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 №6
дисциплина «Теория цифровых автоматов»
по теме «Синтез и анализ многовыходных комбинационных схем в базисе И-ИЛИ-НЕ с учетом неопределенностей»

Выполнил: студент группы ВТ-31 Макаров Д.С.
Проверил: Рязанов Ю.Д.

Белгород 2019

Лабораторная работа №6

«Синтез и анализ многовыходных комбинационных схем в базисе И-ИЛИ-НЕ с учетом неопределенностей»

Цель работы:научиться строить эффективные по быстродействию и затратам оборудования многовыходные комбинационные схемы с учетом неопределенностей.

Вариант 9

Задание:

1. Составить таблицу истинности заданной частично определенной булевой функции (*см. варианты заданий в таблице 2*). Булева функция здесь задана двумя условиями (*условие 1 и условие 2*), зависящими от значений аргументов. Если на наборе аргументов условие 2 истинно, то значение функции на этом наборе не определено. Если же на наборе аргументов условие 2 ложно, то значение функции на этом наборе равно значению условия 1 на этом наборе аргументов. В условии значение аргумента отождествляется с двоичной цифрой, а последовательность аргументов — с двоичным числом. Для составления таблицы истинности рекомендуется написать программу.
2. Решить задачу минимизации частично определенной булевой функции в классе дизъюнктивных нормальных форм.
3. Написать программу, строящую таблицу истинности булевой функции, полученной при выполнении п. 2 Сравнить полученную таблицу с таблицей истинности исходной частично определенной булевой функции.
4. Применить факторизационный метод синтеза многоярусной комбинационной схемы в базисе И-ИЛИ-НЕ с двухвходовыми элементами И и ИЛИ по полученной при выполнении п. 2 минимальной дизъюнктивной нормальной форме булевой функции.
5. Решить задачу минимизации частично определенной булевой функции в классе конъюнктивных нормальных форм.
6. Написать программу, строящую таблицу истинности булевой функции, полученной при выполнении п. 5 Сравнить полученную таблицу с таблицей истинности исходной частично определенной булевой функции.
7. Применить факторизационный метод синтеза многоярусной комбинационной схемы в базисе И-ИЛИ-НЕ с двухвходовыми элементами И и ИЛИ по полученной при выполнении п. 5 минимальной конъюнктивной нормальной форме булевой функции.

Ход работы

№	$x_1x_2x_3x_4x_5$	f1	f2	f3
1	00000	0	0	0
2	00001	0	0	0
3	00010	0	0	0
4	00011	-	-	-
5	00100	-	-	-
6	00101	-	-	-
7	00110	-	-	-
8	00111	-	-	-
9	01000	0	0	0
10	01001	0	0	0
11	01010	1	1	1
12	01011	1	1	1
13	01100	-	-	-
14	01101	-	-	-
15	01110	-	-	-
16	01111	-	-	-
17	10000	-	-	-
18	10001	1	1	0
19	10010	1	1	1
20	10011	1	0	1
21	10100	1	1	0
22	10101	-	-	-
23	10110	-	-	-
24	10111	-	-	-
25	11000	-	-	-
26	11001	-	-	-
27	11010	0	0	1
28	11011	0	0	0
29	11100	1	0	0
30	11101	0	0	1
31	11110	-	-	-
32	11111	-	-	-

№	$x_1x_2x_3x_4x_5$	f1	f2	f3
{1}	00000	0	0	0
{2}	00001	0	0	0
{3}	00010	0	0	0
{9}	01000	0	0	0
{10}	01001	0	0	0
{11}	01010	1	1	1

№	$x_1x_2x_3x_4x_5$	f1	f2	f3
{12}	01011	1	1	1
{18}	10001	1	1	0
{19}	10010	1	1	1
{20}	10011	1	0	1
{21}	10100	1	1	0
{27}	11010	0	0	1
{28}	11011	0	0	0
{29}	11100	1	0	0
{30}	11101	0	0	1

№	x1	x2	x3	x4	x5
{1, 11}	0	1	0	1	0
{1, 12}	0	1	0	1	1
{1, 18}	1	0	0	0	1
{1, 19}	1	0	0	1	0
{1, 20}	1	0	0	1	1
{1, 21}	1	0	1	0	0
{1, 27}	1	1	0	1	0
{1, 29}	1	1	1	0	0
{1, 30}	1	1	1	0	1
{2, 11}	0	1	0	1	1
{2, 12}	0	1	0	1	0
{2, 18}	1	0	0	0	0
{2, 19}	1	0	0	1	1
{2, 20}	1	0	0	1	0
{2, 21}	1	0	1	0	1
{2, 27}	1	1	0	1	1
{2, 29}	1	1	1	0	1
{2, 30}	1	1	1	0	0
{11, 3}	0	1	0	0	0
{3, 12}	0	1	0	0	1
{18, 3}	1	0	0	1	1
{19, 3}	1	0	0	0	0
{3, 20}	1	0	0	0	1
{3, 21}	1	0	1	1	0
{27, 3}	1	1	0	0	0
{3, 29}	1	1	1	1	0
{3, 30}	1	1	1	1	1
{9, 11}	0	0	0	1	0
{9, 12}	0	0	0	1	1

№	x1	x2	x3	x4	x5
{9, 18}	1	1	0	0	1
{9, 19}	1	1	0	1	0
{9, 20}	1	1	0	1	1
{9, 21}	1	1	1	0	0
{9, 27}	1	0	0	1	0
{9, 29}	1	0	1	0	0
{9, 30}	1	0	1	0	1
{10, 11}	0	0	0	1	1
{10, 12}	0	0	0	1	0
{10, 18}	1	1	0	0	0
{10, 19}	1	1	0	1	1
{10, 20}	1	1	0	1	0
{10, 21}	1	1	1	0	1
{10, 27}	1	0	0	1	1
{10, 29}	1	0	1	0	1
{10, 30}	1	0	1	0	0
{18, 11}	1	1	0	1	1
{11, 20}	1	1	0	0	1
{11, 21}	1	1	1	1	0
{27, 11}	1	0	0	0	0
{11, 28}	1	0	0	0	1
{11, 29}	1	0	1	1	0
{11, 30}	1	0	1	1	1
{18, 12}	1	1	0	1	0
{12, 20}	1	1	0	0	0
{12, 21}	1	1	1	1	1
{27, 12}	1	0	0	0	1
{12, 28}	1	0	0	0	0
{12, 29}	1	0	1	1	1
{12, 30}	1	0	1	1	0
{18, 19}	0	0	0	1	1
{18, 20}	0	0	0	1	0
{18, 27}	0	1	0	1	1
{18, 28}	0	1	0	1	0
{18, 29}	0	1	1	0	1
{18, 30}	0	1	1	0	0
{19, 20}	0	0	0	0	1
{19, 21}	0	0	1	1	0
{27, 19}	0	1	0	0	0
{19, 28}	0	1	0	0	1
{19, 29}	0	1	1	1	0

№	x1	x2	x3	x4	x5
{19, 30}	0	1	1	1	1
{20, 21}	0	0	1	1	1
{27, 20}	0	1	0	0	1
{20, 28}	0	1	0	0	0
{20, 29}	0	1	1	1	1
{20, 30}	0	1	1	1	0
{27, 21}	0	1	1	1	0
{28, 21}	0	1	1	1	1
{29, 21}	0	1	0	0	0
{21, 30}	0	1	0	0	1
{27, 28}	0	0	0	0	1
{27, 29}	0	0	1	1	0
{28, 29}	0	0	1	1	1
{28, 30}	0	0	1	1	0
{29, 30}	0	0	0	0	1

Убираем дублирующиеся строки

№	x1	x2	x3	x4	x5
{1, 11}	0	1	0	1	0
{1, 12}	0	1	0	1	1
{1, 18}	1	0	0	0	1
{1, 19}	1	0	0	1	0
{1, 20}	1	0	0	1	1
{1, 21}	1	0	1	0	0
{1, 27}	1	1	0	1	0
{1, 29}	1	1	1	0	0
{1, 30}	1	1	1	0	1
{2, 18}	1	0	0	0	0
{2, 21}	1	0	1	0	1
{2, 27}	1	1	0	1	1
{11, 3}	0	1	0	0	0
{3, 12}	0	1	0	0	1
{3, 21}	1	0	1	1	0
{27, 3}	1	1	0	0	0
{3, 29}	1	1	1	1	0
{3, 30}	1	1	1	1	1
{9, 11}	0	0	0	1	0
{9, 12}	0	0	0	1	1
{9, 18}	1	1	0	0	1
{11, 30}	1	0	1	1	1

№	x1	x2	x3	x4	x5
{18, 29}	0	1	1	0	1
{18, 30}	0	1	1	0	0
{19, 20}	0	0	0	0	1
{19, 21}	0	0	1	1	0
{19, 29}	0	1	1	1	0
{19, 30}	0	1	1	1	1
{20, 21}	0	0	1	1	1

Вошедшие аргументы x_1, x_2, x_4, x_5

№	$x_1x_2x_4x_5$	f1	f2	f3
{1}	0000	0	0	0
{2}	0001	0	0	0
{3}	0010	0	0	0
{9}	0100	0	0	0
{10}	0101	0	0	0
{11}	0110	1	1	1
{12}	0111	1	1	1
{18}	1001	1	1	0
{19}	1010	1	1	1
{20}	1011	1	0	1
{21}	1000	1	1	0
{27}	1110	0	0	1
{28}	1111	0	0	0
{29}	1100	1	0	0
{30}	1101	0	0	1

Конституенты 1

- 0110 {1,2,3}
- 0111 {1,2,3}
- 1001 {1,2}
- 1010 {1,2,3}
- 1011 {1,3}
- 1000 {1,2}
- 1110 {3}
- 1100 {1}
- 1101 {3}

Конституенты 0

- 0000 {1,2,3}

- 0001 {1,2,3}
- 0010 {1,2,3}
- 0100 {1,2,3}
- 0101 {1,2,3}
- 1001 {3}
- 1011 {2}
- 1000 {3}
- 1110 {1,2}
- 1111 {1,2,3}
- 1100 {2,3}
- 1101 {1,2}

Импликанты первого порядка

- 011- {1,2,3}
- 0-11 {1,2,3}
- 100- {1,2}
- -011 {1,3}
- 101- {1,3}
- 10-0 {1,2}
- -110 {3}
- 1-10 {3}
- 1-00 {1}

	x_1	$\overline{x}_1$	x_2	$\overline{x}_2$	x_4	$\overline{x}_4$	x_5	$\overline{x}_5$	z_1	z_2	z_3	z_4	z_5	z_6
u_1		-	*		-				*					
u_2		-			-		*		*					
u_3	*			-		-						*		
u_4				-	-		*						*	
u_5	-			-	*					*				
u_6	-			-				*		*				
u_7			-		-			*						*
u_8	-				*			-			*			
u_9	-					*		-			*			
z_1		*			*									
z_2	*			*										
z_3	*							*						
z_4				*		*								
z_5				*	*									
z_6			*		*									

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
f_1	-	-	-	-	-	-			-	-			*	*			
f_2	-	-	-			-				*							
f_3	-	-		-	-		*	*			*		*				
v_1	-	-	-			-						*			*		
v_2	-	-										*					
v_3	*	*															
v_4				*	*												
v_5									*	*							
v_6			*			*											

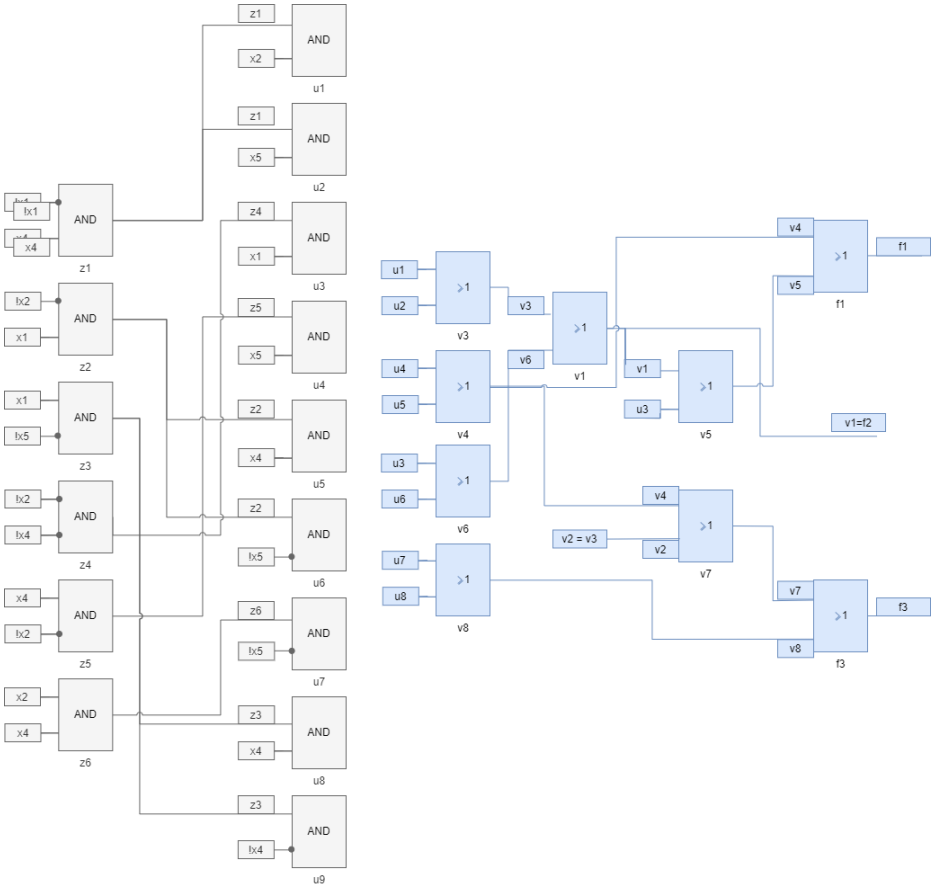

Рис. 1: Схема

Приложение

Содержимое файла funcTest.py

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

def del_dup(list):
    res = []
    for i in list:
        if i not in res:
            res.append(i)
    return res

def diff_pos(str_a,str_b):
    list_a=list(str_a)
    list_b=list(str_b)
    i=0
    while(i<len(list_a)):
        if(list_a[i]!=list_b[i]):
            return i
        i = i + 1

def hamming_dist(s1, s2):
    assert len(s1) == len(s2)
    return sum(ch1 != ch2 for ch1, ch2 in zip(s1, s2))

def truth_table(vector):
    result = []
    for i in range(0,len(vector)):
        args = vector[i][0]
        sch = test_schema(args)
        result.append([
            i+1,
            args,
            f(f_v9,f_v12,args),
            int(sch[0]),
            f(f_v10,f_v12,args),
            int(sch[1]),
            f(f_v11,f_v12,args),
            int(sch[2])
            #int(f(f_v16,f_v19,args)),
            #int(f(f_v17,f_v19,args)),
            #int(f(f_v18,f_v19,args)),
        ])
    return result


def search_str(arr,str):
    for index,i in enumerate(arr):
        if i[1]==str[1] and i[2]==str[2] and i[3]==str[3] and i[4]==str[4] and i[5]==str[5]:
            return index
    return None

def step_one_simplify(table):
    result_arr = []
    for i in table:
        if search_str(result_arr,i) == None:
            result_arr.append(i)
    return result_arr
```

```
def f_v9(str_val):
    x1 = str_val[0]
    x2 = str_val[1]
    x3 = str_val[2]
    x4 = str_val[3]
    x5 = str_val[4]
    return (
        3 < (int(x4 + x5,2) + int(x1 + x2 + x3,2)) < 8
    )

def f_v10(str_val):
    x1 = str_val[0]
    x2 = str_val[1]
    x3 = str_val[2]
    x4 = str_val[3]
    x5 = str_val[4]
    return (
        4 <= (int(x4 + x5,2) + int(x1 + x2 + x3,2)) <= 6
    )

def f_v11(str_val):
    x1 = str_val[0]
    x2 = str_val[1]
    x3 = str_val[2]
    x4 = str_val[3]
    x5 = str_val[4]
    return (
        5 <= (int(x2 + x3,2) + int(x4 + x5 + x1,2)) <= 8
    )

def f_v12(str_val):
    x1 = str_val[0]
    x2 = str_val[1]
    x3 = str_val[2]
    x4 = str_val[3]
    x5 = str_val[4]
    return (
        -2 <= (int(x1 + x2,2) - int(x3 + x4 + x5,2)) <= 1
    )

def f(f1,f2,str_val):
    if not f2(str_val):
        return '-'
    else:
         return str(int(f1(str_val)))

def test_schema(str_val):
    x1 = bool(int(str_val[0]))
    x2 = bool(int(str_val[1]))
    x4 = bool(int(str_val[3]))
    x5 = bool(int(str_val[4]))

    z1 = not x1 and x4
    z2 = not x2 and x1
    z3 = x1 and not x5
    z4 = not x2 and not x4
    z5 = x4 and not x2
    z6 = x2 and x4
```

```
    u1 = z1 and x2
    u2 = z1 and x5
    u3 = z4 and x1
    u4 = z5 and x5
    u5 = z2 and x4
    u6 = z2 and not x5
    u7 = z6 and not x5
    u8 = z3 and x4
    u9 = z3 and not x4

    v3 = u1 or u2
    v4 = u4 or u5
    v6 = u3 or u6
    v8 = u7 or u8

    v1 = v3 or v6
    v2 = v3

    v5 = v1 or u3
    v7 = v4 or v2

    f1 = v4 or v5
    f2 = v1
    f3 = v7 or u8
    return (f1,f2,f3)
table_head = ["№","$x_1x_2x_3x_4x_5$","f1",'sch_f1',"f2",'sch_f2',"f3",'sch_f3']
table = truth_table(gen_bin_vector())
print(tabulate(table,table_head,tablefmt="simple"))
```