Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

‘Функциональная схемотехника’

Вариант №6

*Выполнил:*

Студент группы P3333

Анисимов М. Д.

*Преподаватель:*

Табунщик С. М.



Санкт-Петербург, 2024

# Цели работы:

1. Получить базовые знания о принципах построения цифровых интегральных схем с использованием технологии КМОП.

2. Познакомиться с технологией SPICE-моделирования схем на транзисторах.

3. Получить навыки описания схем базовых операционных элементов (БОЭ) комбинационного типа на вентильном уровне с использованием языка описания аппаратуры Verilog HDL.

# Задание:

Логический базис – NAND; БОЭ – позиционный дешифратор «3 в 8».

# Часть №1. LTSpice.

## **Разработка вентиля.**

Схема разработанного вентиля:

VIN{1, 2} – входы, VOUT - выход, VDD – напряжение питания; использовано по 2 транзистора PMOS и NMOS.

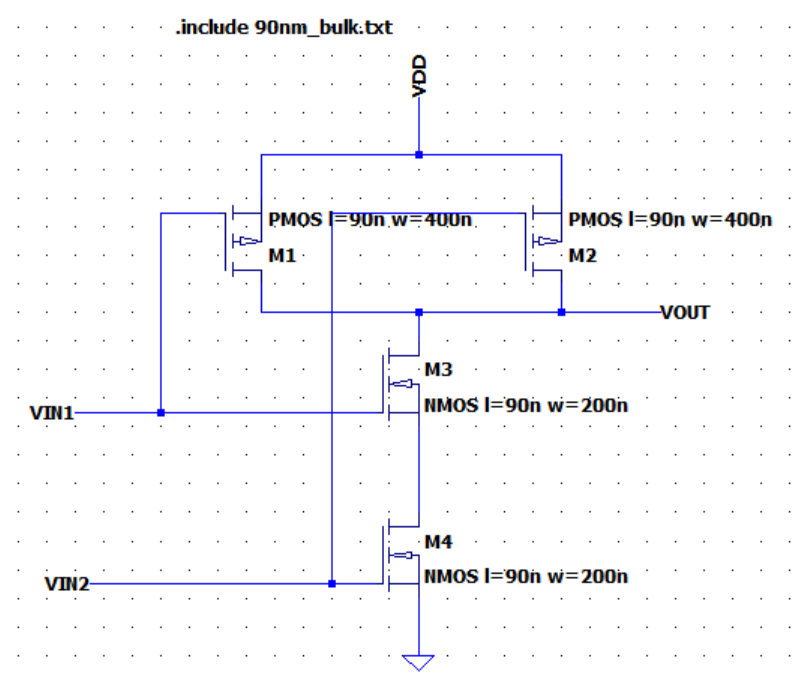


Рисунок 1. Схема NAND

Символ вентиля:

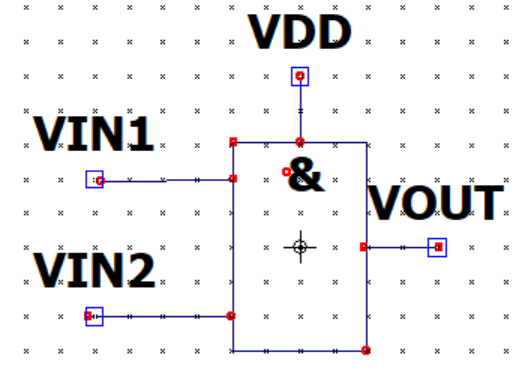


Рисунок 2. Символ NAND элемента

Схема тестирования:

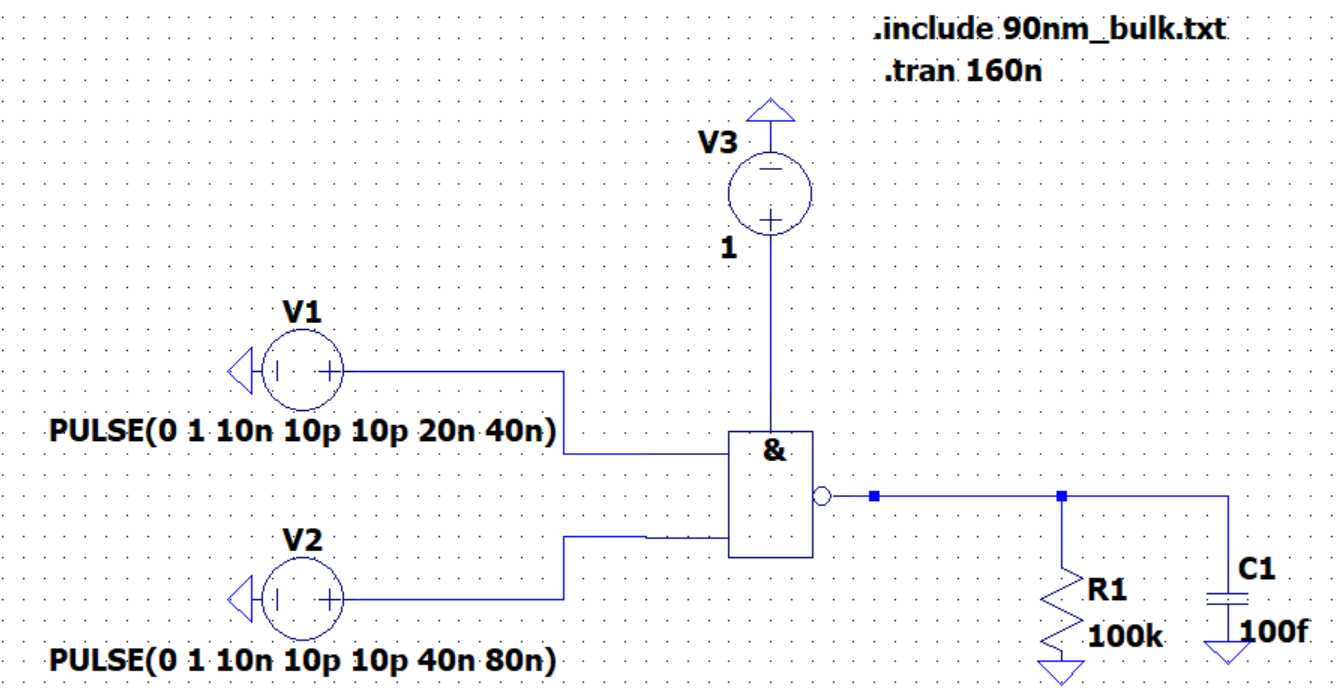


Рисунок 3. Схема тестирования

Временная диаграмма процесса тестирования вентиля:

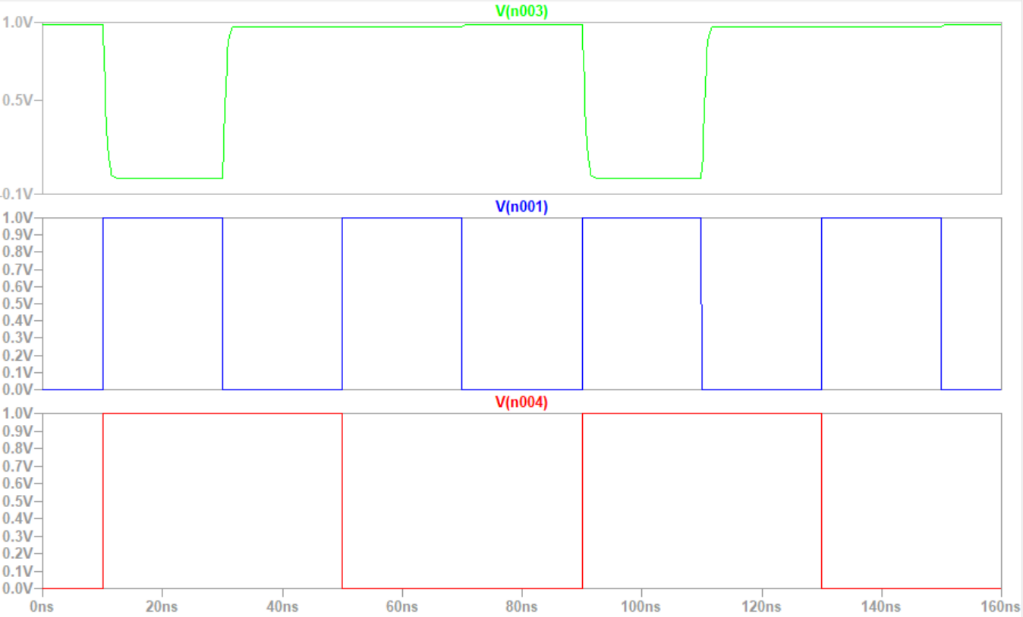


Рисунок 4. Временная диаграмма работы NOR элемента

1 В – логическая единица, 0 В – логический ноль; NAND подаёт на выход сигнал 1 во всех случаях, кроме случая, где на два входа подаётся 1. На графике красным обозначен выход NAND элемента. В период с 10 нс до 30 нс при двух входных елиничных сигналах напряжение спало до 0В, а затем резко возросло на 1 В при смене одного из входов на ноль

Результат измерения задержки распространения сигнала через вентиль:

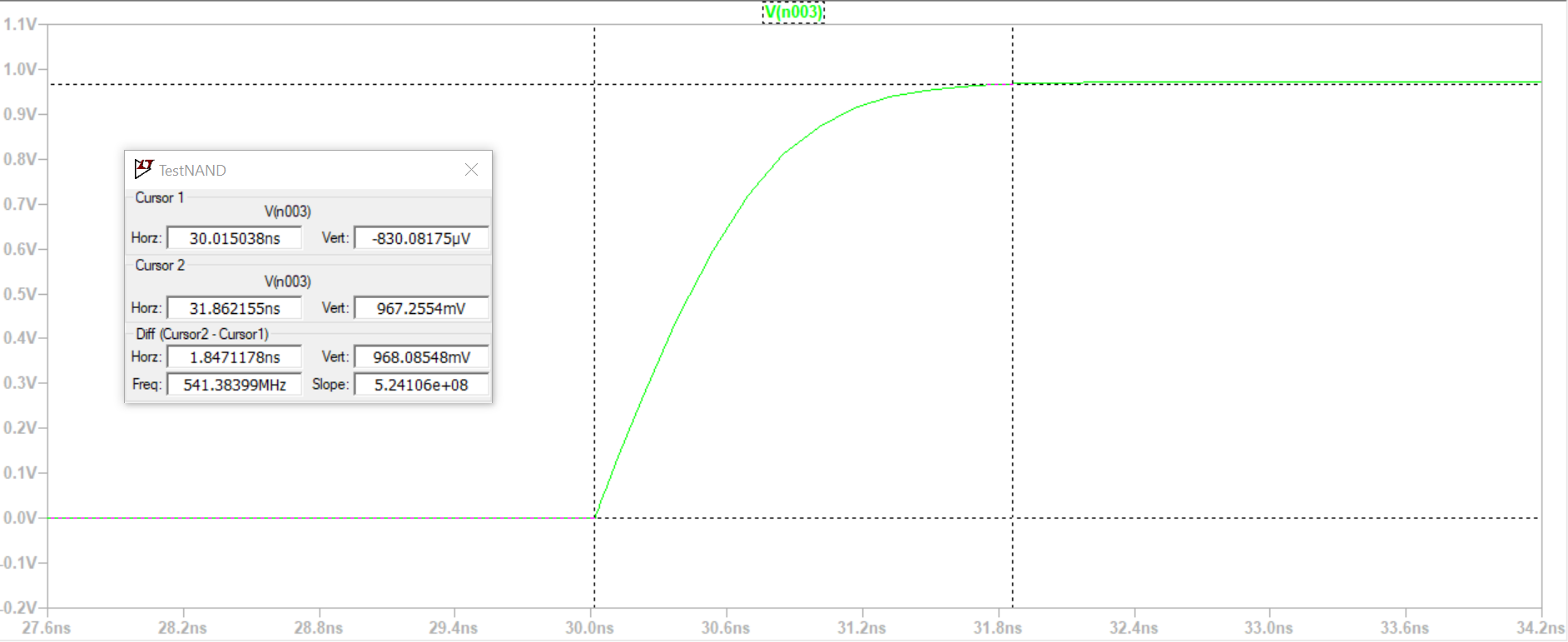


Рисунок 5. Задержка фронта

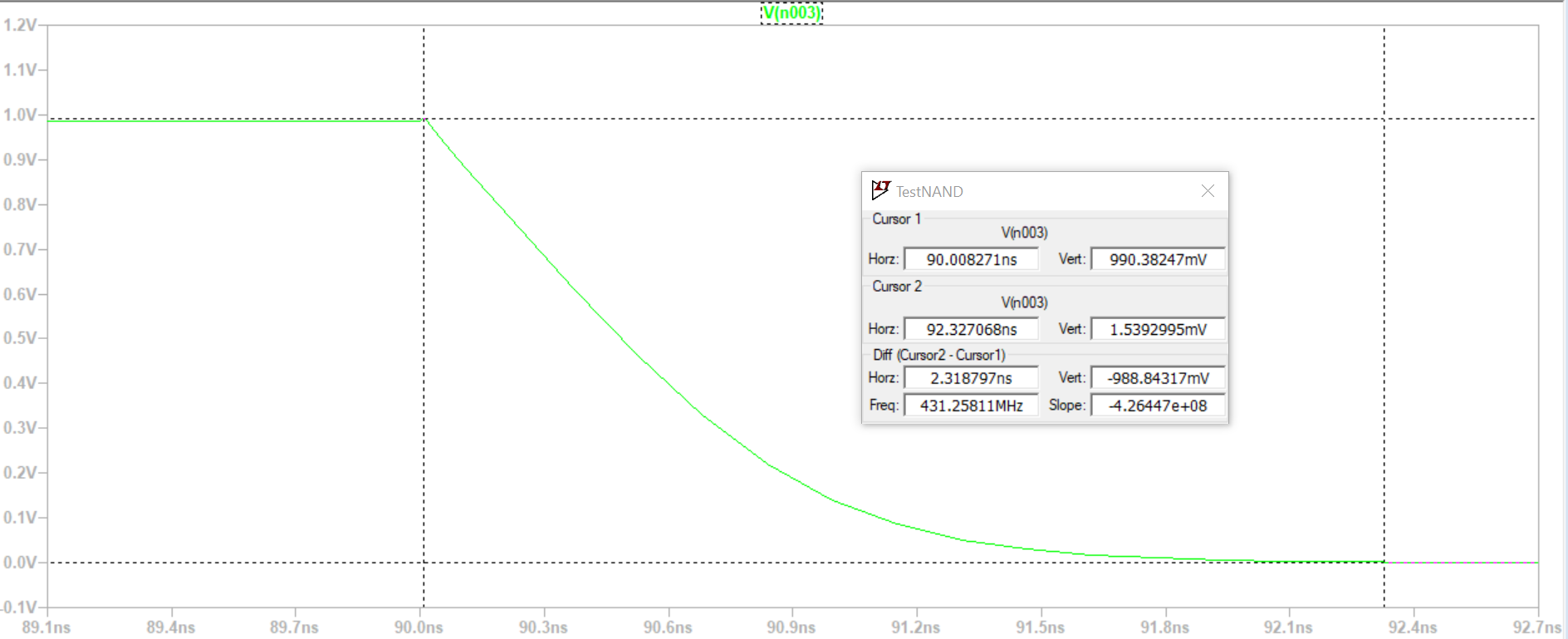


Рисунок 6. Задержка спада

Задержка фронта равна T1 ~ 1.84 нс

Задержка спада равна T2 ~ 2.31 нс

Тогда максимальная частота работы вентиля равна f = 1/(T1+Т2) = 240 МГц

## **Разработка БОЭ.**

Для разработки дешифратора «3 в 8» так же был построен инвертор и элемент NAND с 4 входами (3 для входных данных, 1 для сигнала разрешения). Инвертор нужен для результирующего инвертирования, так как в стандартной реализации дешифратора «3-8» используются элементы AND, поэтому для получения корректного результата необходимо инвертировать выход элементов NAND

Схема инвертора.

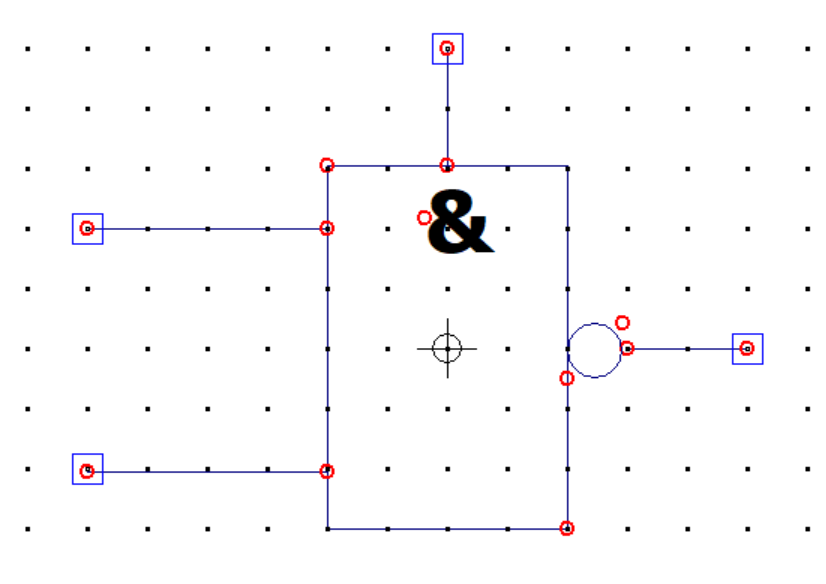


Рисунок 7 – схема инвертора

Символ инвертора

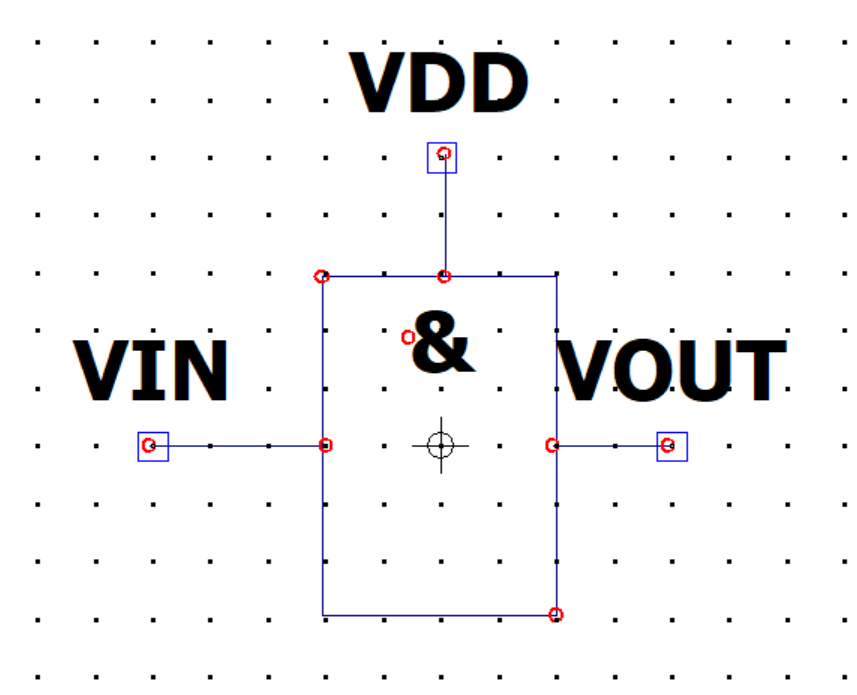


Рисунок 8 – символ инвертора

Схема NAND с 4 входами.

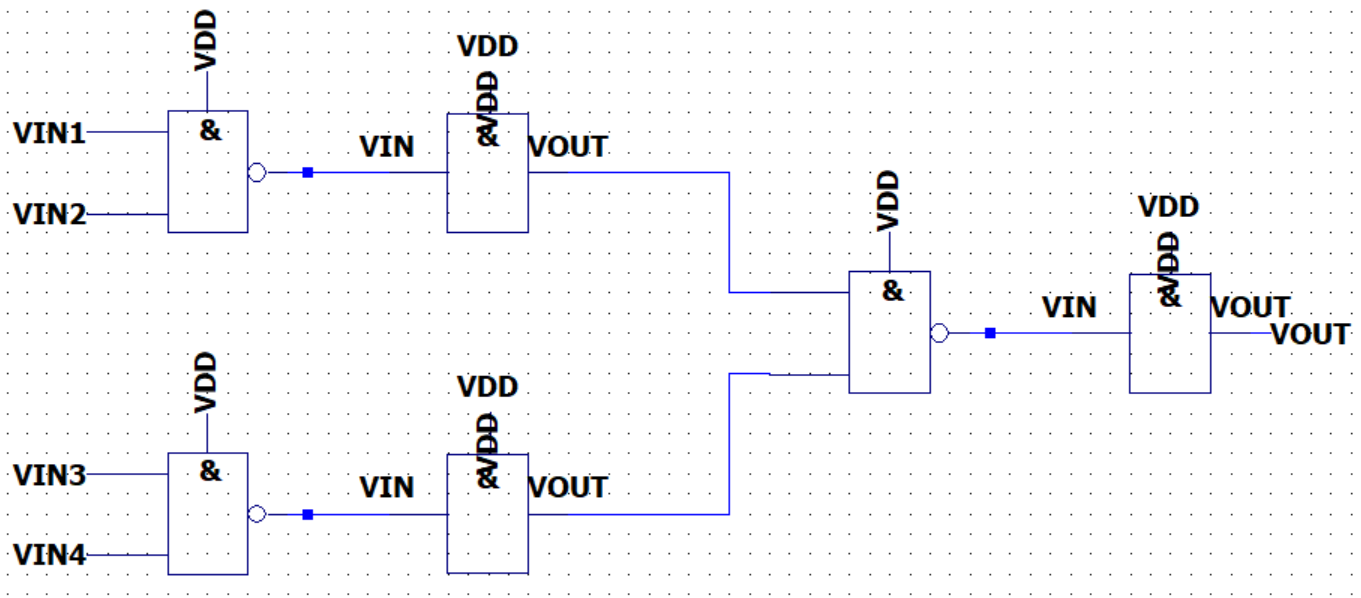


Рисунок 9 – схема NAND с 4 входами

Символ NAND с 4 входами

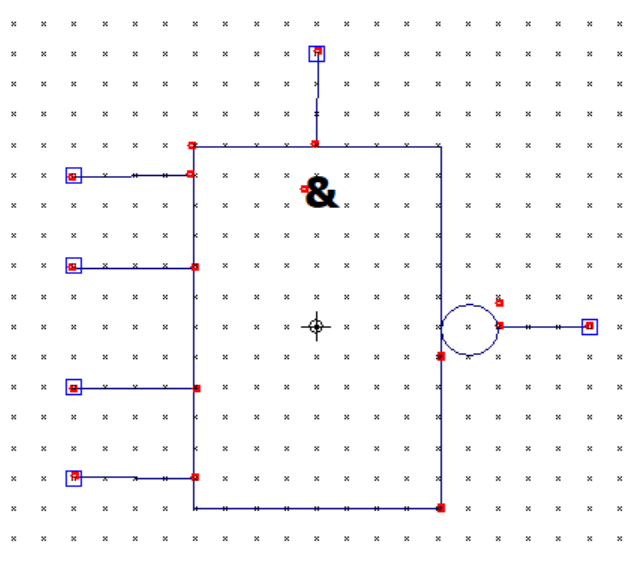
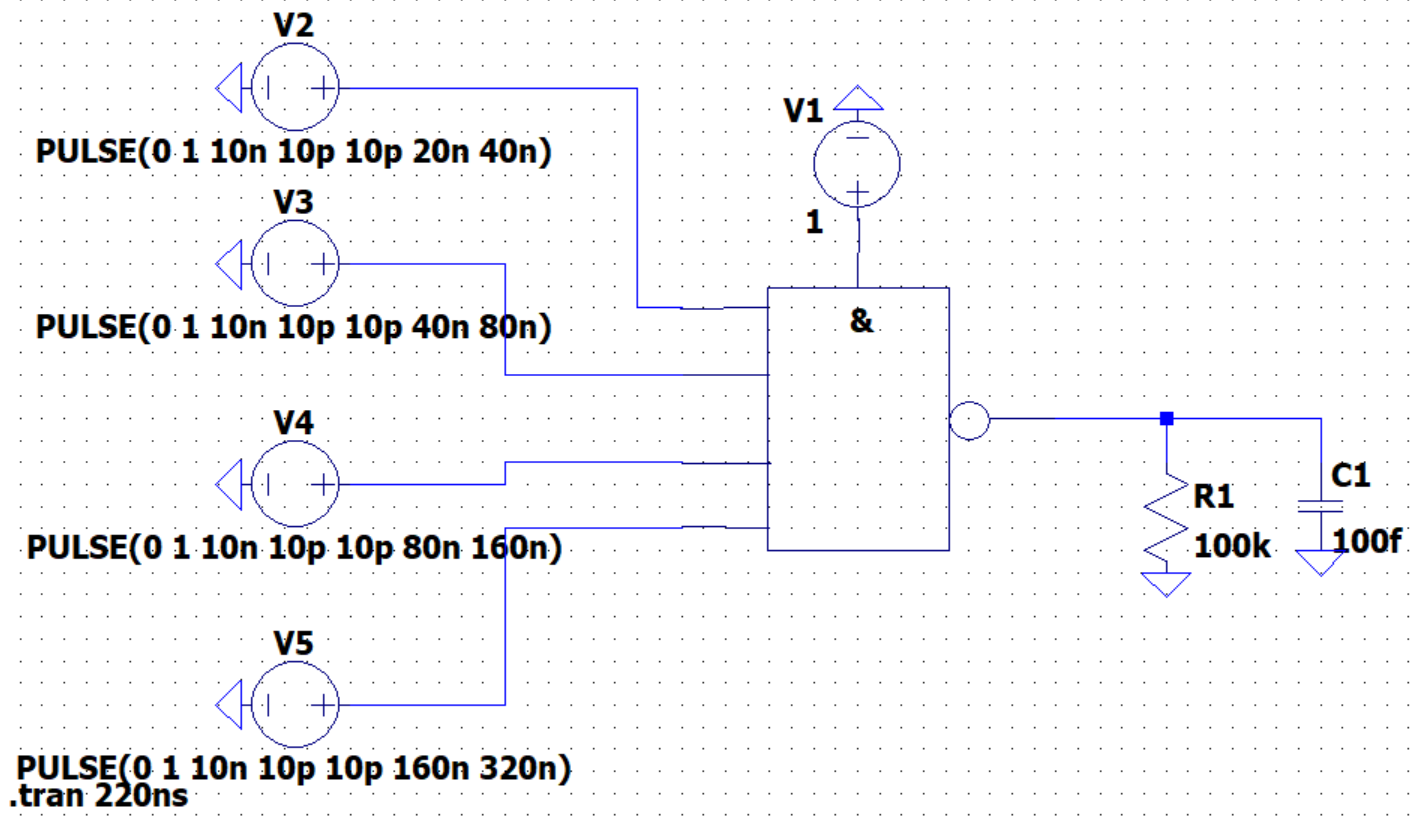


Рисунок 10 – символ NAND с 4 входами

Схема тестирования элемента 4NAND:



Временная диаграмма процесса тестирования вентиля:

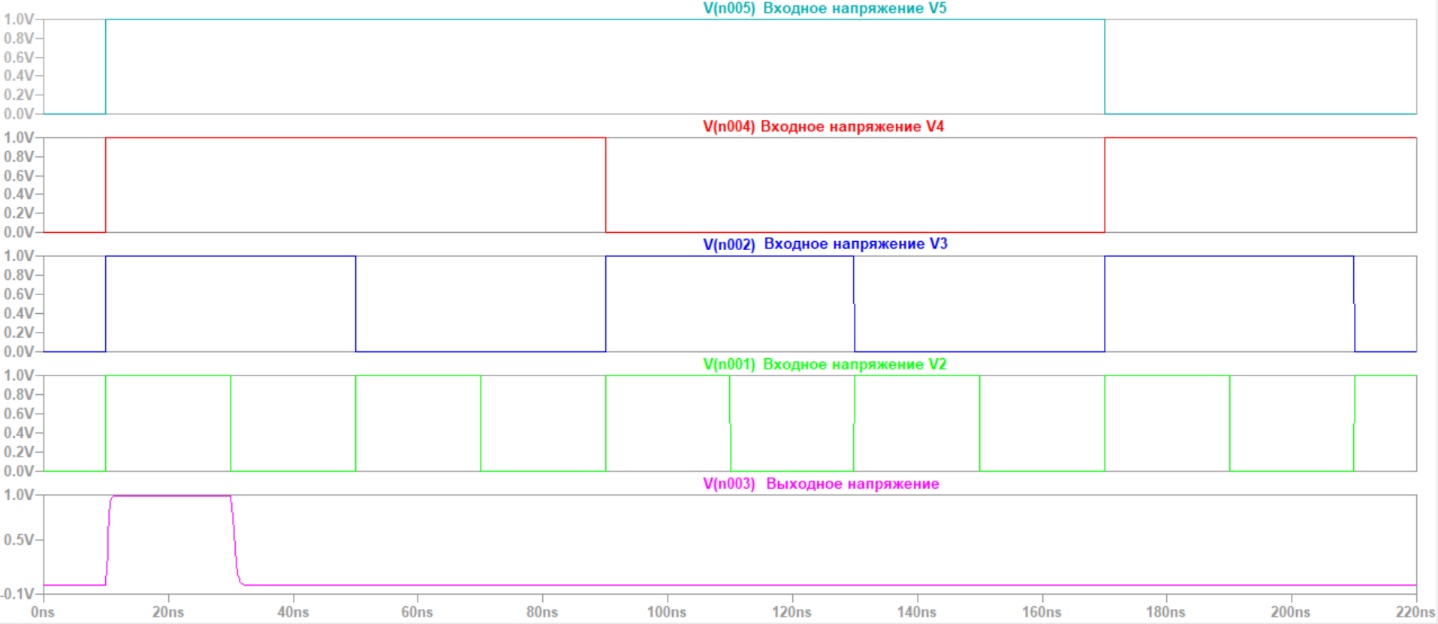


Рисунок 11 – временная диаграмма тестирования NAND с 4 входами

Схема разработанного БОЭ:

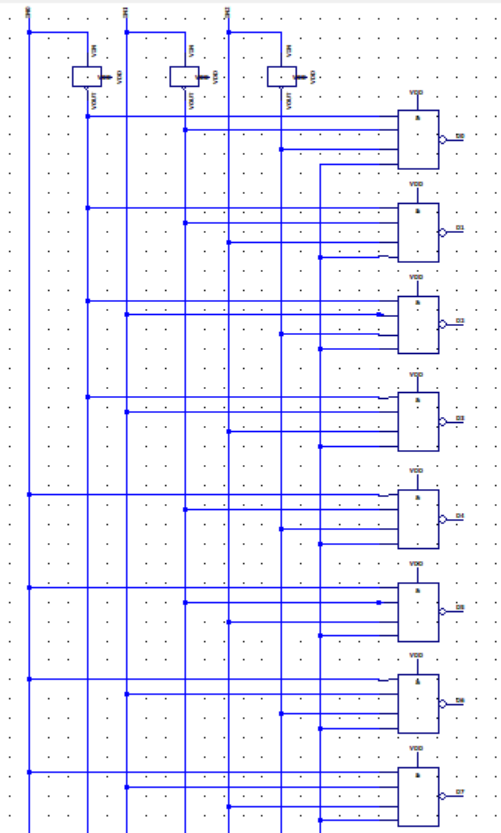


Рисунок 12. Схема БОЭ

Символ разработанного БОЭ:

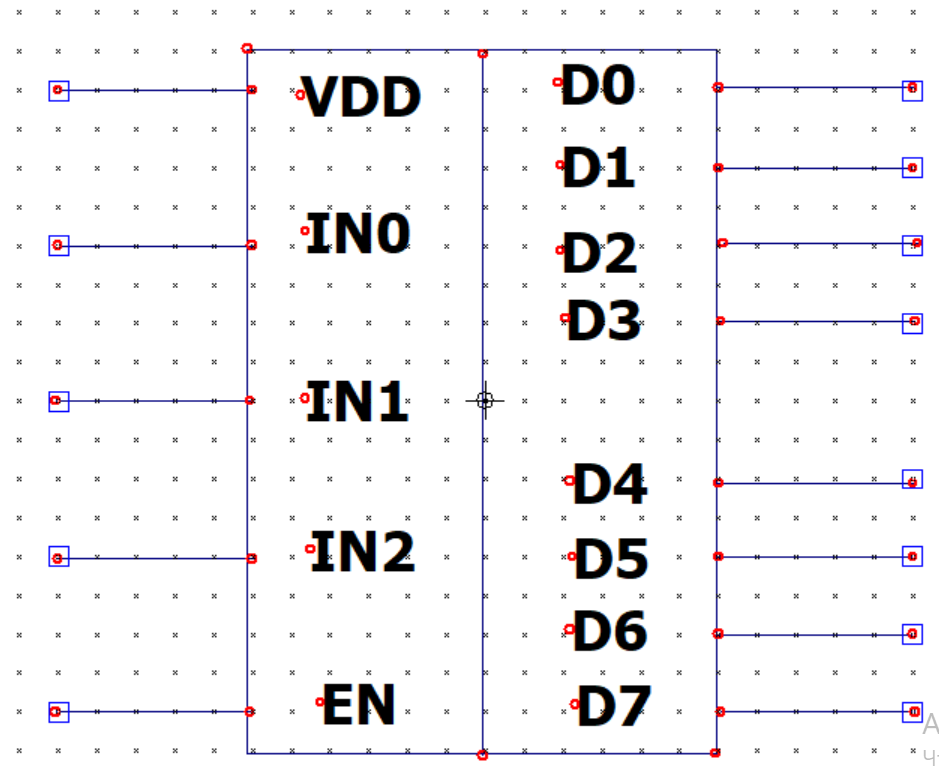


Рисунок 14 – символ БОЭ

Схема тестирования:

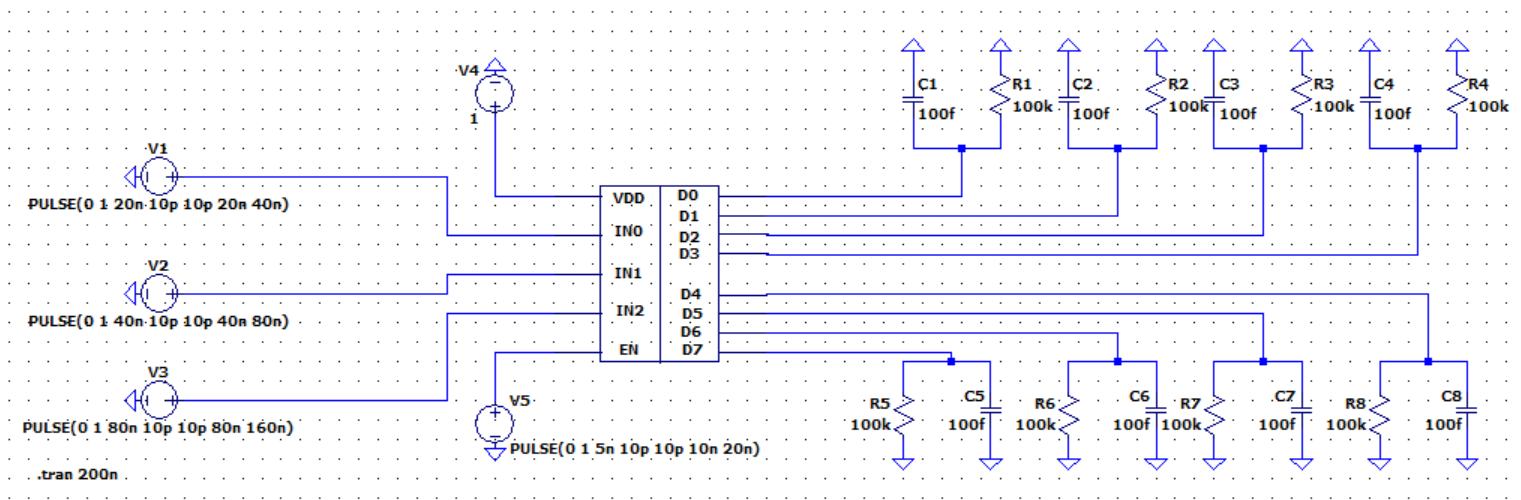
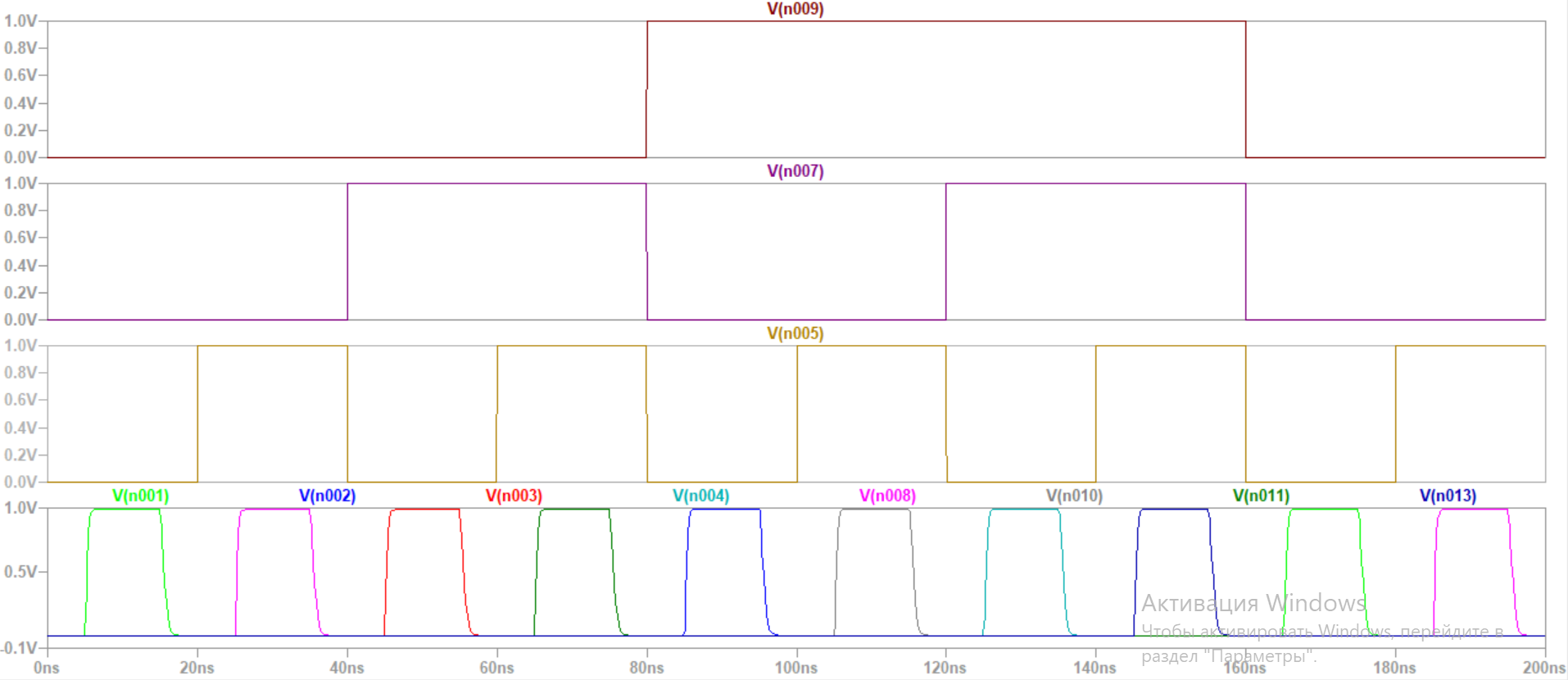


Рисунок 15 – схема тестирования БОЭ

Временная диаграмма процесса тестирования БОЭ:



Выходные напряжения

Вход V1

Вход V1

Вход V2

Рисунок 16 – временная диаграмма

Результат измерения задержки распространения сигнала через БОЭ:

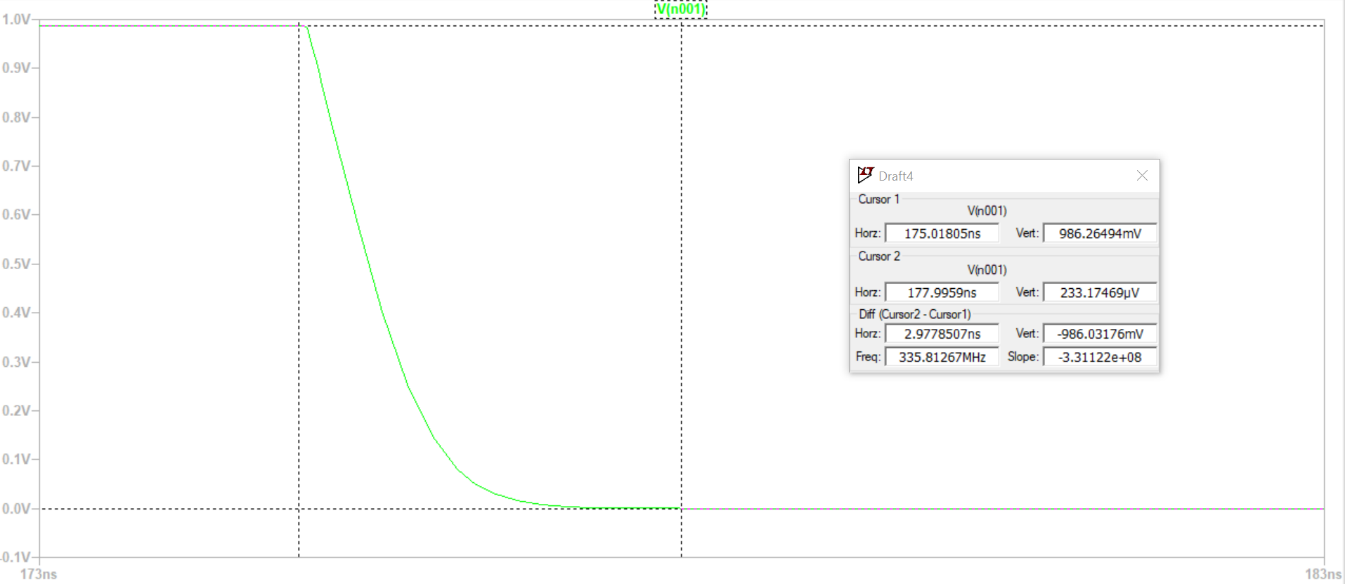


Рисунок 17 - Задержка спада

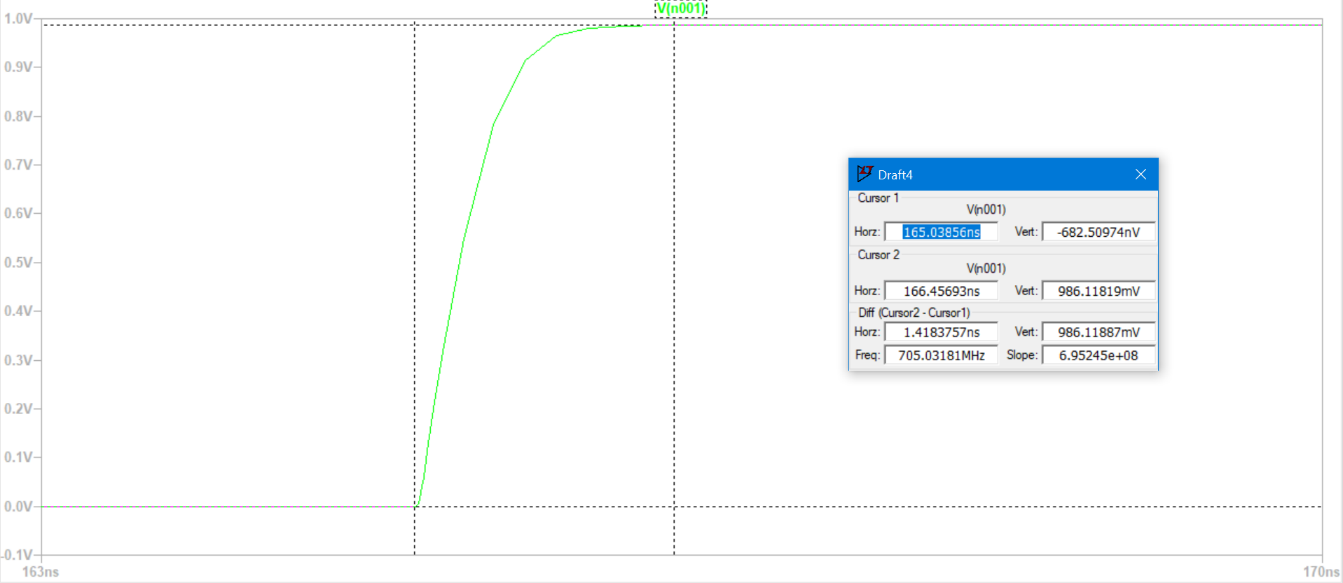


Рисунок 18 - Задержка фронта

Два курсора на верхней (~ 1 В) и нижней (~ 0 В) границах заднего фронта.

Задержка спада равна T1 ~ 2,97 нс

Задержка фронта равна T2 ~ 1,41 нс

Тогда максимальная частота работы БОЭ равна f = 1/(T1+Т2) = 228 МГц

# Часть № 2. Verilog.

Код разработанного модуля БОЭ:

`timescale 1ns/1ps

module decoder(

input [2:0] in,

input en,

output [7:0] d

);

wire [2:0] not\_in;

wire [7:0] not\_d;

wire [7:0] in\_2\_1;

wire [7:0] not\_in\_2\_1;

wire [7:0] in\_0\_en;

wire [7:0] not\_in\_0\_en;

nand(not\_in[2], in[2], in[2]);

nand(not\_in[1], in[1], in[1]);

nand(not\_in[0], in[0], in[0]);

nand(not\_in\_2\_1[0], not\_in[2], not\_in[1]);

nand(not\_in\_0\_en[0], not\_in[0], en);

nand(in\_2\_1[0], not\_in\_2\_1[0], not\_in\_2\_1[0]);

nand(in\_0\_en[0], not\_in\_0\_en[0], not\_in\_0\_en[0]);

nand(not\_d[0], in\_2\_1[0], in\_0\_en[0]);

nand(not\_in\_2\_1[1], not\_in[2], not\_in[1]);

nand(not\_in\_0\_en[1], in[0], en);

nand(in\_2\_1[1], not\_in\_2\_1[1], not\_in\_2\_1[1]);

nand(in\_0\_en[1], not\_in\_0\_en[1], not\_in\_0\_en[1]);

nand(not\_d[1], in\_2\_1[1], in\_0\_en[1]);

nand(not\_in\_2\_1[2], not\_in[2], in[1]);

nand(not\_in\_0\_en[2], not\_in[0], en);

nand(in\_2\_1[2], not\_in\_2\_1[2], not\_in\_2\_1[2]);

nand(in\_0\_en[2], not\_in\_0\_en[2], not\_in\_0\_en[2]);

nand(not\_d[2], in\_2\_1[2], in\_0\_en[2]);

nand(not\_in\_2\_1[3], not\_in[2], in[1]);

nand(not\_in\_0\_en[3], in[0], en);

nand(in\_2\_1[3], not\_in\_2\_1[3], not\_in\_2\_1[3]);

nand(in\_0\_en[3], not\_in\_0\_en[3], not\_in\_0\_en[3]);

nand(not\_d[3], in\_2\_1[3], in\_0\_en[3]);

nand(not\_in\_2\_1[4], in[2], not\_in[1]);

nand(not\_in\_0\_en[4], not\_in[0], en);

nand(in\_2\_1[4], not\_in\_2\_1[4], not\_in\_2\_1[4]);

nand(in\_0\_en[4], not\_in\_0\_en[4], not\_in\_0\_en[4]);

nand(not\_d[4], in\_2\_1[4], in\_0\_en[4]);

nand(not\_in\_2\_1[5], in[2], not\_in[1]);

nand(not\_in\_0\_en[5], in[0], en);

nand(in\_2\_1[5], not\_in\_2\_1[5], not\_in\_2\_1[5]);

nand(in\_0\_en[5], not\_in\_0\_en[5], not\_in\_0\_en[5]);

nand(not\_d[5], in\_2\_1[5], in\_0\_en[5]);

nand(not\_in\_2\_1[5], in[2], not\_in[1]);

nand(not\_in\_0\_en[5], in[0], en);

nand(in\_2\_1[5], not\_in\_2\_1[5], not\_in\_2\_1[5]);

nand(in\_0\_en[5], not\_in\_0\_en[5], not\_in\_0\_en[5]);

nand(not\_d[5], in\_2\_1[5], in\_0\_en[5]);

nand(not\_in\_2\_1[6], in[2], in[1]);

nand(not\_in\_0\_en[6], not\_in[0], en);

nand(in\_2\_1[6], not\_in\_2\_1[6], not\_in\_2\_1[6]);

nand(in\_0\_en[6], not\_in\_0\_en[6], not\_in\_0\_en[6]);

nand(not\_d[6], in\_2\_1[6], in\_0\_en[6]);

nand(not\_in\_2\_1[7], in[2], in[1]);

nand(not\_in\_0\_en[7], in[0], en);

nand(in\_2\_1[7], not\_in\_2\_1[7], not\_in\_2\_1[7]);

nand(in\_0\_en[7], not\_in\_0\_en[7], not\_in\_0\_en[7]);

nand(not\_d[7], in\_2\_1[7], in\_0\_en[7]);

nand(d[0], not\_d[0], not\_d[0]);

nand(d[1], not\_d[1], not\_d[1]);

nand(d[2], not\_d[2], not\_d[2]);

nand(d[3], not\_d[3], not\_d[3]);

nand(d[4], not\_d[4], not\_d[4]);

nand(d[5], not\_d[5], not\_d[5]);

nand(d[6], not\_d[6], not\_d[6]);

nand(d[7], not\_d[7], not\_d[7]);

endmodule

Вход: 3-битная шина s с начальным числом и сигнал разрешения.

Выход: 8-битная шина d.

not\_s, not\_d – инвертированные биты шин s и d.

s\_2\_1[n] - инвертированные биты результата s[2] NAND

not\_s\_2\_1[n] – не инвертированные биты результата not\_s[2] NAND not\_s[1] (т.е. выполнение NAND для старшего и среднего битов).

s\_0\_en[n] - инвертированные биты результата s[0] NAND

not\_s\_0\_en[n] - не инвертированные биты результата not\_s[0] NAND not\_en (т.е. выполнение NAND для младшего бита и сигнала разрешения).

Код разработанного тестового окружения БОЭ:

`timescale 1ns / 1ps

module decoder\_tb;

reg [2:0] in;

wire [7:0] d;

reg en;

integer i;

decoder decoder\_1(

.in(in),

.d(d),

.en(en)

);

initial begin

for(i=0; i<8; i=i+1) begin

in = i;

en = 1;

#10

if(d == 2\*\*i) begin

$display("Task completed successfully! in=%b, d=%b, en=%b, i=%0d", in, d, en, i);

end else begin

$display("Task failed! in=%b, d=%b, en=%b, i=%0d", in, d, en, i);

end

en = 0;

#10

if(d == 0) begin

$display("Task completed successfully! in=%b, d=%b, en=%b, i=%0d", in, d, en, i);

end else begin

$display("Task failed! in=%b, d=%b, en=%b, i=%0d", in, d, en, i);

end

end

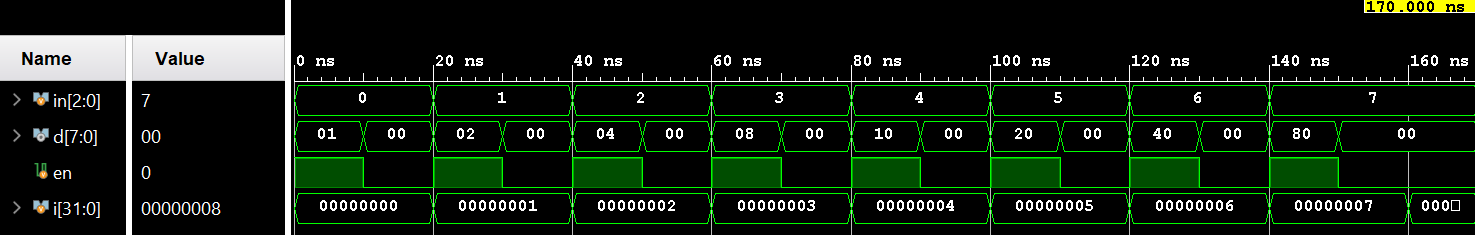
#10 $stop;

end

endmodule

В тестовом окружении задаются переменные входа (3-битное число и сигнал разрешения) и выхода (8-битное число). целочисленная переменная i будет изменяться в цикле от 0 до 8 и её значение будет присваиваться шине s. Затем проверяется работоспособность дешифратора при включенном и выключенном сигнале разрешения

Временная диаграмма процесса тестирования БОЭ:



Вывод в консоль:

Task completed successfully! in=000, d=00000001, en=1, i=0

Task completed successfully! in=000, d=00000000, en=0, i=0

Task completed successfully! in=001, d=00000010, en=1, i=1

Task completed successfully! in=001, d=00000000, en=0, i=1

Task completed successfully! in=010, d=00000100, en=1, i=2

Task completed successfully! in=010, d=00000000, en=0, i=2

Task completed successfully! in=011, d=00001000, en=1, i=3

Task completed successfully! in=011, d=00000000, en=0, i=3

Task completed successfully! in=100, d=00010000, en=1, i=4

Task completed successfully! in=100, d=00000000, en=0, i=4

Task completed successfully! in=101, d=00100000, en=1, i=5

Task completed successfully! in=101, d=00000000, en=0, i=5

Task completed successfully! in=110, d=01000000, en=1, i=6

Task completed successfully! in=110, d=00000000, en=0, i=6

Task completed successfully! in=111, d=10000000, en=1, i=7

Task completed successfully! in=111, d=00000000, en=0, i=7

# Выводы по работе:

В процессе выполнения данной работы я познакомился со средой Ltspice, Vivado и языком Verilog. В качестве опытного образца я создал собственный вентиль NAND и на его основе создал и протестировал позиционный шифратор «3 в 8».