrl = "000" then tmp <= ri+si+c0;

elsif ctrl = "001" then tmp <= si-ri-1+c0;

elsif ctrl = "010" then tmp <= ri-si-1+c0;

elsif ctrl = "011" then tmp <= ri or si;

elsif ctrl = "100" then tmp <= ri and si;

elsif ctrl = "101" then tmp <= (not ri) and si;

elsif ctrl = "110" then tmp <= ri xor si;

elsif ctrl = "111" then tmp <= not (ri xor si);

end if;

f\_out <= tmp (3 downto 0);

end process calc;

f3\_p: process(tmp)

begin

f3 <= tmp(3);

end process;

ovr\_p: process(tmp)

begin

ovr <= tmp(3) xor tmp(2);

end process;

c4\_p: process(tmp)

begin

c4 <= tmp(4);

end process;

z\_p: process(tmp)

begin

z <= (not tmp(0) and not tmp(1) and not tmp(2) and not tmp(3));

end process;

end alu;

Данное АЛУ реализует 8 функций, аргументами которых являютя два 4 битных числа R и S, результат заносится в выходы f\_out, а так же в специальные выходы f3 (заносится 3 разряд результата), ovr (заносится сумма по модулю 2 второго и третьего разряда результата), с4 (заносится перенос, если было переполнение), z (принимает значение 1 если все цифры разряда равны 0). На рис. 4.1 представленна диаграмма состояний АЛУ.

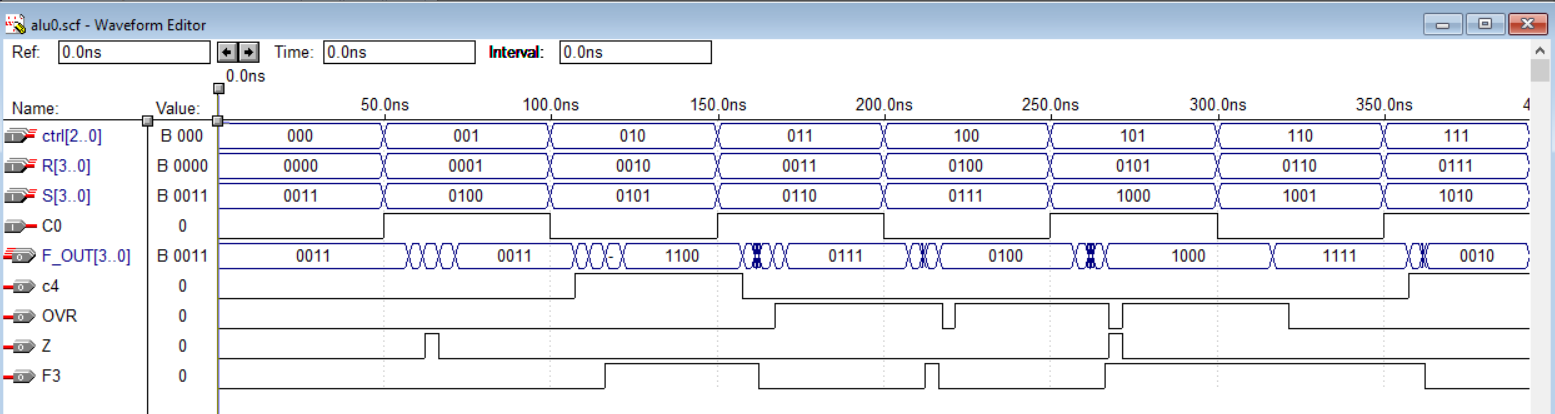


Рис. 4.1 – Диаграмма состояний АЛУ

**4.1 Вычислительное устройство на основе АЛУ и ОЗУ**

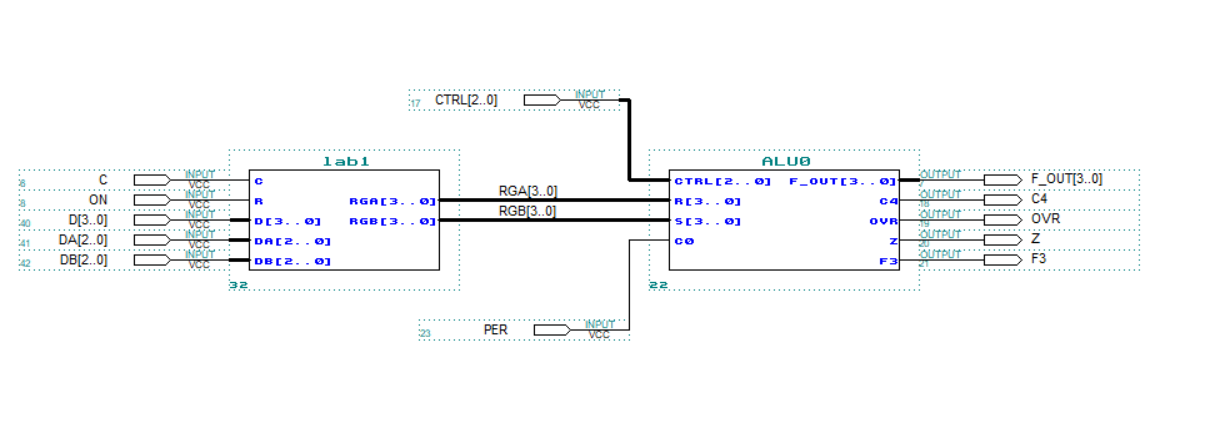
ВУ (рис. 4.2) позволяет производить арифметически-логические действия и сохранять их результат в ОЗУ.

Рис. 4.2 – Вычислительное устройство

На рис. 4.3 представлена диаграмма состояний ВУ.

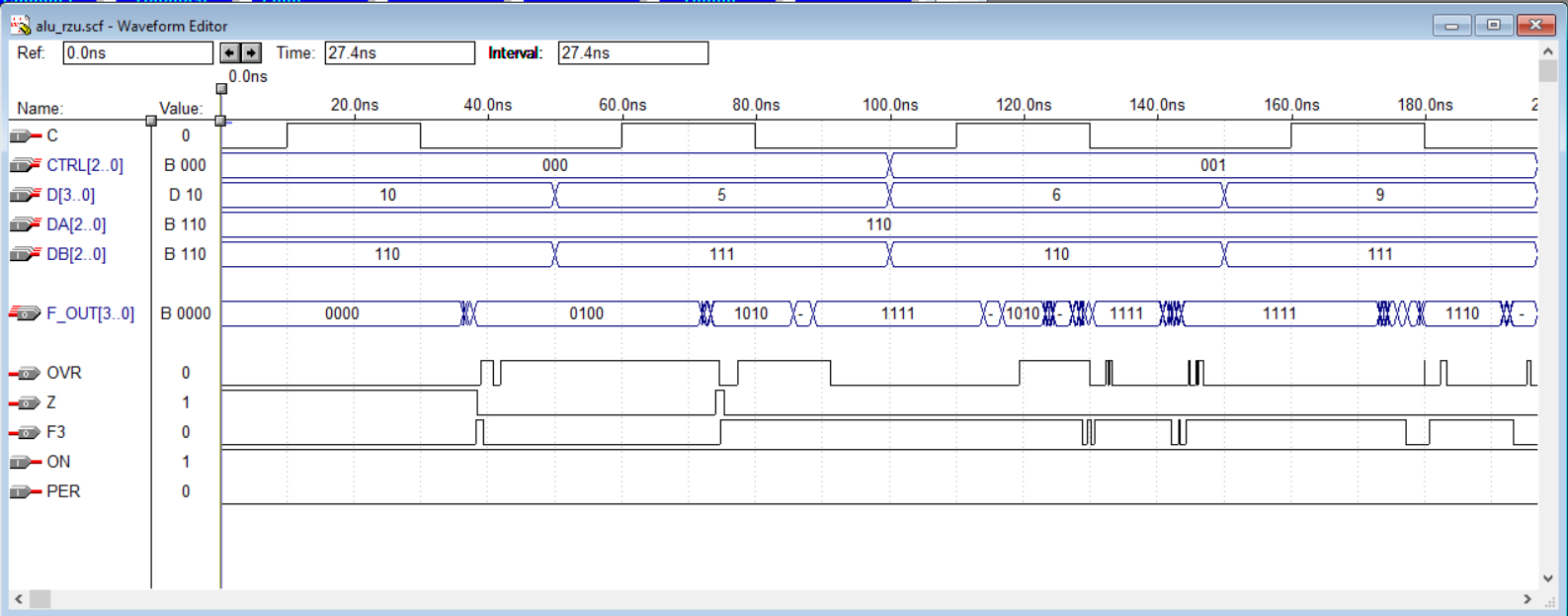


Рис. 4.3 – Диаграмма состояний ВУ