

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра № 705Б

Бортовая автоматика беспилотных космических
и атмосферных летательных аппаратов

Б.Н. ПОПОВ, Е.Э. ЦАЛКОВА, Д.Н. ХАНИН

ПРОЕКТИРОВАНИЕ
ЦИФРОВЫХ СХЕМ И УСТРОЙСТВ
БОРТОВОЙ АВТОМАТИКИ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Под редакцией Б.Н. Попова

Утверждено
на заседании редсовета
факультета № 7
28.01.2016 г.
(Протокол № 1)

Москва
Издательство МАИ
2016

УДК 621.38.049.77

ББК 34.447

П 58

П 58 Попов Б.Н., Цалкова Е.Э., Ханин Д.Н.

Проектирование цифровых схем и устройств бортовой автоматики. Лабораторный практикум / Под ред. Б.Н. Попова. – М.: Изд-во МАИ, 2016. – 92 с.: ил.

Учебное пособие содержит описание четырех практических занятий и двух лабораторных работ. Кратко изложен необходимый теоретический материал.

Дано описание учебной платы DE2 фирмы Altera, используемой при выполнении лабораторных работ. На простом примере изложен маршрут проектирования цифровых устройств на основе программируемых логических интегральных схем в среде Quartus II. Для последовательных логических схем приведены примеры выполнения заданий.

Пособие предназначено для студентов дневной и вечерней форм обучения авиационных вузов, изучающих курсы «Цифровая микроэлектроника», «Аналоговые, дискретные и микропроцессорные устройства систем приводов».

Рецензенты:

кафедра «Системы автоматического управления» Тульского государственного университета» (зав. кафедрой д-р техн. наук, проф. Горячев О.В.);

канд. техн. наук Довгиленко С.В.

ISBN 978-5-4316-0316-7

© Московский авиационный институт
(национальный исследовательский университет), 2016

© ФГУП МОКБ «Марс», 2016

СОДЕРЖАНИЕ

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ	4
Практическое занятие 1.	
Позиционные системы счисления. Перевод чисел из одной позиционной системы счисления в другую	4
Контрольные вопросы и варианты заданий	10
Практическое занятие 2.	
Прямой, обратный и дополнительный коды. Действия над числами со знаком.	17
Контрольные вопросы и варианты заданий	20
Практическое занятие 3.	
Минимизация логических функций. Реализация логических функций в заданном базисе	23
Контрольные вопросы и варианты заданий	25
Практическое занятие 4.	
Минимизация неполностью определенных логических функций с помощью карт Карно	33
Контрольные вопросы и варианты заданий	38
ЛАБОРАТОРНЫЙ ПРАКТИКУМ	48
Маршрут проектирования в среде Quartus II	51
Лабораторная работа 1.	
Практикум по комбинационным схемам	63
Лабораторная работа 2.	
Практикум по последовательностным схемам	69
Приложение 1	81
Приложение 2	86
Библиографический список	89

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Практическое занятие 1.

ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ. ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ ПОЗИЦИОННОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

Цель – знакомство с наиболее распространенными в цифровой технике позиционными системами счисления. Получение и закрепление навыков перевода чисел из одной позиционной системы счисления в другую.

Системой счисления называют совокупность приемов и правил, позволяющих установить взаимно однозначное соответствие между любым числом и его представлением в виде конечного числа символов. Множество символов, используемых для такого представления, называют *цифрами*.

Существуют непозиционные и позиционные системы счисления. В непозиционных системах любое число определяется как некоторая функция от численных значений совокупности цифр, представляющих это число. Если в качестве этой функции используется функция сложения, то систему называют *аддитивной*, если используется функция умножения – *мультипликативной*. Наиболее известным примером непозиционной аддитивной системы является римская система счисления.

В римской системе счисления числа записываются с помощью литер латинского алфавита: I – 1; V – 5; X – 10; L – 50; C – 100; D – 500; M – 1000.

Ряд чисел от 1 до 9:

1 – I; 2 – II = I + I; 3 – III = I + I + I; 4 – IV = V – I; 5 – V;

6 – VI = V + I; 7 – VII = V + I + I; 8 – VIII = V + I + I + I; 9 – IX = X – I.

Десятки (от 10 до 90) обозначаются с помощью комбинации литер X, L и C:

$10 - X$; $20 - XX = X + X$; $30 - XXX = X + X + X$; $40 - XL = L - X$; $50 - L$;
 $60 - LX = L + X$; $70 - LXX = L + X + X$; $80 - LXXX = L + X + X + X$;
 $90 - XC = C - X$.

Сотни обозначаются с помощью комбинации литер C, D и M:

$100 - C$; $200 - CC = C + C$; $300 - CCC = C + C + C$; $400 - CD = D - C$;
 $500 - D$; $600 - DC = D + C$; $700 - DCC = D + C + C$; $800 - DCCC =$
 $= D + C + C + C$; $900 - CM = M - C$; $1000 - M$.

Максимальное число в римской системе счисления:

3999 – MMMCMXCIX.

Пример: год основания Москвы 1147 – MCXLVII.

Непозиционные системы счисления в силу ряда причин (сложные алгоритмы представления чисел и выполнения арифметических операций) в цифровой технике распространения не получили.

Позиционной называют такую систему счисления, в которой положение (позиция) каждой цифры в числе определяет ее значение (вес). То есть одна и та же цифра может принимать различные численные значения в зависимости от местоположения (разряда) этой цифры в совокупности цифр, представляющей заданное число. Позиционные системы разделяют на однородные и смешанные. Во всех разрядах числа, представленного в однородной системе, используются цифры из одного и того же множества. Например, в обычной десятичной системе во всех разрядах любого числа используются цифры из множества 0, 1, ..., 9. В смешанных системах множества цифр различны для разных разрядов. Примером смешанной системы является система измерения времени (различное количество цифр в секундах, минутах, часах, сутках). Наибольшее применение в цифровой технике нашли двоичная, восьмеричная, десятичная и шестнадцатеричная системы счисления. Все перечисленные системы относятся к однородным позиционным системам счисления.

Сформулируем три основных свойства таких систем.

1. Количество цифр, используемых системой, равно ее основанию. Так, например:

- в десятичной системе 10 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;

- в двоичной системе 2 цифры: 0, 1;
- в восьмеричной системе 8 цифр: 0, 1, 2, 3, 4, 5, 6, 7;
- в шестнадцатеричной системе 16 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

2. Старшая цифра на единицу меньше основания. Так, для десятичной это 9, для двоичной – 1, для восьмеричной – 7, для шестнадцатеричной – F (15).

Табл. 1 иллюстрирует соотношение между перечисленными системами счисления.

Таблица 1

Десятичное число	Двоичное число	Восьмеричное число	Шестнадцате- ричное число
10^2 10^1 10^0	2^6 2^5 2^4 2^3 2^2 2^1 2^0	8^2 8^1 8^0	16^1 16^0
0	0	0	0
1	1	1	1
2	1 0	2	2
3	1 1	3	3
4	1 0 0	4	4
5	1 0 1	5	5
6	1 1 0	6	6
7	1 1 1	7	7
8	1 0 0 0	1 0	8
9	1 0 0 1	1 1	9
1 0	1 0 1 0	1 2	A
1 1	1 0 1 1	1 3	B
1 2	1 1 0 0	1 4	C
1 3	1 1 0 1	1 5	D
1 4	1 1 1 0	1 6	E
1 5	1 1 1 1	1 7	F
1 6	1 0 0 0 0	2 0	1 0
.....			
3 1	1 1 1 1 1	3 7	1 F
3 2	1 0 0 0 0 0	4 0	2 0
.....			
6 4	1 0 0 0 0 0 0	1 0 0	4 0
.....			
1 2 6	1 1 1 1 1 1 0	1 7 6	7 E
1 2 7	1 1 1 1 1 1 1	1 7 7	7 F

3. Каждая цифра числа умножается на основание в степени, значение которой определяется положением цифры, т.е. в такой системе любое число можно представить в виде

$$X = x_n k^n + x_{n-1} k^{n-1} + \dots + x_1 k^1 + x_0 k^0 + x_{-1} k^{-1} + \dots + x_{-m} k^{-m} = \sum_{i=-m}^n x_i k^i, \quad (1)$$

где k – основание системы счисления ($k=2, 3, \dots$); x_i – цифры i -го разряда представления числа в системе с основанием k , $x_i = [0, 1, \dots, k-1]$.

Величину k^i принято называть весом i -го разряда. Поскольку значение k известно заранее, то выражение (1) обычно записывают в упрощенной форме:

$$X = x_n x_{n-1} \dots x_1 x_0, x_{-1} \dots x_{-m}. \quad (2)$$

В выражении (2) запятая отделяет целую часть числа от дробной, а значение цифры i -го разряда в k раз больше значения такой же цифры в $(i-1)$ -м разряде. Такую систему называют системой с естественным порядком весов. Существуют также системы с искусственным порядком весов, для которых указанное соотношение одинаковых цифр в соседних разрядах не является обязательным.

В качестве примера представим двоичное число 11010101, восьмеричное число 325 и шестнадцатеричное число B9D, записанные в форме (2), в соответствии с (1):

$$11010101_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0;$$

$$325_8 = 3 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0;$$

$$B9D_{16} = B \cdot 16^2 + 9 \cdot 16^1 + D \cdot 16^0.$$

Нижний индекс здесь и далее указывает на основание позиционной системы счисления.

Перевод чисел из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную

Перевод чисел из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную основан на последовательном делении исходного числа на основание той системы, в которую осуществляется перевод. В результате каждого деления образуется про-

межуточное частное и остаток. Процесс деления продолжается до тех пор, пока промежуточное частное не станет меньше основания. Старшим разрядом искомого числа является результат последнего деления, младшим – остаток от первого деления. Остатки от деления выписываются в обратном порядке. В качестве примера переведем десятичное число 157 в двоичную, восьмеричную и шестнадцатеричную системы счисления.

	остаток
$157 : 2 = 78$	$1 \leftarrow$ младший разряд
$78 : 2 = 39$	0
$39 : 2 = 19$	1
$19 : 2 = 9$	1
$9 : 2 = 4$	1
$4 : 2 = 2$	0
$2 : 2 = 1$	$0 \leftarrow$ второй по старшинству разряд
$\uparrow$ старший разряд	

Таким образом, можно записать: $157_{10} = 10011101_2$

	остаток
$157 : 8 = 19$	$5 \leftarrow$ младший разряд
$19 : 8 = 2$	$3 \leftarrow$ второй по старшинству разряд
$\uparrow$ старший разряд	

Таким образом, можно записать: $157_{10} = 235_8$

	остаток
$157 : 16 = 9$	$13 \leftarrow$ младший разряд
$\uparrow$ старший разряд	

Отметим, что если промежуточное частное либо остаток (как в приведенном примере) окажутся больше десятичной цифры 9, они заменяются соответствующей шестнадцатеричной цифрой ($A \dots F$). Таким образом, можно записать: $157_{10} = 9D_{16}$.

Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную

Для перевода чисел из двоичной системы счисления в восьмеричную либо шестнадцатеричную надо исходное двоичное число, начиная с младшего (правого) разряда, разбить на триады (по три двоичных разряда) либо тетрады (по четыре двоичных разряда). А затем каждую триаду (тетраду) заменить соответствующей восьмеричной (шестнадцатеричной) цифрой. При этом старшая триада (тетрада) может оказаться неполной. В этом случае она дополняется слева незначащими нулями.

В качестве примера переведем двоичное число 10011101_2 в восьмеричное и шестнадцатеричное:

$$10011101_2 = 010\ 011\ 101_2 = 235_8;$$

$$10011101_2 = 1001\ 1101_2 = 9D_{16}.$$

Перевод чисел из восьмеричной системы счисления в шестнадцатеричную и обратно

Перевод чисел из восьмеричной системы счисления в шестнадцатеричную и обратно осуществляется через двоичную систему счисления. В качестве примера переведем числа 235_8 и $9D_{16}$ соответственно в шестнадцатеричную и восьмеричную:

$$235_8 = 010\ 011\ 101_2 = 10011101_2 = 1001\ 1101_2 = 9D_{16};$$

$$9D_{16} = 1001\ 1101_2 = 10011101_2 = 010\ 011\ 101_2 = 235_8.$$

Перевод чисел из двоичной, восьмеричной и шестнадцатеричной систем счисления в десятичную

Перевод чисел из двоичной, восьмеричной и шестнадцатеричной систем счисления в десятичную основан на использовании третьего свойства позиционных систем счисления и состоит из двух этапов:

- 1) соответствующее двоичное, восьмеричное или шестнадцатеричное число представляется в форме (1);
- 2) правая часть выражения суммируется:

$$11010101_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = \\ = 128 + 0 + 0 + 16 + 8 + 4 + 0 + 1 = 157_{10};$$

$$235_8 = 2 \cdot 8^2 + 3 \cdot 8^1 + 5 \cdot 8^0 = 128 + 24 + 5 = 157_{10};$$

$$9D_{16} = 9 \cdot 16^1 + D \cdot 16^0 = 9 \cdot 16^1 + 13 \cdot 16^0 = 144 + 13 = 157_{10}.$$

Контрольные вопросы

1. Что называется системой счисления?
2. В чем различие позиционных и непозиционных систем счисления?
3. В чем различие однородных и смешанных систем счисления?
4. Напишите первые десять чисел (цифр) для троичной, шестеричной, двенадцатеричной систем счисления.
5. Приведите примеры однородных и неоднородных систем счисления.
6. В чем различие аддитивных и мультипликативных систем счисления?

Варианты заданий

Задание: выполнить пять задач по переводу чисел из одной позиционной системы счисления в другую.

Вариант № 1

1. Перевести десятичное число **15**₁₀ в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число **123**₈ в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число **11001100**₂ в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число **AB**₁₆ в восьмеричное, двоичное, десятичное.
5. Перевести двоичное число **100010010**₂ в десятичное, восьмеричное, шестнадцатеричное.

Вариант № 2

1. Перевести десятичное число 144_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 134_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 10101010_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A18_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести двоичное число 1010101101_2 в восьмеричное, шестнадцатеричное, десятичное.

Вариант № 3

1. Перевести десятичное число 16_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 435_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 11000011_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A9_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести десятичное число 20_{10} в восьмеричное, шестнадцатеричное, двоичное.

Вариант № 4

1. Перевести двоичное число 1010101101_2 в восьмеричное, шестнадцатеричное, десятичное.
2. Перевести десятичное число 18_{10} в восьмеричное, шестнадцатеричное, двоичное.
3. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.
4. Перевести шестнадцатеричное число $A9_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести двоичное число 11000011_2 в восьмеричное, шестнадцатеричное, десятичное.

Вариант № 5

1. Перевести двоичное число 11001100_2 в восьмеричное, шестнадцатеричное, десятичное.
2. Перевести десятичное число 20_{10} в восьмеричное, шестнадцатеричное, двоичное.
3. Перевести восьмеричное число 435_8 в десятичное, двоичное, шестнадцатеричное.
4. Перевести шестнадцатеричное число $A12_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести двоичное число 100010010_2 в десятичное, восьмеричное, шестнадцатеричное.

Вариант № 6

1. Перевести десятичное число 15_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 11001100_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A9_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести десятичное число 144_{10} в восьмеричное, шестнадцатеричное, двоичное.

Вариант № 7

1. Перевести десятичное число 16_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 134_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 1010101101_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число AB_{16} в восьмеричное, двоичное, десятичное.
5. Перевести десятичное число 80_{10} в шестнадцатеричное, восьмеричное, двоичное.

Вариант № 8

1. Перевести двоичное число 100010010_2 в десятичное, восьмеричное, шестнадцатеричное.
2. Перевести шестнадцатеричное число $A18_{16}$ в восьмеричное, двоичное, десятичное.
3. Перевести десятичное число 435_{10} в восьмеричное, шестнадцатеричное, двоичное.
4. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.
5. Перевести двоичное число 10101010_2 в восьмеричное, шестнадцатеричное, десятичное.

Вариант № 9

1. Перевести десятичное число 40_{10} в шестнадцатеричное, восьмеричное, двоичное.
2. Перевести восьмеричное число 721_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 11001100_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A12_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.

Вариант № 10

1. Перевести двоичное число 10101010_2 в восьмеричное, шестнадцатеричное, десятичное.
2. Перевести шестнадцатеричное число $A18_{16}$ в восьмеричное, двоичное, десятичное.
3. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.
4. Перевести двоичное число 11001100_2 в восьмеричное, шестнадцатеричное, десятичное.
5. Перевести десятичное число 40_{10} в шестнадцатеричное, восьмеричное, двоичное.

Вариант № 11

1. Перевести десятичное число 144_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 435_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 1010101101_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A9_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести десятичное число 16_{10} в восьмеричное, шестнадцатеричное, двоичное.

Вариант № 12

1. Перевести десятичное число 18_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 721_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 100010010_2 в десятичное, восьмеричное, шестнадцатеричное.
4. Перевести шестнадцатеричное число $A9_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести шестнадцатеричное число AB_{16} в восьмеричное, двоичное, десятичное.

Вариант № 13

1. Перевести десятичное число 80_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 11001100_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A12_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести восьмеричное число 134_8 в десятичное, двоичное, шестнадцатеричное.

Вариант № 14

1. Перевести десятичное число 20_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 134_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 10101010_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число AB_{16} в восьмеричное, двоичное, десятичное.
5. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.

Вариант № 15

1. Перевести десятичное число 15_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 435_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 11000011_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A9_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести десятичное число 144_{10} в восьмеричное, шестнадцатеричное, двоичное.

Вариант № 16

1. Перевести десятичное число 144_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 721_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 1010101101_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число $A18_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести десятичное число 15_{10} в восьмеричное, шестнадцатеричное, двоичное.

Вариант № 17

1. Перевести десятичное число 16_{10} в восьмеричное, шестнадцатеричное, двоичное.
2. Перевести восьмеричное число 123_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 100010010_2 в десятичное, восьмеричное, шестнадцатеричное.
4. Перевести шестнадцатеричное число $A18_{16}$ в восьмеричное, двоичное, десятичное.
5. Перевести двоичное число 1010101101_2 в восьмеричное, шестнадцатеричное, десятичное.

Вариант № 18

1. Перевести шестнадцатеричное число $A18_{16}$ в восьмеричное, двоичное, десятичное.
2. Перевести восьмеричное число 134_8 в десятичное, двоичное, шестнадцатеричное.
3. Перевести двоичное число 11001100_2 в восьмеричное, шестнадцатеричное, десятичное.
4. Перевести шестнадцатеричное число AB_{16} в восьмеричное, двоичное, десятичное.
5. Перевести десятичное число 80_{10} в восьмеричное, шестнадцатеричное, двоичное.

Практическое занятие 2.

ПРЯМОЙ, ОБРАТНЫЙ И ДОПОЛНИТЕЛЬНЫЙ КОДЫ. ДЕЙСТВИЯ НАД ЧИСЛАМИ СО ЗНАКОМ

Цель – знакомство с различными формами представления чисел со знаком, используемыми в цифровой технике. Получение навыков сложения и вычитания двоичных чисел на примере дополнительного кода.

Все системы счисления допускают использование как положительных, так и отрицательных чисел. Поскольку в вычислительной технике используется двоичная система счисления, то для представления знака числа применяют символы 0 (знак +) и 1 (знак –). Здесь будем рассматривать так называемую форму представления числа с фиксированной точкой. Знаковым при таком представлении считается старший (левый) разряд машинного слова. Напомним, что оставшаяся часть машинного слова называется мантиссой.

Рассмотрим три основных способа представления чисел со знаком: прямой, обратный и дополнительный коды. Значения перечисленных кодов для диапазона (+7...–8) приведены в табл. 2. Знаковый разряд для наглядности отделен десятичной точкой.

Наиболее простым способом представления числа со знаком является прямой двоичный код: в разрядах мантиссы записывается абсолютное значение числа в двоичной системе. Прямому коду присущи два недостатка:

- для реализации операций сложения и вычитания требуются две различные схемы;
- для нуля имеются два представления.

Обратный и дополнительный коды позволяют реализовать операции сложения и вычитания с помощью одной и той же схемы, что обуславливает преимущественное использование этих кодов по сравнению с прямым. Операция вычитания заменяется операцией алгебраического сложения.

Обратный код двоичного положительного числа совпадает с его прямым кодом. Обратный код двоичного отрицательного числа фор-

Таблица 2

Десятичное число	Двоичное число		
	Прямой код	Обратный код	Дополнительный код
+7	0.111	0.111	0.111
+6	0.110	0.110	0.110
+5	0.101	0.101	0.101
+4	0.100	0.100	0.100
+3	0.011	0.011	0.011
+2	0.010	0.010	0.010
+1	0.001	0.001	0.001
+0	0.000	0.000	0.000
-0	1.000	1.111	0.000
-1	1.001	1.110	1.111
-2	1.010	1.101	1.110
-3	1.011	1.100	1.101
-4	1.100	1.011	1.100
-5	1.101	1.010	1.011
-6	1.110	1.001	1.010
-7	1.111	1.000	1.001
-8	—	—	1.000

мируется заменой всех нулей мантиссы соответствующего числа в прямом коде на единицы, а единиц – на нули. Как и в случае прямого кода, недостатком является двойственность в представлении нуля.

Дополнительный код наиболее часто применяется в вычислительной технике. При его использовании исчезает двусмысленность представления нуля. Дополнительный код двоичного положительного числа совпадает с его прямым кодом. Дополнительный код отрицательного числа равен соответствующему обратному коду плюс единица. Прежде чем рассмотреть примеры сложения чисел в дополнительном коде, приведем правила сложения двух одноразрядных двоичных чисел:

$$\begin{array}{r}
 0 \\
 + 0 \\
 \hline
 0
 \end{array}
 \begin{array}{r}
 0 \\
 + 1 \\
 \hline
 1
 \end{array}
 \begin{array}{r}
 1 \\
 + 0 \\
 \hline
 1
 \end{array}
 \begin{array}{r}
 1 \\
 + 1 \\
 \hline
 10
 \end{array}$$

В последнем случае при сложении двух одноразрядных чисел сумма оказалась двухразрядным числом. Если предположить, что суммировались не одноразрядные числа, а i -е разряды двух n -разрядных чисел, то равный единице старший разряд двухразрядной суммы является переносом в следующий по старшинству разряд. В том случае, если в i -х разрядах обоих чисел находились единицы и произошел перенос из предыдущего разряда, имеет место следующая схема сложения:

$$\begin{array}{r} 1 \\ + \\ 1 \\ + \\ 1 \\ \hline 11 \end{array}$$

В качестве примера рассмотрим варианты суммирования двух чисел:

(1)

$$\begin{array}{l} (+9)+(-4)=5 \rightarrow \\ \begin{array}{r|l} 0.1001 \\ + \\ 1.1100 \\ \hline 0.0101_{\text{пр}} \end{array} \end{array}$$

теряется 1

(2)

$$\begin{array}{l} (+9)+(+4)=13 \rightarrow \\ \begin{array}{r|l} 0.1001 \\ + \\ 0.0100 \\ \hline 0.1101_{\text{пр}} \end{array} \end{array}$$

(3)

$$\begin{array}{l} (-9)+(+4)=-5 \rightarrow \\ \begin{array}{r|l} 1.0111 \\ + \\ 0.0100 \\ \hline 1.1011_{\text{доп}} \\ + \\ 1.1111 \\ \hline 1.1010_{\text{обр}} \\ 1.0101_{\text{пр}} \end{array} \end{array}$$

теряется 1

(4)

$$\begin{array}{l} (-9)+(-4)=-13 \rightarrow \\ \begin{array}{r|l} 1.0111 \\ + \\ 1.1100 \\ \hline 1.0011_{\text{доп}} \\ + \\ 1.1111 \\ \hline 1.0010_{\text{обр}} \\ 1.1101_{\text{пр}} \end{array} \end{array}$$

теряется 1

Для получения правильного ответа необходимо, чтобы число разрядов под операнды выбиралось из условия числа разрядов, необходимых под кодирование результата операции, либо большего из операндов.

В том случае, если результат отрицательный (в знаковом разряде 1), полученное число следует представить в прямом коде. Для этого из мантиссы результата надо вычесть единицу (при этом результат получается в обратном коде). Вычитание единицы можно заменить сложением с «-1». Затем все разряды мантиссы нужно проинвертировать (при этом результат получается в прямом коде). Этому варианту соответствуют примеры (3) и (4).

Контрольные вопросы

1. В чем преимущество дополнительного кода по сравнению с обратным?
2. В чем преимущество обратного кода по сравнению с прямым?
3. Почему число разрядов, необходимое для операндов, надо выбирать исходя из числа разрядов, необходимого для представления предполагаемого результата или большего из операндов? Приведите пример.

Варианты заданий

Задание: сложить числа в дополнительном коде, результат дать в прямом коде.

Вариант № 1

1. -5 и $+7$;
2. $+5$ и -7 ;
3. $+7$ и $+7$;
4. $+6$ и -6 ;
5. -4 и -5 .

Вариант № 2

1. $+11$ и -9 ;
2. -22 и $+7$;
3. -5 и -3 ;
4. $+5$ и -5 ;
5. $+6$ и $+7$.

Вариант № 3

1. $+7$ и -3 ;
2. $+5$ и -8 ;
3. -4 и -5 ;
4. $+5$ и -5 ;
5. $+7$ и $+7$.

Вариант № 4

1. -5 и $+7$;
2. $+5$ и -8 ;
3. $+5$ и $+3$;
4. $+7$ и -7 ;
5. -7 и -5 .

Вариант № 5

1. $+20$ и -10 ;
2. $+5$ и -6 ;
3. $+5$ и $+7$;
4. $+7$ и -7 ;
5. -5 и $+7$.

Вариант № 7

1. $+19$ и -18 ;
2. $+10$ и -11 ;
3. -5 и -18 ;
4. $+21$ и -21 ;
5. $+5$ и $+7$.

Вариант № 9

1. $+17$ и -8 ;
2. $+16$ и -21 ;
3. -7 и -5 ;
4. $+6$ и -6 ;
5. $+16$ и -21 .

Вариант № 11

1. $+17$ и -8 ;
2. $+16$ и -21 ;
3. -4 и -5 ;
4. $+7$ и -7 ;
5. -5 и -3 .

Вариант № 13

1. -5 и $+15$;
2. $+5$ и -7 ;
3. $+5$ и $+7$;
4. $+21$ и -21 ;
5. $+19$ и -18 .

Вариант № 6

1. $+7$ и -3 ;
2. $+5$ и -7 ;
3. -4 и -8 ;
4. $+21$ и -21 ;
5. $+11$ и -9 .

Вариант № 8

1. $+20$ и -10 ;
2. $+5$ и -6 ;
3. $+17$ и $+16$;
4. $+6$ и -6 ;
5. $+5$ и -5 .

Вариант № 10

1. $+19$ и -18 ;
2. $+10$ и -11 ;
3. $+7$ и $+7$;
4. $+5$ и -5 ;
5. $+5$ и -7 .

Вариант № 12

1. $+19$ и -18 ;
2. $+16$ и -21 ;
3. -5 и -18 ;
4. $+21$ и -21 ;
5. $+7$ и -3 .

Вариант № 14

1. $+17$ и -8 ;
2. -22 и $+7$;
3. -4 и -8 ;
4. $+21$ и -21 ;
5. $+5$ и -6 .

Вариант № 15

1. -5 и $+7$;
2. $+5$ и -8 ;
3. $+6$ и $+7$;
4. $+6$ и -6 ;
5. -4 и -8 .

Вариант № 17

1. $+7$ и -3 ;
2. $+5$ и -6 ;
3. -7 и -5 ;
4. $+5$ и -5 ;
5. $+17$ и $+16$.

Вариант № 16

1. $+11$ и -9 ;
2. $+5$ и -8 ;
3. $+24$ и $+8$;
4. $+7$ и -7 ;
5. $+5$ и $+3$.

Вариант № 18

1. -5 и $+7$;
2. $+5$ и -6 ;
3. $+7$ и $+7$;
4. $+5$ и -5 ;
5. -5 и -18 .

Практическое занятие 3.

МИНИМИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ.

РЕАЛИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ В ЗАДАННОМ БАЗИСЕ

Цель – получение навыков минимизации логических функций с использованием операций склеивания, поглощения и неполного склеивания. Получение навыков реализации логических функций в различных базисах.

Минимизация логических функций

Минимизация логических функций обеспечивает уменьшение веса, габаритов, потребляемой мощности, понижение стоимости электронных изделий.

Все методы минимизации в той или иной форме используют основные соотношения булевой алгебры. Напомним их.

Двойное отрицание (отрицание отрицания) переменной равно самой переменной: $\overline{\overline{x}} = x$. Остальные 20 соотношений имеют вид:

$$x \vee 1 = 1; \quad x \vee 0 = x; \quad x \vee x = x; \quad x \vee \overline{x} = 1;$$

$$x \cdot 1 = x; \quad x \cdot 0 = 0; \quad x \cdot x = x; \quad x \cdot \overline{x} = 0;$$

$$\overline{x \vee 1} = 0; \quad \overline{x \vee 0} = \overline{x}; \quad \overline{x \vee x} = \overline{x}; \quad \overline{x \vee \overline{x}} = 0;$$

$$\overline{x \cdot 1} = \overline{x}; \quad \overline{x \cdot 0} = 1; \quad \overline{x \cdot x} = \overline{x}; \quad \overline{x \cdot \overline{x}} = 1;$$

$$x \oplus 1 = \overline{x}; \quad x \oplus 0 = x; \quad x \oplus x = 0; \quad x \oplus \overline{x} = 1.$$

Большинство известных методов минимизации основано на применении следующих трех операций, которые в свою очередь базируются на основных соотношениях булевой алгебры [1].

1. Операция склеивания:

$$p \cdot x \vee p \cdot \overline{x} = p, \text{ где } p - \text{произвольная конъюнкция.}$$

2. Операция поглощения в общем виде записывается как

$$p \cdot x \vee p = p.$$

3. Операции неполного склеивания:

$$p \cdot x \vee p \cdot \overline{x} = p \vee p \cdot x \vee p \cdot \overline{x};$$

$$p \cdot x \vee p \cdot \overline{x} = p \vee p \cdot x;$$

$$p \cdot x \vee p \cdot \overline{x} = p \vee p \cdot \overline{x}.$$

Из операции неполного склеивания следует, что каждая из склеиваемых конъюнкций может быть использована для склеивания с другими соседними конъюнкциями.

Практический интерес для минимизации представляют два следствия из свойства дистрибутивности:

$$x_1 \vee \bar{x}_1 \cdot x_2 = x_1 \vee x_2;$$

$$\bar{x}_1 \vee x_1 \cdot x_2 = \bar{x}_1 \vee x_2.$$

Результатом минимизации логической функции является выражение, дальнейшее упрощение которого невозможно. Оно называется *табулковой дизъюнктивной нормальной формой* (ТДНФ).

При использовании операций минимизации может получиться несколько различных ТДНФ. Самая простая из всех возможных ТДНФ называется *минимальной дизъюнктивной нормальной формой* (МДНФ).

Номенклатура интегральных микросхем в каждой из серий элементов такова, что реализация любого устройства является многоальтернативной задачей. В состав любой серии наряду со схемами средней степени интеграции элементов входят простейшие элементы, реализующие функции 2И-НЕ и 2ИЛИ-НЕ, образующие минимальный базис, так как любую сколь угодно сложную логическую функцию или цифровое устройство можно реализовать с помощью ИС одного типа – двухвходовых элементов И-НЕ (ИЛИ-НЕ).

Приведение логической функции к заданному базису осуществляется с помощью формул де Моргана, которые позволяют выражать дизъюнкцию через конъюнкцию и отрицание, а конъюнкцию – через дизъюнкцию и отрицание:

$$\overline{x_2 \vee x_1} = \bar{x}_2 \cdot \bar{x}_1; x_2 \vee x_1 = \overline{\bar{x}_2 \cdot \bar{x}_1};$$

$$\overline{x_2 \cdot x_1} = \bar{x}_2 \vee \bar{x}_1; x_2 \cdot x_1 = \overline{\bar{x}_2 \vee \bar{x}_1}.$$

Формулы де Моргана являются справедливыми и для функций n переменных. Например, $\overline{x_3 \vee x_2 \vee x_1 \vee x_0} = \bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0$.

Выполнение каждого задания состоит из трех последовательных этапов.

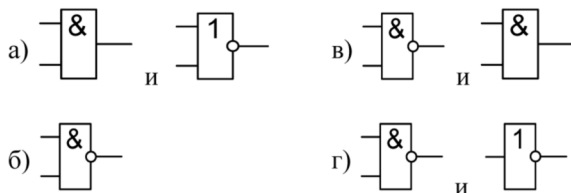
1. Минимизация (получение МДНФ) с помощью операций склеивания и поглощения.

2. Приведение полученной функции к заданному базису с использованием формул де Моргана. Другими словами, если необходимо реализовать некоторую логическую функцию, заданную в виде МДНФ, с помощью элементов И-НЕ (ИЛИ-НЕ), предварительно следует с помощью формул де Моргана привести функцию к виду, в котором исключена ДИЗЬЮНКЦИЯ (КОНЬЮНКЦИЯ).

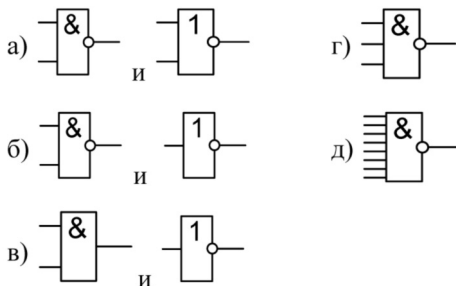
3. Реализация. В результате минимизации может получиться вырожденное выражение, например $f=x_3$. В этом случае следует написать: «Выражение вырожденное и реализации не подлежит».

Контрольные вопросы

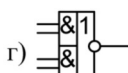
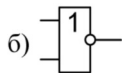
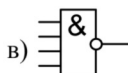
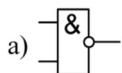
1. Какие две конъюнкции называют соседними?
2. Докажите одну из формул де Моргана.
3. Реализуйте функцию $f=\overline{x_1 \cdot x_2 \vee x_3 \cdot x_4}$, используя элементы:



4. Реализуйте функцию $f=\overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4}$, используя элементы:



5. Реализуйте функцию $f=\overline{x}$, используя элементы:



6. Приведите варианты реализации функций 2И, 3И, 4И, 2ИЛИ, 3ИЛИ, 4ИЛИ, используя элементы из пп. 3, 4 и 5.

Варианты заданий

Задание: минимизировать пять логических функций и затем реализовать их в заданном базисе.

Вариант № 1

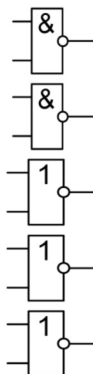
$$1. f(x_4 \dots x_1) = x_4 x_3 \vee x_2 x_1 \vee x_3 x_2 x_1 \vee x_4 x_3 x_2 x_1$$

$$2. f(x_4 \dots x_1) = x_4 \vee x_4 x_3 x_2 x_1 \vee x_4 \bar{x}_2 x_1 \vee x_4 \bar{x}_1$$

$$3. f(x_5 \dots x_1) = x_4 x_3 \bar{x}_3 x_2 x_1 \vee x_5 x_4 x_3$$

$$4. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2$$

$$5. f(x_5 \dots x_0) = x_5 x_4 x_3 x_2 x_1 x_0 \vee x_5 x_4 x_3 \bar{x}_3$$



Вариант № 2

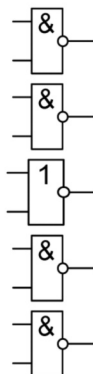
$$1. f(x_4 \dots x_1) = x_4 \vee x_3 x_2 \vee x_3 x_2 x_1$$

$$2. f(x_4 \dots x_1) = x_4 x_3 \vee x_4 x_3 x_2 \vee x_4 x_3 x_2 x_1 \vee x_4$$

$$3. f(x_4 \dots x_1) = x_4 x_3 \bar{x}_2 x_1 \vee x_4 x_3 x_2 x_1 \vee x_3 x_1$$

$$4. f(x_5 \dots x_0) = x_5 \vee \bar{x}_5 x_4 x_3 \vee x_4 x_3 x_2 \vee x_3 x_2 x_1 x_0$$

$$5. f(x_4 \dots x_1) = x_4 x_3 \vee x_2 x_1 \vee x_3 x_2 x_1 \vee x_4 x_3 x_2 x_1$$



Вариант № 3

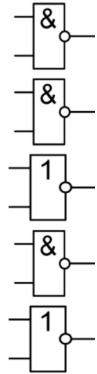
$$1. f(x_4 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_4 x_3 x_2 \bar{x}_1 \vee x_3 x_2$$

$$2. f(x_5 \dots x_0) = x_5 x_4 \bar{x}_3 \bar{x}_2 x_1 \vee x_4 \bar{x}_3 x_2 x_1 \vee x_2 x_1 \vee x_2$$

$$3. f(x_4 \dots x_1) = \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 \vee x_4 x_3 \bar{x}_2 \bar{x}_1$$

$$4. f(x_5 \dots x_1) = \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee \bar{x}_4 x_3 x_1 \vee x_3 \vee \bar{x}_3$$

$$5. f(x_5 \dots x_1) = x_4 x_3 \bar{x}_3 x_2 x_1 \vee x_5 x_4 x_3$$

**Вариант № 4**

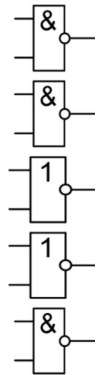
$$1. f(x_4 \dots x_1) = x_4 \bar{x}_3 x_2 x_1 \vee x_4 \bar{x}_3 x_2 x_1 \vee x_3 x_2 x_1 \vee x_4 x_3 x_2 x_1$$

$$2. f(x_5 \dots x_1) = x_5 x_4 \bar{x}_3 x_3 x_2 x_1 \vee x_4 \bar{x}_3 \bar{x}_2 x_1 \vee x_1$$

$$3. f(x_5 \dots x_1) = x_5 x_4 x_3 x_2 \vee x_3 x_2 \vee x_2 \vee x_3 \vee \bar{x}_4 \bar{x}_3$$

$$4. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2 \vee x_4 x_3 \bar{x}_2$$

$$5. f(x_4 \dots x_1) = x_4 \vee x_3 x_2 \vee x_3 x_2 x_1$$

**Вариант № 5**

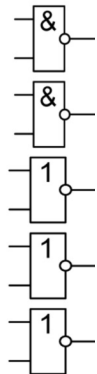
$$1. f(x_4 \dots x_1) = x_4 x_2 x_1 \vee x_4 x_3 x_1 \vee x_3 x_1 \vee \bar{x}_4 x_2 x_1$$

$$2. f(x_5 \dots x_0) = \bar{x}_5 \bar{x}_4 x_1 \vee \bar{x}_5 \vee \bar{x}_4 \vee x_4$$

$$3. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2 \vee x_4 x_3 \bar{x}_2$$

$$4. f(x_5 \dots x_0) = x_5 \vee x_4 \vee x_3 \vee x_2 \vee x_1 \vee \bar{x}_0$$

$$5. f(x_4 \dots x_1) = x_4 x_3 \bar{x}_2 x_1 \vee x_4 x_3 x_2 x_1 \vee x_3 x_1$$



Вариант № 6

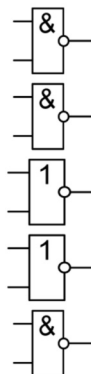
$$1. f(x_4 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_4 x_3 x_2 \bar{x}_1 \vee x_3 x_2$$

$$2. f(x_5 \dots x_1) = \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee \bar{x}_4 x_3 x_1 \vee x_3 \vee \bar{x}_3$$

$$3. f(x_5 \dots x_1) = x_5 x_4 \bar{x}_3 \bar{x}_2 x_1 \vee \bar{x}_4 \bar{x}_3 \bar{x}_2 \vee x_3 x_2 x_1 \vee x_1$$

$$4. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2$$

$$5. f(x_4 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_4 x_3 x_2 \bar{x}_1 \vee x_3 x_2$$



Вариант № 7

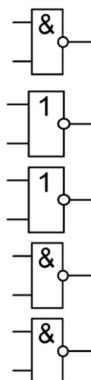
$$1. f(x_4 \dots x_1) = x_1 x_2 x_3 \bar{x}_4 \vee x_2 x_3 \bar{x}_4 \vee x_1 x_2 \vee x_2 x_3 x_4$$

$$2. f(x_5 \dots x_0) = x_5 x_4 x_3 \vee x_4 x_3 x_2 \vee x_3 x_2 x_1 \vee x_3$$

$$3. f(x_5 \dots x_1) = x_5 \bar{x}_4 \bar{x}_3 x_2 \vee \bar{x}_4 \bar{x}_3 \vee x_5 x_2 \vee \bar{x}_5 \bar{x}_2$$

$$4. f(x_5 \dots x_0) = x_5 \vee \bar{x}_5 x_4 x_3 \vee x_4 x_3 x_2 \vee x_3 x_2 x_1 x_0$$

$$5. f(x_4 \dots x_1) = x_4 \bar{x}_3 x_2 x_1 \vee x_4 \bar{x}_3 x_2 x_1 \vee x_3 x_2 x_1 \vee x_4 x_3 x_2 x_1$$



Вариант № 8

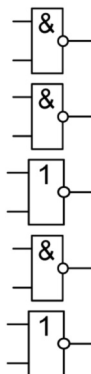
$$1. f(x_5 \dots x_0) = \bar{x}_5 x_4 \bar{x}_3 x_2 \vee x_3 \bar{x}_2 x_1 \vee x_2 x_1 x_0 \vee \bar{x}_2$$

$$2. f(x_4 \dots x_1) = x_4 \vee x_4 x_3 x_2 x_1 \vee x_4 \bar{x}_2 x_1 \vee x_4 \bar{x}_1$$

$$3. f(x_5 \dots x_0) = \bar{x}_5 \bar{x}_4 \bar{x}_3 x_2 x_1 x_0 \vee \bar{x}_4 x_1 \vee \bar{x}_5 x_0$$

$$4. f(x_5 \dots x_1) = \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee \bar{x}_4 x_3 x_1 \vee x_3 \vee \bar{x}_3$$

$$5. f(x_4 \dots x_1) = \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 \vee x_4 x_3 \bar{x}_2 \bar{x}_1$$



Вариант № 9

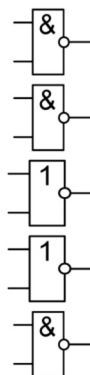
$$1. f(x_5 \dots x_0) = x_5 x_4 x_3 x_2 \vee x_4 x_3 x_2 \vee x_2 x_1 x_0 \vee \bar{x}_5$$

$$2. f(x_4 \dots x_1) = x_4 x_3 \vee x_4 x_3 x_2 \vee x_4 x_3 x_2 x_1 \vee x_4$$

$$3. f(x_5 \dots x_1) = x_5 x_1 \vee x_4 x_2 \vee x_3 \bar{x}_3 \vee \bar{x}_5 \bar{x}_4 \bar{x}_3$$

$$4. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2 \vee x_4 x_3 \bar{x}_2$$

$$5. f(x_4 \dots x_1) = x_4 x_2 x_1 \vee x_4 x_3 x_1 \vee x_3 x_1 \vee \bar{x}_4 x_2 x_1$$



Вариант № 10

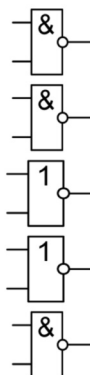
$$1. f(x_5 \dots x_0) = \bar{x}_5 \vee \bar{x}_4 \vee \bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1 \vee \bar{x}_1 x_0$$

$$2. f(x_5 \dots x_0) = x_5 x_4 \bar{x}_3 x_2 x_1 \vee x_4 \bar{x}_3 x_2 x_1 \vee x_2 x_1 \vee x_2$$

$$3. f(x_5 \dots x_0) = x_4 \vee x_3 \vee x_5 x_4 \bar{x}_3 \bar{x}_2 x_1 x_0 \vee \bar{x}_3 \bar{x}_2$$

$$4. f(x_5 \dots x_0) = x_5 \vee x_4 \vee x_3 \vee x_2 \vee x_1 \vee \bar{x}_0$$

$$5. f(x_4 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_4 x_3 x_2 \bar{x}_1 \vee x_3 x_2$$



Вариант № 11

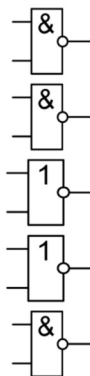
$$1. f(x_5 \dots x_2) = x_5 x_4 x_3 \vee \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee x_4 \vee \bar{x}_3 \vee \bar{x}_4 x_2$$

$$2. f(x_5 \dots x_1) = x_5 x_4 \bar{x}_3 x_3 x_2 x_1 \vee x_4 \bar{x}_3 \bar{x}_2 x_1 \vee x_1$$

$$3. f(x_5 \dots x_0) = \bar{x}_5 x_4 \bar{x}_3 x_2 \bar{x}_1 x_0 \vee \bar{x}_3 x_2 \vee \bar{x}_2$$

$$4. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2$$

$$5. f(x_4 \dots x_1) = x_1 x_2 x_3 \bar{x}_4 \vee x_2 x_3 \bar{x}_4 \vee x_1 x_2 \vee x_2 x_3 x_4$$



Вариант № 12

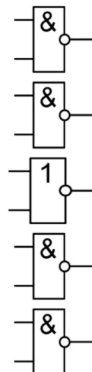
$$1. f(x_5 \dots x_1) = x_5 x_4 \bar{x}_3 x_2 x_1 \vee \bar{x}_5 \bar{x}_4 \vee \bar{x}_5 \bar{x}_4 \bar{x}_3 x_2$$

$$2. f(x_5 \dots x_0) = \bar{x}_5 \bar{x}_4 x_1 \vee \bar{x}_5 \vee \bar{x}_4 \vee x_4$$

$$3. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_4 x_2 x_1 \vee x_4 x_2 x_1 \vee \bar{x}_4 \bar{x}_2 \bar{x}_1$$

$$4. f(x_5 \dots x_0) = x_5 \vee \bar{x}_5 x_4 x_3 \vee x_4 x_3 x_2 \vee x_3 x_2 x_1 x_0$$

$$5. f(x_5 \dots x_0) = \bar{x}_5 x_4 \bar{x}_3 x_2 \vee x_3 \bar{x}_2 x_1 \vee x_2 x_1 x_0 \vee \bar{x}_2$$



Вариант № 13

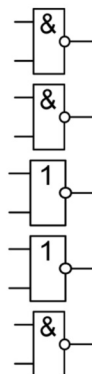
$$1. f(x_5 \dots x_0) = x_5 x_4 x_3 \bar{x}_2 x_2 \vee x_5 \vee x_4 \vee \bar{x}_2$$

$$2. f(x_5 \dots x_1) = \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee \bar{x}_4 x_3 x_1 \vee x_3 \vee \bar{x}_3$$

$$3. f(x_5 \dots x_1) = \bar{x}_5 \bar{x}_4 x_3 \bar{x}_3 x_2 x_1 \vee x_3 \vee x_4 \vee x_4 x_3 x_2 x_1$$

$$4. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2 \vee x_4 x_3 \bar{x}_2$$

$$5. f(x_5 \dots x_0) = x_5 x_4 x_3 x_2 \vee x_4 x_3 x_2 \vee x_2 x_1 x_0 \vee \bar{x}_5$$



Вариант № 14

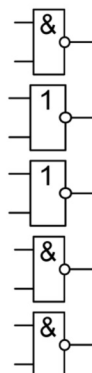
$$1. f(x_5 \dots x_0) = x_5 x_4 \vee x_4 x_3 \vee x_3 x_2 \vee x_5$$

$$2. f(x_5 \dots x_0) = x_5 x_4 x_3 \vee x_4 x_3 x_2 \vee x_3 x_2 x_1 \vee x_3$$

$$3. f(x_5 \dots x_0) = x_5 x_4 x_3 x_2 x_1 x_0 \vee x_5 x_4 x_3 \bar{x}_3$$

$$4. f(x_5 \dots x_1) = \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee \bar{x}_4 x_3 x_1 \vee x_3 \vee \bar{x}_3$$

$$5. f(x_5 \dots x_0) = \bar{x}_5 \vee \bar{x}_4 \vee \bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1 \vee \bar{x}_1 x_0$$



Вариант № 15

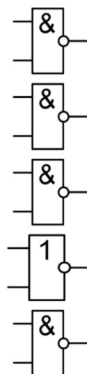
$$1. f(x_5 \dots x_0) = x_5 x_4 x_3 \bar{x}_2 \bar{x}_1 x_0 \vee x_4 x_3 \vee \bar{x}_2 \bar{x}_1 x_0 \vee \bar{x}_4 \vee \bar{x}_3$$

$$2. f(x_4 \dots x_1) = x_4 \vee x_4 x_3 x_2 x_1 \vee x_4 \bar{x}_2 x_1 \vee x_4 \bar{x}_1$$

$$3. f(x_5 \dots x_1) = x_4 x_3 \bar{x}_3 x_2 x_1 \vee x_5 x_4 x_3$$

$$4. f(x_5 \dots x_0) = x_5 \vee x_4 \vee x_3 \vee x_2 \vee x_1 \vee \bar{x}_0$$

$$5. f(x_5 \dots x_2) = x_5 x_4 x_3 \vee \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee x_4 \vee \bar{x}_3 \vee \bar{x}_4 x_2$$



Вариант № 16

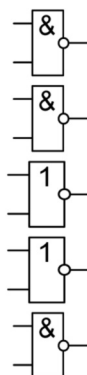
$$1. f(x_5 \dots x_0) = x_5 x_4 \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0 \vee x_4 \vee x_3$$

$$2. f(x_4 \dots x_1) = x_4 x_3 \vee x_4 x_3 x_2 \vee x_4 x_3 x_2 x_1 \vee x_4$$

$$3. f(x_4 \dots x_1) = x_4 x_3 \bar{x}_2 x_1 \vee x_4 x_3 x_2 x_1 \vee x_3 x_1$$

$$4. f(x_5 \dots x_1) = x_4 x_3 x_2 x_1 \vee x_5 x_4 x_3 x_2$$

$$5. f(x_5 \dots x_1) = x_5 x_4 \bar{x}_3 x_2 x_1 \vee \bar{x}_5 \bar{x}_4 \vee \bar{x}_5 \bar{x}_4 \bar{x}_3 x_2$$



Вариант № 17

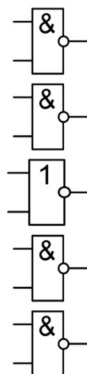
$$1. f(x_4 \dots x_1) = x_4 x_3 \vee x_2 x_1 \vee x_3 x_2 x_1 \vee x_4 x_3 x_2 x_1$$

$$2. f(x_5 \dots x_0) = x_5 x_4 \bar{x}_3 x_2 x_1 \vee x_4 \bar{x}_3 x_2 x_1 \vee x_2 x_1 \vee x_2$$

$$3. f(x_4 \dots x_1) = \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 \vee x_4 x_3 \bar{x}_2 \bar{x}_1$$

$$4. f(x_5 \dots x_0) = x_5 \vee \bar{x}_5 x_4 x_3 \vee x_4 x_3 x_2 \vee x_3 x_2 x_1 x_0$$

$$5. f(x_5 \dots x_0) = x_5 x_4 x_3 \bar{x}_2 x_2 \vee x_5 \vee x_4 \vee \bar{x}_2$$



Вариант № 18

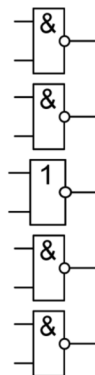
$$1. f(x_4 \dots x_1) = x_4 \vee x_3 x_2 \vee x_3 x_2 x_1$$

$$2. f(x_5 \dots x_1) = x_5 x_4 \bar{x}_3 x_3 x_2 x_1 \vee x_4 \bar{x}_3 \bar{x}_2 x_1 \vee x_1$$

$$3. f(x_5 \dots x_1) = x_5 x_4 x_3 x_2 \vee x_3 x_2 \vee x_2 \vee x_3 \vee \bar{x}_4 \bar{x}_3$$

$$4. f(x_5 \dots x_1) = \bar{x}_5 \bar{x}_4 \bar{x}_3 \vee \bar{x}_4 x_3 x_1 \vee x_3 \vee \bar{x}_3$$

$$5. f(x_5 \dots x_0) = x_5 x_4 \vee x_4 x_3 \vee x_3 x_2 \vee x_5$$



Практическое занятие 4.

МИНИМИЗАЦИЯ НЕПОЛНОСТЬЮ ОПРЕДЕЛЕННЫХ ЛОГИЧЕСКИХ ФУНКЦИЙ С ПОМОЩЬЮ КАРТ КАРНО

Цель – получение навыков минимизации неполностью определенных логических функций с помощью карт Карно.

Карты Карно реализуют графический метод представления и упрощения булевых выражений, с их помощью можно быстро и просто отыскивать МДНФ.

Карта Карно – это таблица специального вида, имеющая 2^n клеток, где n – число логических переменных. Каждой клетке карты Карно соответствует уникальный минтерм [6], десятичный эквивалент которого является номером (адресом) клетки. При переходе от каждой клетки к соседней изменяется состояние только одной переменной в минтерме (рис. 1). Напомним, что в карте Карно n переменных имеется $2n$ зон. *Зоной* называют часть карты Карно, в которую соответствующая переменная входит либо только без знака инверсии (x_i), либо только со знаком инверсии ($\bar{x}_i$). Здесь и далее на рисунках называются только прямые зоны.

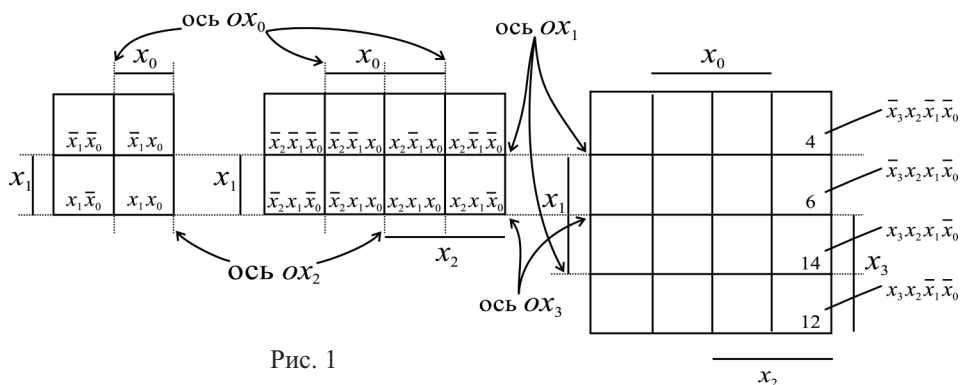


Рис. 1

Линии, разделяющие карту Карно на зоны x_i и $\bar{x}_i$, называют *осями* карты Карно. Все соседние клетки являются симметричными относительно какой-либо оси карты. На рис. 2 для карт Карно 2, 3 и 4 переменных показаны варианты соседних клеток, которые используются при отыскании максимальных покрытий.

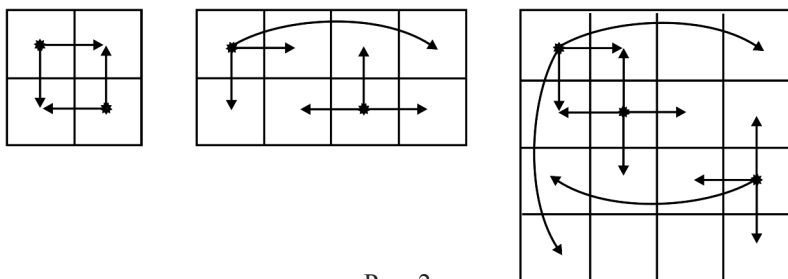


Рис. 2

Покрытием единичных значений логической функции в карте Карно называют совокупность клеток карты, в которых проставлены единицы. При этом покрытие должно состоять из 2^R клеток, где $R=0, 1, 2, \dots, (n-1)$ называют *рангом покрытия карты Карно n переменных*.

Произвольная единичная клетка является покрытием нулевого ранга. Две единичные клетки, симметричные относительно какой-либо оси карты, составляют покрытие 1-го ранга. Два покрытия 1-го ранга (четыре единичные клетки), симметричные относительно какой-либо оси карты, образуют покрытие 2-го ранга. Два покрытия 2-го ранга (восемь единичных клеток), симметричные относительно какой-либо оси карты, образуют покрытие 3-го ранга.

Каждому покрытию соответствует минтерм, причем чем выше ранг покрытия, тем меньшее число переменных в конъюнкции. На рис. 3 приведены примеры покрытий 1-го, 2-го и 3-го рангов для карты Карно четырех переменных, наиболее часто используемой в практике. Для каждого из покрытий там же указаны минтермы.

Отыскание МДНФ по карте Карно состоит из двух этапов:

- покрытие всех единичных значений функции минимальным числом покрытий максимально высокого ранга, при этом одна и та же единичная клетка может входить в разные покрытия;
- объединение всех найденных покрытий (конъюнкций) знаком дизъюнкции.

Сформулируем два правила, согласно которым каждому максимальному покрытию ранга R можно поставить в соответствие конъюнкцию ТДНФ.

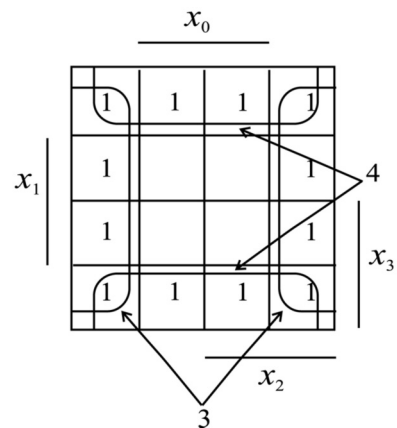
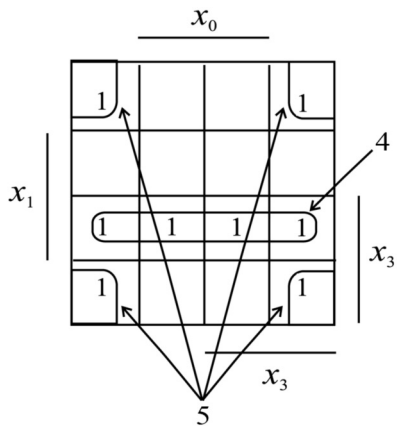
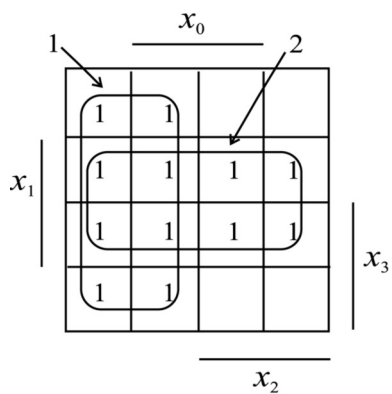
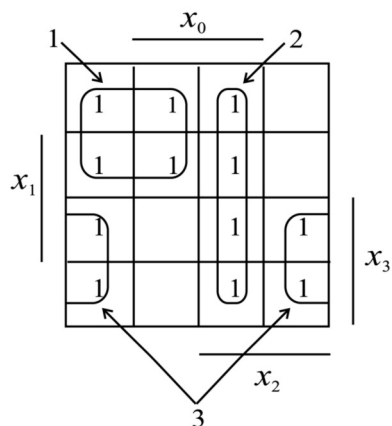
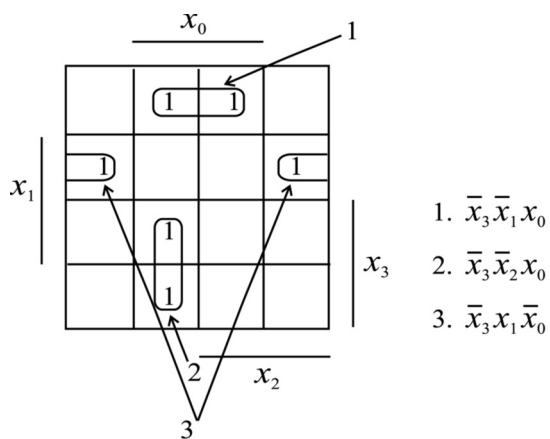


Рис. 3

2. Если покрытие целиком лежит в зоне x_i ($\bar{x}_i$), то в соответствующую этому покрытию конъюнкцию ТДНФ войдет переменная x_i ($\bar{x}_i$).

Неполностью определенная логическая функция должна быть доопределена таким образом, чтобы все единичные значения функции накрывались минимальным числом покрытий возможно более высокого ранга.

Общее число вариантов доопределения -2^k , где k – число наборов переменных, на которых функция не определена. Для примера (табл. 3): число вариантов будет $2^6 = 64$, из которых следует выбрать один.

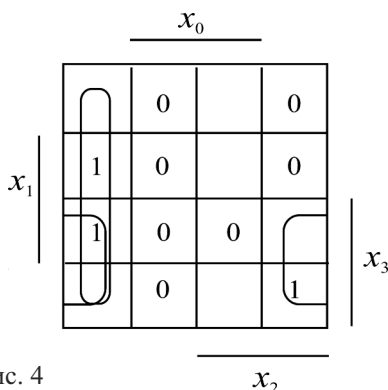


Таблица 3

i	x_3	x_2	x_1	x_0	f
0	0	0	0	0	$\sim$
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	$\sim$
6	0	1	1	0	0
7	0	1	1	1	$\sim$
8	1	0	0	0	$\sim$
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	$\sim$
14	1	1	1	0	$\sim$
15	1	1	1	1	0

По табл. 3 заполняем карту Карно (рис. 4).

Алгебраическое (аналитическое) выражение для функции f (минимальная дизъюнктивная нормальная форма), полученное по карте Карно (рис. 4), имеет вид: $f = \bar{x}_2 \cdot \bar{x}_0 \vee x_3 \cdot \bar{x}_0$.

На рис. 5 приведены рекомендованная нумерация клеток и зоны карт Карно для 2, 3, 4, 5 и 6 переменных.

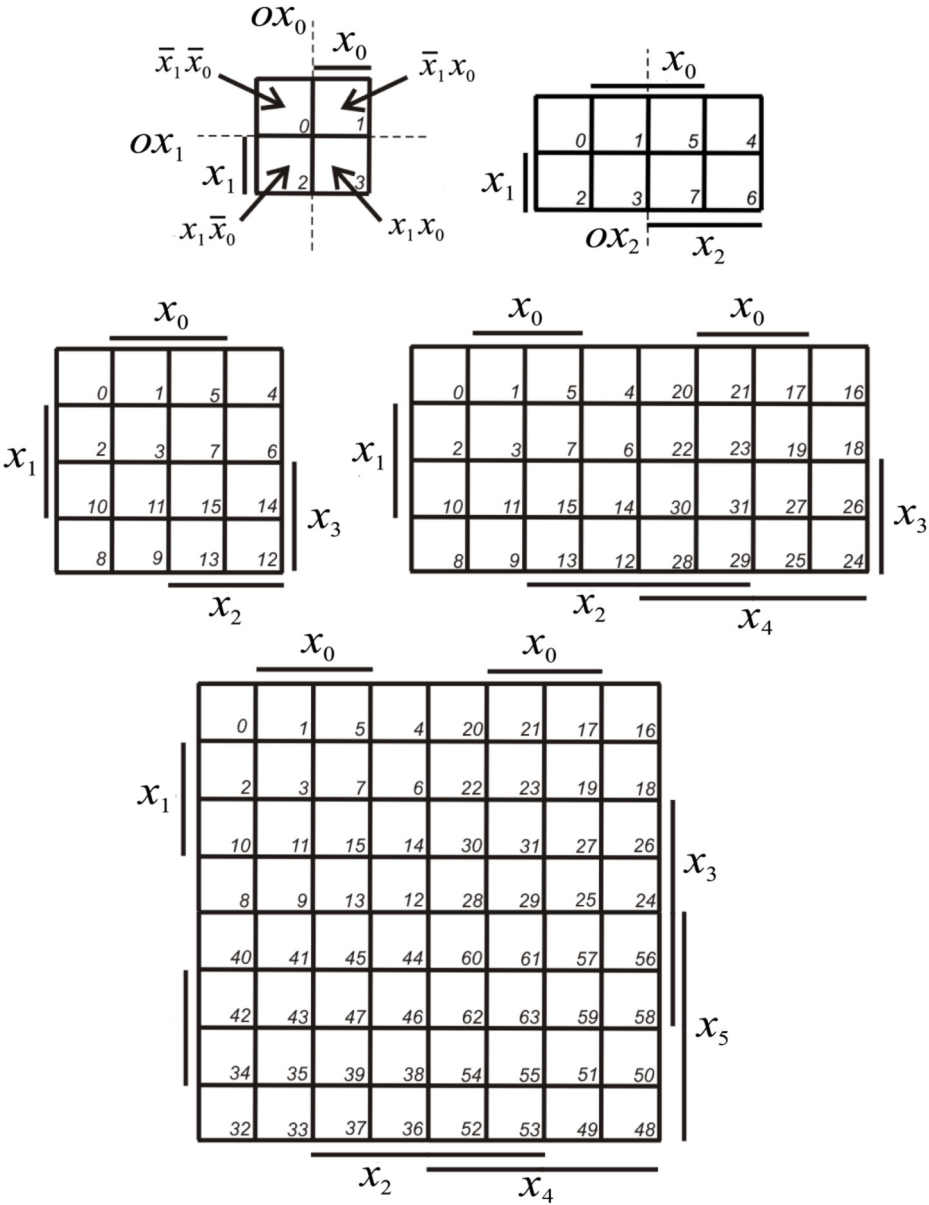


Рис. 5

Контрольные вопросы

1. В чем суть минимизации с использованием карт Карно?
2. Пронумеруйте карты Карно 3, 4 и 5 переменных.
3. Что означает понятие «покрытие n -го ранга»?
4. Как связаны размерность карты Карно и максимальный ранг покрытия?
5. Какая логическая функция называется неполностью определенной?
6. Какова методика отыскания МДНФ с помощью карт Карно?

Варианты заданий

Задание: минимизировать неполностью определенные логические функции, заданные картой Карно.

Вариант № 1

		0	0
1	0		1
1	1		0
	1		

	1	1	1
1		0	0
		1	
0	0	1	1

	1	0	
	1	0	
	0	1	
	1	0	0

0		1	1
			1
0			1
1	0		

0	1	1	
		1	
1	1	1	0
		0	

Вариант № 2

	1	0	1
0		1	0
0	1		0
1			

	0	1	
1		0	1
1	0		
		0	1

	1	0	1
		1	0
0		0	0
1	1		

	1	0	1
0		1	0
0	1		0
1			

		1	0
	1	0	1
1	0	1	
0	1		

Вариант № 3

		0	0
1			0
1	1	0	0

1			0
	1		
0		1	
0	0		1

1	0	0	
0	1		
0	0		
0	0	1	1

1		1	
0		0	1
0	0		
		1	

1	0		
0	1		0
0		1	0
		0	1

Вариант № 4

	0	0	1
0		1	0
1		1	
		0	1

	0	0	1
	1		1
0			0
1	0	0	

1	0	0	
	1	1	
0			0
	0	0	1

0			1
	1	0	
1		0	
		0	1

			1
0	1	0	
1	1	1	
0	1	0	

Вариант № 5

	1	1	
	0		
1		0	0
0	1	0	

	1		
1		1	0
1	0	1	1

	1		0
	1		0
1	0		0

0	1	1	0
		0	
	0		
0	1	1	

	0	1	
	0	1	
	1	0	
	0	1	

Вариант № 6

1		1	
0			
1		1	
0	0		0

	0	1	
1		0	1
1	0		
		0	1

1		1	
0		0	1
0	0		
		1	

		0	1
	0	1	1
0	1	1	
1	1		0

			1
	0	1	
0	1	1	0
	1		

Вариант № 7

		0	0
1	0		1
1	1		0
	1		

	1	0	1
0		1	0
0	1		0
1			

		0	0
1			0
1	1	0	0

	0	0	1
0		1	0
1		1	
		0	1

	1	1	
	0		
1		0	0
0	1	0	

Вариант № 8

	0	1	
1		0	1
1	0		
		0	1

1			0
	1		
0		1	
0	0		1

	0	0	1
	1		1
0			0
1	0	0	

	1		
1		1	0
1	0	1	1

	0	1	
1		0	1
1	0		
		0	1

Вариант № 9

	1	0	
	1	0	
	0	1	
	1	0	0

	1	0	1
		1	0
0		0	0
1	1		

1	0	0	
0	1		
0	0		
0	0	1	1

1	0	0	
	1	1	
0			0
	0	0	1

	1		0
	1		0
1	0		0

Вариант № 10

	1	0	1
0		1	0
0	1		0
1			

1		1	
0		0	1
0	0		
		1	

0			1
	1	0	
1		0	
		0	1

0	1	1	0
		0	
	0		
0	1	1	

		0	1
	0	1	1
0	1	1	
1	1		0

Вариант № 11

0	1	1	
		1	
1	1	1	0
		0	

		1	0
	1	0	1
1	0	1	
0	1		

1	0		
0	1		0
0		1	0
		0	1

			1
0	1	0	
1	1	1	
0	1	0	

	0	1	
	0	1	
	1	0	
	0	1	

Вариант № 12

		0	0
1	0		1
1	1		0
	1		

	0	1	
1		0	1
1	0		
		0	1

1	0	0	
0	1		
0	0		
0	0	1	1

0			1
	1	0	
1		0	
		0	1

	0	1	
	0	1	
	1	0	
	0	1	

Вариант № 13

	1	0	1
0		1	0
0	1		0
1			

1			0
	1		
0		1	
0	0		1

1	0	0	
	1	1	
0			0
	0	0	1

0	1	1	0
		0	
	0		
0	1	1	

			1
	0	1	
0	1	1	0
	1		

Вариант № 14

		0	0
1			0
1	1	0	0

	0	0	1
	1		1
0			0
1	0	0	

	1		0
	1		0
1	0		0

		0	1
	0	1	1
0	1	1	
1	1		0

0	1	1	
		1	
1	1	1	0
		0	

Вариант № 15

	0	0	1
0		1	0
1		1	
		0	1

	1		
1		1	0
1	0	1	1

1		1	
0		0	1
0	0		
		1	

0		1	1
			1
0			1
1	0		

		1	0
	1	0	1
1	0	1	
0	1		

Вариант № 16

	1	1	
	0		
1		0	0
0	1	0	

	0	1	
1		0	1
1	0		
		0	1

	1	0	
	1	0	
	0	1	
	1	0	0

	1	0	1
0		1	0
0	1		0
1			

1	0		
0	1		0
0		1	0
		0	1

Вариант № 17

1		1	
0			
1		1	
0	0		0

	1	1	1
1		0	0
		1	
0	0	1	1

	1	0	1
		1	0
0		0	0
1	1		

1		1	
0		0	1
0	0		
		1	

			1
0	1	0	
1	1	1	
0	1	0	

Вариант № 18

	1	0	1
0		1	0
0	1		0
1			

	1	1	1
1		0	0
		1	
0	0	1	1

1		1	
0		0	1
0	0		
		1	

0	1	1	0
		0	
	0		
0	1	1	

			1
0	1	0	
1	1	1	
0	1	0	

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

При проведении лабораторных работ используется персональный компьютер с установленной средой разработки Quartus II. Аппаратная часть представлена демонстрационной платой DE2 фирмы Altera с кабелем питания и USB-кабелем (USB Blaster) для программирования и управления программируемой логической интегральной схемой (ПЛИС, см. приложение 1).

На рис. 6 показан общий вид платы DE2. Плата включает следующие компоненты:

- ПЛИС Altera Cyclone® II 2C35 FPGA;
- последовательное устройство конфигурации – EPCS16;
- USB Blaster – устройство для программирования и пользовательского управления API; поддерживаются режимы программирования JTAG и Active Serial (AS);
- 512 Кб SRAM;
- 8 Мб SDRAM;
- 4 Мб Flash-память;
- гнездо карты SD;
- 4 переключателя-кнопки (*KEY0... KEY3*);
- 18 переключателей-тумблеров (*SW0...SW17*);
- 18 красных светодиодов (*LEDR0...LEDR17*);
- 9 зеленых светодиодов (*LEDG0...LEDG8*);
- 50 МГц осциллятор и 27 МГц (от TV-декодера) для генераторов тактовых сигналов;
- 24-разрядный аудиокодек с разъемами line-in (линейный вход), line-out (линейный выход) и microphone-in (микрофон);
- VGA DAC (10-разрядные высокоскоростные тройные ЦАП) с выходным VGA-коннектором;
- TV-декодер (NTSC/PAL) и входной TV-коннектор;
- 10/100 Ethernet-контроллер;
- USB Host/Slave Controller (USB-контроллер хост/слэйв) с USB-коннекторами типа A и B;

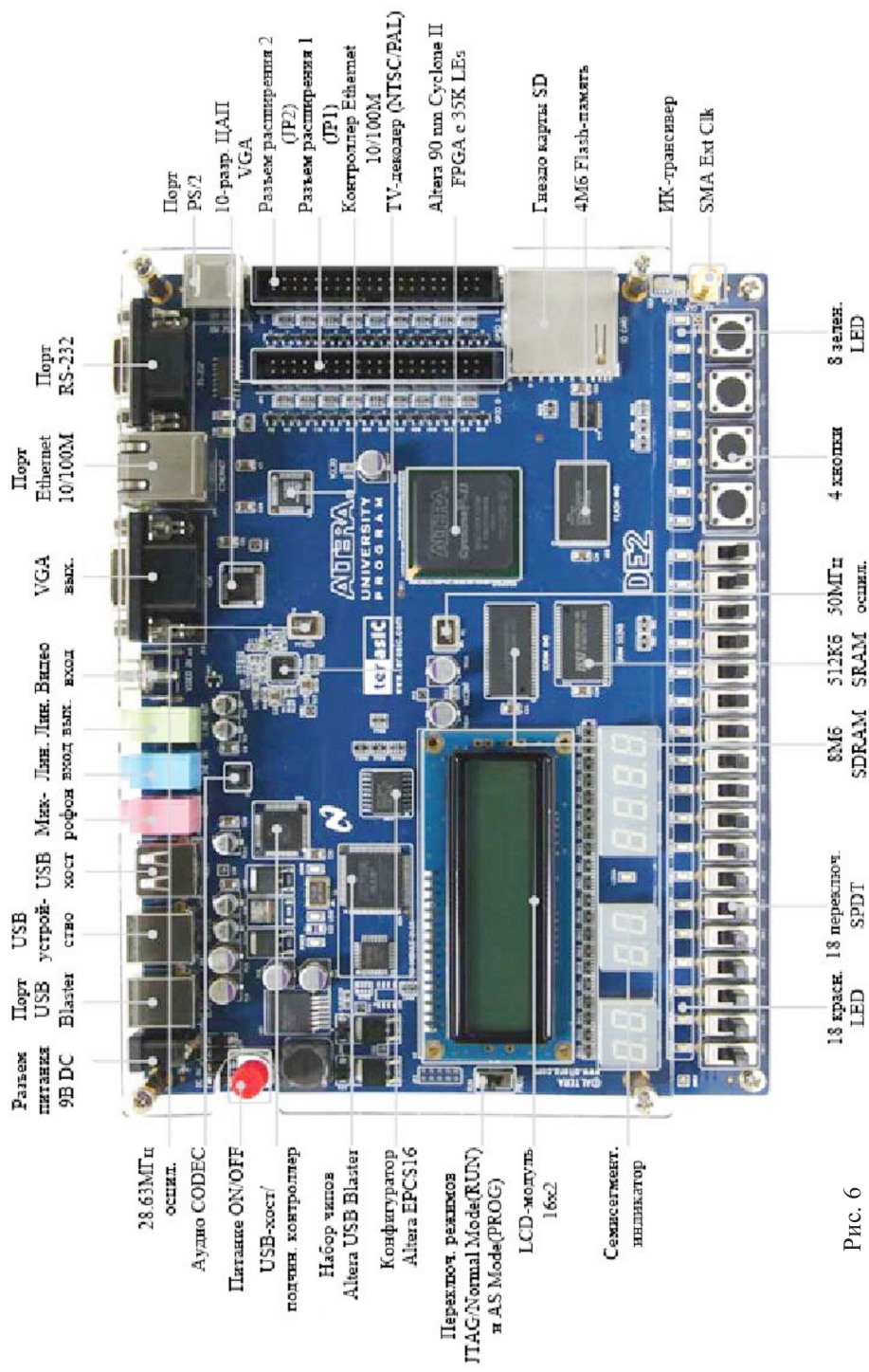
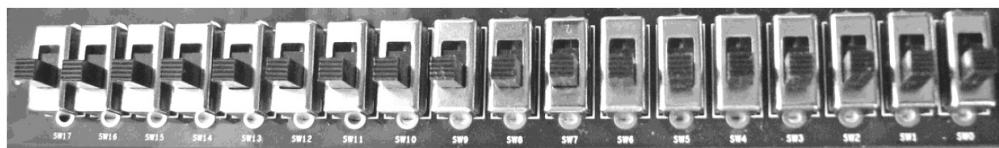


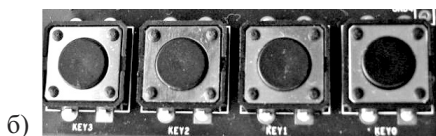
Рис. 6

- трансивер RS-232 и 9-разрядный коннектор;
- трансивер IrDA;
- два разъема расширения по 40 выводов с диодной защитой.

В лабораторном практикуме для задания входных статических сигналов используются переключатели-тумблеры (рис. 7, а). При этом верхнее положение тумблера соответствует логической единице (высокий уровень), а нижнее – логическому нулю (низкий уровень).



а)



б)

Рис. 7

Для задания входных динамических сигналов используются переключатели-кнопки (рис. 7, б).

Здесь следует отметить, что исходное состояние сигнала, поступающего с кнопки, – высокий уровень. При нажатии кнопки формируется активный обратный фронт.

Наблюдение за выходными сигналами осуществляется с помощью красных светодиодных индикаторов (рис. 8, а), находящихся над тумблерами, и зеленых светодиодных индикаторов, расположенных над переключателями-кнопками (рис. 8, б).



а)



б)

Рис. 8

Если индикатор загорается, то это соответствует логической единице, отсутствие светового сигнала свидетельствует о логическом нуле.

Маршрут проектирования в среде Quartus II

Порядок проведения лабораторных работ определяется маршрутом проектирования в среде Quartus II.

Для изучения маршрута проектирования исследуем работу стандартного библиотечного элемента 2ИЛИ-НЕ с помощью программной среды Quartus II. Ход работы будет включать следующие этапы:

1. Ввод проекта.
2. Анализ и синтез схемы.
3. Размещение и разводка.
4. Компиляция проекта.
5. Моделирование.
6. Прошивка проекта в ПЛИС.

Как отмечено выше, реализация проекта будет осуществляться с помощью демонстрационной платы DE2 с использованием ПЛИС Cyclone II. Схема эксперимента показана на рис. 9. Помимо ПЛИС, на схеме обозначены ключи, соответствующие переключателям-тумблерам на плате *SW0* и *SW1*, и светодиодный индикатор *LEDR0*.

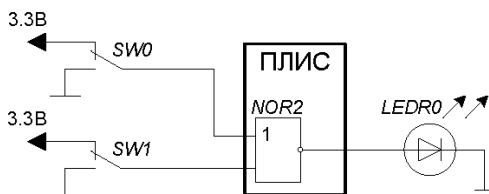


Рис. 9

Работа логического элемента 2ИЛИ-НЕ описана таблицей истинности (табл. 4), где i – номер набора переменных. После прошивки проекта в ПЛИС при задании разных положений переключателей *SW0* x_0 и x_1 индикатор *LEDR0* покажет значение выходного сигнала (y), которое должно соответствовать таблице истинности.

Таблица 4

i	Входы		Выход
	x_0	x_1	y
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0

1. Ввод проекта

Создание проекта

Запускаем Quartus II, при этом появляется окно (рис. 10), в котором выбираем *Create a New Project*.

Затем нужно выбрать папку, в которой будет размещаться проект, и указать его название (рис. 11).

В следующем окне выбираем устройство, в которое будет прошиваться проект (рис. 12): Family: *Cyclone II*; Name: *EP2C35F672C6*.



Рис. 10

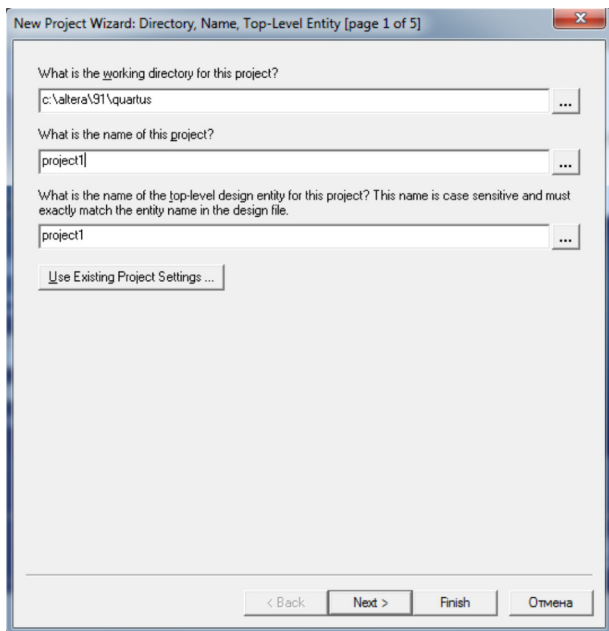


Рис. 11

Далее можно выбрать файл из списка уже существующих файлов для включения в проект. Пропустим этот шаг, так как необходимо создать новый файл.

На следующем шаге предлагается задать дополнительные параметры синтеза, моделирования и временного анализа. Эти поля остаются пустыми.

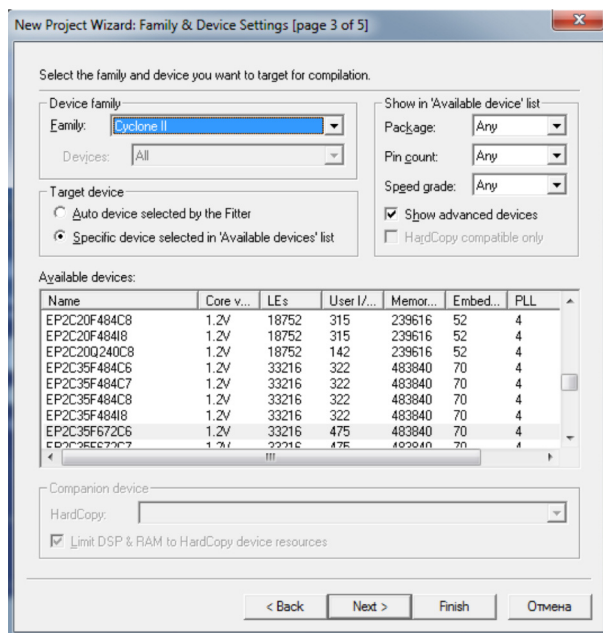


Рис. 12

Создание файла схемы

Перейдем к созданию файла, описывающего разрабатываемое устройство. Выбираем *File=>New=>Block Diagram/Schematic File* (рис. 13).

Так как предполагается рассмотреть работу простейшего элемента 2ИЛИ-НЕ, применение текстового описания на языке VHDL (Verilog) не очень оправдано. В данном случае целесообразнее использовать ввод устройства с помощью схемы, т.е. графическое описание.

И в заключительном окне нажимаем *Finish*.

Таким образом, был получен новый проект, который следует сразу же сохранить (*File=>Save Project*).

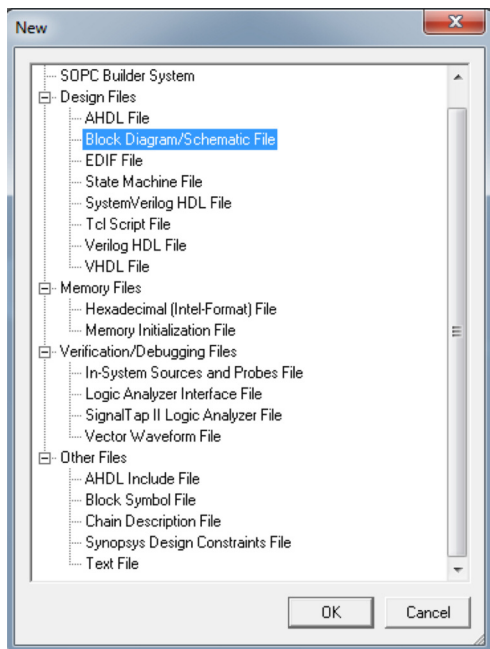


Рис. 13

В результате проделанных шагов открывается лист для создания схемы, который также нужно сохранить.

Размещение компонентов и выводов

Приступим к созданию самой схемы. Для этого понадобится один двухвходовой элемент 2ИЛИ-НЕ, а также два входных и один выходной порты. Чтобы открыть библиотеку символов, нужно щелкнуть на пиктограмме  (*Symbol Tool*) либо нажать правую клавишу мыши и в выпадающем меню выбрать *Insert=>Symbol*. В левой части появившегося окна *Symbol* находятся библиотеки Quartus II (рис. 14).

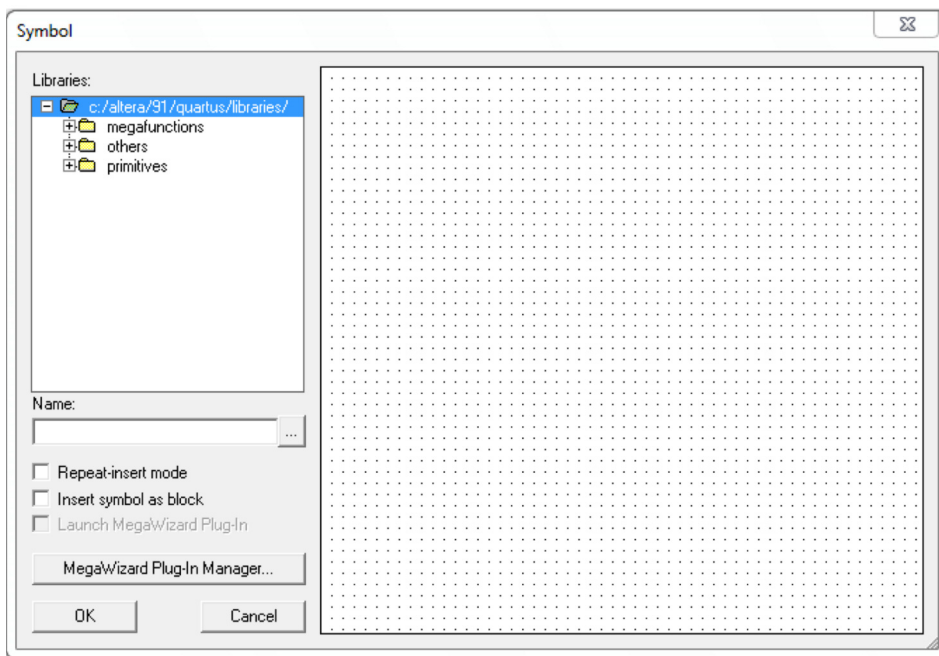


Рис. 14

Нужный элемент 2ИЛИ-НЕ – `\primitives\logic\nor2`. Также выбираются входы и выходы (`\primitives\pin\input` и `\primitives\pin\output`), которые размещаются на листе таким образом, что два входных порта (*input*) соединены с входами элемента 2ИЛИ-НЕ, а его выход соединен

с портом выхода (*output*). Порт представляет собой точку подключения, предназначенную для осуществления обмена данными между устройством, реализованным в ПЛИС, и внешними устройствами. Двойной щелчок мыши на элементе позволяет менять его свойства. В этом случае нужно задать имена портов: *input1*, *input2*, *output* (рис. 15).

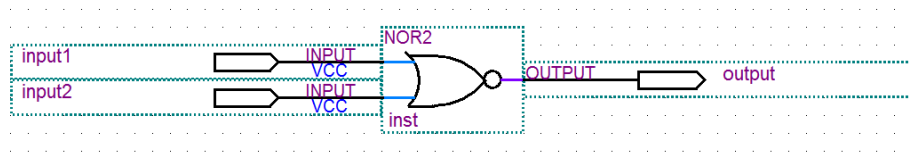


Рис. 15

2. Анализ и синтез схемы

Данный этап позволяет произвести проверку на наличие ошибок и впоследствии получить возможность назначить выводы ПЛИС в соответствии со схемой. В меню *Processing* выбираем *Start=>Start Analysis & Synthesis*. Начинается процесс поиска ошибок в созданной схеме, по результатам генерируется отчет. В случае если все верно, появится сообщение об удачном синтезе и отсутствии ошибок.

3. Размещение и разводка

Перед загрузкой проекта в ПЛИС необходимо задать номера выводов для входов и выходов. Это делается при помощи команды *Assignments=>Pins*. В появившемся окне *Pin Planner* для портов нужно назначить конкретные выводы ПЛИС. В данном проекте предполагается использовать два переключателя-тумблера *SW0* и *SW1* для задания входных сигналов *input1* и *input2*, а также красный светодиод *LED0* для определения значения выходного сигнала *output*. Используя таблицы соответствия выводов ПЛИС переключателям и светодиодам (см. приложение 2), введем в поле *Location* соответствующие выводы для портов (рис. 16).

4. Компиляция проекта

Прежде чем приступить к моделированию, необходимо откомпилировать проект. Компиляция представляет собой процесс преобразо-

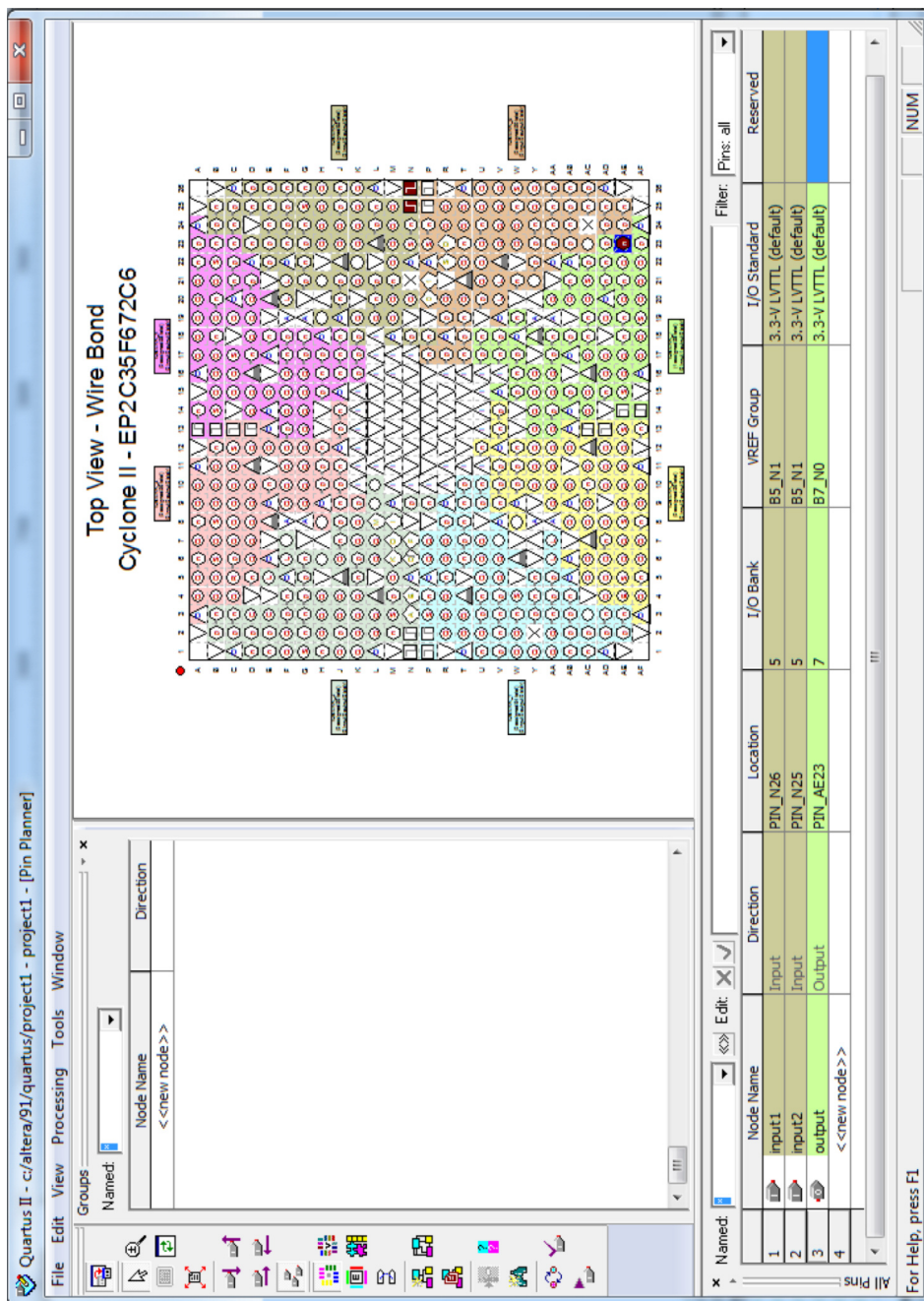


Рис. 16

вания введенного проекта (в рассматриваемом случае заданного в виде схемы с помощью графического редактора) в представление проекта на вентиляном уровне, т.е. на базе элементов И-НЕ, ИЛИ-НЕ. Логическое описание на вентиляном уровне формируется в виде списка связей (*Netlist*). Список связей (*Netlist*) является текстовой формой кодирования системы, т.е. текстовым эквивалентом графического представления схемы, введенной разработчиком или синтезированной на языке VHDL. В меню *Processing* выбираем *Start Compilation*. Это запускает процесс компиляции, по итогам которой формируется отчет об ошибках и предупреждениях.

5. Моделирование

Создание временной диаграммы

Для моделирования работы созданного проекта и визуализации сигналов создадим и сохраним файл временной диаграммы: *File=>New=>Vector Waveform File* (рис. 17).

Затем нужно определить сигналы, которые планируется выводить на график. Правой клавишей мыши щелкаем на панели в левой части экрана и выбираем *Insert=>Insert Node or Bus* (рис. 18).

В появившемся окне можно набрать имя вывода какого-либо сигнала, однако это не всегда удобно.

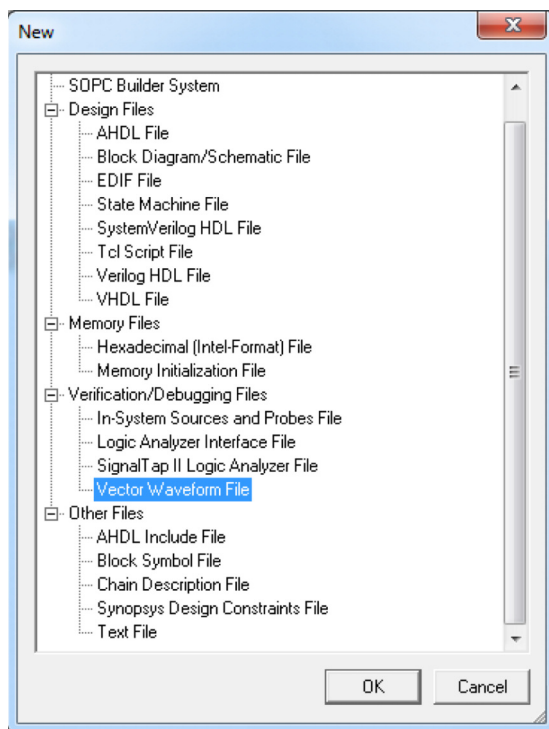


Рис. 17

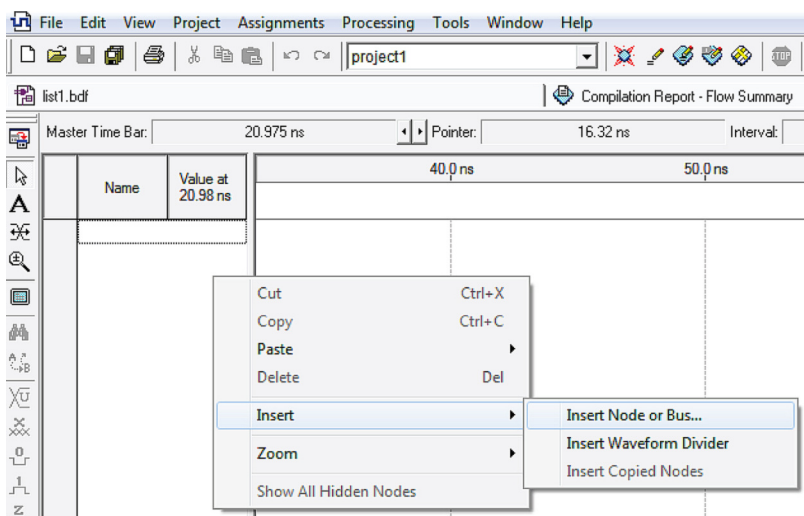


Рис. 18

Поэтому лучше воспользоваться поиском сигнала в проекте, нажав *Node Finder* (рис. 19).

В следующем окне *Node Finder* нажимаем кнопку *List*, при этом появляется список портов. С помощью кнопки **>>** нужно перенести все выводы в область

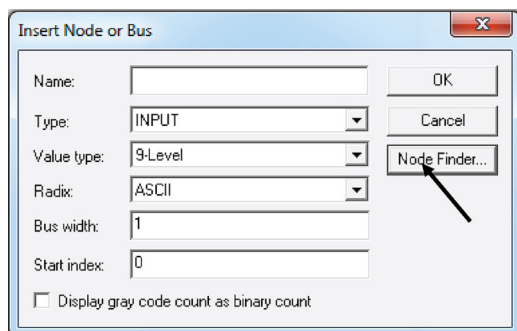


Рис. 19

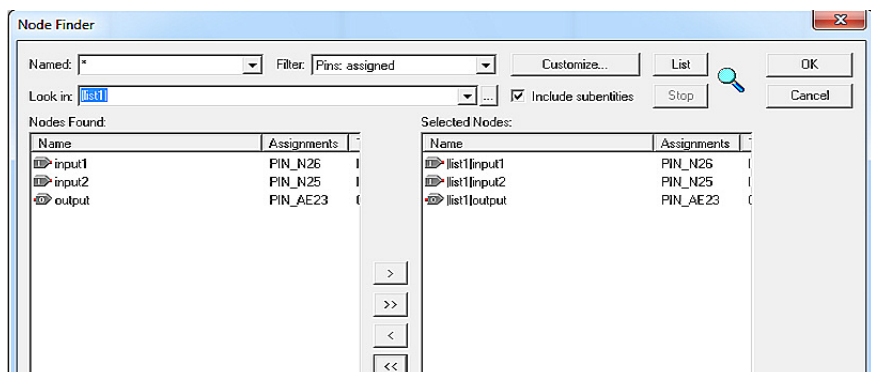


Рис. 20

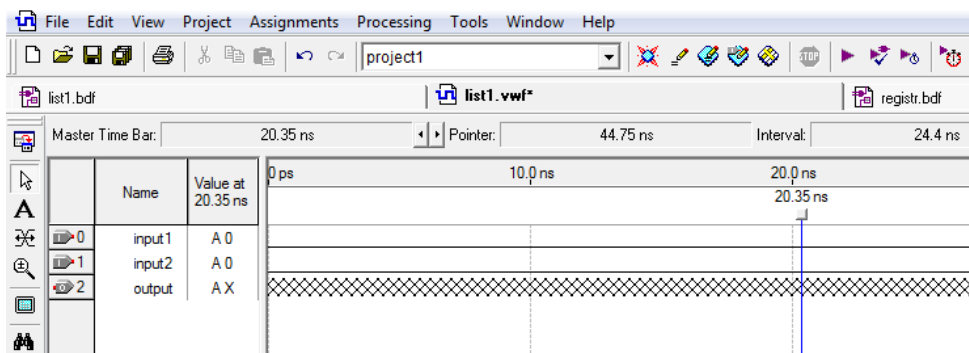


Рис. 21

Selected Nodes (рис. 20) и нажать ОК. Теперь в файле временных диаграмм появились выбранные сигналы (рис. 21).

Задание значений сигналов

Редактировать входные сигналы можно при помощи кнопок на панели инструментов. Например, кнопка *Forcing High* позволит установить выбранный сигнал в значение логической единицы на всем отрезке времени. Также можно выделить мышью определенный участок и установить сигнал логической единицы только на нем.

В рассматриваемом случае удобнее задать входные сигналы как сигналы тактовой частоты (сигналы генератора импульсов). Выбираем сигнал *input1* и нажимаем кнопку *Overwrite Clock* . При этом открывается окно задания временных параметров тактового сигнала (рис. 22), которые можно оставить заданными по умолчанию.

Таким же образом устанавливаются значения сигнала *input2*, но в параметрах увеличим его период до 20 ns. И затем сохраняем файл.

Временная диаграмма приобрела вид, представленный на рис. 23.

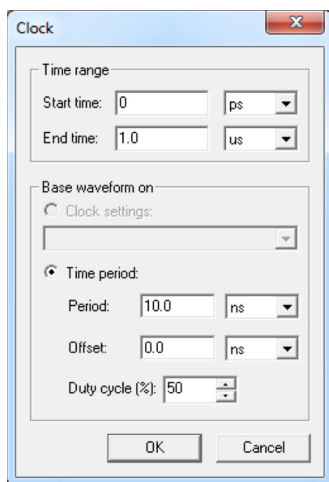


Рис. 22

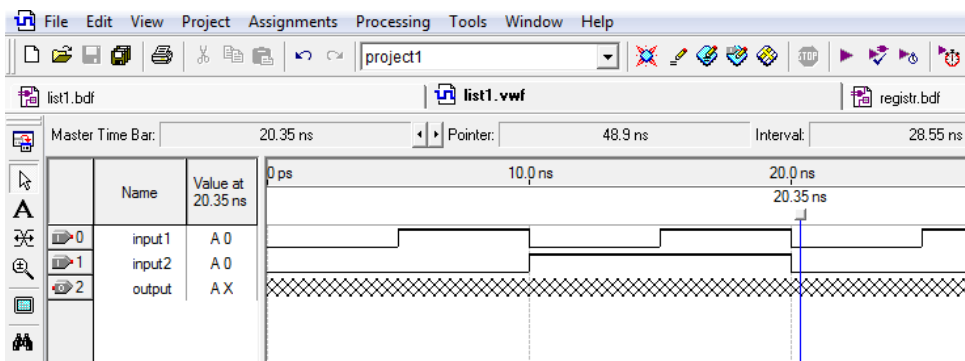


Рис. 23

Запуск модели

Переходим к моделированию работы устройства. Следует отметить, что моделирование бывает временным и функциональным. Временное моделирование позволяет увидеть сигналы с учетом всех возникающих задержек между элементами, входами и выходами. Функциональное моделирование позволяет проверить именно логику работы. С его помощью можно убедиться, что проект работает так,

как положено. В данном проекте будет использоваться именно функциональное моделирование: меню *Processing* => *Simulator Tool*. В строке режима моделирования *Simulation mode* устанавливаем *Functional* и создаем специальный список цепей для функционального моделирования: *Generate Functional Simulation Netlist* (рис. 24).

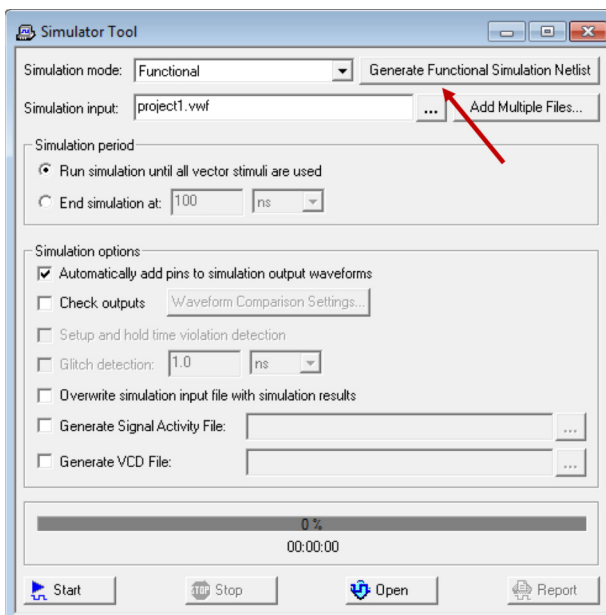


Рис. 24

После завершения процесса создания списка цепей для функционального моделирования на экране появится соответствующее сообщение.

Запуск моделирования осуществляется из меню *Processing=>Start Simulation*. По окончании моделирования формируются временные диаграммы, по которым можно судить о работе проекта (рис. 25). Полученная временная диаграмма соответствует таблице истинности логического элемента (табл. 4).

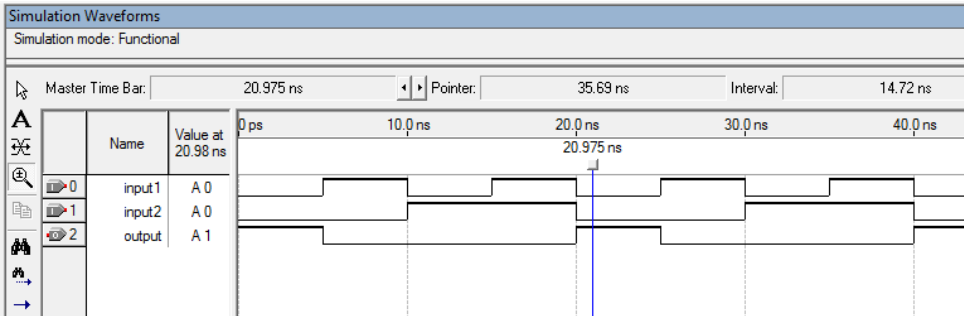


Рис. 25

6. Прошивка проекта в ПЛИС

Включение платы осуществляется при помощи красной кнопки на плате (см. рис. 6). Запускаем работу программатора, который запишет проект в ПЛИС: *Tools=>Programmer*. Появляется окно, в котором нажимаем кнопку *Hardware Setup* и выбираем в качестве канала, по которому будет осуществляться программирование, *USB-Blaster* (рис. 26).

Далее нажимаем кнопку *Start*, и начинается прошивка ПЛИС. Когда прошивка будет завершена, на плате загорятся семисегментные индикаторы и некоторые светодиоды (рис. 27).

Следует обратить внимание, что на плате горит и индикатор *LEDR0*, что говорит о единичном выходном сигнале при нулевых входных значениях *input1* и *input2* (т.к. переключатели, задающие входные сигналы, находятся в нижнем положении). Изменяя положение переключателей *SW0* и *SW1*, можно убедиться, что проект работает в соответствии с логикой элемента 2ИЛИ-НЕ (см. табл. 4).

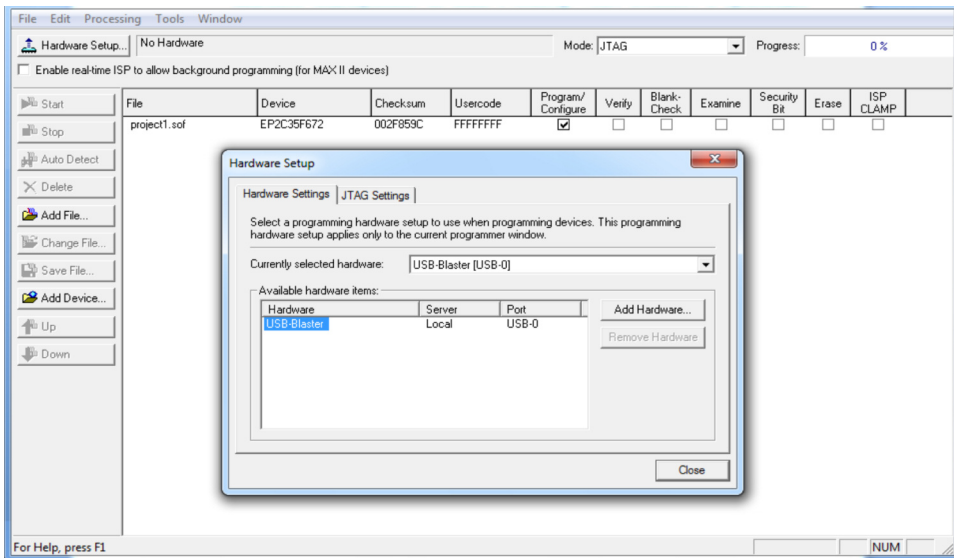


Рис. 26

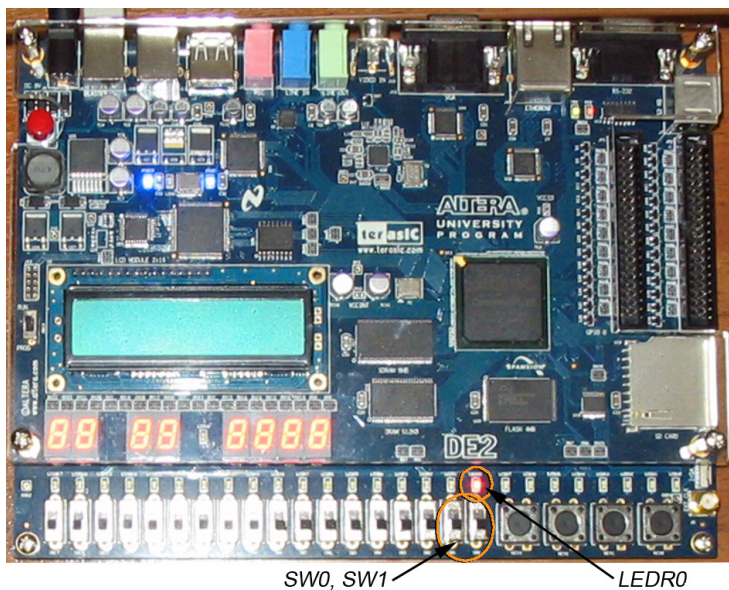


Рис. 27

Лабораторная работа 1.

ПРАКТИКУМ ПО КОМБИНАЦИОННЫМ СХЕМАМ

Цель работы – изучение комбинационных логических схем, освоение маршрута проектирования в программной среде Quartus II и приобретение навыков работы с ПЛИС (программируемые логические интегральные схемы).

Логические схемы и устройства комбинационного типа

Логической схемой называют такую схему, которая выполняет логические операции над одной или несколькими логическими переменными.

Комбинационной логической схемой называют такую схему, выходное состояние которой определяется только комбинацией входных логических переменных в данный момент времени.

Логические схемы являются наиболее простым и одновременно наиболее универсальным и многочисленным классом цифровых устройств. Они различаются по типу выполняемой функции (инверсия НЕ, логическое умножение И, логическое сложение ИЛИ и их инверсии И-НЕ, ИЛИ-НЕ, сумма по модулю 2), числу элементов (от одного до шести) и количеству входов.

К логическим схемам комбинационного типа относятся: мажоритарные логические схемы, мультиплексоры и демультиплексоры, сумматоры, преобразователи кодов, цифровые компараторы, шифраторы и дешифраторы, например часто применяемый дешифратор семисегментного индикатора.

Семисегментный индикатор

Семисегментные индикаторы на светодиодах являются одним из наиболее распространенных устройств отображения десятичных чисел. На рис. 28, а показан общий вид индикатора,

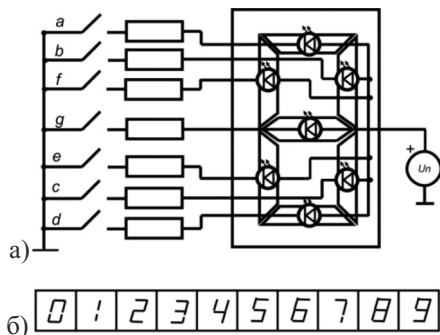


Рис. 28

а на рис. 28, b – все десять цифр, получаемые с его помощью. Семь сегментов индикатора обозначаются латинскими буквами от a до g .

Когда ключ замкнут, ток от источника течет через светодиод, вызывая его свечение. Последовательно включенный резистор ограничивает ток до определенного уровня ≈ 20 мА. Без ограничивающего резистора светодиод может выйти из строя. Обычно работой семисегментного индикатора управляют не механические, а электронные ключи. На практике их роль реализуют дешифраторы семисегментного индикатора.

Часть I

В рамках одного проекта в Quartus II рассмотреть работу логических элементов: сумма по модулю 2, 4И-НЕ, 8И-НЕ.

1. Нарисовать одну из схем эксперимента для исследуемых элементов аналогично схеме на рис. 9.

2. Составить таблицы истинности для каждого проверяемого элемента и записать совершенную дизъюнктивную нормальную форму (СДНФ) для элементов сумма по модулю 2 и 4И-НЕ. Для элемента 8И-НЕ таблица будет иметь следующий вид:

i	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	f
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	
2	0	0	0	0	0	0	1	0	
3	0	0	0	0	0	0	1	1	
...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	
127	0	1	1	1	1	1	1	1	
128	1	0	0	0	0	0	0	0	
129	1	0	0	0	0	0	0	1	
...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	
255	1	1	1	1	1	1	1	1	

3. Минимизировать функции, используя карты Карно.

4. С помощью графического редактора программы Quartus II и описанного маршрута проектирования создать схемы для каждого проверяемого элемента. Для отработки проекта на плате предполагается использовать тумблеры $SW0$, $SW1$ и светодиодный индикатор $LEDR0$ – для элемента сумма по модулю 2; $SW2$ – $SW5$, $LEDR2$ – для элемента 4И-НЕ; $SW6$ – $SW13$ и $LEDR6$ – для элемента 8И-НЕ.

5. Промоделировать работу каждого элемента с получением временных диаграмм.

6. Прошить проект в ПЛИС платы DE2 и провести проверку его работоспособности.

Часть II

1. Исходя из задания, составить таблицу истинности четырехрядного комбинационного устройства. Номера в задании указывают номера наборов, на которых функция обращается в логическую единицу, на остальных наборах функция равна нулю.

2. Записать СДНФ функции, заданной таблично.

3. Минимизировать логическую функцию с помощью карты Карно.

4. Используя все доступные элементы библиотеки Quartus II, создать в графическом редакторе схему устройства.

5. Доказать работоспособность схемы с помощью моделирования по временным диаграммам.

6. Прошить проект в ПЛИС и, переключая тумблеры, проверить его работу на соответствие таблице истинности.

Варианты заданий

Вариант 1	0, 2, 4, 7, 10, 12, 13, 15	Вариант 5	0, 1, 3, 4, 6, 8, 11, 14
Вариант 2	0, 1, 3, 5, 8, 11, 13, 14	Вариант 6	1, 2, 4, 5, 7, 9, 12, 15
Вариант 3	1, 2, 4, 6, 9, 12, 14, 15	Вариант 7	0, 1, 3, 5, 8, 11, 13, 14
Вариант 4	0, 2, 3, 5, 7, 10, 13, 15	Вариант 8	1, 3, 4, 6, 7, 9, 11, 14

Вариант 9	0, 1, 3, 5, 8, 11, 13, 14	Вариант 14	1, 4, 7, 9, 10, 12, 13, 15
Вариант 10	0, 3, 5, 6, 8, 9, 11, 13	Вариант 15	0, 2, 5, 8, 10, 11, 13, 14
Вариант 11	1, 4, 6, 7, 9, 10, 12, 15	Вариант 16	0, 1, 4, 8, 11, 11, 14, 15
Вариант 12	2, 5, 7, 8, 10, 11, 13, 15	Вариант 17	1, 2, 3, 5, 9, 13, 14, 15
Вариант 13	0, 3, 6, 8, 9, 11, 12, 14	Вариант 18	0, 2, 3, 4, 9, 10, 11, 13

Пример выполнения

Исходные данные: номера наборов – 0, 2, 4, 6, 8, 10, 11, 15.

1. Таблица истинности данной функции будет иметь вид:

i	x_3	x_2	x_1	x_0	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

2. СДНФ запишется как

$$f = \bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0 + \bar{x}_3 \cdot \bar{x}_2 \cdot x_1 \cdot \bar{x}_0 + \bar{x}_3 \cdot x_2 \cdot \bar{x}_1 \cdot \bar{x}_0 + \bar{x}_3 \cdot x_2 \cdot x_1 \cdot \bar{x}_0 + \\ + x_3 \cdot \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0 + x_3 \cdot \bar{x}_2 \cdot x_1 \cdot \bar{x}_0 + x_3 \cdot \bar{x}_2 \cdot x_1 \cdot x_0 + x_3 \cdot x_2 \cdot x_1 \cdot x_0.$$

3. С помощью таблицы истинности формируется карта Карно для минимизации заданной функции (рис. 29).

МДНФ данной функции запишется как

$$f = x_3 \cdot x_1 \cdot x_0 \vee \bar{x}_3 \cdot \bar{x}_0 \vee \bar{x}_2 \cdot \bar{x}_0.$$

	x_0		x_2	
	0	1	5	4
x_1	2	3	7	6
	10	11	15	14
x_3	8	9	13	12

Рис. 29

4. В программе Quartus II создается схема, описывающая работу данного устройства (рис. 30).

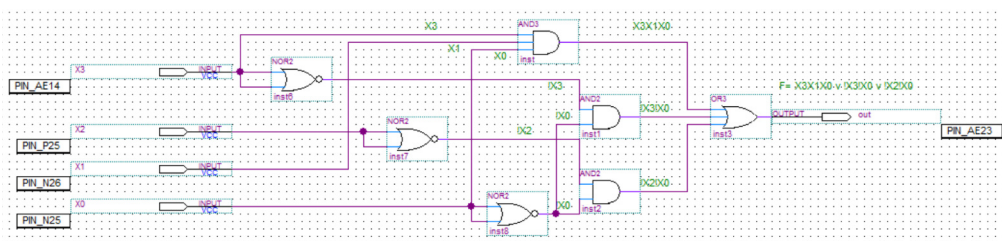


Рис. 30

5. Далее, с использованием пп. 2–5 маршрута проектирования, производится моделирование проекта и получаются временные диаграммы (рис. 31).

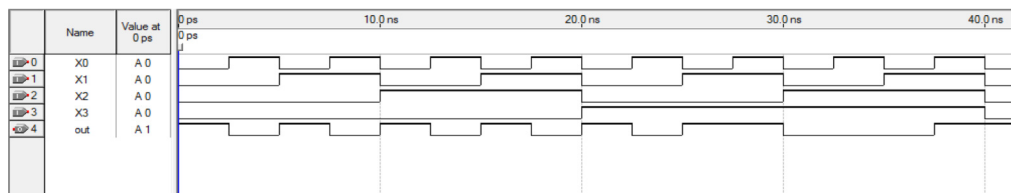


Рис. 31

6. В заключение проект прошивается в ПЛИС и производится его проверка на соответствие таблице истинности.

Часть III

Исследовать работу дешифратора семисегментного индикатора.

1. Используя готовый проект для управления дешифратором семисегментного индикатора, созданный в среде Quartus II (проект выдается преподавателем), прошить его в ПЛИС демонстрационной платы DE2.

2. Задавая входные воздействия переключателями $SW0-SW3$, составить таблицу истинности, включающую значения функций $f_a, f_b, f_c, f_d, f_e, f_f$ и f_g .

3. Для одной из семи логических функций (по выбору преподавателя) найти МДНФ с помощью карты Карно.

Отчет по лабораторной работе должен содержать: титульный лист, содержание, цель работы, задание, теоретическую и практическую часть, выводы и ссылки на использованную литературу.

Контрольные вопросы

1. Назовите основные логические функции двух переменных.
2. Приведите 5 из 20 основных соотношений булевой алгебры (по усмотрению преподавателя).
3. Приведите 4 формулы де Моргана.
4. Приведите примеры мультиплексоров и демультимплексоров.
5. Расскажите о принципах работы шифраторов и дешифраторов.
6. Какова методика минимизации логических функций с помощью карт Карно?
7. Какая логическая схема называется мажоритарной?
8. Какая логическая схема называется комбинационной?

Лабораторная работа 2.

ПРАКТИКУМ ПО ПОСЛЕДОВАТЕЛЬНОСТНЫМ СХЕМАМ

Цель работы – изучение принципов построения последовательностных схем, закрепление знаний по маршруту проектирования в среде Quartus II и навыков работы с ПЛИС.

Последовательностной логической схемой называют такую схему, выходное состояние которой зависит не только от текущих значений входных сигналов, но и от последовательности значений входных и выходных сигналов в предыдущие моменты времени.

В данной лабораторной работе будут рассмотрены такие типы последовательностных схем, как триггеры, регистры и счетчики.

Триггеры

Триггер – это устройство с двумя устойчивыми взаимоинверсными состояниями, которые устанавливаются при подаче на вход управляющих сигналов и могут хранить это состояние в течение бесконечно долгого времени после снятия этих сигналов. Триггер имеет прямой Q и инверсный $\bar{Q}$ выходы. У триггера может быть один или несколько информационных входов, и он может иметь синхронизирующий (тактовый) вход. Условное графическое обозначение обобщенного триггера, тактируемого по переднему фронту, представлено на рис. 32. В за-

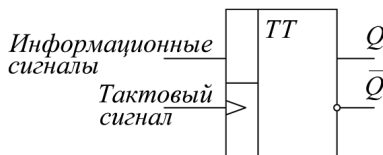


Рис. 32

висимости от того, используется синхронизирующий сигнал или нет, триггеры делятся на синхронные (тактируемые) и асинхронные.

Говорят, что триггер установлен, если выполнено условие: $Q=1$ ($\bar{Q}=0$). При этом $Q=0$ ($\bar{Q}=1$) – состояние сброса.

В практике наибольшее применение нашли следующие триггеры: R - S , D , J - K , T .

В основе всех этих типов триггеров лежит один и тот же базовый элемент – **бистабильная ячейка** (БЯ) или $\bar{R}$ - $\bar{S}$ -триггер.

Пример бистабильной ячейки, построенной на основе элементов 2И-НЕ, показан на рис. 33. БЯ представляет собой два элемента 2И-НЕ, объединенных положительными обратными связями. Реализация бистабильной ячейки в Quartus II и ее временные диаграммы показаны в примере выполнения (см. лаборат. работа 2, часть I).

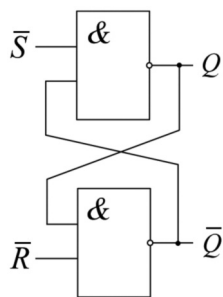


Рис. 33

Следует отметить, что заполнение таблиц состояний для триггеров происходит в произвольном порядке, с учетом текущего состояния выхода Q^n .

Регистры

Регистром называют устройство для приема, временного хранения и передачи данных, реализованное в виде совокупности триггеров, объединенных по некоторым управляющим входам. Данные в регистре хранятся в виде двоичного числа (кода), каждому разряду которого соответствует отдельный триггер.

Основными классификационными признаками регистров являются способ записи и способ считывания данных. Согласно этим признакам различают следующие типы регистров:

- с параллельными записью и считыванием;
- с последовательной записью и параллельным считыванием;
- с параллельной записью и последовательным считыванием;
- с последовательными записью и считыванием.

Реальные схемы регистров могут быть комбинированными и сочетать различные способы записи и считывания данных.

В зависимости от типа используемых в регистре триггеров различают регистры однофазного и парафазного вида. В парафазных регистрах, построенных, как правило, на основе R - S -триггеров, данные поступают одновременно по двум каналам: прямому и инверсному. Однофазные регистры реализуются на базе D -триггеров, и данные поступают только по одному каналу: прямому или инверсному.

Счетчики

Подсчет импульсов является одной из наиболее распространенных операций, выполняемых в устройствах цифровой автоматики. Такая операция осуществляется с помощью счетчиков. Счетчики делятся на простые и сложные. Простые счетчики бывают двух типов: суммирующие (счет в прямом направлении) и вычитающие (счет в инверсном направлении). Сложные счетчики являются реверсивными и осуществляют счет в оба направления. По организации цепей переноса счетчики бывают параллельные, последовательные и комбинированные.

Основной характеристикой счетчиков является модуль счета K_C . Модуль счета определяет максимальное число импульсов, которое может быть подсчитано счетчиком. После поступления K_C импульсов счетчик должен возвратиться в исходное состояние. Модуль счета K_C и число разрядов счетчика n связаны соотношением $n = E(3,3 \lg K_C + 1)$, где E — целая часть числа, заключенного в скобки.

Обычно счетчики имеют дополнительные коды установки: S , позволяющий предварительно записать в счетчик некоторое число, и R , который переводит счетчик в исходное состояние. Частота импульсов на выходе последнего разряда счетчика в K_C раз меньше, чем частота импульсов на выходе счетчика. Поэтому счетчики часто используются в качестве делителей частоты.

Особенностью реализации цифровых устройств в ПЛИС является то, что все процессы тактируются внешним генератором импульсов. Именно поэтому имеется некоторая сложность при проектировании асинхронных устройств, например счетчиков с произвольным модулем счета. Сигнал сброса, формируемый дешифратором и являющийся асинхронным, должен быть обработан синхронным триггером, на тактовый вход которого подается сигнал от внешнего генератора импульсов.

Часть I

1. В соответствии с заданием, построить триггер, используя логические элементы библиотеки Quartus II.
2. Промоделировать работу триггера с получением временных диаграмм.
3. Прошить проект в ПЛИС и провести проверку его работоспособности.

Пример выполнения

Построение и исследование работы бистабильной ячейки.

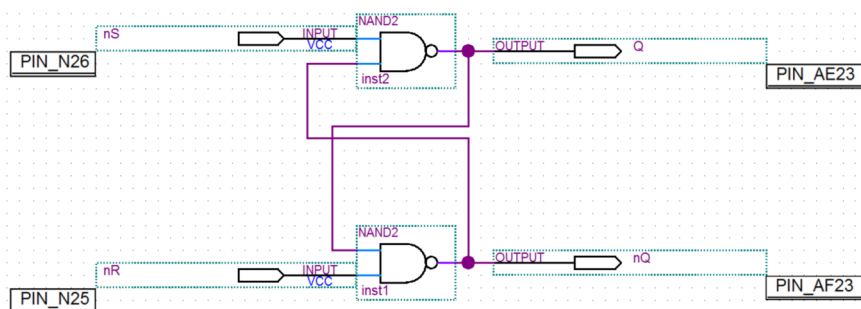
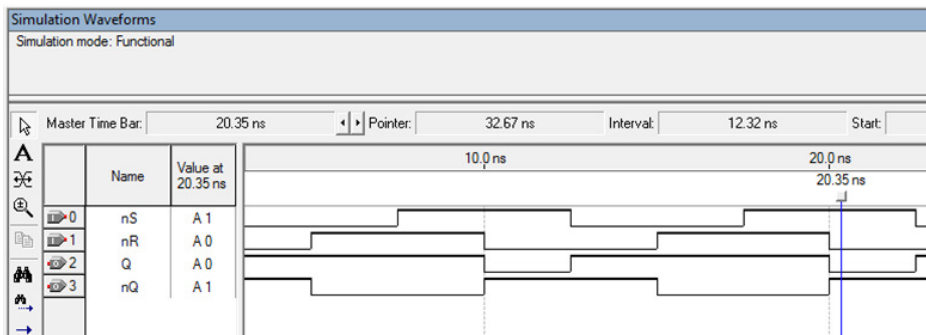


Схема бистабильной ячейки, созданная в Quartus II



Результат моделирования

Часть II

1. Используя триггеры из библиотеки, построить регистр в соответствии с заданием.
2. Промоделировать работу регистра с получением временных диаграмм.

3. Прощить проект в ПЛИС и провести проверку его работоспособности.

Пример выполнения

Исследование работы трехразрядного регистра с параллельной записью и считыванием.

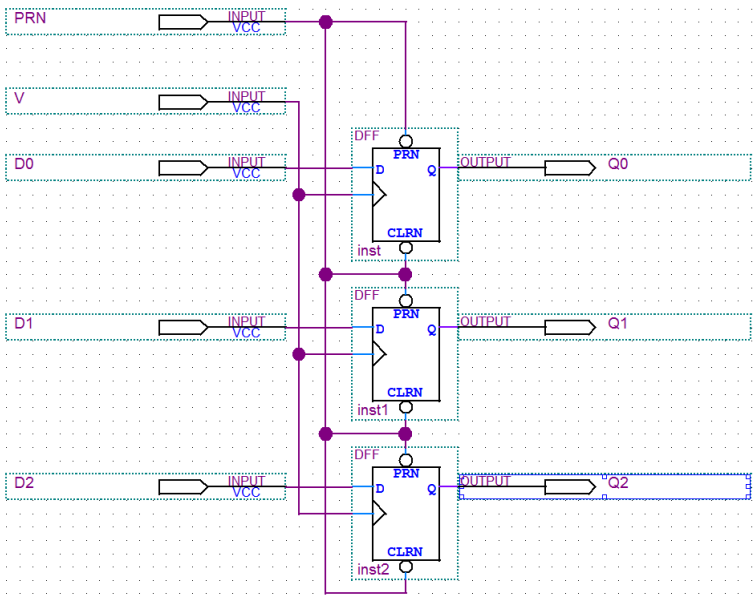
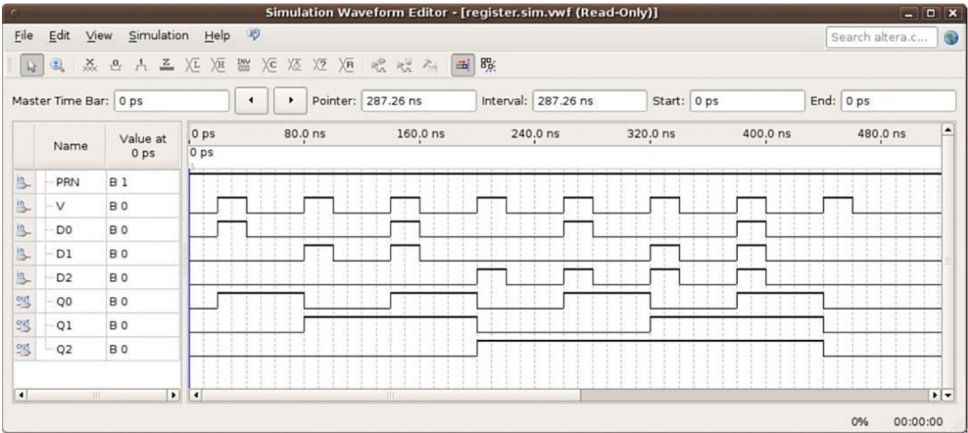


Схема регистра, созданная в Quartus II



Результат моделирования

Часть III

1. Согласно заданию осуществить синтез счетчика.
2. Промоделировать работу счетчика с получением временных диаграмм.
3. Прощить проект в ПЛИС и провести проверку его работоспособности.

Пример выполнения

Синтез суммирующего асинхронного счетчика по модулю 11.

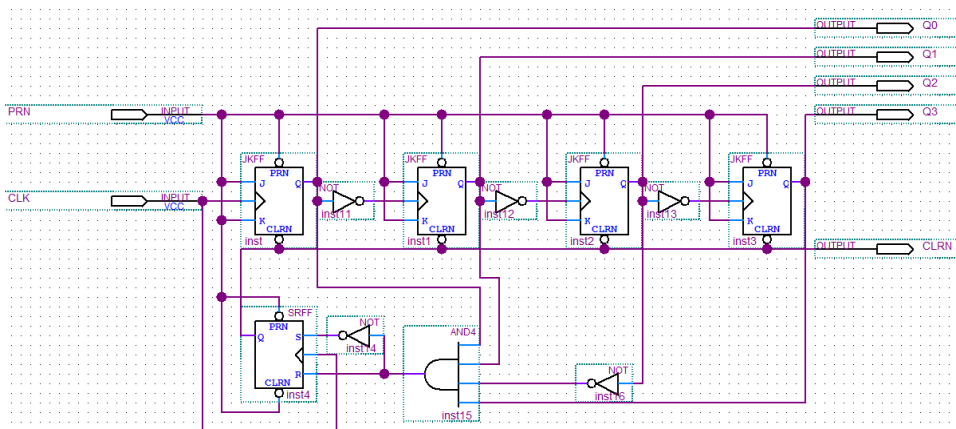
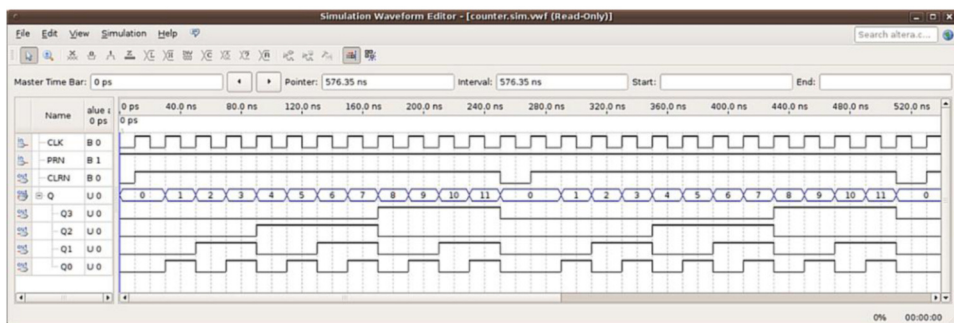


Схема счетчика, созданная в Quartus II



Результат моделирования

Варианты заданий (части I–III)

Вариант № 1

- 1) асинхронный R - S -триггер;
- 2) 4-разрядный регистр с параллельной записью и считыванием;
- 3) суммирующий асинхронный счетчик по модулю 12.

Вариант № 2

- 1) синхронный R - S -триггер;
- 2) 4-разрядный регистр с последовательной записью и параллельным считыванием;
- 3) суммирующий синхронный счетчик по модулю 13.

Вариант № 3

- 1) D -триггер;
- 2) 4-разрядный регистр с параллельной записью и последовательным считыванием;
- 3) вычитающий асинхронный счетчик по модулю 14.

Вариант № 4

- 1) асинхронный R - S -триггер;
- 2) 4-разрядный регистр с параллельной записью и последовательным считыванием;
- 3) суммирующий синхронный счетчик по модулю 19.

Вариант № 5

- 1) синхронный R - S -триггер;
- 2) 4-разрядный регистр с параллельной записью и считыванием;
- 3) вычитающий асинхронный счетчик по модулю 20.

Вариант № 6

- 1) D -триггер;
- 2) 4-разрядный регистр с последовательной записью и последовательным считыванием;
- 3) вычитающий синхронный счетчик по модулю 21.

Вариант № 7

- 1) асинхронный R - S -триггер;
- 2) 4-разрядный регистр с последовательной записью и параллельным считыванием;
- 3) вычитающий асинхронный счетчик по модулю 24.

Вариант № 8

- 1) синхронный R - S -триггер;
- 2) 4-разрядный регистр с последовательной записью и последовательным считыванием;
- 3) вычитающий синхронный счетчик по модулю 25.

Вариант № 9

- 1) D -триггер;
- 2) 4-разрядный регистр с последовательной записью и параллельным считыванием;
- 3) суммирующий асинхронный счетчик по модулю 26.

Вариант № 10

- 1) асинхронный R - S -триггер;
- 2) 4-разрядный регистр с последовательной записью и последовательным считыванием;
- 3) вычитающий синхронный счетчик по модулю 29.

Вариант № 11

- 1) синхронный R - S -триггер;
- 2) 4-разрядный регистр с параллельной записью и последовательным считыванием;
- 3) вычитающий синхронный счетчик по модулю 17.

Вариант № 12

- 1) D -триггер;
- 2) 4-разрядный регистр с параллельной записью и считыванием;
- 3) вычитающий асинхронный счетчик по модулю 28.

Вариант № 13

- 1) асинхронный R - S -триггер;
- 2) 4-разрядный регистр с параллельной записью и последовательным считыванием;
- 3) суммирующий синхронный счетчик по модулю 23.

Вариант № 14

- 1) синхронный R - S -триггер;
- 2) 4-разрядный регистр с последовательной записью и последовательным считыванием;
- 3) вычитающий синхронный счетчик по модулю 22.

Вариант № 15

- 1) D -триггер;
- 2) 4-разрядный регистр с параллельной записью и последовательным считыванием;
- 3) вычитающий асинхронный счетчик по модулю 16.

Вариант № 16

- 1) асинхронный R - S -триггер;
- 2) 4-разрядный регистр с параллельной записью и считыванием;
- 3) суммирующий асинхронный счетчик по модулю 30.

Вариант № 17

- 1) синхронный R - S -триггер;
- 2) 4-разрядный регистр с параллельной записью и последовательным считыванием;
- 3) суммирующий синхронный счетчик по модулю 27.

Вариант № 18

- 1) D -триггер;
- 2) 4-разрядный регистр с последовательной записью и параллельным считыванием;
- 3) вычитающий асинхронный счетчик по модулю 16.

Часть IV

1. В соответствии с заданием, используя триггеры из библиотеки, синтезировать четырехразрядный счетчик словарным методом. В задании указаны тип триггера, на котором нужно реализовать счетчик, и таблица истинности устройства.

2. Промоделировать работу счетчика с получением временных диаграмм.

3. Прощить проект в ПЛИС и провести проверку его работоспособности.

Варианты заданий (часть IV)

Вариант 1	<i>D</i> -триггер, таблица ЛР.2.1
Вариант 2	<i>D</i> -триггер, таблица ЛР.2.2
Вариант 3	<i>D</i> -триггер, таблица ЛР.2.3
Вариант 4	<i>D</i> -триггер, таблица ЛР.2.4
Вариант 5	<i>D</i> -триггер, таблица ЛР.2.5
Вариант 6	<i>D</i> -триггер, таблица ЛР.2.6
Вариант 7	<i>J-K</i> -триггер, таблица ЛР.2.1
Вариант 8	<i>J-K</i> -триггер, таблица ЛР.2.2
Вариант 9	<i>J-K</i> -триггер, таблица ЛР.2.3
Вариант 10	<i>J-K</i> -триггер, таблица ЛР.2.4
Вариант 11	<i>J-K</i> -триггер, таблица ЛР.2.5
Вариант 12	<i>J-K</i> -триггер, таблица ЛР.2.6
Вариант 13	<i>R-S</i> -триггер, таблица ЛР.2.1
Вариант 14	<i>R-S</i> -триггер, таблица ЛР.2.2
Вариант 15	<i>R-S</i> -триггер, таблица ЛР.2.3
Вариант 16	<i>R-S</i> -триггер, таблица ЛР.2.4
Вариант 17	<i>R-S</i> -триггер, таблица ЛР.2.5
Вариант 18	<i>R-S</i> -триггер, таблица ЛР.2.6

Таблица ЛР.2.1

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0

Таблица ЛР.2.2

Q_3	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	0
1	0	0	1
1	0	1	1
1	0	1	0
1	1	1	0
1	1	1	1
1	1	0	1
1	1	0	0
0	1	0	0
0	1	0	1
0	1	1	1
0	1	1	0
0	0	1	0
0	0	1	1
0	0	0	1

Таблица ЛР.2.3

Q_3	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	0
1	1	0	0
0	1	0	0
0	1	1	0
1	1	1	0
1	0	1	0
0	0	1	0
0	0	1	1
1	0	1	1
1	1	1	1
0	1	1	1
0	1	0	1
1	1	0	1
1	0	0	1
0	0	0	1

Таблица ЛР.2.4

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Таблица ЛР.2.5

Q_3	Q_2	Q_1	Q_0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0

Таблица ЛР.2.6

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	1

Отчет по лабораторной работе должен содержать: титульный лист, содержание, цель работы, задание, теоретическую и практическую часть, выводы и ссылки на использованную литературу.

Контрольные вопросы

1. *Какая логическая схема называется последовательностной?*
2. *Дайте определение триггеру.*
3. *Какие типы асинхронных триггеров вы знаете?*
4. *Какие типы синхронных триггеров вы знаете?*
5. *Какие устройства можно построить на основе триггеров?*
6. *Приведите классификацию регистров.*
7. *Приведите классификацию счетчиков.*

ПРИЛОЖЕНИЕ 1

ПЛИС – программируемая логическая интегральная схема (FPGA – field programmable gate array) – представляет собой цифровую интегральную микросхему, состоящую из программируемых логических блоков и программируемых соединений между ними.

ПЛИС применяются главным образом для:

- цифровой обработки сигналов;
- создания микропроцессорных систем и систем на кристалле;
- организации сетевых и коммуникационных функций при передаче данных;
- прототипирования различных цифровых систем и заказных интегральных микросхем.

Основными функциональными характеристиками ПЛИС являются:

- типовая логическая емкость в эквивалентных вентилях;
- количество логических элементов;
- количество триггеров;
- емкость встроенной памяти;
- наличие встроенных умножителей, процессорных ядер, различных контроллеров интерфейсов и т.д. и их количество.

Программируемость ПЛИС обеспечивается управлением (замыканием и размыканием) связями внутри и между логическими блоками. В зависимости от технологии изготовления ПЛИС управление осуществляется либо наращиваемыми перемычками, либо управляющими транзисторами, соединенными с ячейками памяти. В зависимости от содержимого ячейки памяти («0» или «1») управляющий транзистор будет закрыт либо открыт. Таким образом, программирование ПЛИС на ячейках памяти сводится к записи в ПЛИС битового массива, каждый элемент которого отражает состояние какого-то конкретного управляемого соединения и тем самым формирует схему электрических соединений элементов внутри ПЛИС.

Рассмотрим внутреннее устройство ПЛИС на примере семейства Cyclone II фирмы Altera. Расположим составные части ПЛИС иерархически в порядке укрупнения структур:

1. Логический элемент (ЛЭ) – базовый элемент ПЛИС. Состоит из 4-входовой таблицы соответствия, которая представляет собой логический блок табличного типа, триггера и управляющих мультиплексоров. Конкретное строение логического элемента может меняться в зависимости от конкретной модели ПЛИС (рис. П.1.1).

2. Логический массив состоит из 16 логических элементов и связей между ними. Количество логических элементов зависит от конкретной модели ПЛИС (рис. П.1.2).

3. Блок логических массивов (БЛМ) состоит из собственно логических массивов и связей как между ними внутри блока, так и между соседними блоками (рис. П.1.3).

4. Кристалл ПЛИС (рис. П.1.4). На кристалле ПЛИС расположены:

- блоки логических массивов, объединенные в столбцы;
- встроенные блоки памяти М4К;
- встроенные блоки умножителей;
- элементы ввода/вывода (IOE);
- схемы фазовой автоподстройки частоты (PLL).

Главным элементом ПЛИС, присутствующим во всех типах устройств всех производителей, является блок логических массивов. Остальные же элементы могут присутствовать или отсутствовать в зависимости от конкретной модели устройства.

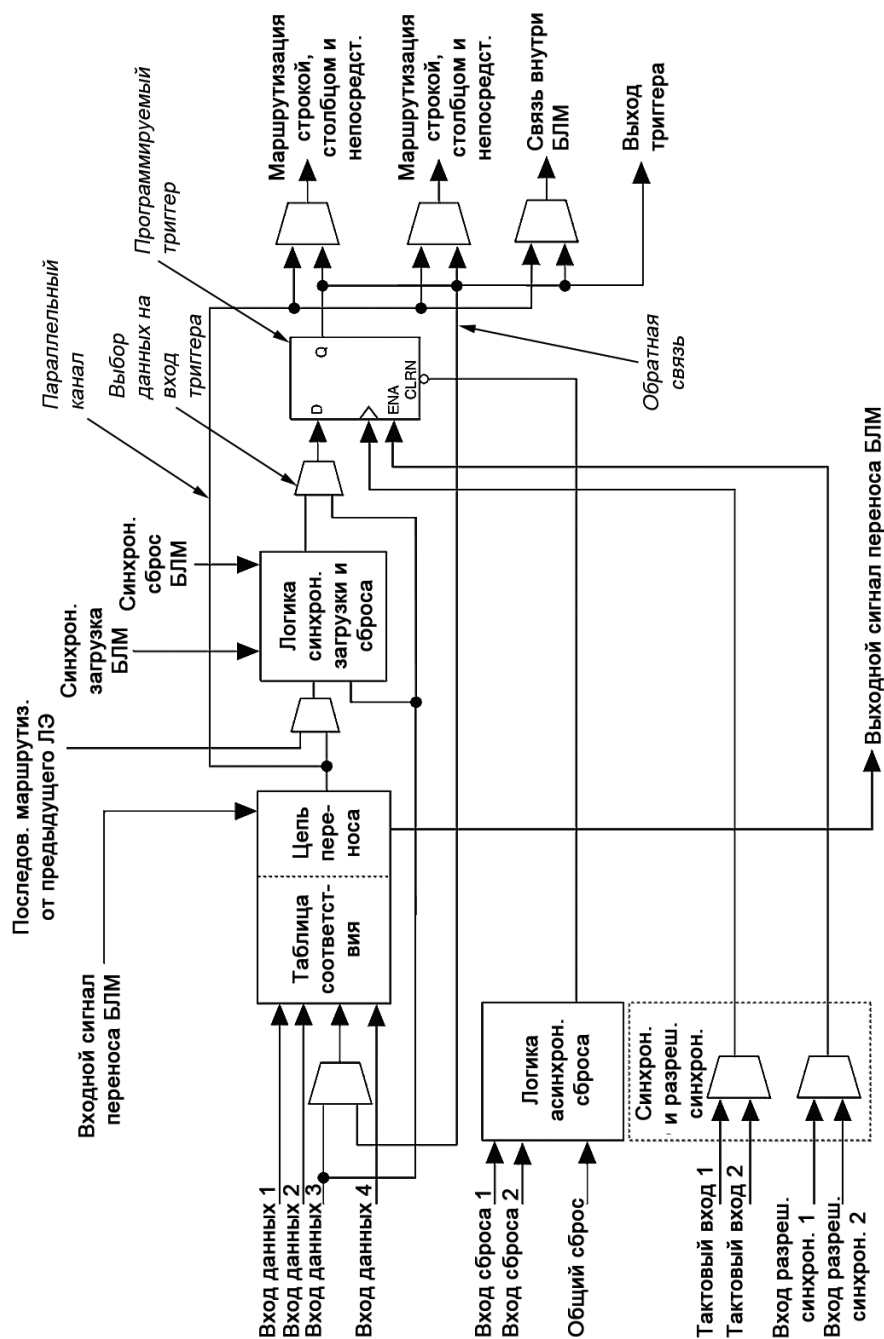


Рис. П.1.1.1. Логический элемент

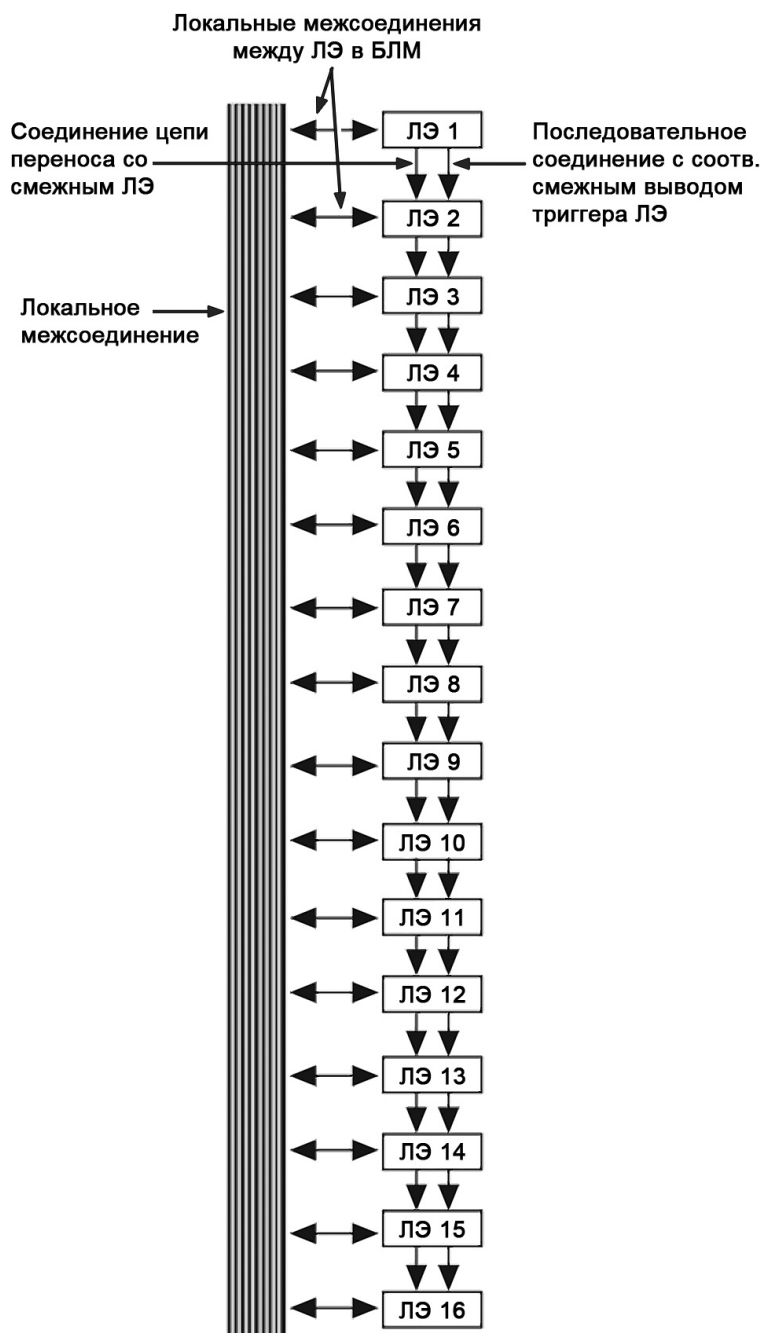


Рис. П.1.2. Логический массив

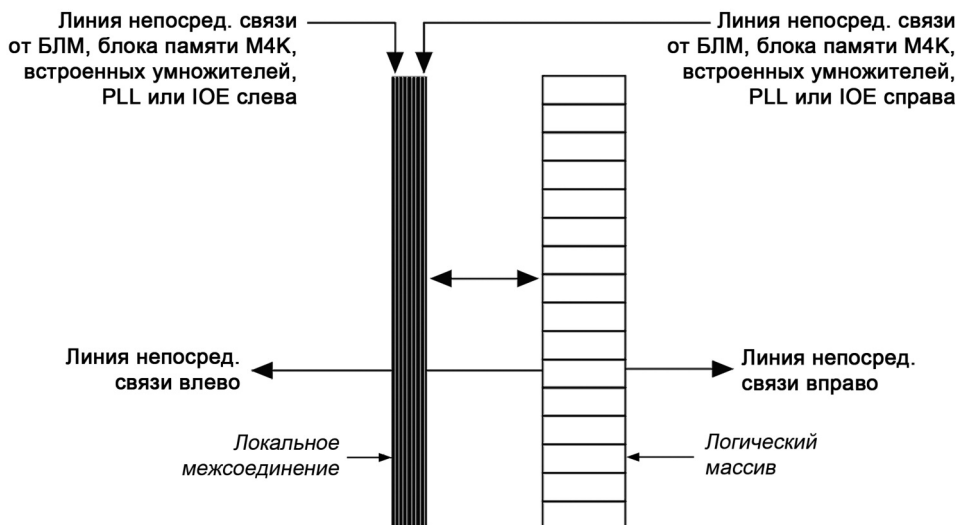


Рис. П.1.3. Блок логических массивов

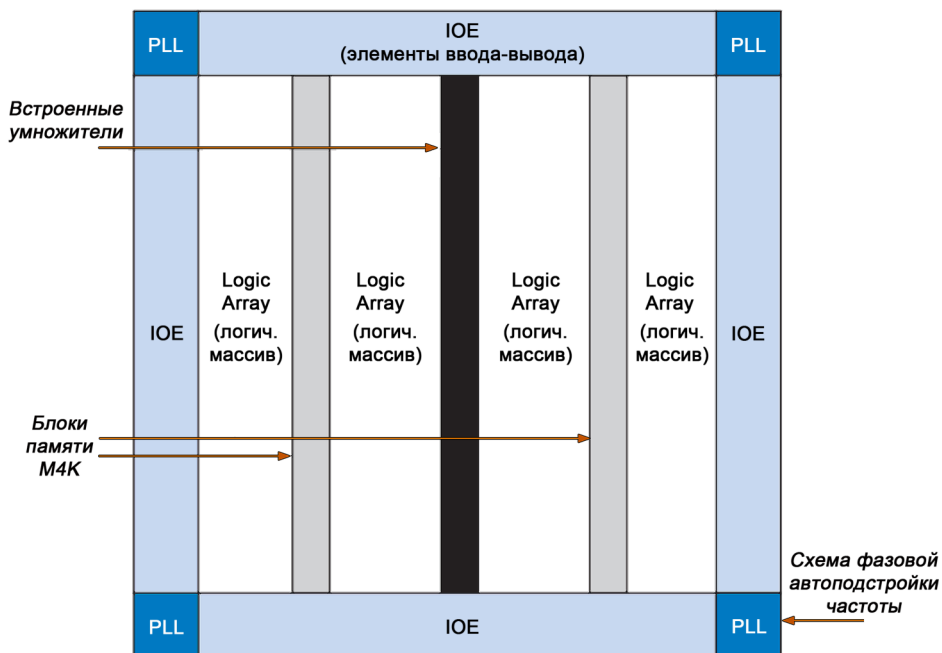


Рис. П.1.4. Кристалл ПЛИС

ПРИЛОЖЕНИЕ 2

Таблица П.2.1

Назначение выводов для переключателей-тумблеров

Имя сигнала	Номер вывода ПЛИС	Описание
SW [0]	PIN_N25	Переключатель-тумблер [0]
SW [1]	PIN_N26	Переключатель-тумблер [1]
SW [2]	PIN_P25	Переключатель-тумблер [2]
SW [3]	PIN_AE14	Переключатель-тумблер [3]
SW [4]	PIN_AF14	Переключатель-тумблер [4]
SW [5]	PIN_AD13	Переключатель-тумблер [5]
SW [6]	PIN_AC13	Переключатель-тумблер [6]
SW [7]	PIN_C13	Переключатель-тумблер [7]
SW [8]	PIN_B13	Переключатель-тумблер [8]
SW [9]	PIN_A13	Переключатель-тумблер [9]
SW [10]	PIN_N1	Переключатель-тумблер [10]
SW [11]	PIN_P1	Переключатель-тумблер [11]
SW [12]	PIN_P2	Переключатель-тумблер [12]
SW [13]	PIN_T7	Переключатель-тумблер [13]
SW [14]	PIN_U3	Переключатель-тумблер [14]
SW [15]	PIN_U4	Переключатель-тумблер [15]
SW [16]	PIN_V1	Переключатель-тумблер [16]
SW [17]	PIN_V2	Переключатель-тумблер [17]

Таблица П.2.2

Назначение выводов для переключателей-кнопок

Имя сигнала	Номер вывода ПЛИС	Описание
KEY [0]	PIN_G26	Кнопка [0]
KEY [1]	PIN_N23	Кнопка [1]
KEY [2]	PIN_P23	Кнопка [2]
KEY [3]	PIN_W26	Кнопка [3]

Таблица П.2.3

Назначение выводов для LED

Имя сигнала	Номер вывода ПЛИС	Описание
LEDR [0]	PIN_AE23	Красный светодиод [0]
LEDR [1]	PIN_AF23	Красный светодиод [1]
LEDR [2]	PIN_AB21	Красный светодиод [2]
LEDR [3]	PIN_AC22	Красный светодиод [3]
LEDR [4]	PIN_AD22	Красный светодиод [4]
LEDR [5]	PIN_AD23	Красный светодиод [5]
LEDR [6]	PIN_AD21	Красный светодиод [6]
LEDR [7]	PIN_AC21	Красный светодиод [7]
LEDR [8]	PIN_AA14	Красный светодиод [8]
LEDR [9]	PIN_Y13	Красный светодиод [9]
LEDR [10]	PIN_AA13	Красный светодиод [10]
LEDR [11]	PIN_AC14	Красный светодиод [11]
LEDR [12]	PIN_AD15	Красный светодиод [12]

Окончание таблицы П.2.3

Имя сигнала	Номер вывода ПЛИС	Описание
LEDR [13]	PIN_AE15	Красный светодиод [13]
LEDR [14]	PIN_AF13	Красный светодиод [14]
LEDR [15]	PIN_AE13	Красный светодиод [15]
LEDR [16]	PIN_AE12	Красный светодиод [16]
LEDR [17]	PIN_AD12	Красный светодиод [17]
LEDG [0]	PIN_AE22	Зеленый светодиод [0]
LEDG [1]	PIN_AF22	Зеленый светодиод [1]
LEDG [2]	PIN_W19	Зеленый светодиод [2]
LEDG [3]	PIN_V18	Зеленый светодиод [3]
LEDG [4]	PIN_U18	Зеленый светодиод [4]
LEDG [5]	PIN_U17	Зеленый светодиод [5]
LEDG [6]	PIN_AA20	Зеленый светодиод [6]
LEDG [7]	PIN_Y18	Зеленый светодиод [7]
LEDG [8]	PIN_Y12	Зеленый светодиод [8]

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексенко А.Г. Основы микросхемотехники. – М.: Юнимед-стайл, 2002. – 448 с.
2. Бусурин В.И., Можаяев В.А., Шеленков В.М. Микроэлектронные устройства систем управления летательных аппаратов: Учебное пособие / Под ред. В.И. Бусурина. – М.: Изд-во МАИ-ПРИНТ, 2011. – 200 с.
3. ГОСТ 17021-88. Микросхемы интегральные. Термины и определения. – М.: Изд-во стандартов, 1995.
4. ГОСТ 2.743-91. Обозначения условные графические в схемах. Элементы цифровой техники. – М.: Изд-во стандартов, 1995.
5. ГОСТ 2.755-87. Обозначения условные графические в схемах. Устройства коммутационные и контактные соединения. – М.: Изд-во стандартов, 1995.
6. Попов Б.Н. Цифровые устройства систем приводов летательных аппаратов: Учебное пособие. – М.: Изд-во МАИ-ПРИНТ, 2008. – 124 с.
7. Проектирование и испытание бортовых систем управления: Учебное пособие / Под редакцией А.С. Сырова. – М.: Изд-во МАИ-ПРИНТ, 2011. – 344 с.
8. Прянишников В.А. Электроника: Полный курс лекций. – 4-е изд. – СПб.: КОРОНА ПРИНТ, 2004.
9. Суворова Е.А., Шейнин Ю.Е. Проектирование цифровых систем на VHDL. – СПб.: БХВ-Петербург, 2003. – 576 с.
10. Угрюмов Е.П. Цифровая схемотехника. Учебное пособие. – 2-е изд. – СПб.: БХВ-Петербург, 2007. – 800 с.
11. Altera DE-2 Development and Education Board. User Manual. – Altera Corporation, 2012.

ДЛЯ ЗАМЕТОК

.....

ДЛЯ ЗАМЕТОК

.....

Тем. план 2016

Попов Борис Николаевич,
Цалкова Елизавета Эдуардовна,
Ханин Дмитрий Николаевич

**ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ СХЕМ И УСТРОЙСТВ
БОРТОВОЙ АВТОМАТИКИ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

Редакторы: М.С. Винниченко, Т.В. Кособокова
Компьютерная верстка Е.Э. Качаловой

Подписано в печать 19.08.2016
Бум. офсетная. Формат 60х84 1/16. Печать на ризографе.
Усл. печ. л. 5,35. Уч.-изд. л. 5,75. Тираж 100 экз.
Изд. № 427. Заказ 34.

Издательство МАИ
(МАИ), Волоколамское ш., д. 4, Москва, А-80, ГСП-3 125993

Отпечатано в МОКБ «Марс»
127473, г. Москва, 1-й Щемилловский переулок, 16