

Министерство образования Республики Беларусь

Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра вычислительных машин, систем и сетей

Дисциплина: Арифметические и логические основы  
цифровых устройств

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_ Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-  
УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 202 ПЗ

Студент

М. А. Бекетова

Руководитель

Ю. А. Луцик

МИНСК 2023

Министерство образования Республики Беларусь

Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы  
цифровых устройств

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Б. В. Никульшин  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

ЗАДАНИЕ  
по курсовой работе студента  
Бекетовой Марии Александровны

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-десятичных чисел»
2. Срок сдачи студентом законченной работы: до 20 мая 2023г.
3. Исходные данные к работе:
  - 3.1. Исходные сомножители:  $M_n = 61,47$ ;  $M_t = 74,34$ ;
  - 3.2. Алгоритм умножения: В;
  - 3.3. Метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в дополнительных кодах;
  - 3.4. Коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования:  $0_4 - 00$ ,  $1_4 - 01$ ,  $2_4 - 11$ ,  $3_4 - 10$ ;
  - 3.5. Тип синтезируемого умножителя: 2;
  - 3.6. Логический базис для реализации ОЧС: ИЛИ, Сумма по модулю 2, Тожественная единица; метод минимизации – карты Карно-Вейча;
  - 3.7. Логический базис для реализации ОЧУС: И, Сумма по модулю 2, Тожественная единица; метод минимизации – алгоритм Рота, карты Карно-Вейча.

**4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):**

Введение. 1. Разработка алгоритма умножения. 2. Разработка структурной схемы сумматора-умножителя. 3. Разработка функциональных схем основных узлов сумматора-умножителя. 4. Синтез комбинационных схем устройств на основе мультиплексоров. 5. Оценка результатов разработки. Заключение. Список литературы.

**5. Перечень графического материала:**

**5.1.** Умножитель-сумматор 2 типа. Схема электрическая структурная.

**5.2.** Однозарядный четверичный умножитель-сумматор. Схема электрическая функциональная.

**5.3.** Однозарядный четверичный сумматор. Схема электрическая функциональная.

**5.4.** Однозарядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.

**5.5.** Преобразователь множителя. Схема электрическая функциональная.

### КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов курсовой работы	Объем этапа, %	Срок выполнения этапа	Примечания
Разработка алгоритма умножения	10	10.02–20.02	
Разработка структурной схемы сумматора-умножителя	10	21.02–09.03	С выполнением чертежа
Разработка функциональных схем основных узлов сумматора- умножителя	50	10.03–30.04	С выполнением чертежей
Синтез комбинационных схем устройств на основе мультиплексоров	10	01.05–15.05	С выполнением чертежа
Завершение оформления пояснительной записки	20	15.05–20.05	

Дата выдачи задания: 10 февраля 2022г.

Руководитель

\_\_\_\_\_ Луцик Ю.А.

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ

\_\_\_\_\_ Бекетова М.А.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ .....	5
2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ .....	8
3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ .....	9
3.1. Логический синтез одноразрядного четверичного умножителя .....	9
3.2. Логический синтез одноразрядного четверичного сумматора-умножителя .....	28
3.3. Логический синтез преобразователя множителя .....	31
4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ .....	33
5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ .....	35
ЗАКЛЮЧЕНИЕ .....	36
СПИСОК ЛИТЕРАТУРЫ .....	37
ПРИЛОЖЕНИЕ А .....	38
ПРИЛОЖЕНИЕ Б .....	39
ПРИЛОЖЕНИЕ В .....	40
ПРИЛОЖЕНИЕ Г .....	41
ПРИЛОЖЕНИЕ Д .....	42
ПРИЛОЖЕНИЕ Е .....	43

## ВВЕДЕНИЕ

Курсовое проектирование является обязательным элементом подготовки специалиста с высшим образованием и одной из форм текущей аттестации студента по учебной дисциплине. Для студентов это первая работа такого рода и объёма. Она содержит результаты теоретических и экспериментальных исследований по дисциплине “Арифметические и логические основы вычислительной техники”, включает совокупность аналитических, расчётных, экспериментальных заданий и предполагает выполнение конструкторских работ и разработку графической документации.

Целью данной курсовой работы является проектирование такого цифрового устройства, как двоично-четверичный сумматор-умножитель (СУ). Сумматор является одним из центральных узлов арифметико-логического устройства (АЛУ) вычислительной машины, поэтому глубокое понимание принципов его работы критически важно для современного инженера. Для того чтобы спроектировать данное устройство, необходимо пройти несколько последовательных этапов разработки:

- Разработка алгоритма умножения чисел, по которому работает СУ
- Разработка структурной схемы СУ
- Разработка функциональной схемы основных узлов структурной схемы СУ
- Оценка результатов проделанной работы
- Оформление документации по проделанной работе

В ходе выполнения курсовой работы автором были пройдены все эти этапы. В настоящей пояснительной записке изложено краткое описание процесса проектирования и приведена разработанная автором графическая документация по структурной схеме и функциональным схемам основных её узлов.

## 1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

1. Перевод сомножителей из десятичной системы счисления в четверичную.

## Множественное

$$\begin{array}{r} \begin{array}{r} \begin{array}{r} - \quad 61 \end{array} \bigg| \begin{array}{r} 4 \end{array} \\ \begin{array}{r} 60 \end{array} \quad \begin{array}{r} - \quad 15 \end{array} \bigg| \begin{array}{r} 4 \end{array} \\ \hline \begin{array}{r} 1 \end{array} \quad \begin{array}{r} 12 \end{array} \quad \begin{array}{r} 3 \end{array} \\ \hline \begin{array}{r} 3 \end{array} \end{array} \qquad \begin{array}{r} 0,47 \\ * \quad 4 \\ \hline 1,88 \\ * \quad 4 \\ \hline 3,52 \\ * \quad 4 \\ \hline 2,08 \end{array}$$

$$M_{H_4} = 331,132.$$

В соответствии с заданной кодировкой множимого:

$$M_{H_{2/4}} = 101001,011011.$$

# Множитель

[illegible]

$$M_{T_4} = 1022,11.$$

В соответствии с обычной весомозначной кодировкой множителя:

$$M_{T_{2/4}} = 01001010,0101$$

2. Запишем сомножители в форме с плавающей запятой в дополнительном коде:

$$M_H = 0,101001011011 \quad P_{M_H} = 0.0010 + 03_{10} - \text{закодировано по заданию,}$$

$$M_T = 0,010010100101 \quad P_{M_T} = 0.0100 + 04_{10} - \text{закодировано традиционно.}$$

3. Умножение двух чисел с плавающей запятой на два разряда множителя одновременно в дополнительных кодах. Сложение порядков, формирование знака произведения, преобразование разрядов множителя согласно алгоритму и перемножение мантисс сомножителей.

Порядок произведения:

$$\begin{aligned} P_{M_H} &= 0.0010 \ 03_4 \\ P_{M_T} &= \frac{0.0100 \ 10_4}{0.0110 \ 13_4} \\ P_{M_H \cdot M_T} &= 0.0110 \ 13_4 \end{aligned}$$

Знак произведения определяется суммой по модулю два знаков сомножителей:

$$\text{зн } M_H \oplus \text{зн } M_T = 0 \oplus 0 = 0.$$

При умножении чисел в дополнительных кодах диада  $11(3_4)$  заменяется на триаду  $10\bar{1}$ , диада  $10(2_4)$  заменяется на триаду  $1\bar{1}0$ . Преобразованный множитель имеет вид  $M_T^p = 11\bar{1}211$ .

Перемножение мантисс по алгоритму «В» представлено в таблице 1.1

Таблица 1.1 — Перемножение мантисс

Четверичная С/С		Двоично-четверичная С/С		Комментарии
1		2		3
0,	000000000000	00,	000000000000000000000000	$\sum_0^q$
0,	000000331132	00,	000000000000101001011011	$\Pi_1^q = [M_H]_д$
0,	000000331132	00,	000000000000101001011011	$\sum_1^q$
0,	000003311320	00,	000000000010100101101100	$\sum_1^q * 4^1$
0,	000000331132	00,	000000000000101001011011	$\Pi_2^q = [M_H]_д$
0,	000010303112	00,	000000000100100010010111	$\sum_2^q$
0,	000103031120	00,	000000010010001001011100	$\sum_2^q * 4^1$
3,	333333002202	10,	101010101010000011110011	$\Pi_3^q = [-M_H]_д$
0,	000102033322	00,	000000010011001010101111	$\sum_3^q$
0,	001020333220	00,	000001001100101010111100	$\sum_3^q * 4^1$
3,	333332011010	10,	101010101011000101000100	$\Pi_4^q = [-2M_H]_д$
0,	001013010230	00,	000001000110000100111000	$\sum_4^q$
0,	010130102300	00,	000100011000010011100000	$\sum_4^q * 4^1$
0,	000000331132	00,	000000000000101001011011	$\Pi_5^q = [M_H]_д$
0,	010131100032	00,	000100011001010000001011	$\sum_5^q$
0,	101311000320	00,	010001100101000000101100	$\sum_5^q * 4^1$
0,	000000331132	00,	000000000000101001011011	$\Pi_6^q = [M_H]_д$
0,	101311332112	00,	010001100101101011010111	$\sum_6^q$

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение ( $M_{H4} \cdot M_{T4} = 0,101311332112$ ,  $P_{M_H} \cdot M_T = 7$ ) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

$$\begin{aligned} M_{H4} \cdot M_{T4} &= 1013113,32112 & P_{M_H} \cdot M_T &= 0; \\ M_{H10} \cdot M_{T10} &= 4567,8964. \end{aligned}$$

Результат прямого перемножения операндов дает следующее:  
 $M_{H10} \cdot M_{T10} = 61,47 * 74,34 = 4569,6798$ .

Абсолютная погрешность:  
 $\Delta = 4569,6798 - 4567,8964 = 1,7834$ .

Относительная погрешность:  
$$\delta = \frac{\Delta}{M_H \cdot M_T} = \frac{1,7834}{4569,6798} = 0,000390268 \quad (\delta = 0,0390268\%)$$

Эта погрешность получена за счёт приближённого перевода из десятичной системы счисления в четверичную обоих сомножителей, а также за счёт округления полученного результата произведения.



## 2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ

Структурная схема сумматора-умножителя второго типа для алгоритма умножения «В» представлена в приложении А.

Структурная схема второго типа строится на базе заданных узлов ОЧУС, ОЧС, формирователя дополнительного кода и регистра результата. Управление режимами работы схемы осуществляется внешним сигналом Mul/sum, который определяет вид текущей арифметической операции (умножение или суммирование).

Если устройство работает как сумматор, то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода  $F_2$  поступает «1».

Если устройство работает как умножитель, то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК  $F_2$  поступает «0».

Принцип работы ФДК, в зависимости от управляющих сигналов, приведён в таблице 2.1.

Таблица 2.1 – Режимы работы формирователя дополнительного кода

Сигналы на входах ФДК		Результат на выходах ФДК
$F_1$	$F_2$	
0	0	Дополнительный код множимого
0	1	Дополнительный код слагаемого
1	0	Меняется знак Мн
1	1	Меняется знак слагаемого

### 3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ

#### 3.1 Логический синтез одноразрядного четверичного сумматора-умножителя

ОЧУС – это комбинационное устройство, имеющее 6 входов (2 разряда из регистра Мн, 2 разряда из регистра Мт, вход переноса и управляющий вход  $H$ ) и 3 выхода.

Разряды множимого закодированы: 0 – 00, 1 – 01, 2 – 11, 3 – 10;

Разряды множителя закодированы: 0 – 00, 1 – 01, 2 – 10, 3 – 11;

Управляющий вход  $H$  определяет тип операции: «0» – умножение закодированных цифр, поступивших на информационные входы, и добавление переноса; «1» – вывод на выходы без изменения значений разрядов, поступивших из регистра Мн.

В таблице 3.1.1 выделено 36 безразличных наборов, т.к. на входы ОЧУС из разрядов Мт не может поступить код «11», при работе ОЧУС как сумматора на вход переноса не может поступить 1, а при умножении на 0 или 1 на вход переноса также не может поступить 1.

Таблица 3.1.1 — Таблица истинности ОЧУС

Пер.	Мн.		Мт.		Упр.	Перенос	Результат		Пример операции в четверичной с/с
$p$	$x_1$	$x_2$	$y_1$	$y_2$	$H$	$P$	$Q_1$	$Q_2$	
1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	$0 * 0 + 0 = 00$
0	0	0	0	0	1	0	0	0	Выход – код «00»
0	0	0	0	1	0	0	0	0	$0 * 1 + 0 = 00$
0	0	0	0	1	1	0	0	0	Выход – код «00»
0	0	0	1	0	0	0	0	0	$0 * 2 + 0 = 00$
0	0	0	1	0	1	0	0	0	Выход – код «00»
0	0	0	1	1	0	х	х	х	$0 * 3 + 0 = 00$
0	0	0	1	1	1	х	х	х	Выход – код «00»
0	0	1	0	0	0	0	0	0	$1 * 0 + 0 = 00$
0	0	1	0	0	1	0	0	1	Выход – код «01»
0	0	1	0	1	0	0	0	1	$1 * 1 + 0 = 01$
0	0	1	0	1	1	0	0	1	Выход – код «01»
0	0	1	1	0	0	0	1	1	$1 * 2 + 0 = 02$
0	0	1	1	0	1	0	0	1	Выход – код «01»
0	0	1	1	1	0	х	х	х	$1 * 3 + 0 = 03$
0	0	1	1	1	1	х	х	х	Выход – код «01»
0	1	0	0	0	0	0	0	0	$3 * 0 + 0 = 00$
0	1	0	0	0	1	0	1	0	Выход – код «03»
0	1	0	0	1	0	0	1	0	$3 * 1 + 0 = 03$

Продолжение таблицы 3.1.1

0	1	0	0	1	1	0	1	0	Выход – код «03»
0	1	0	1	0	0	1	1	1	$3 * 2 + 0 = 12$
0	1	0	1	0	1	0	1	0	Выход – код «03»
0	1	0	1	1	0	х	х	х	$3 * 3 + 0 = 21$
0	1	0	1	1	1	х	х	х	Выход – код «03»
0	1	1	0	0	0	0	0	0	$2 * 0 + 0 = 00$
0	1	1	0	0	1	0	1	1	Выход – код «02»
0	1	1	0	1	0	0	1	1	$2 * 1 + 0 = 02$
0	1	1	0	1	1	0	1	1	Выход – код «02»
0	1	1	1	0	0	1	0	0	$2 * 2 + 0 = 10$
0	1	1	1	0	1	0	1	1	Выход – код «02»
0	1	1	1	1	0	х	х	х	$2 * 3 + 0 = 12$
0	1	1	1	1	1	х	х	х	Выход – код «02»
1	0	0	0	0	0	х	х	х	$0 * 0 + 1 = 01$
1	0	0	0	0	1	х	х	х	Выход – код «00»
1	0	0	0	1	0	х	х	х	$0 * 1 + 1 = 01$
1	0	0	0	1	1	х	х	х	Выход – код «00»
1	0	0	1	0	0	0	0	1	$0 * 2 + 1 = 01$
1	0	0	1	0	1	х	х	х	Выход – код «00»
1	0	0	1	1	0	х	х	х	$0 * 3 + 1 = 01$
1	0	0	1	1	1	х	х	х	Выход – код «00»
1	0	1	0	0	0	х	х	х	$1 * 0 + 1 = 01$
1	0	1	0	0	1	х	х	х	Выход – код «01»
1	0	1	0	1	0	х	х	х	$1 * 1 + 1 = 02$
1	0	1	0	1	1	х	х	х	Выход – код «01»
1	0	1	1	0	0	0	1	0	$1 * 2 + 1 = 03$
1	0	1	1	0	1	х	х	х	Выход – код «01»
1	0	1	1	1	0	х	х	х	$1 * 3 + 1 = 10$
1	0	1	1	1	1	х	х	х	Выход – код «01»
1	1	0	0	0	0	х	х	х	$3 * 0 + 1 = 01$
1	1	0	0	0	1	х	х	х	Выход – код «03»
1	1	0	0	1	0	х	х	х	$3 * 1 + 1 = 10$
1	1	0	0	1	1	х	х	х	Выход – код «03»
1	1	0	1	0	0	1	1	0	$3 * 2 + 1 = 13$
1	1	0	1	0	1	х	х	х	Выход – код «03»
1	1	0	1	1	0	х	х	х	$3 * 3 + 1 = 22$
1	1	0	1	1	1	х	х	х	Выход – код «03»
1	1	1	0	0	0	х	х	х	$2 * 0 + 1 = 01$
1	1	1	0	0	1	х	х	х	Выход – код «02»
1	1	1	0	1	0	х	х	х	$2 * 1 + 1 = 03$
1	1	1	0	1	1	х	х	х	Выход – код «02»

Продолжение таблицы 3.1.1

1	1	1	1	0	0	1	0	1	$2 * 2 + 1 = 11$
1	1	1	1	0	1	x	x	x	Выход – код «02»
1	1	1	1	1	0	x	x	x	$2 * 3 + 1 = 13$
1	1	1	1	1	1	x	x	x	Выход – код «02»

### Минимизация функции $P$ :

Минимизацию функции  $P$  проведем с помощью карт Вейча. Для функции  $P$  заполненная карта приведена на рисунке 3.1.1. В рисунках 3.1.1 – 3.1.3 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

$x_1$									
$p$	x	x	x	x	x	x	x	$y_1$	$H$
	x	x	x	x	x	x	x		
	x	x	x	x	x	x	x		
	1	x	x	1	0	x	x		
	1	x	x	1	0	x	x	0	$H$
	0	x	x	0	0	x	x	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
$x_2$									
$y_2$				$y_2$					

Рисунок 3.1.1 — Минимизация функции  $P$  картой Вейча

Следовательно:

$$P = x_1 y_1 \bar{H}$$

Запишем результат базисе И – Тожественная единица – Сумма по модулю 2:

$$P = x_1 y_1 (H \oplus 1)$$

Эффективность минимизации можно оценить отношением числа входов схем, реализующих переключательную функцию до и после минимизации:

$$K = \frac{6*4+4+2}{4} = 7,5$$

### Минимизация функции $Q_I$ :

Определим множество единичных кубов:

$$L = \left\{ \begin{array}{l} 001100, 010001, 010010, \\ 010011, 010100, 010101, \\ 011001, 011010, 011011, \\ 011101, 101100, 110100 \end{array} \right\}$$

Определим множество безразличных кубов:

$$N = \left\{ \begin{array}{l} 000110, 000111, 001110, 001111, \\ 010110, 010111, 011110, 011111, \\ 100000, 100001, 100010, 100011, \\ 100101, 100110, 100111, 101000, \\ 101001, 101010, 101011, 101101, \\ 101110, 101111, 110000, 110001, \\ 110010, 110011, 110101, 110110, \\ 110111, 111000, 111001, 111010, \\ 111011, 111101, 111110, 111111 \end{array} \right\}$$

После предварительного склеивания в исходных множествах кубов получим:

$$L = \{ 011101, 110100, x01100, \\ 01x010, 01010x, 01x0x1 \}$$

$$N = \{ 1xx110, 0xx11x, \\ 1xx1x1, 1xx0xx \}$$

Сформируем множество  $C_0 = L \cup N$ :

$$C_0 = \{ 011101, 110100, x01100, 01x010, 01010x, \\ 01x0x1, 1xx1x1, 1xx0xx, 1xx1x1, 1xx0xx \}$$

Первым этапом алгоритма Рота является нахождение множества простых импликант.

Для реализации этого этапа будем использовать операцию умножения (\*) над множествами  $C_0$ ,  $C_I$  и т. д., пока в результате операции будут образовываться новые кубы большей размерности.

Первый шаг умножения ( $C_0 * C_0$ ) приведён в таблице 3.1.2.

Таблица 3.1.2 – Поиск простых импликант ( $C_0 * C_0$ )

$C_0 * C_0$	011101	110100	x01100	01x010	01010x	01x0x1	1xx110	0xx11x	1xx1x1	1xx0xx
011101	-									
110100		-								
x01100			-							
01x010				-						
01010x	01y101	y10100			-					
01x0x1	011y01			01x01y	010y01	-				
1xx110		1101y0	1011y0				-			
0xx11x	0111y1		0011y0	01xy10	0101yx	01xy11	yxx110	-		
1xx1x1	y11101	11010y	10110y		y10101		1xx11y	yxx111	-	
1xx0xx		110y00	101y00	y1x010		y1x0x1	1xxy10		1xxyx1	-
$A_1$	01x101 011x01 0111x1 x11101	x10100 1101x0 11010x 110x00	1011x0 0011x0 10110x 101x00	01x01x 01xx10 x1x010	010x01 0101xx x10101	01xx11 x1x0x1	xxx110 1xx11x 1xxx10	xxx111	1xxxx1	$\emptyset$

Множество  $Z_0$  кубов, не участвовавших в образовании новых кубов, пустое.

$$B_1 = C_0 = \{ 011101, 110100, x01100, 01x010, 01010x, 01x0x1, 1xx110, 0xx11x, 1xx1x1, 1xx0xx \}$$

В результате этой операции сформируется новое множество кубов:

$$C_1 = \{ 01x101, 011x01, 0111x1, x11101, x10100, 1101x0, 11010x, 110x00, 1011x0, 0011x0, 10110x, 101x00, 01x01x, 01xx10, x1x010, 010x01, 0101xx, x10101, 01xx11, x1x0x1, xxx110, 1xx11x, 1xxx10, xxx111, 1xxxx1, x01100, 0xx11x, 1xx0xx \}$$

В таблице 3.1.3 приведён следующий шаг поиска простых импликант с помощью операции  $C_1 * C_1$ .

Таблица 3.1.3 – Поиск простых импликант ( $C_1 * C_1$ )

$C_1 * C_1$	01x101	011x01	0111x1	x11101	x10100	1101x0	11010x	110x00	1011x0	0011x0	10110x	11x00	01x01x
01x101	-												
011x01		-											
0111x1			-										
x11101				-									
x10100					-								
1101x0						-							
11010x							-						
110x00								-					
1011x0									-				
0011x0									y011x0	-			
10110x											-		
101x00												-	
01x01x													-
01xx10													
x1x010													
010x01		01yx01											
0101xx			01y1x1			y101x0	y1010x						010y1x
x10101				x1y101	x1010y								
01xx11	01x1y1	011xy1											
x1x0x1	01xy01		011yx1	x11y01									
xxx110					x101y0								01xy10
1xx11x							1101yx				1011yx		
1xxx10								110xy0				101xy0	y1x010
xxx111	01x1y1			x111y1									01xy11
1xxxx1	y1x101	y11x01	y111x1			1101xy		110x0y	1011xy			101x0y	y1x011
x01100													
0xx11x	01x1y1												01xy1x
1xx0xx						110yx0	110y0x		101yx0		101y0x		y1x01x
$A_2$	01x1x1 01xx01 01x1x1 x1x101 01x1x1	01xx01 011xx1 x11x01	01x1x1 011xx1 x111x1	x1x101 x11x01 x111x1	x1010x x101x0	x101x0 1101xx 110xx0	x1010x 1101xx 110x0x	110xx0 110x0x	x011x0 1011xx 101xx0	∅	1011xx 101x0x	101xx0 101x0x	010x1x 01xx10 x1x010 01xx11 x1x011 01xx1x x1x01x



### Продолжение таблицы 3.1.3

C1*C1	01xx10	x1x010	010x01	0101xx	x10101	01xx11	x1x0x1	xxx110	1xx11x	1xxx10	xxx111	1xxxx1
01x101												
011x01												
0111x1												
x11101												
x10100												
1101x0												
11010x												
110x00												
1011x0												
0011x0												
10110x												
101x00												
01x01x												
01xx10	-											
x1x010		-										
010x01			-									
0101xx				-								
x10101					-							
01xx11	01xx1y	01x01y	010xy1			-						
x1x0x1	01x01y	x1x01y		010yx1	x10y01		-					
xxx110		x1xy10				01x11y		-				
1xx11x	y1x110	11xy10		y1011x		y1x111	11xy11		-			
1xxx10	y1xx10						11x01y			-		
xxx111	01x11y				x101y1		x1xy11	xxx11y		1xx11y	-	
1xxxx1		11x01y	y10x01	y101x1		y1xx11		1xx11y		1xxx1y		-
x01100								x011y0				
0xx11x		01xy10					01xy11		yxx11x	yxx110		yxx111
1xx0xx	y1x010					y1x011		1xxy10	1xxy1x		1xxy11	
A2	01xx1x 01x01x x1x110 x1xx10 01x11x x1x010	01x01x x1x01x x1xx10 11xx10 11x01x 01xx10	010xx1 x10x01	010xx1 x1011x x101x1	x10x01 x101x1	01x11x x1x111 x1xx11 x1x011	11xx11 11x01x x1xx11 01xx11	xxx11x 1xx11x x011x0 1xxx10	xxx11x 1xxx1x	1xx11x 1xxx1x xxx110	1xxx11	xxx111

Множество  $Z_1$  кубов, не участвовавших в образовании новых кубов, пустое.

$B_2 = C_1 = \{ 01x101, 011x01, 0111x1, x11101, x10100, 1101x0, 11010x, 110x00, 1011x0, 0011x0, 10110x, 101x00, 01x01x, 01xx10, x1x010, 010x01, 0101xx, x10101, 01xx11, x1x0x1, xxx110, 1xx11x, 1xxx10, xxx111, 1xxxx1, x01100, 0xx11x, 1xx0xx \}$

В результате образовалось множество  $C_2$ :

$C_2 = \{ 01x1x1; 01xx01; x1x101; 011xx1; x11x01; x111x1; x1010x; x101x0; 1101xx; 110xx0; 110x0x; x011x0; 1011xx; 101xx0; 101x0x; 01xx1x; x1x01x; x1xx10; 010xx1; x10x01; x101x1; x1xx11; xxx11x; 1xxx1x; 0101xx; x1x0x1; 1xxxx1; 1xx0xx \}$

В таблице 3.1.4 приведён следующий шаг поиска простых импликант – операция  $C_2 * C_2$ .

Таблица 3.1.4 – Поиск простых импликант ( $C_2 * C_2$ )

$C_2 * C_2$	01x1x1	01xx01	x1x101	011xx1	x11x01	x111x1	x1010x	x101x0	1101xx	110xx0	110x0x	x011x0
01x1x1	-											
01xx01		-										
x1x101			-									
011xx1				-								
x11x01					-							
x111x1						-						
x1010x							-					
x101x0								-				
1101xx									-			
110xx0										-		
110x0x											-	
x011x0												-
1011xx												
101xx0												
101x0x												
01xx1x		01xxy1										
x1x01x												
x1xx10												
010xx1				01yxx1								
x10x01					x1yx01							
x101x1						x1y1x1		x101xy				
x1xx11		01xxy1	x1x1y1		x11xy1							
xxx11x			x1x1y1				x101yx					
1xxx1x											110хух	
0101xx									y101xx			
x1x0x1	01xyx1		x1xy01			x11yx1						
1xxxx1	y1x1x1	y1xx01		y11xx1						110хху		
1xx0xx									110yxx			
A3	01xxx1 x1x1x1	01xxx1 01xxx1 x1xx01	x1x1x1 x1x1x1 x1xx01	01xxx1 x11xx1	x1xx01 x11xx1	x1x1x1 x11xx1	x101xx	x101xx	x101xx 110xxx	110xxx	110xxx	Ø

# Продолжение таблицы 3.1.4

C2*C2	1011xx	101xx0	101x0x	01xx1x	x1x01x	x1xx10	010xx1	x10x01	x101x1	x1xx11	xxx11x	1xxx1x
01x1x1												
01xx01												
x1x101												
011xx1												
x11x01												
x111x1												
x1010x												
x101x0												
1101xx												
110xx0												
110x0x												
x011x0												
1011xx	-											
101xx0		-										
101x0x			-									
01xx1x				-								
x1x01x					-							
x1xx10						-						
010xx1							-					
x10x01								-				
x101x1									-			
x1xx11						x1xx1y		x10xy1		-		
xxx11x					x1xy1x						-	
1xxx1x			101xyx	y1xx1x								-
0101xx												
x1x0x1						x1x01y			x10yx1		x1xy11	
1xxxx1		101xxy		y1xx11		11xx1y	y10xx1					
1xx0xx	101yxx			y1x01x							1xxy1x	
A3	110xxx	110xxx	110xxx	x1xx1x x1xx11 x1x01x	x1xx1x	x1xx1x x1x01x 11xx1x	x10xx1	x10xx1	x10xx1	Ø	x1xx11 1xxx1x	Ø

Получено множество  $Z_2 = \{ x011x0 \}$ .

$B_3 = \{ 01x1x1, 01xx01, x1x101, 011xx1, x11x01, x111x1, x1010x, x101x0, 1101xx, 110xx0, 110x0x, 1011xx, 101xx0, 101x0x, 01xx1x, x1x01x, x1xx10, 010xx1, x10x01, x101x1, x1xx11, xxx11x, 1xxx1x, 0101xx, x1x0x1, 1xxxx1, 1xx0xx \}$

В результате образовалось множество  $C_3$ :

$C_3 = \{ 01xxx1, x1x1x1, x1xx01, x11xx1, x101xx, 110xxx, 101xxx, x1xx1x, x10xx1, 1xxx1x, xxx11x, x1x0x1, 1xxxx1, 1xx0xx \}$

В таблице 3.1.5 приведён следующий шаг поиска простых импликант – операция  $C_3 * C_3$ .

Таблица 3.1.5 – Поиск простых импликант ( $C_3 * C_3$ )

$C_3 * C_3$	01xxx1	x1x1x1	x1xx01	x11xx1	x101xx	110xxx	101xxx	x1xx1x	x10xx1
01xxx1	-								
x1x1x1		-							
x1xx01			-						
x11xx1				-					
x101xx					-				
110xxx						-			
101xxx							-		
x1xx1x			x1xxy1					-	
x10xx1				x1yxx1					-
1xxx1x									
xx11x									
x1x0x1		x1xyx1							
1xxxx1	y1xxx1								
1xx0xx									
$A_4$	x1xxx1	x1xxx1	x1xxx1	x1xxx1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Получено множество  $Z_3 = \{ x011x0, x101xx, 110xxx, 101xxx, 1xxx1x, xxx11x, 1xx0xx \}$

$B_4 = \{ 01xxx1, x1x1x1, x1xx01, x11xx1, x1xx1x, x10xx1, x1x0x1, 1xxxx1 \}$

В результате образовалось множество  $C_4$ :

$C_4 = \{ x1xxx1, x1xx1x, 1xxxx1 \}$

В таблице 3.1.6 приведён следующий шаг поиска простых импликант – операция  $C_4 * C_4$ .

Таблица 3.1.6 – Поиск простых импликант ( $C_4 * C_4$ )

$C_4 * C_4$	x1xxx1
x1xxx1	-
x1xx1x	
1xxxx1	
$A_5$	$\emptyset$

Новых кубов не образовалось.

Получено множество  $Z_4 = \{ x1xxx1, x1xx1x, 1xxxx1 \}$

На этом заканчивается этап поиска простых импликант.

Конечное множество простых импликант:

$Z = \{ x011x0, x101xx, 110xxx, 101xxx, 1xxx1x, xxx11x, 1xx0xx, x1xxx1, x1xx1x, 1xxxx1 \}$

Следующий этап – поиск L-экстремалей на множестве простых импликант (таблица 3.1.7). Для этого используется операция # (вычитание).

Таблица 3.1.7 – Поиск L-экстремалей ( $Z \# (Z - z)$ )

$Z \# (Z - z)$	x011x0	x101xx	110xxx	101xxx	1xxx1x	xxx11x	1xx0xx	x1xxx1	x1xx1x	1xxxx1
x011x0	-	zyyzz1 x101xx	zyy0z1 110xxx	zzz0z1 1010xx 101xx1	z100z1 11xx1x 1x0x1x 1xx01x 1xxx11	z10zz1 x1x11x xx011x xxx111	z10yz1 1xx0xx	zy00zy x1xxx1	zy00z1 x1xx1x	z100zy 1xxxx1
x101xx	zyyzzz x011x0	-	zzz0zz 1100xx	zyyyzz zyy0zz 1010xx 101xx1	zz10zz z0z0zz z01yzz z010zz 111x1x 11x01x 100x1x 1x001x 1xx01x 10xx11 1x1x11 1xx011	zz1zzz z0zzzz z01zzz x1111x x0011x x0x111 xx1111	z01yzz 1xx0xx	zz10zz x11xx1 x1x0x1	zz10zz x11x1x x1x01x	z010zz 10xxx1 1x1xx1 1xx0x1
110xxx	0yyzzz x011x0	0zzzzz 0101xx	-	zyyzzz zyyzzz 1010xx 101xx1	zzyzzz zz1zzz zyzzzz z0zzzz z01zzz zy1zzz z0yzzz z01zzz 111x1x	0zyzzz 0yzzzz 0y1zzz 00yzzz x1111x x0011x x0x111 xx1111	z01zzz 10x0xx 1x10xx	0zyzzz 0z1zzz x11xx1 01x0x1 x110x1	0zyzzz 0z1zzz x11x1x 01x01x x1101x	zy1zzz z0yzzz z01zzz 10xxx1 1x1xx1 10x0x1 1x10x1

Продолжение таблицы 3.1.7

					11101x 100x1x 10001x 10x01x 1x101x 10xx11 1x1x11 10x011 1x1011					
101xxx	0zzzzz 0011x0	yyyzzz 0101xx	zyyzzz 1100xx	-	zyzzzz zyzzzz zzyzzz zzyzzz zz0zzz z1zzzz zz0zzz z1zzzz zz0zzz z0zzzz x1111x 111x1x 11101x 100x1x 10001x 10001x 11101x 100x11 111x11 100011 111011	0yzzzz 0zyzzz 0z0zzz 01zzzz x1111x x0011x 00x111 x00111 0x1111 x11111	zz0zzz z1zzzz 1000xx 1110xx	0yzzzz yy0zzz 0yzzzz x11xx1 01x0x1 x110x1	0yzzzz yy0zzz 0yzzzz x11x1x 01x01x x1101x	zz0zzz z1zzzz zz0zzz z1zzzz 100xx1 111xx1 1000x1 1110x1
1xxx1x	yzzz0z 0011x0	yzzz0z 0101xx	zzzz0z 11000x	zzzz0z zzzz0z 10100x 101x01	-	0zzzzz 0zzzzz yzzzzz 0zzzzz yzzzzz 0zzzzz 01111x 00011x 00x111 000111 0x1111 011111	zzzz0z zzzz0z 10000x 11100x	0zzz0z yzzz0z 0zzz0z 011xx1 x11x01 01x0x1 0110x1 x11001	0zzzzz yzzzzz 0zzzzz 011x1x 01x01x 01101x	zzzz0z zzzz0z zzzz0z zzzz0z 100x01 111x01 100001 111001
xxx11x	zzzz0z 001100	zzzz0z 01010x	zzzyyz 11000x	zzzyyz zzz0yz 10100x 101x01	-	zzz0zz zzzyzz zzz0zz zzzyzz zzzyzz zzz0zz zzz0zz zzzyzz zzzyzz 11101x 11101x 10001x 10001x 10001x 11101x 100011 111011 100011 111011	zzzyyz zzzyyz 10000x 11100x	zzz00z zzz0yz zzzy0z zzzy0z zzzyyz 0110x1 011x01 x11x01 01x0x1 0110x1 x11001	zzz0zz zzzyzz zzzyzz 01101x 01x01x 01101x	zzz0yz zzz0yz zzzyyz zzzyyz 100x01 111x01 100001 111001

## Продолжение таблицы 3.1.7

[illegible]



В таблице 3.1.7 из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты  $Z\#(Z-z)$ .

Результат операции (последняя строка таблицы) указывает на то, что  $L$ -экстремальными стали следующие простые импликанты:

$\{x011x0, x101xx, xxx11x, 1xx0xx, x1xxx1, x1xx1x, 1xxxx1\}$ .

Необходимо проверить, нет ли среди полученных  $L$ -экстремалей таких, которые стали  $L$ -экстремальными за счёт безразличных кубов. Для этого в таблице 3.1.8 из кубов множества  $L$  вычитаются остатки простых импликант, полученные в таблице 3.1.7 (результат выполнения операции  $Z\#(Z-z)$ ).

По результатам таблицы 3.1.8  $L$ -экстремальными, не связанными с безразличными наборами, стали кубы  $x011x0, x101xx, x1xxx1, x1xx1x$  (остаток от вычитания из них всех остальных простых импликант –  $001100, 010100, 011001, 011101, 010001, 011010$  и  $010010$  соответственно – относятся к множеству единичных наборов  $L$  исходного задания функции). Эти кубы обязательно должны войти в минимальное покрытие.

Таблица 3.1.8 – Проверка L-экстремалей

$z\#(Z-z)$ $\cap L$	001100	010001	010010	010011	010100	010101	011001	011010	011011	011101	101100	110100
001100	001100	0yуу0y Ø	0уууу0 Ø	0ууууу Ø	0уу100 Ø	0уу10у Ø	0у1у0у Ø	0у1уу0 Ø	0у1ууу Ø	0у110у Ø	у01100 Ø	ууу100 Ø
010100	0уу100 Ø	010у0у Ø	010уу0 Ø	010ууу Ø	010100	01010у Ø	01уу0у Ø	01ууу0 Ø	01уууу Ø	01у10у Ø	ууу100 Ø	у10100 Ø
00011x	00у1у0 Ø	0у0уу1 Ø	0у0у10 Ø	0у0у11 Ø	0у01у0 Ø	0у01у1 Ø	0уууу1 Ø	0ууу10 Ø	0ууу11 Ø	0уу1у1 Ø	у0у1у0 Ø	уу01у0 Ø
00x111	0011уу Ø	0у0уу1 Ø	0у0у1у Ø	0у0у11 Ø	0у01уу Ø	0у01у1 Ø	0у1уу1 Ø	0у1у1у Ø	0у1у11 Ø	0у11у1 Ø	у011уу Ø	уу01уу Ø
000111	00у1уу Ø	0у0уу1 Ø	0у0у1у Ø	0у0у11 Ø	0у01уу Ø	0у01у1 Ø	0уууу1 Ø	0ууу1у Ø	0ууу11 Ø	0уу1у1 Ø	у0у1уу Ø	уу01уу Ø
001111	0011уу Ø	0уууу1 Ø	0ууу1у Ø	0ууу11 Ø	0уу1уу Ø	0уу1у1 Ø	0у1уу1 Ø	0у1у1у Ø	0у1у11 Ø	0у11у1 Ø	у011уу Ø	ууу1уу Ø
100000	у0уу00 Ø	уу000у Ø	уу00у0 Ø	уу00уу Ø	уу0у00 Ø	уу0у0у Ø	ууу00у Ø	ууу0у0 Ø	ууу0уу Ø	уууу0у Ø	10уу00 Ø	1у0у00 Ø
111000	уу1у00 Ø	у1у00у Ø	у1у0у0 Ø	у1у0уу Ø	у1уу00 Ø	у1уу0у Ø	у1100у Ø	у110у0 Ø	у110уу Ø	у11у0у Ø	1у1у00 Ø	11уу00 Ø
011001	0у1у0у Ø	01у001 Ø	01у0уу Ø	01у0у1 Ø	01уу0у Ø	01уу01 Ø	011001	0110уу Ø	0110у1 Ø	011у01 Ø	уу1у0у Ø	у1уу0у Ø
011x01	0у110у Ø	01у001 Ø	01у0уу Ø	01у0у1 Ø	01у10у Ø	01у101 Ø	011001	0110уу Ø	0110у1 Ø	011101	уу110у Ø	у1у10у Ø
011101	0у110у Ø	01уу01 Ø	01уууу Ø	01ууу1 Ø	01у10у Ø	01у101 Ø	011у01 Ø	011ууу Ø	011уу1 Ø	011101	уу110у Ø	у1у10у Ø
01x001	0у1у0у Ø	010001	0100уу Ø	0100у1 Ø	010у0у Ø	010у01 Ø	011001	0110уу Ø	0110у1 Ø	011у01 Ø	уу1у0у Ø	у10у0у Ø
011010	0у1уу0 Ø	01у0уу Ø	01у010 Ø	01у01у Ø	01ууу0 Ø	01уууу Ø	0110уу Ø	011010	01101у Ø	011ууу Ø	уу1уу0 Ø	у1ууу0 Ø
01x010	0у1уу0 Ø	0100уу Ø	010010	01001у Ø	010уу0 Ø	010ууу Ø	0110уу Ø	011010	01101у Ø	011ууу Ø	уу1уу0 Ø	у10уу0 Ø
100101	у0у10у Ø	уу0у01 Ø	уу0ууу Ø	уу0уу1 Ø	уу010у Ø	уу0101 Ø	уууу01 Ø	уууууу Ø	ууууу1 Ø	ууу101 Ø	10у10у Ø	1у010у Ø

Таблица 3.1.9 – Поиск непокрытых исходных наборов

L#E	001100	010001	010010	010011	010100	010101
x011x0	∅	010001	010010	010011	010100	010101
x101xx	∅	010001	010010	010011	∅	∅
x1xxx1	∅	∅	010010	∅	∅	∅
x1xx1x	∅	∅	∅	∅	∅	∅
Остаток	∅	∅	∅	∅	∅	∅
L#E	011001	011010	011011	011101	101100	110100
x011x0	011001	011010	011011	011101	∅	110100
x101xx	011001	011010	011011	011101	∅	∅
x1xxx1	∅	011010	∅	∅	∅	∅
x1xx1x	∅	∅	∅	∅	∅	∅
Остаток	∅	∅	∅	∅	∅	∅

Из таблицы 3.1 .9 видно, что  $L$ -экстремалью покрываются все кубы.

Минимальное покрытие - множество  $L$ -экстремалей  $E = \{ x011x0; x101xx; x1xxx1; x1xx1x \}$

Следовательно:

$$Q_1 = x_1 y_2 + x_1 \bar{x}_2 y_1 + \bar{x}_1 x_2 y_1 \bar{H} + x_1 H$$

Запишем результат базисе И – Тожественная единица – Сумма по модулю 2:

$$Q_1 = (x_1 y_2 \oplus 1)((x_1(x_2 \oplus 1)y_1) \oplus 1)((x_1 \oplus 1)x_2 y_1(H \oplus 1)) \oplus 1)(x_1 H \oplus 1) \oplus 1$$

**Минимизация функции  $Q_1$ :**

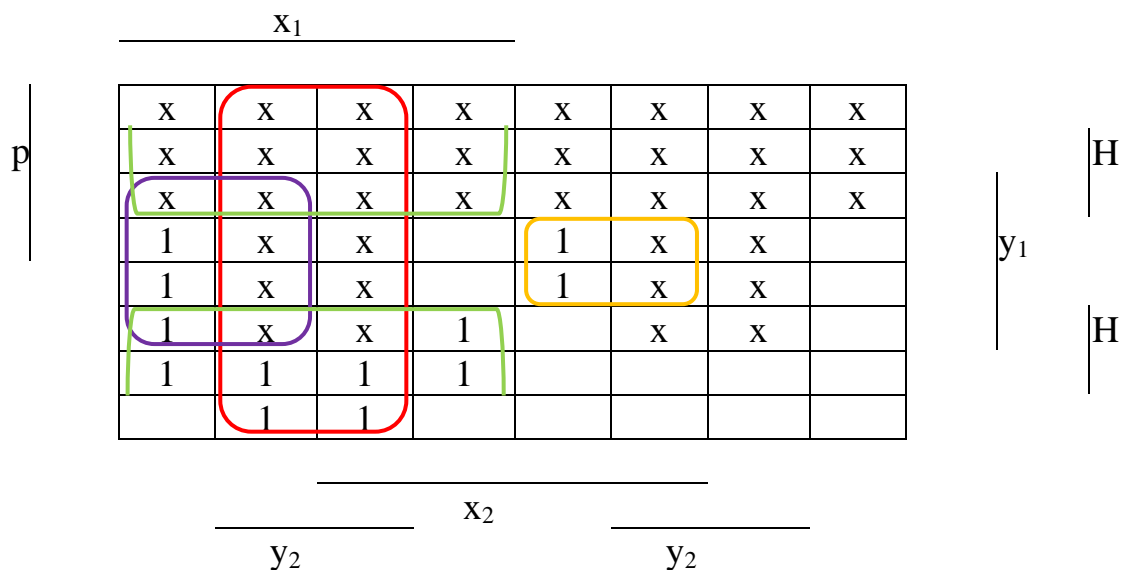


Рисунок 3.1.2 — Минимизация функции  $Q_1$  картой Вейча

Следовательно:

$$Q_1 = x_1 y_2 + x_1 \bar{x}_2 y_1 + \bar{x}_1 x_2 y_1 \bar{H} + x_1 H$$

Запишем результат базисе И – Тожественная единица – Сумма по модулю 2:

$$Q_1 = [ (x_1 y_2 \oplus 1)((x_1(x_2 \oplus 1)y_1) \oplus 1)((x_1 \oplus 1)x_2 y_1(H \oplus 1)) \oplus 1)(x_1 H \oplus 1) ] \oplus 1$$

Эффективность минимизации:

$$K = \frac{12 \cdot 6 + 12 + 6}{18} = 5$$

**Минимизация функции  $Q_2$ :**

		$x_1$									
$p$		x	x	x	x	x	x	x	x	$y_1$	$H$
		x	x	x	x	x	x	x	x		
		x	x	x	x	x	x	x	x		
			x	x	1		x	x	1		
		1	x	x		1	x	x			$H$
			x	x	1	1	x	x			
				1	1	1	1				
				1			1				
		$x_2$									
		$y_2$				$y_2$					

Рисунок 3.1.3 — Минимизация функции  $Q_2$  картой Вейча

Следовательно:

$$Q_2 = \bar{p} x_1 \bar{x}_2 y_1 \bar{H} + p x_1 x_2 + p \bar{x}_1 \bar{x}_2 + x_2 H + x_2 y_2 + \bar{p} \bar{x}_1 x_2 y_1$$

Запишем результат базисе И – Тожественная единица – Сумма по модулю 2:

$$Q_2 = [ (((p \oplus 1)x_1(x_2 \oplus 1)y_1(H \oplus 1)) \oplus 1)(p x_1 x_2 \oplus 1)(p(\bar{x}_1 \oplus 1)(x_2 \oplus 1)) \oplus 1)(x_2 H \oplus 1)(x_2 y_2 \oplus 1)((p \oplus 1)(x_1 \oplus 1)x_2 y_1) \oplus 1 ] \oplus 1$$

Эффективность минимизации:

$$K = \frac{12 \cdot 6 + 12 + 6}{29} = 3,1$$

Функциональная схема ОЧУС приведена в приложении Б.

### 3.2 Логический синтез одноразрядного четверичного сумматора

Одноразрядный четверичный сумматор – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда одного слагаемого, 2 разряда второго слагаемого и вход переноса) и 3 двоичных выхода.

Принцип работы ОЧС представлен с помощью таблицы истинности (таблица 3.2.1)

Кодировка слагаемых обоих разрядов: 0 – 00, 1 – 01, 2 – 11, 3 – 10;

Таблица 3.2.1 — Таблица истинности ОЧС

$a_1$	$a_2$	$b_1$	$b_2$	$p$	$\Pi$	$S_1$	$S_2$	<i>Пример операции в четверичной с/с</i>
1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	$0 + 0 + 0 = 00$
0	0	0	0	1	0	0	1	$0 + 0 + 1 = 01$
0	0	0	1	0	0	0	1	$0 + 1 + 0 = 01$
0	0	0	1	1	0	1	1	$0 + 1 + 1 = 02$
0	0	1	0	0	0	1	0	$0 + 3 + 0 = 03$
0	0	1	0	1	1	0	0	$0 + 3 + 1 = 10$
0	0	1	1	0	0	1	1	$0 + 2 + 0 = 02$
0	0	1	1	1	0	1	0	$0 + 2 + 1 = 03$
0	1	0	0	0	0	0	1	$1 + 0 + 0 = 01$
0	1	0	0	1	0	1	1	$1 + 0 + 1 = 02$
0	1	0	1	0	0	1	1	$1 + 1 + 0 = 02$
0	1	0	1	1	0	1	0	$1 + 1 + 1 = 03$
0	1	1	0	0	1	0	0	$1 + 3 + 0 = 10$
0	1	1	0	1	1	0	1	$1 + 3 + 1 = 11$
0	1	1	1	0	0	1	0	$1 + 2 + 0 = 03$
0	1	1	1	1	1	0	0	$1 + 2 + 1 = 10$
1	0	0	0	0	0	1	0	$3 + 0 + 0 = 03$
1	0	0	0	1	1	0	0	$3 + 0 + 1 = 10$
1	0	0	1	0	1	0	0	$3 + 1 + 0 = 10$
1	0	0	1	1	1	0	1	$3 + 1 + 1 = 11$
1	0	1	0	0	1	1	1	$3 + 3 + 0 = 12$
1	0	1	0	1	1	1	0	$3 + 3 + 1 = 13$
1	0	1	1	0	1	0	1	$3 + 2 + 0 = 11$
1	0	1	1	1	1	1	1	$3 + 2 + 1 = 12$
1	1	0	0	0	0	1	1	$2 + 0 + 0 = 02$
1	1	0	0	1	0	1	0	$2 + 0 + 1 = 03$
1	1	0	1	0	0	1	0	$2 + 1 + 0 = 03$
1	1	0	1	1	1	0	0	$2 + 1 + 1 = 10$
1	1	1	0	0	1	0	1	$2 + 3 + 0 = 11$

Продолжение таблицы 3.2.1

1	1	1	0	1	1	1	1	$2 + 3 + 1 = 12$
1	1	1	1	0	1	0	0	$2 + 2 + 0 = 10$
1	1	1	1	1	1	0	1	$2 + 2 + 1 = 11$

**Минимизация функции П:**

$\frac{b_1 b_2 p}{a_1 a_2}$	000	001	011	010	110	111	101	100
00							1	
01						1	1	1
11			1		1	1	1	1
10		1	1	1	1	1	1	1

Рисунок 3.2.1 — Минимизация функции П картой Карно

Следовательно:

$$П = a_1 \bar{a}_2 p + a_1 \bar{a}_2 b_2 + a_1 b_2 p + a_2 b_1 \bar{b}_2 + a_1 b_1 + a_2 b_1 p + b_1 \bar{b}_2 p$$

Запишем результат в базисе ИЛИ - Тожественная единица - Сумма по модулю 2:

$$П = \{((a_1 \oplus 1) + a_2 + (p \oplus 1)) \oplus 1\} + \{((a_1 \oplus 1) + a_2 + (b_2 \oplus 1)) \oplus 1\} + \{((a_1 \oplus 1) + (b_2 \oplus 1) + (p \oplus 1)) \oplus 1\} + \{((a_2 \oplus 1) + (b_1 \oplus 1) + b_2) \oplus 1\} + \{((a_1 \oplus 1) + (b_1 \oplus 1)) \oplus 1\} + \{((a_2 \oplus 1) + (b_1 \oplus 1) + (p \oplus 1)) \oplus 1\} + \{((b_1 \oplus 1) + b_2 + (p \oplus 1)) \oplus 1\}$$

Эффективность минимизации:

$$K = \frac{16 \cdot 5 + 16 + 5}{29} = 3,5$$

**Минимизация функции S<sub>1</sub>**

$\frac{b_1 b_2 p}{a_1 a_2}$	000	001	011	010	110	111	101	100
00			1		1	1		1
01		1	1	1	1			
11	1	1		1			1	
10	1					1	1	1

Рисунок 3.2.2 — Минимизация функции S<sub>1</sub> картой Карно

Следовательно:

$$S_1 = a_1 \bar{b}_1 \bar{b}_2 \bar{p} + a_2 \bar{b}_1 \bar{b}_2 p + \bar{a}_1 \bar{b}_1 b_2 p + a_2 \bar{b}_1 b_2 \bar{p} + \bar{a}_1 b_1 b_2 \bar{p} + \bar{a}_1 b_1 \bar{b}_2 p + \bar{a}_2 b_1 b_2 p + \bar{a}_2 b_1 \bar{b}_2 \bar{p}$$

Запишем результат в базисе ИЛИ - Тожественная единица - Сумма по модулю 2:

$$S_1 = \{((a_1 \oplus 1) + b_1 + b_2 + p) \oplus 1\} + \{((a_2 \oplus 1) + b_1 + b_2 + (p \oplus 1)) \oplus 1\} + \{(a_1 + b_1 + (b_2 \oplus 1) + (p \oplus 1)) \oplus 1\} + \{((a_2 \oplus 1) + b_1 + (b_2 \oplus 1) + p) \oplus 1\} + \{(a_1 + (b_1 \oplus 1) + (b_2 \oplus 1) + p) \oplus 1\} + \{(a_1 + (b_1 \oplus 1) + b_2 + (p \oplus 1)) \oplus 1\} + \{(a_2 + (b_1 \oplus 1) + (b_2 \oplus 1) + (p \oplus 1)) \oplus 1\} + \{(a_2 + (b_1 \oplus 1) + b_2 + p) \oplus 1\}$$

Эффективность минимизации:

$$K = \frac{16 \cdot 5 + 16 + 5}{45} = 2,24$$

### Минимизация функции $S_2$

$b_1 b_2 p$ $a_1 a_2$	000	001	011	010	110	111	101	100
00		1	1	1	1			
01	1	1		1			1	
11	1					1	1	1
10			1		1	1		1

Рисунок 3.2.3 — Минимизация функции  $S_2$  картой Карно

Следовательно:

$$S_2 = a_2 \bar{b}_1 \bar{b}_2 \bar{p} + \bar{a}_1 \bar{b}_1 \bar{b}_2 p + \bar{a}_2 \bar{b}_1 b_2 p + \bar{a}_1 \bar{b}_1 b_2 \bar{p} + \bar{a}_2 b_1 b_2 \bar{p} + a_2 b_1 \bar{b}_2 p + a_1 b_1 b_2 p + a_1 b_1 \bar{b}_2 \bar{p}$$

Запишем результат в базисе ИЛИ - Тожественная единица - Сумма по модулю 2:

$$S_2 = \{((a_2 \oplus 1) + b_1 + b_2 + p) \oplus 1\} + \{(a_1 + b_1 + b_2 + (p \oplus 1)) \oplus 1\} + \{(a_2 + b_1 + (b_2 \oplus 1) + (p \oplus 1)) \oplus 1\} + \{(a_1 + b_1 + (b_2 \oplus 1) + p) \oplus 1\} + \{(a_2 + (b_1 \oplus 1) + (b_2 \oplus 1) + p) \oplus 1\} + \{((a_2 \oplus 1) + (b_1 \oplus 1) + b_2 + (p \oplus 1)) \oplus 1\} + \{((a_1 \oplus 1) + (b_1 \oplus 1) + (b_2 \oplus 1) + (p \oplus 1)) \oplus 1\} + \{(a_1 + (b_1 \oplus 1) + b_2 + p) \oplus 1\}$$

Эффективность минимизации:

$$K = \frac{16 \cdot 5 + 16 + 5}{45} = 2,24$$

Функциональная схема ОЧС приведена в приложении В.

### 3.3. Логический синтез преобразователя множителя

Преобразователь множителя (ПМ) – это устройство, которое преобразовывает диады множителя в соответствии с методом умножения.

При умножении в дополнительных кодах ПМ заменяет диады 11 ( $3_4$ ) и 10 ( $2_4$ ) на триады  $1\bar{0}\bar{1}$  и  $1\bar{1}\bar{0}$  соответственно.

Принцип работы ПМ представлен с помощью таблицы истинности (таблица 3.3.1).

Таблица 3.3.1 – Таблица истинности ПМ

Входная диада		Младший разряд	Знак	Выходная диада	
$a_1$	$a_2$	$p$	$Q$	$S_1$	$S_2$
1	2	3	4	5	6
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	0	0	0

Минимизацию переключательных функций произведём с помощью карт Карно, карт Вейча и реализуем их в базисе И, ИЛИ, НЕ.

#### Функция Q

Для функции Q заполненная карта приведена на рисунке 3.3.1.

		$p$	
	$a_1$		
$a_2$		1	
	1	1	

Рисунок 3.3.1 - Минимизация функции Q картой Вейча

Следовательно:

$$Q = a_1\bar{p} + a_1\bar{a}_2 = a_1(\bar{p} + \bar{a}_2)$$

#### Функция S<sub>1</sub>

Для функции S<sub>1</sub> заполненная карта приведена на рисунке 3.3.2.



		$a_1a_2$			
$p$		00	01	11	10
	0				1
	1		1		

Рисунок 3.3.2 - Минимизация функции  $S_1$  картой Карно

Следовательно:

$$S_1 = a_1 \bar{a}_2 \bar{p} + \bar{a}_1 a_2 p$$

### Функция $S_2$

Для функции  $S_2$  заполненная карта приведена на рисунке 3.3.3.

		$a_1a_2$			
$p$		00	01	11	10
	0		1	1	
	1	1			1

Рисунок 3.3.3 - Минимизация функции  $S_2$  картой Карно

Следовательно:

$$S_2 = a_2 \bar{p} + \bar{a}_2 p$$

Функциональная схема ПМ приведена в приложении Д.

#### 4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ

*Мультиплексор* – это логическая схема, которая имеет  $n$  информационных входов,  $m$  управляющих входов и один выход. При этом должно выполняться условие  $m = 2^n$ .

На выход мультиплексора может быть пропущен без изменений один любой логический сигнал, поступающий на один из информационных входов. Порядковый номер информационного входа, значение которого в данный момент должно быть передано на выход, определяется двоичным кодом, подаваемым на управляющие входы.

Переключательные функции (ПФ) от пяти переменных (как, например, ОЧС) можно реализовать на мультиплексоре «один из восьми». Управляющее поле такого мультиплексора будет определяться тремя переменными, следовательно, число групп с одинаковыми значениями этих переменных будет равно восьми. Также, реализация нескольких ПФ требует для каждой ПФ отдельного мультиплексора.

Для определения управляющего поля мультиплексора возьмём переменные  $a_1$ ,  $a_2$  и  $b_1$ .

Таблица истинности для синтеза ПФ ОЧС приведена в таблице 4.1.

Таблица 4.1 – Таблица истинности для синтеза ПФ ОЧС

$a_1$	$a_2$	$b_1$	$b_2$	$p$	$\Pi$	Функция	$S_1$	Функция	$S_2$	Функция
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	$b_2 \cdot p$	0	$b_2 + p$
0	0	0	0	1	0		0		1	
0	0	0	1	0	0		0		1	
0	0	0	1	1	0		1		1	
0	0	1	0	0	0	$\bar{b}_2 \cdot p$	1	$b_2 + \bar{p}$	0	$b_2 \cdot \bar{p}$
0	0	1	0	1	1		0		0	
0	0	1	1	0	0		1		1	
0	0	1	1	1	0		1		0	
0	1	0	0	0	0	0	0	$b_2 + p$	1	$\bar{b}_2 + \bar{p}$
0	1	0	0	1	0		1		1	
0	1	0	1	0	0		1		1	
0	1	0	1	1	0		1		0	
0	1	1	0	0	1	$\bar{b}_2 + p$	0	$b_2 \cdot \bar{p}$	0	$\bar{b}_2 \cdot p$
0	1	1	0	1	1		0		1	
0	1	1	1	0	0		1		0	
0	1	1	1	1	1		0		0	
1	0	0	0	0	0	$b_2 + p$	1	$\bar{b}_2 \cdot \bar{p}$	0	$b_2 \cdot p$
1	0	0	0	1	1		0		0	

Продолжение таблицы 4.1

1	0	0	1	0	1		0		0	
1	0	0	1	1	1		0		1	
1	0	1	0	0	1	1	1	$\bar{b}_2 + p$	1	$b_2 + \bar{p}$
1	0	1	0	1	1		1		0	
1	0	1	1	0	1		0		1	
1	0	1	1	1	1		1		1	
1	1	0	0	0	0	$b_2 \cdot p$	1	$\bar{b}_2 + \bar{p}$	1	$\bar{b}_2 \cdot \bar{p}$
1	1	0	0	1	0		1		0	
1	1	0	1	0	0		1		0	
1	1	0	1	1	1		0		0	
1	1	1	0	0	1	1	0	$\bar{b}_2 \cdot p$	1	$\bar{b}_2 + p$
1	1	1	0	1	1		1		1	
1	1	1	1	0	1		0		0	
1	1	1	1	1	1		0		1	

При синтезе ОЧС на основе мультиплексоров получается более эффективная схема (если считать, что количество входов мультиплексора =  $n + 2^n$ , не включая входы элементов, из которых он состоит).

Функциональная схема ОЧС на основе мультиплексоров представлена в приложении Г.

## 5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ

Формула расчёта временных затрат на умножение:

$$T = n * (T_{\text{ПМ}} + T_{\text{ФДК}} + m * T_{\text{ОЧУС}} + (n + 2) * T_{\text{ОЧС}} + T_{\text{сдвига}}), \text{ где}$$

$T_{\text{ПМ}}$  – время преобразования множителя;

$T_{\text{ФДК}}$  – время формирования дополнительного кода множимого;

$T_{\text{ОЧУС}}$  – время умножения на ОЧУС;

$T_{\text{ОЧС}}$  – время формирования единицы переноса в ОЧС;

$T_{\text{сдвига}}$  – время сдвига в регистрах;

$n$  – количество разрядов множителя;

$m$  – количество разрядов множимого.

Минимизация функций позволила в несколько раз удешевить схему сумматора-умножителя и уменьшить затраты времени на выполнение за счет уменьшения количества элементов.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы была разработана структурная схема сумматора-умножителя второго типа, а также функциональные схемы основных узлов данного устройства. Для уменьшения стоимости логических схем были выполнены минимизации переключательных функций различными способами. Такой подход позволил выявить достоинства и недостатки этих алгоритмов.

В качестве главного достоинства минимизации картами Карно-Вейча можно выделить простоту и минимальные затраты времени. Однако применение данного способа для функций многих переменных будет затруднительно. Для минимизации функций многих переменных удобно использовать алгоритм Рота, который полностью формализует алгоритмы минимизации и делает минимизацию доступной для выполнения компьютерной программой.

Функциональные схемы были построены в различных логических базисах. Это позволило закрепить теоретические знания основных законов булевой алгебры, например, правило де Моргана.

Реализация переключательных функций на основе мультиплексоров позволила облегчить процесс минимизации этих функций и упростить функциональную схему одноразрядного четверичного сумматора.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Луцик, Ю. А. Учебное пособие по курсу «Арифметические и логические основы вычислительной техники» / Ю. А. Луцик, И. В. Лукьянова. - Минск : БГУиР, 2014. – 76с.

Искра, Н. А. Арифметические и логические основы вычислительной техники : пособие / Н. А. Искра, И. В. Лукьянова, Ю. А. Луцик. – Минск : БГУИР, 2016. – 75 с.

Единая система конструкторской документации (ЕСКД) : справ. пособие / С. С. Борушек [и др.]. – М. : Изд-во стандартов, 1989. – 352 с.

Лысиков, Б. Г. Цифровая вычислительная техника / Б. Г. Лысиков. – Минск : Выш. шк., 2003. – 242 с.

Савельев, А. Я. Прикладная теория цифровых автоматов / А. Я. Савельев. – М. : Высш. шк., 1987. – 272 с.

## **ПРИЛОЖЕНИЕ А**

(обязательное)

Сумматор-умножитель второго типа. Схема электрическая структурная

## **ПРИЛОЖЕНИЕ Б**

(обязательное)

Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная



## **ПРИЛОЖЕНИЕ В**

(обязательное)

Одноразрядный четверичный сумматор. Схема электрическая  
функциональная

## **ПРИЛОЖЕНИЕ Г**

(обязательное)

Однозарядный четверичный сумматор. Схема электрическая функциональная  
на основе мультиплексоров

**ПРИЛОЖЕНИЕ Д**  
**(обязательное)**

Преобразователь множителя. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Е**  
**(обязательное)**  
**Ведомость документов**