Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота № 7**

**з курсу: «Теорія проектування комп’ютерних систем»**

*Виконав:*

студент 4 курсу

групи ІО-12, ФІОТ

Бута С. О.

Київ, 2014 р.

**ЗАДАНИЕ:**

1. Разработать модуль генерации аналитических форм минимизированных булевых функций из предыдущей работы.

2. Разработать средства сохранения результатов в файл формата VHDL.

**Зачётка: 1205**

**ВАРИАНТ: 1205 = 10000010000010**

**010 =>** 3AND 2NOR

**Программа:**

Функции переключения и выходных сигналов:

y1 = !(!Qt0 !Qt1 !Qt2 v !Qt0 !Qt1 Qt2 v !Qt0 Qt1 !Qt2 v Qt0 Qt1 Qt2 v Qt0 Qt1 !Qt2 v Qt0 !Qt1 !Qt2 )

y2 = !(!Qt0 !Qt1 !Qt2 v !Qt0 !Qt1 Qt2 v !Qt0 Qt1 Qt2 x1 v !Qt0 Qt1 Qt2 !x1 v !Qt0 Qt1 !Qt2 v Qt0 Qt1 Qt2 v Qt0 Qt1 !Qt2 v Qt0 !Qt1 !Qt2 )

y56 = !(!Qt0 !Qt1 !Qt2 v !Qt0 !Qt1 Qt2 v !Qt0 Qt1 Qt2 x1 v !Qt0 Qt1 Qt2 !x1 v Qt0 Qt1 Qt2 v Qt0 Qt1 !Qt2 v Qt0 !Qt1 !Qt2 )

J0 = !(!Qt0 !Qt1 !Qt2 v !Qt0 !Qt1 Qt2 v !Qt0 Qt1 Qt2 x1 v Qt0 Qt1 Qt2 v Qt0 Qt1 !Qt2 v Qt0 !Qt1 !Qt2 )

K0 = !(!Qt0 !Qt1 !Qt2 v !Qt0 !Qt1 Qt2 v !Qt0 Qt1 Qt2 x1 v !Qt0 Qt1 Qt2 !x1 v !Qt0 Qt1 !Qt2 v Qt0 Qt1 Qt2 v Qt0 Qt1 !Qt2 )

J1 = !(!Qt0 !Qt1 !Qt2 v !Qt0 Qt1 Qt2 x1 v !Qt0 Qt1 Qt2 !x1 v !Qt0 Qt1 !Qt2 v Qt0 Qt1 Qt2 v Qt0 Qt1 !Qt2 v Qt0 !Qt1 !Qt2 )

K1 = !(!Qt0 !Qt1 !Qt2 v !Qt0 !Qt1 Qt2 v !Qt0 Qt1 Qt2 x1 v !Qt0 Qt1 Qt2 !x1 v !Qt0 Qt1 !Qt2 v Qt0 Qt1 Qt2 v Qt0 !Qt1 !Qt2 )

J2 = !(!Qt0 !Qt1 Qt2 v !Qt0 Qt1 Qt2 x1 v !Qt0 Qt1 Qt2 !x1 v Qt0 Qt1 Qt2 v Qt0 Qt1 !Qt2 v Qt0 !Qt1 !Qt2 )

K2 = !(!Qt0 !Qt1 !Qt2 v !Qt0 !Qt1 Qt2 v !Qt0 Qt1 Qt2 !x1 v !Qt0 Qt1 !Qt2 v Qt0 Qt1 !Qt2 v Qt0 !Qt1 !Qt2 )

(тоже что и выше, но в битах):

y1 = !(000-v001-v010-v111-v110-v100-)

y2 = !(000-v001-v0111v0110v010-v111-v110-v100-)

y56 = !(000-v001-v0111v0110v111-v110-v100-)

J0 = !(000-v001-v0111v111-v110-v100-)

K0 = !(000-v001-v0111v0110v010-v111-v110-)

J1 = !(000-v0111v0110v010-v111-v110-v100-)

K1 = !(000-v001-v0111v0110v010-v111-v100-)

J2 = !(001-v0111v0110v111-v110-v100-)

K2 = !(000-v001-v0110v010-v110-v100-)

Минимизированные функции:

y1 = !(!Qt0 !Qt1 v Qt0 Qt1 v !Qt2 )

y2 = !(!Qt0 v !Qt2 v Qt1 )

y56 = !(!Qt0 !Qt1 v !Qt1 !Qt2 v Qt0 Qt1 v Qt1 Qt2 v Qt0 !Qt2 )

J0 = !(!Qt0 Qt1 Qt2 x1 v !Qt0 !Qt1 v !Qt1 !Qt2 v Qt0 Qt1 v Qt0 !Qt2 )

K0 = !(!Qt0 !Qt2 v !Qt0 v Qt1 )

J1 = !(!Qt2 v Qt1 )

K1 = !(!Qt0 !Qt2 v !Qt1 !Qt2 v Qt1 Qt2 v !Qt0 )

J2 = !(!Qt0 !Qt1 Qt2 v Qt0 Qt1 v Qt1 Qt2 v Qt0 !Qt2 )

K2 = !(!Qt0 Qt1 Qt2 !x1 v !Qt0 !Qt1 v !Qt2 )

(тоже что и выше, но в битах):

y1 = !(00--v11--v--0-)

y2 = !(0---v--0-v-1--)

y56 = !(00--v-00-v11--v-11-v1-0-)

J0 = !(0111v00--v-00-v11--v1-0-)

K0 = !(0-0-v0---v-1--)

J1 = !(--0-v-1--)

K1 = !(0-0-v-00-v-11-v0---)

J2 = !(001-v11--v-11-v1-0-)

K2 = !(0110v00--v--0-)

Эффективность минимизации:

1. Эффективность минимизации по количеству входов:

y1 : 4.8

y2 : 10.666666666666666

y56 : 2.8

J0 : 2.0

K0 : 7.0

J1 : 14.0

K1 : 4.0

J2 : 2.6666666666666665

K2 : 3.4285714285714284

2. Эффективность минимизации по количеству блоков:

y1 : 2.2

y2 : 3.0

y56 : 1.4444444444444444

J0 : 1.2222222222222223

K0 : 2.6

J1 : 4.333333333333333

K1 : 1.8571428571428572

J2 : 1.5714285714285714

K2 : 2.2

ФОРМУЛЫ В БАЗИСЕ И, ИЛИ-НЕ

y1 = !(!( ( !( Qt0 v Qt0 ) !( Qt1 v Qt1 ) !( Qt1 v Qt1 ) ) v ( Qt0 Qt1 Qt1 ) ) !( ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) !( ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) )

y2 = !(!( ( !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) ) v ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) !( ( Qt1 Qt1 Qt1 ) v ( Qt1 Qt1 Qt1 ) ) !( ( Qt1 Qt1 Qt1 ) v ( Qt1 Qt1 Qt1 ) ) )

y56 = !(!( ( !( Qt0 v Qt0 ) !( Qt1 v Qt1 ) !( Qt1 v Qt1 ) ) v ( !( Qt1 v Qt1 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) !( ( Qt0 Qt1 Qt1 ) v ( Qt1 Qt2 Qt2 ) ) !( ( Qt0 !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( Qt0 !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) )

J0 = !(!( ( ( !( Qt0 v Qt0 ) Qt1 Qt2 ) x1 x1 ) v ( !( Qt0 v Qt0 ) !( Qt1 v Qt1 ) !( Qt1 v Qt1 ) ) ) !( ( !( Qt1 v Qt1 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( Qt0 Qt1 Qt1 ) ) !( ( Qt0 !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( Qt0 !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) )

K0 = !(!( ( !( Qt0 v Qt0 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) ) ) !( ( Qt1 Qt1 Qt1 ) v ( Qt1 Qt1 Qt1 ) ) !( ( Qt1 Qt1 Qt1 ) v ( Qt1 Qt1 Qt1 ) ) )

J1 = !(( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( Qt1 Qt1 Qt1 ) )

K1 = !(!( ( !( Qt0 v Qt0 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( !( Qt1 v Qt1 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) !( ( Qt1 Qt2 Qt2 ) v ( !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) ) ) !( ( Qt1 Qt2 Qt2 ) v ( !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) !( Qt0 v Qt0 ) ) ) )

J2 = !(!( ( !( Qt0 v Qt0 ) !( Qt1 v Qt1 ) Qt2 ) v ( Qt0 Qt1 Qt1 ) ) !( ( Qt1 Qt2 Qt2 ) v ( Qt0 !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) !( ( Qt1 Qt2 Qt2 ) v ( Qt0 !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) )

K2 = !(!( ( ( !( Qt0 v Qt0 ) Qt1 Qt2 ) !( x1 v x1 ) !( x1 v x1 ) ) v ( !( Qt0 v Qt0 ) !( Qt1 v Qt1 ) !( Qt1 v Qt1 ) ) ) !( ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) !( ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) v ( !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) !( Qt2 v Qt2 ) ) ) )

library ieee; use ieee.std\_logic\_1164.all;

entity APCS\_Lab8 is

port(

Qt0 : in STD\_LOGIC;

Qt1 : in STD\_LOGIC;

Qt2 : in STD\_LOGIC;

x1 : in STD\_LOGIC;

y1 : out STD\_LOGIC;

y2 : out STD\_LOGIC;

y56 : out STD\_LOGIC;

J0 : out STD\_LOGIC;

K0 : out STD\_LOGIC;

J1 : out STD\_LOGIC;

K1 : out STD\_LOGIC;

J2 : out STD\_LOGIC;

K2 : out STD\_LOGIC

);

end APCS\_Lab8;

architecture FULL of APCS\_Lab8 is

component NOR2 is

port (

I0, I1 : in std\_logic;

O : out std\_logic

);

end component;

component AND3

port (

I0, I1, I2 : in std\_logic;

O : out std\_logic

);

end component;

signal A6, A7, A2, A3, A5, A0, A1, A4, A11, A12, A9, A10, A17, A18, A19, A14, A15, A16, A24, A25, A23, A22, A21, A28, A27, A31, A32, A36, A37, A35, A34, A41, A40, A39 : STD\_LOGIC;

begin

Term43: AND3 port map( I0 => A6, I1 => A7, I2 => A7, O => y1);

Term44: AND3 port map( I0 => A11, I1 => A12, I2 => A12, O => y2);

Term45: AND3 port map( I0 => A17, I1 => A18, I2 => A19, O => y56);

Term46: AND3 port map( I0 => A24, I1 => A25, I2 => A19, O => J0);

Term47: AND3 port map( I0 => A28, I1 => A12, I2 => A12, O => K0);

Term48: NOR2 port map( I0 => A5, I1 => A10, O => J1);

Term49: AND3 port map( I0 => A31, I1 => A32, I2 => A32, O => K1);

Term50: AND3 port map( I0 => A36, I1 => A37, I2 => A37, O => J2);

Term51: AND3 port map( I0 => A41, I1 => A7, I2 => A7, O => K2);

end FULL;

**Алгоритмы работы:**

С элементом И никаких проблем не возникает — его можно масштабировать (брать самого от себя) сколько угодно раз. С элементом НЕ-ИЛИ приходится при выделении каждого нового терма ставить над выделенным термом двойное отрицание, но так как инвертора нет в заданной элементной базе, приходится использовать опять элемент НЕ-ИЛИ, у которого на оба входа подан одинаковый сигнал.

Термы построены в виде композитной структуры. Разбиение на термы заданной величины инициируется самым высоким термом, рекурсивно быстро доходит до нижних слоев и начинает подниматься, формируя нужную структуру.

**ВЫВОД:**

В результате проведенной работы я реализовал механизм приведения переключательной функции к элементном базису, у элементов которого заданно количество входов. Для реализации такой функциональности я использовал рекурсивный обход полученной структуры термов функций. Это позволило сравнительно просто привести все функции к нужному базису.

Так же я ознакомился с базой языка VHDL и реализовал сохранение полученной схемы в формат данного языка, с учетом всех его особенностей.