

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ и РАДИОЭЛЕКТРОНИКИ**

Факультет компьютерных систем и сетей

Кафедра ЭВМ

**Контроль и диагностика средств
вычислительной техники**

Лабораторная работа № 2

**Тестирование элементов памяти на уровне поведения.
Тесты КМОП-структур на переключательном уровне
Вариант №7**

**Выполнил:
Студент гр.350501
Пашковский А.А.**

**Проверил:
Золоторевич Л.А.**

Минск 2016

Триггер: D-триггер со сбросом и установкой.

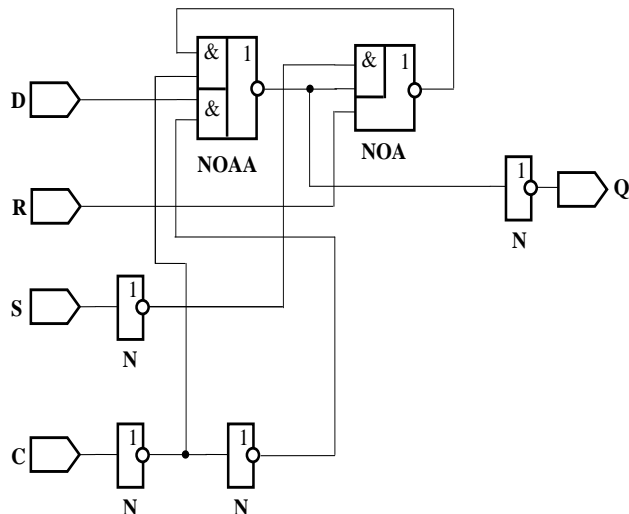


Таблица функционирования
D-триггера со сбросом и установкой

| R | S | D | C | Q |
|---|---|---|---|-----|
| 1 | - | - | 0 | 0 |
| 0 | 1 | - | 0 | 1 |
| 0 | 0 | - | 1 | (D) |
| 0 | 0 | - | 0 | N |

Функциональный тест построенный на базе таблицы функционирования:

| R | S | D | C | Q | Операция |
|---|---|---|---|---|-------------|
| 1 | 0 | 0 | 0 | 0 | Сброс в 0 |
| 0 | 0 | 1 | 1 | 1 | Установка 1 |
| 0 | 0 | 1 | 0 | 1 | Хранение 1 |
| 0 | 0 | 0 | 1 | 0 | Установка 0 |
| 0 | 0 | 0 | 0 | 0 | Хранение 0 |
| 0 | 0 | 1 | 1 | 1 | Установка 1 |
| 1 | 0 | 1 | 0 | 0 | Сброс в 0 |
| 0 | 0 | 0 | 1 | 0 | Установка 0 |
| 0 | 1 | 0 | 0 | 1 | Установка 1 |

Код схемы на VLSI-SIM:

```
circuit trigger;
inputs R(1), S(1), D(1), C(1);
outputs Q(1);
GATES
NC 'NOT'(1) C(1);
NNC 'NOT'(1) NC(1);
NS 'NOT'(1) S(1);
NOA22_R 'NOA22'(1) NOA2_R(1),NC(1),D(1),NNC(1);
NOA2_R 'NOA2'(1) R(1),NOA22_R(1),NS(1);
Q 'NOT'(1) NOA22_R(1);
ENDGATES
END
```

Определение контролирующей способности функционального теста:

Генерация теста

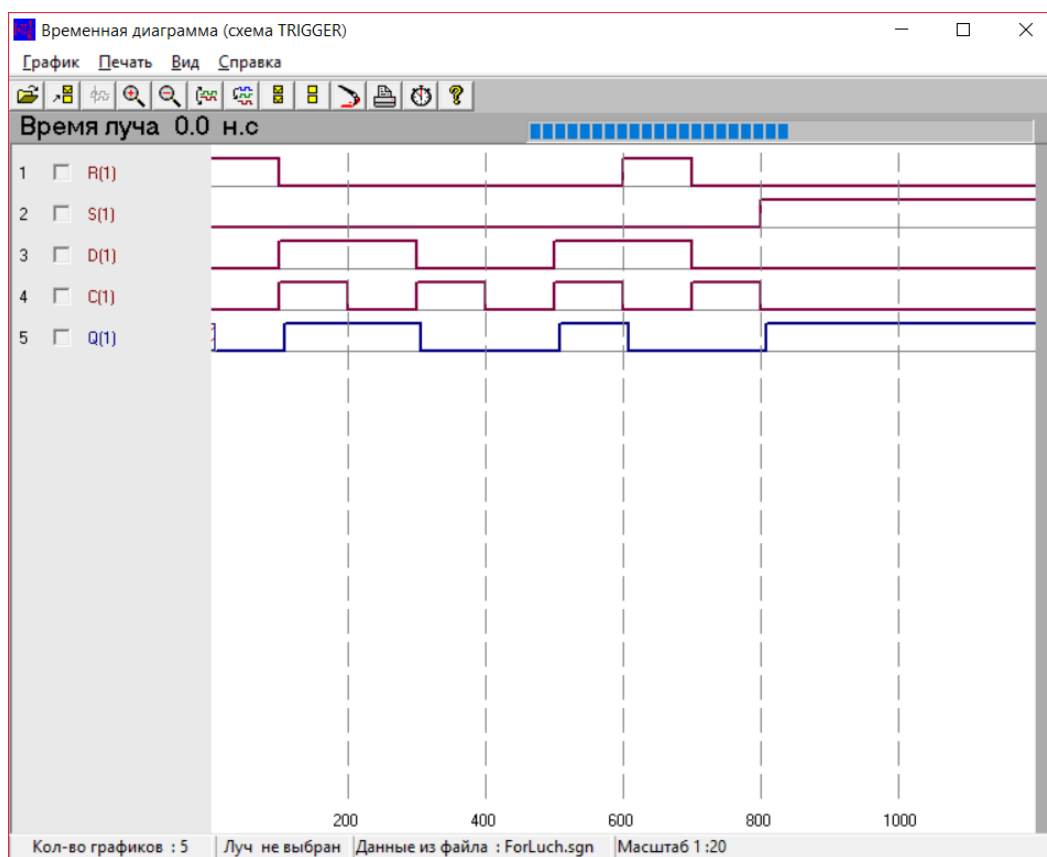
| | |
|----------------------------|--------|
| Требуемая полнота теста: | 100.00 |
| Достигнутая полнота теста: | 100.00 |
| Количество претендентов: | 9 |
| Длина теста: | 9 |

Результаты моделирования схемы

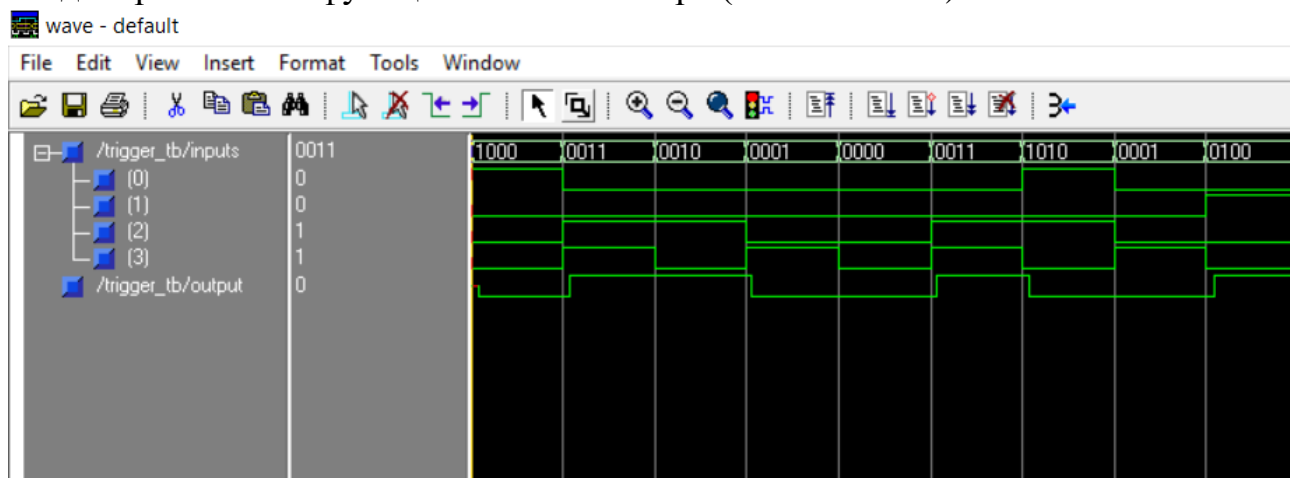
| № наб... | % обнар. ... | R(1) | S(1) | D(1) | C(1) | Q(1) |
|----------|--------------|------|------|------|------|------|
| 1 | 14.71% | 1 | 0 | 0 | 0 | 0 |
| 2 | 41.18% | 0 | 0 | 1 | 1 | 1 |
| 3 | 55.88% | 0 | 0 | 1 | 0 | 1 |
| 4 | 61.76% | 0 | 0 | 0 | 1 | 0 |
| 5 | 73.53% | 0 | 0 | 0 | 0 | 0 |
| 6 | 73.53% | 0 | 0 | 1 | 1 | 1 |
| 7 | 91.18% | 1 | 0 | 1 | 0 | 0 |
| 8 | 91.18% | 0 | 0 | 0 | 1 | 0 |
| 9 | 100.00% | 0 | 1 | 0 | 0 | 1 |

Количество наборов: 9

Моделирование на функциональном наборе (VLSI-SIM):



Моделирование на функциональном наборе (MODEL-SIM):



Построение автоматического теста:

Задание критериев качества теста

Требуемая полнота теста (%)

Покрывание одного набора (%)

Генерация теста

| | |
|----------------------------|--------|
| Требуемая полнота теста: | 100.00 |
| Достигнутая полнота теста: | 67.65 |
| Количество претендентов: | 366 |
| Длина теста: | 3 |

Список необнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%)

Покрывание одного набора (%)

Генерация теста

| | |
|----------------------------|--------|
| Требуемая полнота теста: | 100.00 |
| Достигнутая полнота теста: | 67.65 |
| Количество претендентов: | 640 |
| Длина теста: | 3 |

Список необнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%)

Покрывание одного набора (%)

Генерация теста

| | |
|----------------------------|--------|
| Требуемая полнота теста: | 100.00 |
| Достигнутая полнота теста: | 91.18 |
| Количество претендентов: | 1023 |
| Длина теста: | 5 |

Список необнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%)

Покрывание одного набора (%)

Генерация теста

| | |
|----------------------------|--------|
| Требуемая полнота теста: | 100.00 |
| Достигнутая полнота теста: | 100.00 |
| Количество претендентов: | 1025 |
| Длина теста: | 6 |

Результат построения автоматического теста:

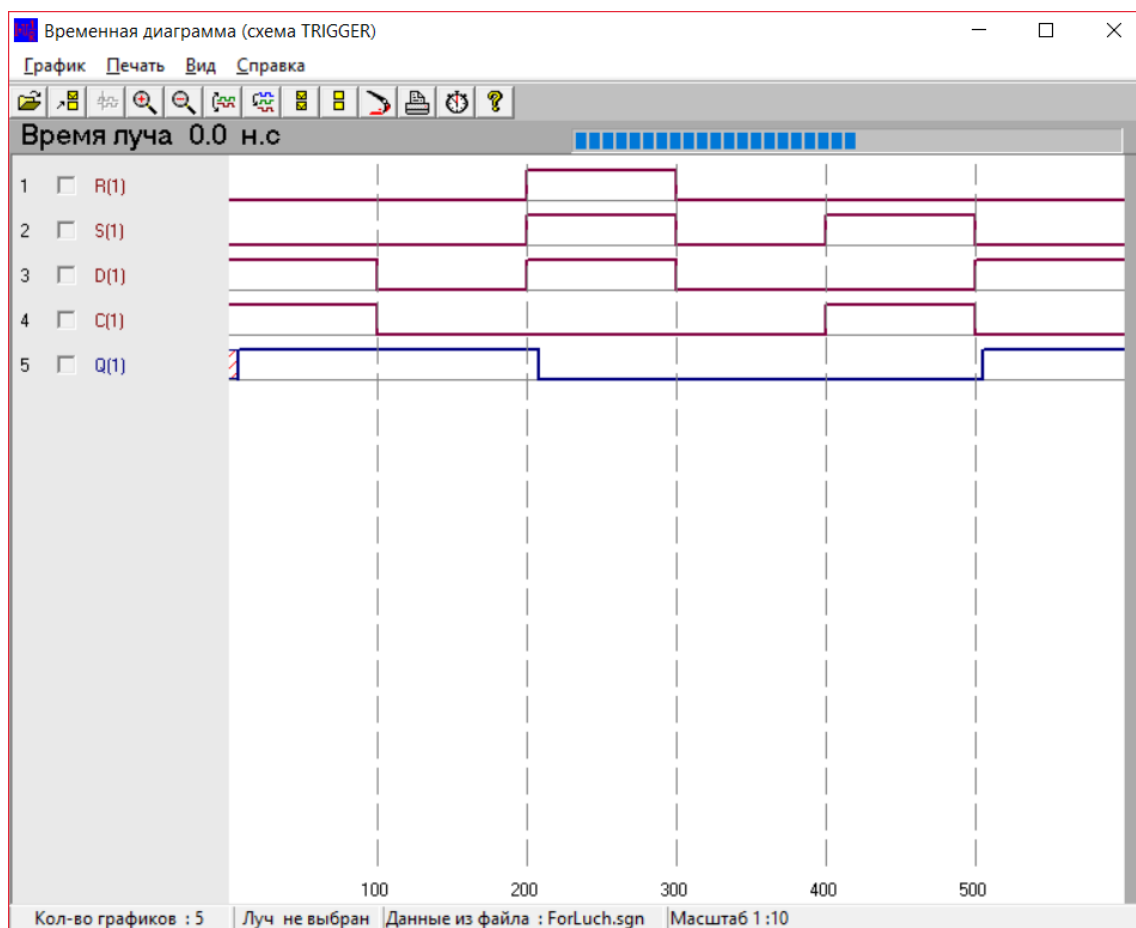
Результаты моделирования схемы

| № наб... | % обнар. ... | R(1) | S(1) | D(1) | C(1) | Q(1) |
|----------|--------------|------|------|------|------|------|
| 1 | 20.59% | 0 | 0 | 1 | 1 | 1 |
| 2 | 41.18% | 0 | 0 | 0 | 0 | 1 |
| 3 | 67.65% | 1 | 1 | 1 | 0 | 0 |
| 4 | 79.41% | 0 | 0 | 0 | 0 | 0 |
| 5 | 91.18% | 0 | 1 | 0 | 1 | 0 |
| 6 | 100.00% | 0 | 0 | 1 | 0 | 1 |

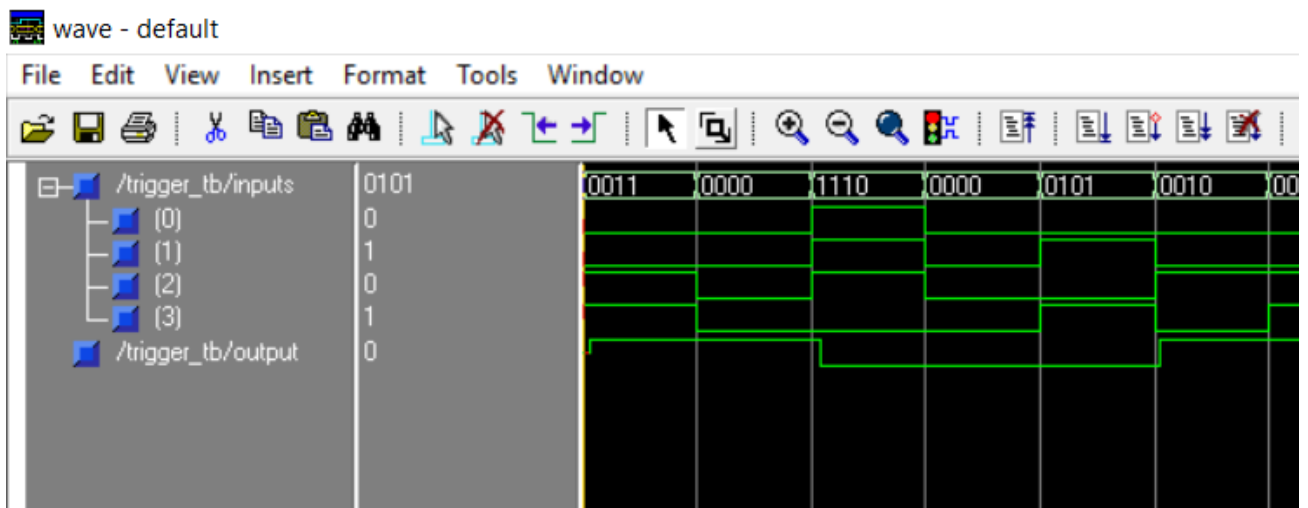
Количество наборов: 6

Сохранить тест

Моделирование на автоматическом тесте (VLSI-SIM):



Моделирование на автоматическом тесте (MODEL-SIM):



Построение автоматизированного теста:

Данные эксперта о входах схемы

Тип входа

- ☐ Общего типа
- ☒ Управляющий
- ☐ "Синхро"
- ☐ Эквивалентный
- ☐ По формату
- ☐ Константный
- ☐ Инверсный

Управляющий вход

Сохранить

Закреть

Просмотр и редактирование наборов, задающих режимы для управляющих входов

0

1

Текущий набор 2 Из 2

Редактир. Добавить Вставить Удалить Удал. все В файл Закреть

Задание критериев качества теста

Требуемая полнота теста (%) 100

Покрытие одного набора (%) 20

OK Отмена

Генерация теста

Требуемая полнота теста: 100.00
Достигнутая полнота теста: 52.94
Количество претендентов: 665
Длина теста: 2

Тест в файл Закреть

Список обнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%) 100

Покрытие одного набора (%) 15

OK Отмена

Генерация теста

Требуемая полнота теста: 100.00
Достигнутая полнота теста: 70.59
Количество претендентов: 1305
Длина теста: 3

Тест в файл Закреть

Список обнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%) 100

Покрытие одного набора (%) 10

OK Отмена

Генерация теста

Требуемая полнота теста: 100.00
Достигнутая полнота теста: 70.59
Количество претендентов: 1863
Длина теста: 3

Тест в файл Закреть

Список обнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%) 100

Покрытие одного набора (%) 5

OK Отмена

Генерация теста

Требуемая полнота теста: 100.00
Достигнутая полнота теста: 76.47
Количество претендентов: 2528
Длина теста: 4

Тест в файл Закреть

Список обнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%) 100

Покрытие одного набора (%) 3

OK Отмена

Генерация теста

Требуемая полнота теста: 100.00
Достигнутая полнота теста: 97.06
Количество претендентов: 3183
Длина теста: 6

Тест в файл Закреть

Список обнаруженных неисправностей в файле 'TRIGGER.UFL'

Задание критериев качества теста

Требуемая полнота теста (%)

Покрывтие одного набора (%)

Генерация теста

Требуемая полнота теста: 100.00
Достигнутая полнота теста: 100.00
Количество претендентов: 3185
Длина теста: 7

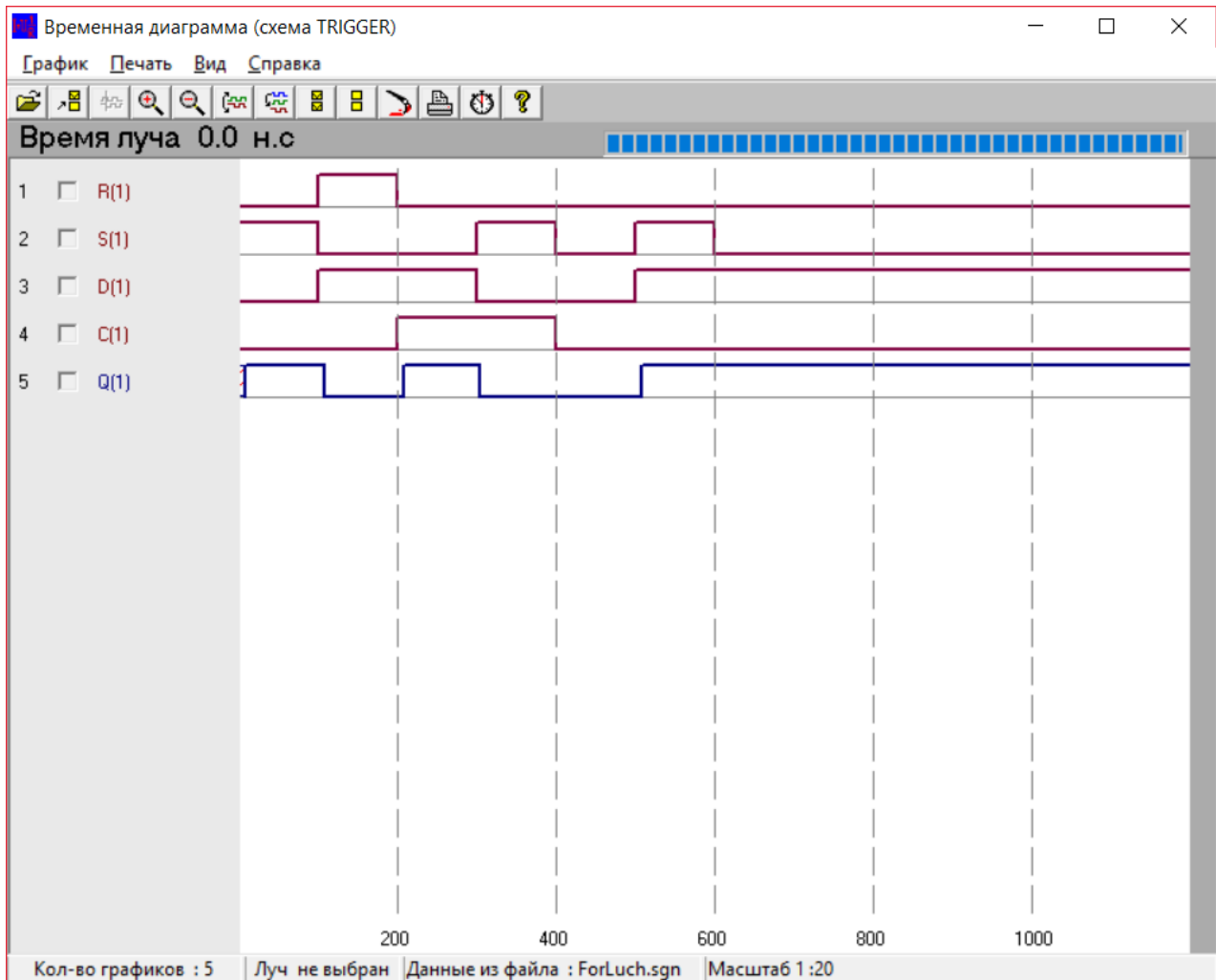
Результат:

Результаты моделирования схемы

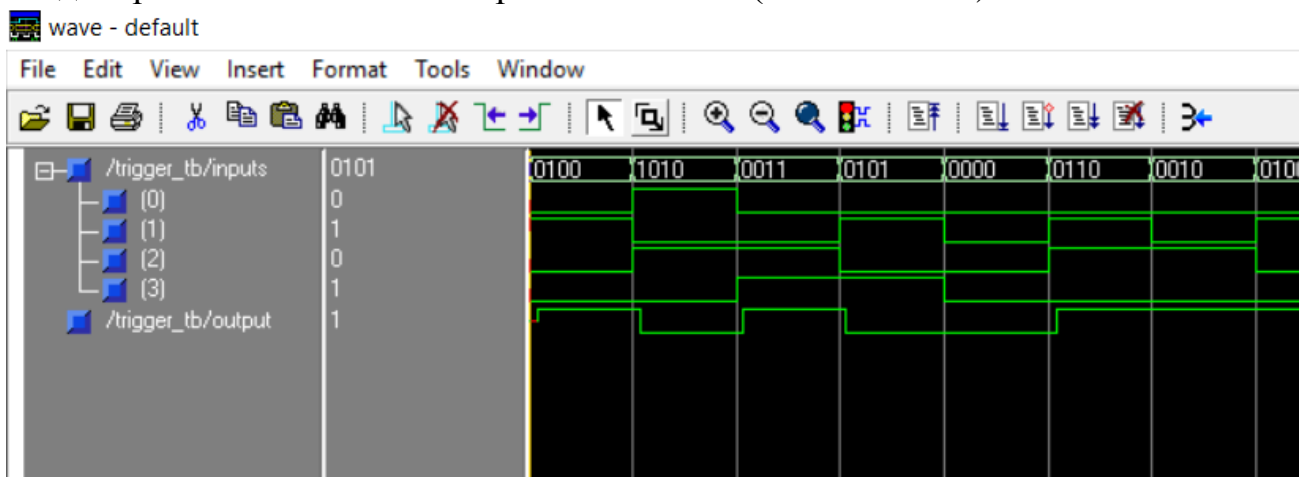
| N° наб... | % обнар. ... | R(1) | S(1) | D(1) | C(1) | Q(1) |
|-----------|--------------|------|------|------|------|------|
| 1 | 26.47% | 0 | 1 | 0 | 0 | 1 |
| 2 | 52.94% | 1 | 0 | 1 | 0 | 0 |
| 3 | 70.59% | 0 | 0 | 1 | 1 | 1 |
| 4 | 76.47% | 0 | 1 | 0 | 1 | 0 |
| 5 | 88.24% | 0 | 0 | 0 | 0 | 0 |
| 6 | 97.06% | 0 | 1 | 1 | 0 | 1 |
| 7 | 100.00% | 0 | 0 | 1 | 0 | 1 |

Количество наборов: 7

Моделирование на автоматизированном тесте (VLSI-SIM):



Моделирование на автоматизированном тесте (MODEL-SIM):



Исходный код схемы на VHDL:

Элемент N:

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

entity N is
    Port (
        INPUT: in STD_LOGIC;
        OUTPUT: out STD_LOGIC
    );
end N;

architecture Behavioral of N is
begin
    OUTPUT <= not INPUT after 1 ns;
end Behavioral;
```

Элемент NOA:

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

entity NOA is
    Port (
        A: in STD_LOGIC;
        B: in STD_LOGIC;
        C: in STD_LOGIC;
        OUTPUT: out STD_LOGIC
    );
end NOA;

architecture Behavioral of NOA is
begin
    OUTPUT <= not(A or (B and C))after 3 ns;
end Behavioral;
```

Элемент NOAA:

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

entity NOAA is
    Port (
```

```

        A: in STD_LOGIC;
        B: in STD_LOGIC;
        C: in STD_LOGIC;
        D: in STD_LOGIC;
        OUTPUT: out STD_LOGIC
    );
end NOAA;

architecture Behavioral of NOAA is
begin
    OUTPUT <= not((A and B) or (C and D)) after 4 ns;
end Behavioral;

```

Структурная схема триггера:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity trigger is
    Port (
        D: in STD_LOGIC;
        R: in STD_LOGIC;
        S: in STD_LOGIC;
        C: in STD_LOGIC;
        Q: out STD_LOGIC
    );
end trigger;

architecture Behavioral of trigger is
    component N
        Port (
            INPUT: in STD_LOGIC;
            OUTPUT: out STD_LOGIC
        );
    end component;

    component NOA
        Port (
            A: in STD_LOGIC;
            B: in STD_LOGIC;
            C: in STD_LOGIC;
            OUTPUT: out STD_LOGIC
        );
    end component;

    component NOAA
        Port (
            A: in STD_LOGIC;

```

```

        B: in STD_LOGIC;
        C: in STD_LOGIC;
        D: in STD_LOGIC;
        OUTPUT: out STD_LOGIC
    );
end component;

signal ns, nc, nnc: STD_LOGIC;
signal noaa_r, noa_r: STD_LOGIC;
begin
    n_first_instance: N port map(C, nc);
    n_second_instance: N port map(nc, nnc);
    n_third_instance: N port map(S, ns);
    noaa_first_instance: NOAA port map(noa_r, nc, D, nnc, noaa_r);
    noa_first_instance: NOA port map(R, ns, noaa_r, noa_r);
    n_four_instance: N port map(noaa_r, Q);
end Behavioral;

```

Поведенческая схема триггера:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity trigger_behavioral is
    Port (
        D: in STD_LOGIC;
        R: in STD_LOGIC;
        S: in STD_LOGIC;
        C: in STD_LOGIC;
        Q: out STD_LOGIC
    );
end trigger_behavioral;

architecture Behavioral of trigger_behavioral is
begin
    process (D, R, S, C)
        variable trigger_state: STD_LOGIC := '0';
    begin
        if (R = '1' and C = '0') then
            trigger_state := '0';
            Q <= trigger_state;
        end if;

        if (R = '0' and S = '1' and C = '0') then
            trigger_state := '1';
            Q <= trigger_state;
        end if;
    end process;
end Behavioral;

```

```

        if (R = '0' and S = '0' and C'event and C = '1') then
            trigger_state := D;
            Q <= trigger_state;
        end if;

        if (R = '0' and S = '0' and C = '0') then
            Q <= trigger_state;
        end if;
    end process;

```

end Behavioral;

Tect:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity trigger_tb is
    --Port ( );
end trigger_tb;

```

```

architecture Behavioral of trigger_tb is
    component trigger
        Port (
            D: in STD_LOGIC;
            R: in STD_LOGIC;
            S: in STD_LOGIC;
            C: in STD_LOGIC;
            Q: out STD_LOGIC
        );
    end component;

```

```

    type TRIGGER_TEST is array (natural range <>) of STD_LOGIC_VECTOR(0
to 3);

```

```

    signal inputs: STD_LOGIC_VECTOR(0 to 3);
    signal output: STD_LOGIC;
    constant manual_functional_test: TRIGGER_TEST(0 to 8) := (
        "1000",
        "0011",
        "0010",
        "0001",
        "0000",
        "0011",
        "1010",
        "0001",
        "0100"
    );

```

```

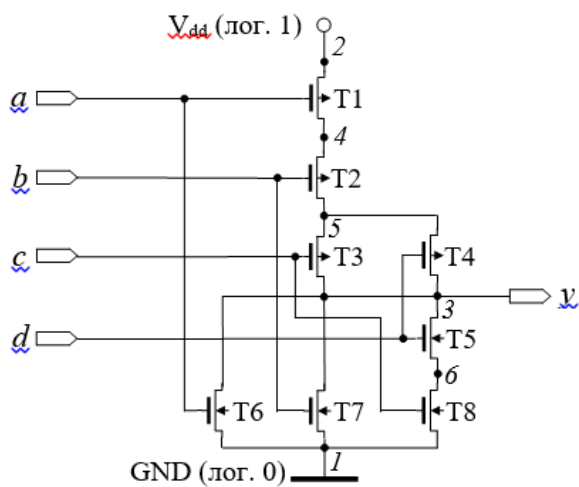
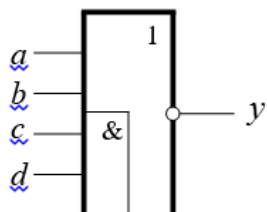
);
constant automatic_functional_test: TRIGGER_TEST(0 to 5) := (
    "0011",
    "0000",
    "1110",
    "0000",
    "0101",
    "0010"
);
constant automatization_functional_test: TRIGGER_TEST(0 to 6) := (
    "0100",
    "1010",
    "0011",
    "0101",
    "0000",
    "0110",
    "0010"
);
begin
    trigger_instance: trigger port map(
        R => inputs(0),
        S => inputs(1),
        D => inputs(2),
        C => inputs(3),
        Q => output
    );

    process
    begin
        for index in 0 to 6 loop
            inputs <= automatization_functional_test(index);
            wait for 100ns;
        end loop;
    end process;

end Behavioral;

```

7 Комбинационный элемент 2И-3ИЛИ-НЕ (NO3A2)

[illegible]

y^* - состояние исправной схемы;

y_i - состояние схемы с неисправностью i ;

Π - предыдущее состояние схемы.

Тесты для проверки:

- неисправности типа «ПЗТ» транзистора T1, T2
{0011-1111} – {0000-0010}

- неисправности типа «ПЗТ» транзистора T3
{0011-1111} – {0001}

- неисправности типа «ПЗТ» транзистора T4
{0011-1111} – {0010}

- неисправности типа «ПЗТ» транзистора T5, T8
{0000-0010} – {0011}

- неисправности типа «ПЗТ» транзистора T6
{0000-0010} – {1000-1010}

- неисправности типа «ПЗТ» транзистора T7
{0000-0010} – {0100-0110}

Тест для проверки всех неисправностей заданного класса:

0011

0010

0011

0001

0000

0011

0010

1010

0010

0110

