



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет
«СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Отчет
по лабораторной работе №2

Выполнил
Студент ИДМ-24-01
Барышников Егор

Москва 2025

Задание 1

Задание представлено на рисунке 1.

2.1) Задача на сложение двухбитного и однобитного чисел:

$$\begin{array}{r} x1\ x2 \\ + \quad x3 \\ \hline y1\ y2\ y3 \end{array}$$

Рисунок 1

Код модуля представлен на рисунке 2.

```
module main(y1,y2,y3,x1,x2,x3);
    output y1,y2,y3;
    input x1,x2,x3;
    wire z;

    xor(y3,x2,x3);
    and(y1,x1,x2,x3);
    and(z,x2,x3);
    xor(y2,z,x1);

endmodule
```

Рисунок 2

Результаты представлены на рисунках 3-4.

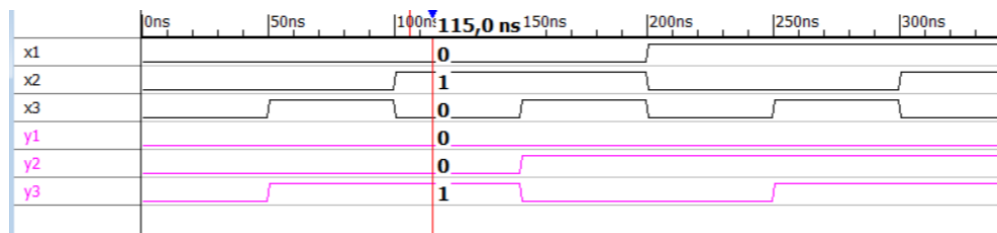


Рисунок 3



Рисунок 4

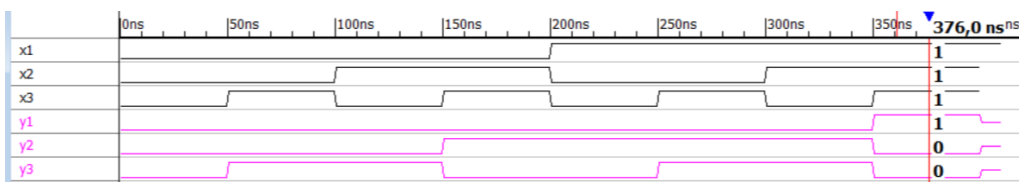


Рисунок 5

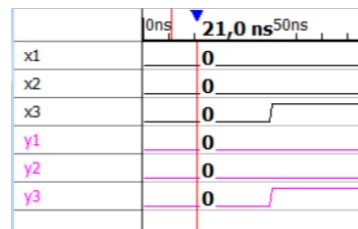


Рисунок 6

Задание 2

Задание представлено на рисунке 7.

2.2) Задача на создание собственного модуля «Импликация»:

$$y = (x_1 \rightarrow x_2) \rightarrow (x_1 \rightarrow x_3), \text{ если } a \rightarrow b = \bar{a} \vee b$$

В главном модуле разрешено только вызывать вспомогательный (можно несколько раз).
Никаких других примитивов/операций производить нельзя!

Рисунок 7

Код модуля представлен на рисунке 8.

```
module impl(y,x1,x2);
    input x1,x2;
    output y;
    wire z;
    not(z,x1);
    or(y,z,x2);
endmodule

module main(y,x1,x2,x3);
    input x1,x2,x3;
    output y;
    wire z1,z2;
    impl xxx(z1,x1,x2);
    impl xxx(z2,x1,x3);
    impl xxx(y,z1,z2);
endmodule
```

Рисунок 8

Результаты представлены на рисунках 9-10.

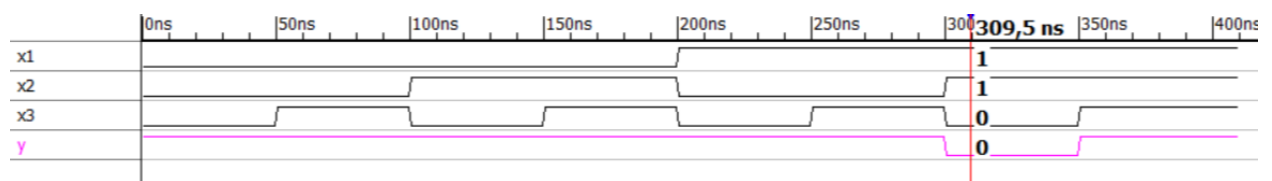


Рисунок 9

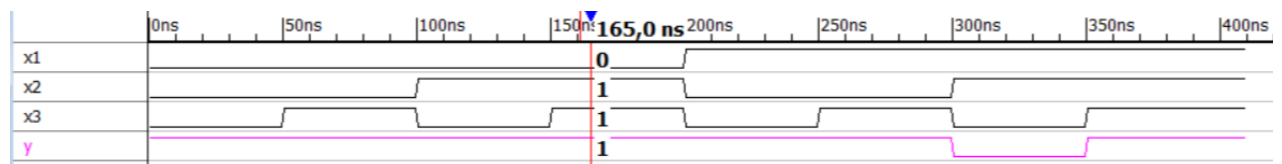


Рисунок 10

Задание 3

Задание представлено на рисунке 11.

2.3) Реализовать функцию без преобразования ее формулы:

Нарисовать логическую схему, обозначить все провода, написать код модуля.
В проекте должно быть строго 2 модуля: вспомогательный и основной.
Основной модуль может содержать только несколько вызовов вспомогательного модуля и никаких других преобразований (надо внимательно изучить формулу и найти закономерность).

Реализовать функцию $y = (x_1 \oplus \overline{x_2} \oplus \overline{x_3}) \oplus \overline{x_1} \oplus \overline{(x_2 \oplus \overline{x_1} \oplus \overline{x_3})}$

Рисунок 11

Код модуля представлен на рисунке 12.

```
module func(y,x1,x2,x3);
    input x1,x2,x3;
    output y;
    wire z1,z2,z3;
    not(z1,x2);
    not(z2,x3);
    xor(z3,z1,z2);
    xor(y,z3,x1);
endmodule

module main(y,x1,x2,x3);
    input x1,x2,x3;
    output y;
    wire z1,z2;
    func xxx(z1,x1,x2,x3);
    func xxx(z2,x2,x1,x3);
    func xxx(y,z1,x1,z2);
endmodule
```

Рисунок 12

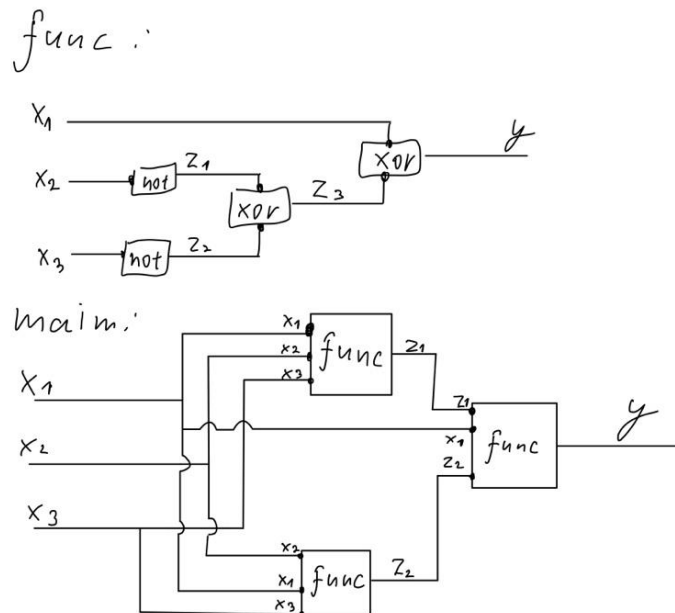


Рисунок 13

Результат представлен на рисунке 14.



Рисунок 14

Задание 5

Задание представлено на рисунке 15.

2.5) Реализовать модуль для поразрядной инверсии 4-битного числа:

Пример: $x_1 x_2 x_3 x_4 \rightarrow y_1 y_2 y_3 y_4$ - 0101 \rightarrow 1010

Рисунок 15

Код модуля представлен на рисунке 16.

```
module inver (y1,y2,y3,y4,x1,x2,x3,x4);  
    output y1,y2,y3,y4;  
    input x1,x2,x3,x4;  
        not (y1,x1);  
        not (y2,x2);  
        not (y3,x3);  
        not (y4,x4);  
endmodule
```

Рисунок 16

Результаты представлены на рисунках 17-18.

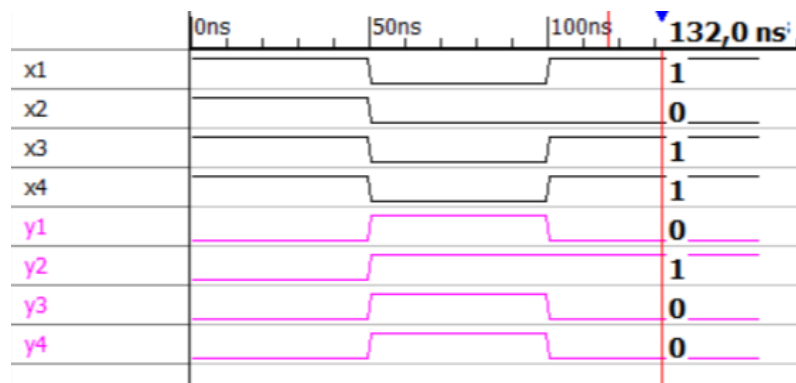


Рисунок 17

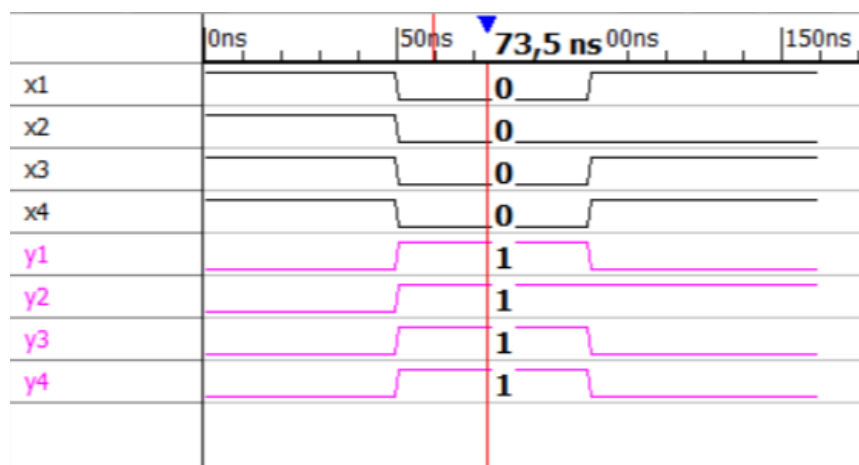


Рисунок 18

Задание 6

Задание представлено на рисунке 19.

2.6) Реализовать модуль для инкремента (увеличения на единицу) 4-битного числа:

Пример: $x_1 x_2 x_3 x_4 \rightarrow y_1 y_2 y_3 y_4$ - $0101 \rightarrow 0110$ (пятый сигнал переполнения не нужен).

Рисунок 19

Код модуля представлен на рисунке 20.

```
module incr(y1,y2,y3,y4,x1,x2,x3,x4);  
    input x1,x2,x3,x4;  
    output y1,y2,y3,y4;  
    wire c1,c2,c3;  
        not(y4,x4);  
  
        xor(y3,x3,x4);  
        and(c3,x3,x4);  
  
        xor(y2,x2,c3);  
        and(c2,x2,c3);  
  
        xor(y1,x1,c2);  
  
endmodule
```

Рисунок 20

Результат представлен на рисунке 21.

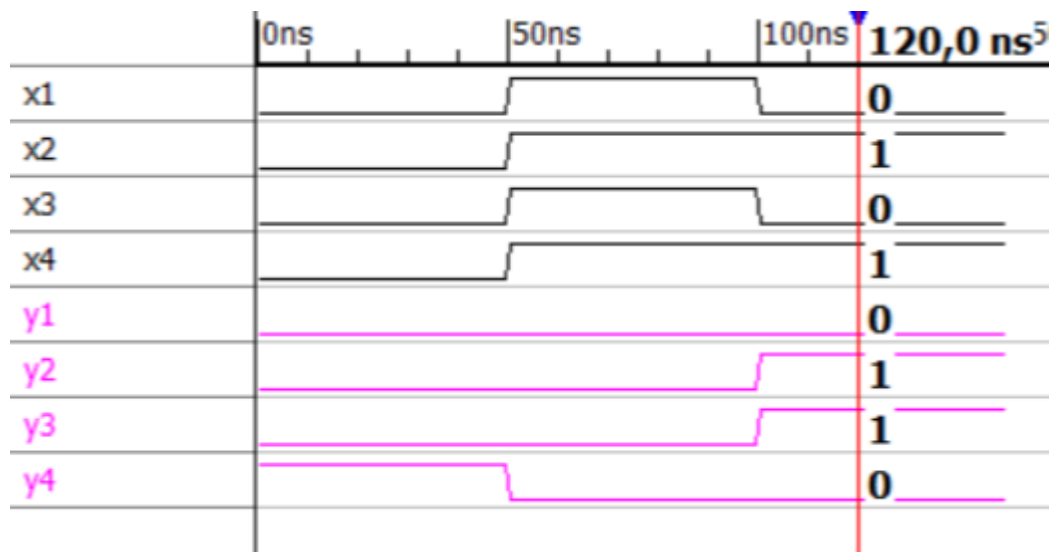


Рисунок 21

Задание 7

Задание представлено на рисунке 22.

2.7) На основе модулей задач 2.5 и 2.6 составить модуль вычитания двух 4-битных чисел:

Пример:

$x_1 x_2 x_3 x_4 - x_5 x_6 x_7 x_8 \rightarrow x_1 x_2 x_3 x_4 + (d_5 d_6 d_7 d_8)$ отрицательное число в дополнительном коде
(дополнительный код = инверсия, затем инкремент)

$x_1 x_2 x_3 x_4 + (d_5 d_6 d_7 d_8) \rightarrow y_1 y_2 y_3 y_4$

$0101 - 0011 \rightarrow 0110 + (-0011) \rightarrow 0110 + (1100) \text{ инверсия} \rightarrow 0110 + (1101) \text{ инкремент} =$

```
  0110
+ 1101
-----
  0011 - ответ
```

Рисунок 22

Код модуля представлен на рисунке 23.

```

module inver(y1,y2,y3,y4,x1,x2,x3,x4);
    output y1,y2,y3,y4;
    input x1,x2,x3,x4;
        not(y1,x1);
        not(y2,x2);
        not(y3,x3);
        not(y4,x4);
endmodule

module incr(y1,y2,y3,y4,x1,x2,x3,x4);
    output y1,y2,y3,y4;
    input x1,x2,x3,x4;
    wire c1,c2,c3;
        not(y4,x4);

        xor(y3,x3,x4);
        and(c3,x3,x4);

        xor(y2,x2,c3);
        and(c2,x2,c3);

        xor(y1,x1,c2);
endmodule

module subsum(y,cn,a,b,c);
    output y,cn;
    input a,b,c;
    wire z1,z2,z3;
        xor(z1,a,b);
        xor(y,c,z1);

        and(z2,a,b);
        and(z3,c,z1);
        or(cn,z2,z3);
endmodule

module sum4(y1,y2,y3,y4,x1,x2,x3,x4,x5,x6,x7,x8);
    output y1,y2,y3,y4;
    input x1,x2,x3,x4,x5,x6,x7,x8;
    wire c1,c2,c3,c0;
        xor(y4,x4,x8);
        and(c3,x4,x8);

        subsum xxx(y3,c2,x3,x7,c3);
        subsum xxx(y2,c1,x2,x5,c2);
        subsum xxx(y1,c0,x1,x6,c1);
endmodule

module main(y1,y2,y3,y4,x1,x2,x3,x4,x5,x6,x7,x8);
    output y1,y2,y3,y4;
    input x1,x2,x3,x4,x5,x6,x7,x8;
    wire d5,d6,d7,d8,i5,i6,i7,i8;
        inver xxx(d5,d6,d7,d8,x5,x6,x7,x8);
        incr xxx(i5,i6,i7,i8,d5,d6,d7,d8);
        sum4 xxx(y1,y2,y3,y4,x1,x2,x3,x4,i5,i6,i7,i8);
endmodule

```

Результат представлен на рисунке 24.

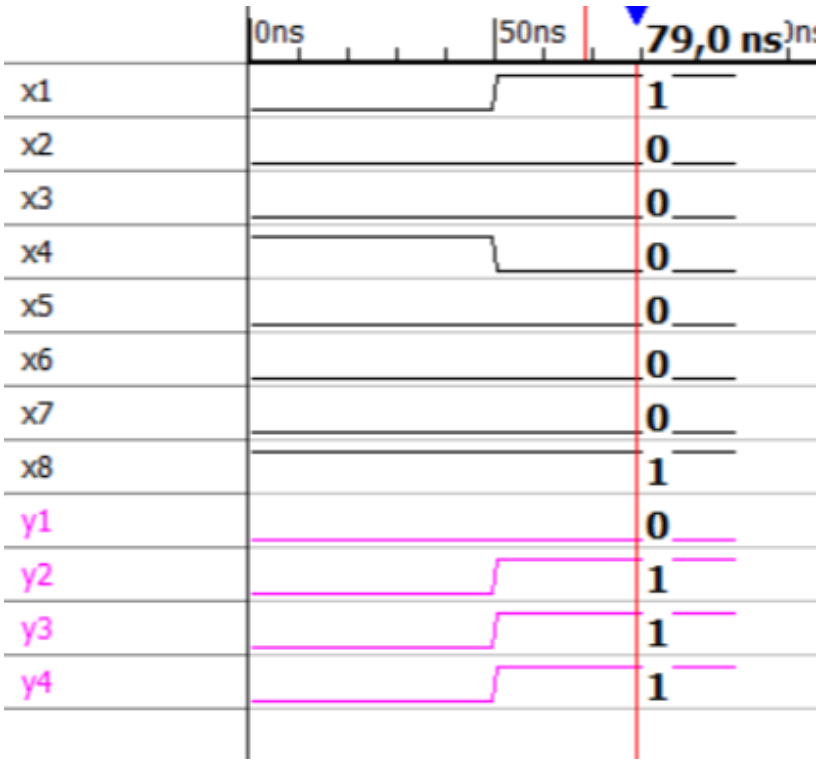


Рисунок 24