

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Лабораторная работа №4  
дисциплина «Теория цифровых автоматов»  
по теме «Диагностика неисправностей комбинационных схем с одним выходом»

Выполнил: студент группы ВТ-31  
Проверил:

Макаров Д.С.  
Рязанов Ю.Д.

Белгород 2019

# Лабораторная работа №4

## «Диагностика неисправностей комбинационных схем с одним выходом»

**Цель работы:** научиться строить диагностические тесты и алгоритмы распознавания неисправностей комбинационных схем с одним выходом..

### Вариант 9

**Задание:**

**Дано:**

1. комбинационная схема с одним выходом, построенная при выполнении лабораторной работы № 3;
2. множество одиночных неисправностей, состоящее из неисправностей «константа 0» и «константа 1» на каждом входе схемы.

**Найти:** диагностический тест для заданного множества неисправностей.

**Построить:** алгоритм распознавания неисправностей.

Для решения задачи нужно выполнить следующие задания:

1. Написать программу моделирования исправной схемы и построить таблицу истинности булевой функции, реализуемой исправной комбинационной схемой.
2. Для каждой неисправности написать программу моделирования схемы с этой неисправностью и построить таблицу истинности функции неисправности.
3. Определить, существуют ли в множестве неисправностей необнаружимые и неразличимые неисправности.
4. Составить матрицу функций неисправностей, содержащей попарно различные строки. Столбцы матрицы соответствуют наборам входных сигналов, а строки — векторам-значений функций неисправности. Каждой строке матрицы поставить в соответствие множество подозреваемых неисправностей.
5. Составить диагностическую матрицу, заменив в матрице функций неисправностей каждую функцию неисправности соответствующей разностной функцией.
6. По диагностической матрице найти минимальный диагностический тест.
7. В матрице функций неисправностей (см. п. 4) оставить только столбцы, соответствующие наборам входных сигналов, принадлежащим диагностическому тесту.
8. По полученной в п. 7 матрице построить алгоритм распознавания неисправностей в виде диагностического дерева.
9. Используя программу моделирования комбинационной схемы с неисправностью и алгоритм распознавания неисправностей написать программу для проведения диагностического эксперимента.

## Ход работы

Матрица функций неисправностей

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30	X31	X32
x1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
f1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	1	1
f2	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0	0
f3	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
f4	0	0	1	1	0	1	1	1	0	0	1	1	0	1	1	1	1	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0
f5	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0
f6	0	1	0	1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
f7	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
f8	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
f9	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0

x3(0), x5(0) неразличимые неисправности

Матрица разностных функций

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30	X31	X32
R1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
R2	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0
R4	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0
R5	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
R6	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
R7	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0
R8	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0
R9	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0

	X10	X31	X7	X8	X11	X24	X2	X5
R1	0	1	0	0	0	1	0	0
R2	1	0	1	1	1	0	1	1
R3	0	1	0	0	1	0	0	0
R4	0	0	1	0	0	0	0	0
R5	0	0	0	1	0	1	0	0
R6	1	0	0	0	0	0	1	0
R7	0	1	0	0	1	1	0	0
R8	1	0	0	0	0	0	0	0
R9	0	0	1	0	0	0	0	1

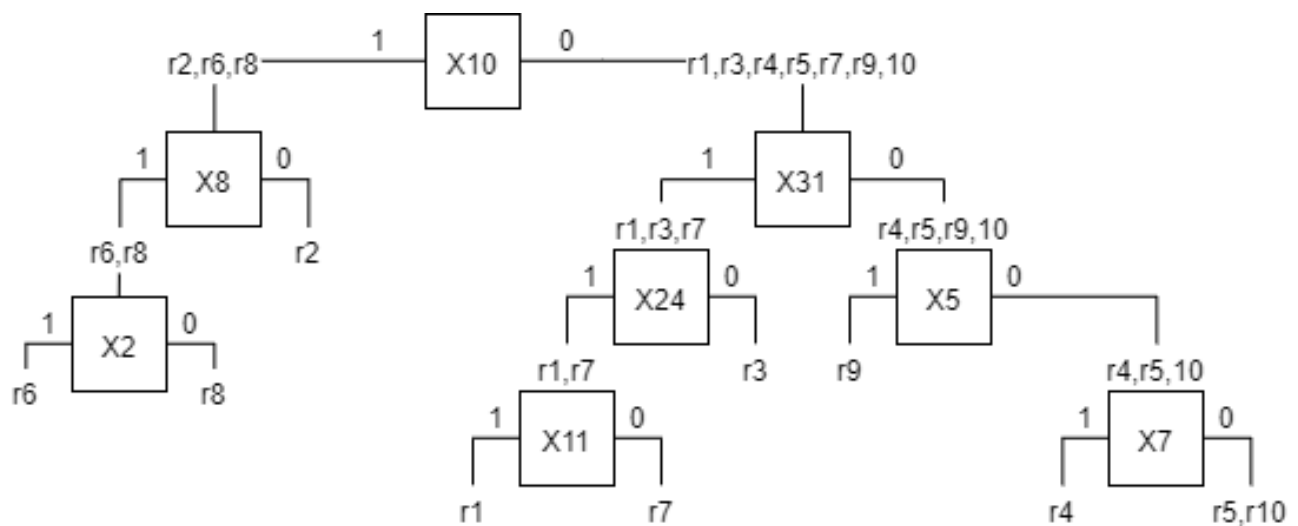


Рис. 1: Диагностическое дерево

# Приложение

## Содержимое файла funcTest.py

```
from binVectors import gen_bin_vector_5 as gen_bin_vector
from tabulate import tabulate

def truth_table(vector,f):
    result = []
    for i in range(0,len(vector)):
        result.append([
            i+1,
            vector[i][0],
            int(f(vector[i][0])),
        ])
    return result

def bad_truth_table(vector,f,var,val):
    result = []
    for i in range(0,len(vector)):
        result.append([
            i+1,
            vector[i][0],
            int(f(vector[i][0],var,val)),
        ])
    return result

def function(str_val):
    x1 = bool(int(str_val[0]))
    x2 = bool(int(str_val[1]))
    x3 = bool(int(str_val[2]))
    x4 = bool(int(str_val[3]))
    x5 = bool(int(str_val[4]))
    z1 = x1 and (not x2)
    z2 = x1 and (not x5)
    z3 = (not x1) and x3
    z4 = x2 and x4
    z5 = (not x3) and (not x4)

    u1 = z3 and x5
    u2 = z4 and (not x1)
    u7 = z2 and (not x4)
    u4 = z1 and (not x4)
    u5 = z5 and x1
    u6 = z2 and (not x2)
    u3 = z1 and (not x3)

    v1 = u1 or u2
    v2 = u3 or u4
    v3 = u5 or u6
    v4 = v1 or u7
    v5 = v2 or v3
    f = v4 or v5
    return f

def bad_schema(vars, fault_gate, fault_val):
    x1 = bool(int(fault_val)) if fault_gate == 'x1' else bool(int(vars[0]))
    x2 = bool(int(fault_val)) if fault_gate == 'x2' else bool(int(vars[1]))
    x3 = bool(int(fault_val)) if fault_gate == 'x3' else bool(int(vars[2]))
    x4 = bool(int(fault_val)) if fault_gate == 'x4' else bool(int(vars[3]))
```

```

x5 = bool(int(fault_val)) if fault_gate == 'x5' else bool(int(vars[4]))
z1 = x1 and (not x2)
z2 = x1 and (not x5)
z3 = (not x1) and x3
z4 = x2 and x4
z5 = (not x3) and (not x4)

u1 = z3 and x5
u2 = z4 and (not x1)
u7 = z2 and (not x4)
u4 = z1 and (not x4)
u5 = z5 and x1
u6 = z2 and (not x2)
u3 = z1 and (not x3)

v1 = u1 or u2
v2 = u3 or u4
v3 = u5 or u6
v4 = v1 or u7
v5 = v2 or v3
f = v4 or v5
return f

def diag_test(var, val):
    table = bad_truth_table(gen_bin_vector(), bad_schema, var, val)
    print("Подаем набор X10")
    if(table[9][2]==1):
        print("Возможные неисправности F2,F6,F8")
        print("Подаем набор X8")
        if(table[7][2]==1):
            print("Возможные неисправности F6,F8")
            print("Подаем набор X2")
            if(table[1][2]==1):
                print("Неисправность определена F6")
            else:
                print("Неисправность определена F8")
        else:
            print("Неисправность определена F2")
    else:
        print("Возможные неисправности F1,F3,F4,F5,F7,F9,F10")
        print("Подаем набор X31")
        if(table[30][2]==1):
            print("Возможные неисправности F1,F3,F7")
            print("Подаем набор X24")
            if(table[23][2]==1):
                print("Возможные неисправности F1,F7")
                print("Подаем набор X11")
                if(table[10][2]==1):
                    print("Неисправность определена F1")
                else:
                    print("Неисправность определена F7")
            else:
                print("Неисправность определена F3")
        else:
            print("Возможные неисправности F4,F5,F9,F10")
            print("Подаем набор X5")
            if(table[4][2]==1):
                print("Неисправность определена F10")
            else:
                print('Возможные неисправности F4,F5,F9')

```

```

print("Подаем набор X7")
if(table[6][2]==1):
    print("Неисправность определена F4")
else:
    print('Возможные неисправности F5,F9')

table_head = ["J", "$x_1x_2x_3x_4x_5$", "f"]
table = truth_table(gen_bin_vector(),function)
print(tabulate(table,table_head,tablefmt="simple"))

var_arr = ['x1','x2','x3','x4','x5']

for var in var_arr:
    for val in ['0','1']:
        print("Неисправность "+var+"("+val+")")
        diag_test(var,val)
        print()

```