

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

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_ Е. И. Ковшер

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-  
УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 6-05-0611-05 616 ПЗ

Студент

Я. К. Морозов

Руководитель

Е. И. Ковшер

МИНСК 2024

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  
«\_\_» \_\_\_\_\_ 2024г.

ЗАДАНИЕ  
по курсовой работе студента  
Морозова Ярослава Константиновича

- 1 Тема работы: «Проектирование и логический синтез сумматора- умножителя двоично-четверичных чисел»
- 2 Срок сдачи студентом законченной работы: 10 мая 2024
- 3 Исходные данные к работе:
  - 3.1 Исходные сомножители:  $M_n = 52,34$ ;  $M_t = 38,63$ ;
  - 3.2 Алгоритм умножения: В;
  - 3.3 Метод умножения: умножение закодированного двоично- четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;
  - 3.4 Коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования:  $0_4 - 11$ ,  $1_4 - 10$ ,  $2_4 - 01$ ,  $3_4 - 00$ ;
  - 3.5 Тип синтезируемого умножителя – 2;
  - 3.6 Логический базис для реализации ОЧС: ИЛИ-НЕ; Метод минимизации – алгоритм Рота;
  - 3.7 Логический базис для реализации ОЧУС: И, сложение по модулю 2, «константа 1»; Метод минимизации – 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** Сумматор-умножитель второго типа. Схема электрическая структурная.
- 5.2** Одноразрядный четверичный сумматор. Схема электрическая функциональная.
- 5.3** Одноразрядный четверичный умножитель-сумматор. Схема электрическая функциональная.
- 5.4** Регистр-аккумулятор. Схема электрическая функциональная.
- 5.5** Одноразрядный четверичный сумматор. Реализация на мультиплексах. Схема электрическая функциональная.

**КАЛЕНДАРНЫЙ ПЛАН**

Наименование этапов курсовой работы	Объём этапа, %	Срок выполнения этапа	Примечания
Разработка алгоритма умножения	10	10.02-29.02	
Разработка структурной схемы сумматора-умножителя	10	29.02-29.03	С выполнением чертежа
Разработка функциональных схем основных узлов сумматора-умножителя	50	29.03-27.04	С выполнением чертежей
Схема комбинационных схем устройств на основе мультиплексов	10		С выполнением чертежа
Завершение оформления пояснительной записки	20		

Дата выдачи задания: 9 февраля 2024

Руководитель

Ковшер Е. И.

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ

\_\_\_\_\_ Морозов Я. К.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ.....	6
2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА – УМНОЖИТЕЛЯ .....	9
3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ.....	10
4.СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКОРОВ.....	26

## ВВЕДЕНИЕ

Современный мир невозможно представить без постоянных вычислений. Каждый день огромное количество суперкомпьютеров по всему миру выполняют огромное количество вычислительных операций. Именно поэтому изучение "Арифметических и логических основ цифровых устройств" становится неотъемлемой частью повседневной жизни. Новые вычислительные системы постоянно создаются, и для квалифицированных специалистов необходимо полное понимание принципов их работы.

Выбор темы проектирования не случаен. Основные арифметические операции - сложение и умножение - являются базовыми элементами вычислительных систем. Эта тема не только доступна для понимания студенту, но и имеет важное значение для его будущей карьеры.

Цель данной курсовой работы заключается не только в проектировании, но и в решении сложных задач с использованием методов, изученных в рамках курса. Эти задачи помогут студенту продемонстрировать свои знания.

В заключение, курсовая работа представляет собой важное звено между теорией и практикой. Студент получит возможность выявить и исправить неточности в своих знаниях, что будет полезно для его будущей профессиональной деятельности.

# 1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

Исходные данные:

- исходные сомножители:  $M_n = 52,34$ ;  $M_t = 38,63$ ;
- алгоритм умножения: В;
- метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;
- кодирование четверичных цифр множимого для перехода к двоично-четверичной системе кодирования:  $0_4 - 11$ ,  $1_4 - 10$ ,  $2_4 - 01$ ,  $3_4 - 00$ ;
- тип синтезируемого умножителя: 2-й;

1. Перевод сомножителей из десятичной системы счисления в четверичную

$$\begin{array}{r|l} 52 & 4 \\ \hline 52 & 13 & 4 \\ \hline 0 & 12 & 3 \\ & \hline & 1 \end{array}$$

$$\begin{array}{r} 0,34 \\ * \quad 4 \\ \hline 1,36 \\ * \quad 4 \\ \hline 1,44 \\ * \quad 4 \\ \hline 1,76 \end{array}$$

$$M_{n4} = 310,111.$$

В соответствии с заданной кодировкой множимого:

$$M_{n2/4} = 001011,101010$$

$$\begin{array}{r|l} 38 & 4 \\ \hline 36 & 9 & 4 \\ \hline 2 & 8 & 2 \\ & \hline & 1 \end{array}$$

$$\begin{array}{r} 0,63 \\ * \quad 4 \\ \hline 2,52 \\ * \quad 4 \\ \hline 2,08 \\ * \quad 4 \\ \hline 0,32 \end{array}$$

$$M_{t4} = 212,220.$$

В соответствие с обычной весомозначной кодировкой  $M_t$ :

$$M_{t2/4} = 110111,111100$$

1. Запишем сомножители в форме с плавающей запятой в прямом коде:

$$\begin{aligned} M_H &= 0,001011101010 & P_{M_H} &= 0.1100 + 03_{10} \\ M_T &= 0,110111111100 & P_{M_T} &= 0.0011 + 03_{10} \end{aligned}$$

Порядок произведения будет такой:

$$\begin{aligned} P_{M_H} &= 0.1100 (+03_4) \\ P_{M_T} &= 0.0011 (+03_4) \\ P_{M_H \cdot M_T} &= 0.1001 (+12_4) \end{aligned}$$

Результат соответствует кодировке множимого по заданию. Знак определяется суммой по модулю 2 знаков сомножителей:

$$znM_H \oplus znM_T = 0 \oplus 0 = 0.$$

При умножении в прямом коде диада  $11(3_4)$  заменяется  $1\overline{0}1$ , а  $10(2_4)$  заменяется в  $10\overline{2}$ . После преобразования для умножения в прямом коде  $M_T^H = 1\overline{2}2\overline{1}1\overline{2}0$ , а  $M_T^H = 1001011000$ . Умножение по алгоритму «В» представлено в таблице 1.1. Так как  $2M_H$  выходит за знаковый разряд нужно увеличить  $P_{M_H \cdot M_T}$  увеличить на один. И будет  $P_{M_H \cdot M_T} = 0.1000(+13_4)$ .

Таблица 1.1 – Умножение мантисс

Четверичная с/с			Двоично-четверичная с/с			Комментарии
0.	0000000	000000	11.	111111111111	1111111111	$\Sigma_0^q = 0$
0.	0000000	310111	11.	111111111111	001011101010	$\Pi_1^q = [M_H]_д$
0.	0000000	310111	11.	111111111111	001011101010	$\Sigma_1^q$
0.	0000003	101110	11.	111111111100	101110101011	$\Sigma_1^q \cdot 4$
3.	3333332	113112	00.	00000000000001	101000101001	$\Pi_2^q = [M_H \cdot -2]_д$
0.	0000001	220222	11.	1111111111110	010111010101	$\Sigma_2^q$
0.	0000012	202220	11.	1111111111001	011101010111	$\Sigma_2^q \cdot 4$
0.	0000001	220222	11.	1111111111110	010111010101	$\Pi_3^q = [M_H \cdot 2]_д$
0.	0000020	023102	11.	1111111110111	110100101101	$\Sigma_3^q$
0.	0000200	231020	11.	1111111011111	010010110111	$\Sigma_3^q \cdot 4$
3.	3333333	023223	00.	00000000000000	110100010100	$\Pi_4^q = [M_H \cdot -1]_д$
0.	0000133	320303	11.	11111111100000	000111001100	$\Sigma_4^q$
0.	0001333	203030	11.	11111110000000	011100110011	$\Sigma_4^q \cdot 4$
3.	3333333	023223	00.	00000000000000	110100010100	$\Pi_5^q = [M_H \cdot -1]_д$
0.	0001332	232313	11.	11111110000001	010001001000	$\Sigma_5^q$
0.	0013322	323130	11.	11111000000101	000100100011	$\Sigma_5^q \cdot 4$
3.	3333332	113112	00.	00000000000001	101000101001	$\Pi_6^q = [M_H \cdot -2]$
0.	0013321	102302	11.	11111000000110	101101001101	$\Sigma_6^q$
0.	0133211	023020	11.	11100000011010	110100110111	$\Sigma_6^q \cdot 4(\Sigma_7^q)$

После умножения результат оценивается на погрешность. Сначала надо привести к нулевому порядку и перевести в десятичную с/с.

$$M_{H4} \cdot M_{T4} = 133211,023020$$

$$(M_H \cdot M_T)_{10} = 2021,1738$$

Результат умножения в десятичной с/с без перехода в другую с/с равен

$$M_{H10} \cdot M_{T10} = 52,34 \cdot 38,63 = 2\,021,8942$$

Абсолютная погрешность при этом:

$$\Delta = 2\,021,8942 - 2021,1738 = 0,7204$$

Относительная погрешность:

$$\delta = \frac{\Delta}{M_H \cdot M_T} = \frac{0,7204}{2\,021,8942} = 0,00035629 (\delta = 0.035629\%)$$

Эта погрешность получается за счёт округления и приближенного перевода из одной с/с в другую.



## 2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА – УМНОЖИТЕЛЯ

Структурная схема сумматора умножителя представлена для заданного алгоритма умножения представлена в приложении А.

Структурная схема второго типа строится на базе узлов ОЧУС и ОЧС, формирования дополнительного кода и результата регистра. Режим работы определяется сигналом *Mul/Sum*, определяющую текущую операцию суммы или умножения.

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

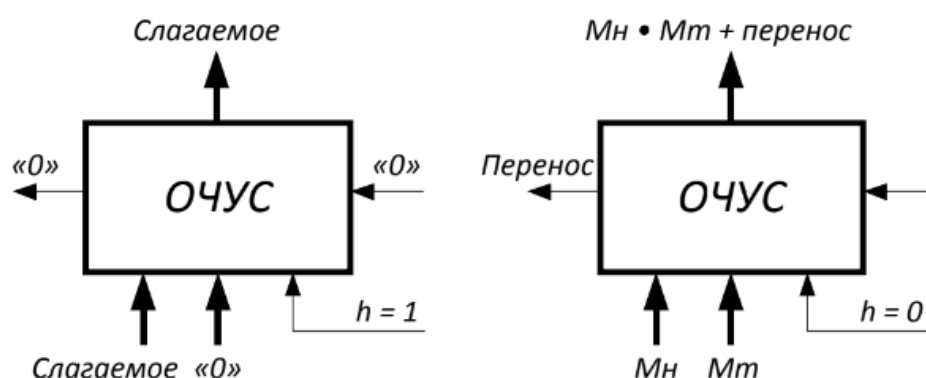


Рисунок 2.1 – Схема работы ОЧУС

Если же устройство работает как умножитель, то множимое и множитель поступает в соответствующие регистры. На управляющий вход ФДК поступает 0.

Таблица 2.2 - Режимы работы ФДК

Сигналы на входах ФДК		Результат на выходах ФДК
$F_1$	$F_2$	
0	0	Дополнительный код Мн
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

Принцип работы ОЧС представлен с помощью таблицы истинности (таблица 3.1.1).

Разряды обоих слагаемых закодированы: 0 – 11; 1 – 10; 2 – 01; 3 – 00.

Если ОЧС синтезируется для схемы второго типа, то безразличные наборы в таблице истинности отсутствуют.

Таблица 3.1 – Таблица истинности ОЧС

a <sub>1</sub>	a <sub>2</sub>	b <sub>1</sub>	b <sub>2</sub>	p	П	S <sub>1</sub>	S <sub>2</sub>	Пример операции в четверичной с/с
1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	0	1	3+3+0=12
0	0	0	0	1	1	0	0	3+3+1=13
0	0	0	1	0	1	1	0	3+2+0=11
0	0	0	1	1	1	0	1	3+2+1=12
0	0	1	0	0	1	1	1	3+1+0=10
0	0	1	0	1	1	1	0	3+1+1=11
0	0	1	1	0	0	0	0	3+0+0=03
0	0	1	1	1	1	1	1	3+0+1=10
0	1	0	0	0	1	1	0	2+3+0=11
0	1	0	0	1	1	0	1	2+3+1=12
0	1	0	1	0	1	1	1	2+2+0=10
0	1	0	1	1	1	1	0	2+2+1=11
0	1	1	0	0	0	0	0	2+1+0=03
0	1	1	0	1	1	1	1	2+1+1=10
0	1	1	1	0	0	0	1	2+0+0=02
0	1	1	1	1	0	0	0	2+0+1=03
1	0	0	0	0	1	1	1	1+3+0=10

Продолжение таблицы 3.1

1	2	3	4	5	6	7	8	9
1	0	0	0	1	1	1	0	$1+3+1=11$
1	0	0	1	0	0	0	0	$1+2+0=03$
1	0	0	1	1	1	1	1	$1+2+1=10$
1	0	1	0	0	0	0	1	$1+1+0=02$
1	0	1	0	1	0	0	0	$1+1+1=03$
1	0	1	1	0	0	1	0	$1+0+0=01$
1	0	1	1	1	0	0	1	$1+0+1=02$
1	1	0	0	0	0	0	0	$0+3+0=03$
1	1	0	0	1	1	1	1	$0+3+1=10$
1	1	0	1	0	0	0	1	$0+2+0=02$
1	1	0	1	1	0	0	0	$0+2+1=03$
1	1	1	0	0	0	1	0	$0+1+0=01$
1	1	1	0	1	0	0	1	$0+1+1=02$
1	1	1	1	0	0	1	1	$0+0+0=00$
1	1	1	1	1	0	1	0	$0+0+1=01$

### Минимизация функции П:

$\begin{matrix} & b_1b_2p \\ a_1a_2 \end{matrix}$		000	001	011	010	110	111	101	100
		00				0			
	01					0	0		0
	11	0		0	0	0	0	0	0
	10				0	0	0	0	0

Рисунок 3.1.1 - Минимизация функции при помощи карты Карно

$$\Pi = (\bar{b}_1 + \bar{b}_2 + p)(\bar{a}_2 + \bar{b}_1 + p)(\bar{a}_2 + \bar{b}_1 + \bar{b}_2)(\bar{a}_1 + \bar{b}_2 + p)(\bar{a}_1 + \bar{b}_1)(\bar{a}_1 + \bar{a}_2 + p)(\bar{a}_1 + \bar{a}_2 + \bar{b}_2)$$
$$\Pi = \frac{\overline{(\bar{b}_1 + \bar{b}_2 + p)} \overline{(\bar{a}_2 + \bar{b}_1 + p)} \overline{(\bar{a}_2 + \bar{b}_1 + \bar{b}_2)} \overline{(\bar{a}_1 + \bar{b}_2 + p)} \overline{(\bar{a}_1 + \bar{b}_2 + \bar{b}_1)}}{\overline{(\bar{a}_1 + \bar{b}_1)} \overline{(\bar{a}_1 + \bar{a}_2 + p)} \overline{(\bar{a}_1 + \bar{a}_2 + \bar{b}_2)}}$$
$$K = \frac{16 \cdot 5 + 16 + 5}{31} = 3,26$$

Diagram illustrating a 2D lattice structure with axes  $a_1$  (vertical) and  $b_2$  (horizontal). The lattice is divided into four quadrants by a vertical line at  $b_2 = 4$  and a horizontal line at  $a_1 = 2$ . The quadrants are labeled  $p$  (bottom-left),  $b_1$  (bottom-right), and  $b_2$  (top-left). The top-right quadrant is unlabeled. The lattice contains 0s in the following cells: (1,1), (2,1), (3,1), (5,1), (2,2), (5,2), (8,2), (1,3), (4,3), (4,4), (1,5), (6,5), (7,5), (8,5). The cell (2,1) is highlighted with a thick border.

$$S_1 = (a_1+a_2+b_1+b_2)(a_1+a_2+b_1+\bar{p})(a_1+\bar{b}_1+\bar{b}_2+p) \\ (a_1+b_1+b_2+\bar{p})(a_1+\bar{a}_2+\bar{b}_1+p)(a_1+\bar{a}_2+\bar{b}_1+\bar{b}_2)(\bar{a}_1+b_1+\bar{b}_2+p)(\bar{a}_1+a_2+\bar{b}_1+b_2) \\ (\bar{a}_1+a_2+\bar{b}_1+\bar{p})(\bar{a}_1+\bar{a}_2+b_1+p)(\bar{a}_1+\bar{a}_2+b_1+\bar{b}_2)(\bar{a}_1+\bar{b}_1+b_2+\bar{p})$$
$$S_1 = \frac{\overline{(a1 + a2 + b1 + b2)} \vee \overline{(a1 + a2 + b1 + \bar{p})} \vee \overline{(a1 + \bar{b}1 + \bar{b}2 + p)}}{\overline{(a1 + b1 + b2 + \bar{p})} \vee \overline{(a1 + \bar{a}2 + \bar{b}1 + p)} \vee \overline{(a1 + \bar{a}2 + \bar{b}1 + \bar{b}2)}} \vee$$
$$\frac{\overline{(\bar{a}1 + b1 + \bar{b}2 + p)} \vee \overline{(\bar{a}1 + a2 + \bar{b}1 + b2)} \vee \overline{(\bar{a}1 + a2 + \bar{b}1 + \bar{p})}}{\quad}$$

$$\overline{(\bar{a}1 + \bar{a}2 + b1 + p)} \vee \overline{(\bar{a}1 + \bar{a}2 + b1 + \bar{b}2)} \vee \overline{(\bar{a}1 + \bar{b}1 + b2 + \bar{p})}$$

Эффективность минимизации:

$$K = \frac{16*5+16+5}{65} = 1,56$$

### Минимизация функции $S_2$ :

Минимизация функции  $S_2$  будет производиться с помощью алгоритма Рота:

Определим множество единичных кубов:

$$L = \left\{ \begin{array}{l} 00000, 00011, 00100, 00111, \\ 01001, 01010, 01101, 01110, \\ 10000, 10011, 10100, 10111, \\ 11001, 11010, 11101, 11110 \end{array} \right\}$$

Множества безразличных наборов в ОЧС нет, потому  $N \equiv \emptyset$ .

Множество  $C_0 = L$

$$C_0 = L = \left\{ \begin{array}{l} 00000, 00011, 00100, 00111, \\ 01001, 01010, 01101, 01110, \\ 10000, 10011, 10100, 10111, \\ 11001, 11010, 11101, 11110 \end{array} \right\}$$

Первым этапом алгоритма Рота является нахождение множества простых импликант.

Для нахождения будем выполнять операцию умножения над множествами  $C_0$ ,  $C_1$  и т.д. Эта операция поможет склеить кубы большей размерности из ныне полученных.

Первый шаг операции умножения приведён в таблице 3.2

В результате этой операции сформируется новое множество кубов:

$$C_1 = \left\{ \begin{array}{l} 00x00, x0000, 00x11, x0011, \\ x0100, x0111, 01x01, x1001, \\ 01x10, x1010, x1101, x1110, \\ 10x00, 10x11, 11x01, 11x10 \end{array} \right\}$$

Таблица 3.2 – Поиск простых импликант  $C_0 * C_0$

$C_0 * C_0$	00000	00011	00100	00111	01001	01010	01101	01110	10000	10011	10100	10111	11001	11010	11101	11110
00000	—															
00011		—														
00100	00y00		—													
00111		00y11		—												
01001					—											
01010						—										
01101					01y01		—									
01110						01y10		—								
10000	y0000								—							
10011		y0011								—						
10100			y0100						10y00		—					
10111				y0111						10y11		—				
11001					y1001								—			
11010						y1010								—		
11101							y1101						11y01		—	
11110								y1110						11y10		—
$A_1$	00x00 x0000	00x11 x0011	x0100	x0111	01x01 x1001	01x10 x1010	x1101	x1110	10x00	10x11	∅	∅	11x01	11x10	∅	∅

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

В таблице 3.3 приведён следующий шаг поиска простых импликант с помощью операции  $C_1 * C_1$ .

В результате образовалось множество  $C_2$  кубов второй размерности:

$$C_2 = \{x_0x_00, x_0x_11, x_1x_01, x_1x_10\}$$

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

В таблице 3.4 приведён следующий шаг поиска простых импликант – операция  $C_2 * C_2$ :

Таблица 3.3 – Поиск простых импликант  $C_1 * C_1$

$C_1 * C_1$	00x00	x0000	00x11	x0011	x0100	x0111	01x01	x1001	01x10	x1010	x1101	x1110	10x00	10x11	11x01	11x10
00x00	—															
x0000		—														
00x11			—													
x0011				—												
x0100		x0y00			—											
x0111				x0y11		—										
01x01							—									
x1001								—								
01x10									—							
x1010										—						
x1101								x1y01			—					
x1110										x1y10		—				
10x00	y0x00												—			
10x11														—		
11x01			y0x11												—	
11x10									y1x10							—
$A_2$	x0x00	x0x00	x0x11	x0x11	$\emptyset$	$\emptyset$	$\emptyset$	x1x01	x1x10	x1x10	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$



Таблица 3.4 - Поиск простых импликант  $C_1 * C_1$

$C_2 * C_2$	x0x00	x0x11	x1x01	x1x10
x0x00	—			
x0x11		—		
x1x01			—	
x1x10				—
$A_3$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Новых кубов не образовалось.

Получено множество  $Z_2 = \{x0x00, x0x11, x1x01, x1x10\}$

Поиск простых импликант закончен. Следующим этапом будет поиск L-экстремалей на множестве простых импликант. Для этого используется операция вычитания (#). Результат приведён в таблице 3.5.

Таблица 3.5 – Поиск L-экстремалей

$z\#(Z-z) \cap L$	x0x00	x0x11	x1x01	x1x10
00000	00000	000yy $\emptyset$	0y00y $\emptyset$	0y0y0 $\emptyset$
00011	000yy $\emptyset$	00011	0y0y1 $\emptyset$	0y01y $\emptyset$
00100	00100	001yy $\emptyset$	0y10y $\emptyset$	0y1y0 $\emptyset$
00111	001yy $\emptyset$	00111	0y1y1 $\emptyset$	0y11y $\emptyset$
01001	0y00y $\emptyset$	0y0y1 $\emptyset$	01001	010yy $\emptyset$
01010	0y0y0 $\emptyset$	0y01y $\emptyset$	010yy $\emptyset$	01010
01101	0y10y $\emptyset$	0y1y1 $\emptyset$	01101	011yy $\emptyset$
01110	0y1y0 $\emptyset$	0y11y $\emptyset$	011yy $\emptyset$	01110
10000	10000	100yy $\emptyset$	1y00y $\emptyset$	1y0y0 $\emptyset$
10011	100yy $\emptyset$	10011	1y0y1 $\emptyset$	1y01y $\emptyset$
10100	10100	101yy $\emptyset$	1y10y $\emptyset$	1y1y0 $\emptyset$
10111	101yy $\emptyset$	10111	1y1y1 $\emptyset$	1y11y $\emptyset$

Продолжение таблицы 3.5

11001	1y00y Ø	1y0y1 Ø	11001	110yy Ø
11010	1y0y0 Ø	1y01y Ø	110yy Ø	11010
11101	1y10y Ø	1y1y1 Ø	11101	111yy Ø
11110	1y1y0 Ø	1y11y Ø	111yy Ø	11110

В таблице из каждой простой импликанты поочерёдно вычитаются другие простые импликанты. L-экстремальными стали следующие импликанты:

$$E=\{x_0x_{00},x_0x_{11},x_1x_{01},x_1x_{10}\}.$$

Минимальное покрытие – множество L-экстремалей E.

Следовательно:

$$S_2=(a_2+b_2+\bar{p})(a_2+\bar{b}_2+p)(\bar{a}_2+b_2+p)(\bar{a}_2+\bar{b}_2+p)$$

Приведём результат к базису ИЛИ-НЕ:

$$S_2=\overline{\overline{(a_2 + b_2 + \bar{p})} \vee \overline{(a_2 + \bar{b}_2 + p)} \vee \overline{(\bar{a}_2 + b_2 + p)} \vee \overline{(\bar{a}_2 + \bar{b}_2 + p)}}$$

Эффективность минимизации:

$$K = \frac{16*5+16+5}{19} = 5,36$$

### 3.2 Логический синтез одноразрядного четверичного сумматора-умножителя

Принцип работы ОЧУС представлен с помощью таблицы истинности (таблица 3.1).

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

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

Таблица 3.1 – Таблица истинности ОЧУС

Пер.	Мн		Мт		Упр	Перенос	Результат		Результат операции в четверичной с/с
$P_1$	$x_1$	$x_2$	$y_1$	$y_2$	$h$	$P$	$Q_1$	$Q_2$	
1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	1	1	$3 \cdot 0 + 0 = 00$
0	0	0	0	0	1	0	0	0	Выход – код «03»
0	0	0	0	1	0	0	0	0	$3 \cdot 1 + 0 = 03$
0	0	0	0	1	1	0	0	0	Выход – код «03»
0	0	0	1	0	0	1	0	1	$3 \cdot 2 + 0 = 12$
0	0	0	1	0	1	0	0	0	Выход – код «03»
0	0	0	1	1	0	х	х	х	$3 \cdot 3 + 0 = 21$
0	0	0	1	1	1	х	х	х	Выход – код «03»
0	0	1	0	0	0	0	1	1	$2 \cdot 0 + 0 = 00$
0	0	1	0	0	1	0	0	1	Выход – код «02»
0	0	1	0	1	0	0	0	1	$2 \cdot 1 + 0 = 02$
0	0	1	0	1	1	0	0	1	Выход – код «02»
0	0	1	1	0	0	1	1	1	$2 \cdot 2 + 0 = 10$
0	0	1	1	0	1	0	0	1	Выход – код «02»
0	0	1	1	1	0	х	х	х	$2 \cdot 3 + 0 = 12$
0	0	1	1	1	1	х	х	х	Выход – код «02»
0	1	0	0	0	0	0	1	1	$1 \cdot 0 + 0 = 00$
0	1	0	0	0	1	0	1	0	Выход – код «01»
0	1	0	0	1	0	0	1	0	$1 \cdot 1 + 0 = 01$
0	1	0	0	1	1	0	1	0	Выход – код «01»
0	1	0	1	0	0	0	0	1	$1 \cdot 2 + 0 = 02$
0	1	0	1	0	1	0	1	0	Выход – код «01»
0	1	0	1	1	0	х	х	х	$1 \cdot 3 + 0 = 03$
0	1	0	1	1	1	х	х	х	Выход – код «01»
0	1	1	0	0	0	0	1	1	$0 \cdot 0 + 0 = 00$
0	1	1	0	0	1	0	1	1	Выход – код «00»
0	1	1	0	1	0	0	1	1	$0 \cdot 1 + 0 = 00$
0	1	1	0	1	1	0	1	1	Выход – код «00»

Продолжение таблицы 3.1

1	2	3	4	5	6	7	8	9	10
0	1	1	1	0	0	0	1	1	$0 \cdot 2 + 0 = 00$
0	1	1	1	0	1	0	1	1	Выход – код «00»
0	1	1	1	1	0	x	x	x	$0 \cdot 3 + 0 = 00$
0	1	1	1	1	1	x	x	x	Выход – код «00»
1	0	0	0	0	0	x	x	x	$3 \cdot 0 + 1 = 01$
1	0	0	0	0	1	x	x	x	Выход – код «03»
1	0	0	0	1	0	x	x	x	$3 \cdot 1 + 1 = 10$
1	0	0	0	1	1	x	x	x	Выход – код «03»
1	0	0	1	0	0	1	0	0	$3 \cdot 2 + 1 = 13$
1	0	0	1	0	1	x	x	x	Выход – код «03»
1	0	0	1	1	0	x	x	x	$3 \cdot 3 + 1 = 22$
1	0	0	1	1	1	x	x	x	Выход – код «03»
1	0	1	0	0	0	x	x	x	$2 \cdot 0 + 1 = 01$
1	0	1	0	0	1	x	x	x	Выход – код «02»
1	0	1	0	1	0	x	x	x	$2 \cdot 1 + 1 = 03$
1	0	1	0	1	1	x	x	x	Выход – код «02»
1	0	1	1	0	0	1	1	0	$2 \cdot 2 + 1 = 11$
1	0	1	1	0	1	x	x	x	Выход – код «02»
1	0	1	1	1	0	x	x	x	$2 \cdot 3 + 1 = 13$
1	0	1	1	1	1	x	x	x	Выход – код «02»
1	1	0	0	0	0	x	x	x	$1 \cdot 0 + 1 = 01$
1	1	0	0	0	1	x	x	x	Выход – код «01»
1	1	0	0	1	0	x	x	x	$1 \cdot 1 + 1 = 02$
1	1	0	0	1	1	x	x	x	Выход – код «01»
1	1	0	1	0	0	0	0	0	$1 \cdot 2 + 1 = 03$
1	1	0	1	0	1	x	x	x	Выход – код «01»
1	1	0	1	1	0	x	x	x	$1 \cdot 3 + 1 = 10$
1	1	0	1	1	1	x	x	x	Выход – код «01»
1	1	1	0	0	0	x	x	x	$0 \cdot 0 + 1 = 01$
1	1	1	0	0	1	x	x	x	Выход – код «00»
1	1	1	0	1	0	x	x	x	$0 \cdot 1 + 1 = 01$
1	1	1	0	1	1	x	x	x	Выход – код «00»
1	1	1	1	0	0	0	1	0	$0 \cdot 2 + 1 = 01$
1	1	1	1	0	1	x	x	x	Выход – код «00»
1	1	1	1	1	0	x	x	x	$0 \cdot 3 + 1 = 01$
1	1	1	1	1	1	x	x	x	Выход – код «00»

## Минимизация функции P:

								$y_2$	
$P_1$					<div>x</div>	x		<div>1</div>	$x_2$
					<div>x</div>	x		<div>1</div>	
					x	x			$x_1$
					x	x			
	x	x	x	x	x	x	x		$x_2$
	x	x	x	x	x	x	x		
	x	x	x	x	<div>x</div>	x	x	<div>1</div>	$x_1$
	x	x	x	x	<div>x</div>	x	x	<div>1</div>	
								$y_1$	
$h$				$h$					

Рисунок 3.2.1 - Минимизация функции при помощи карты Вейча

Следовательно:

$$P = \bar{x}_1 y_1 \bar{h}$$

Приведём функцию к базису И, сумма по модулю два, константная единица:

$$P = (x_1 \oplus 1)y(h \oplus 1)$$

Эффективность минимизации:

$$K = \frac{4*6+6+4}{8} = 4,25$$

## Минимизация функции $Q_1$ :

$y_1 y_2 h$ $P_1 x_1 x_2$	000	001	011	010	110	111	101	100
000		0	0	0	x	x	0	0
001		0	0	0	x	x	0	
011					x	x		
010					x	x		0
110	x	x	x	x	x	x	x	0
111	x	x	x	x	x	x	x	
101	x	x	x	x	x	x	x	
100	x	x	x	x	x	x	x	0

Рисунок 3.2.2 - Минимизация функции при помощи карты Карно

Следовательно:

$$Q_1 = (x_1 + \bar{h})(x_1 + \bar{y}_2)(x_2 + \bar{y}_1 + h)(x_1 + \bar{x}_2)$$

Приведём результат к базису И, сумма по модулю два, константная единица:

$$Q_1 = ((x_1 \oplus \bar{h})x_1\bar{h})((x_1 \oplus \bar{y}_2)x_1\bar{y}_2)((x_2 \oplus \bar{y}_1h)x_2\bar{y}_1h)((x_1 \oplus \bar{x}_2)x_1\bar{x}_2)$$

Эффективность минимизации:

$$K = \frac{12*6+6+12}{13} = 6,92$$

### Минимизация функции $Q_2$ :

								$y_2$	
$P_1$	1				x	x		1	$x_2$
	1	1	1	1	x	x	1	1	
	1	1	1	1	x	x	1	1	
	1				x	x		1	
	x	x	x	x	x	x	x		$x_1$
	x	x	x	x	x	x	x		
	x	x	x	x	x	x	x		
	x	x	x	x	x	x	x		
$h$				$y_1$				$h$	

Рисунок 3.2.3 - Минимизация функции при помощи карты Вейча

Следовательно:

$$Q_2 = \bar{P}_1 \bar{y}_2 \bar{h} + \bar{P}_1 x_2$$

Приведём к базису И, сумма по модулю два, константная единица:

$$Q_2 = ((p_1 \oplus 1)(y_2 \oplus 1)(h \oplus 1) \oplus (p_1 \oplus 1)x_2)((p_1 \oplus 1)x_2(y_2 \oplus 1)(h \oplus 1))$$

Эффективность минимизации:

$$K = \frac{16*6+16+6}{10} = 11,8$$

### 3.3. Логический синтез преобразователя множителя

Преобразователь множителя (ПМ) – это устройство, преобразующее диады множителя в зависимости от метода умножения.

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

Принцип работы показан в таблице истинности(таблица 3.1):

Таблица 3.1 – Таблица истинности ПМ

Входная диада		Младший разряд	Знак	Выходная диада	
$Q_1$	$Q_2$	$p$	$Z$	$S_1$	$S_2$
0	0	0	0	0	0
0	0	1	0	0	1

Продолжение таблицы 3.1

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

Минимизация представлена с помощью карт Карно-Вейча:

**Минимизация функции Z:**

		<b>Q<sub>2</sub>p</b>			
<b>Q<sub>1</sub></b>		<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
	<b>0</b>	1	*	1	1
	<b>1</b>				

Рисунок 3.3.1 - Минимизация функции Z картой Карно

Следовательно:

$$Z = Q_1$$

**Минимизация функции S<sub>1</sub>:**

		<b>p</b>			
		<b>Q<sub>2</sub></b>			
<b>Q<sub>1</sub></b>					1
		1			

Рисунок 3.3.2 - Минимизация функции S<sub>1</sub> с помощью карты Вейча

Следовательно:

$$S_1 = \overline{Q_1}Q_2p + Q_1\overline{Q_2}\overline{p}$$

**Минимизация функции S<sub>2</sub>:**

		<b>Q<sub>2</sub>p</b>			
<b>Q<sub>1</sub></b>		<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
	<b>0</b>		1		1
	<b>1</b>		1		1

Рисунок 3.3.3 - Минимизация функции S<sub>2</sub> с помощью карты Карно



Следовательно:

$$S_2 = \overline{Q_2}p + Q_2\overline{p} = Q_2 \oplus p$$

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

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 любой один 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, поданным на управляющие входы.

Переключательные функции (ПФ) от пяти переменных можно реализовать на мультиплексоре «один из восьми».

Управляющее поле мультиплексора определяется тремя переменными, значит, число групп с одинаковыми значениями этих переменных будет равно восьми.

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

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	1	1	0	$b_2 * \bar{p}$	1	$\overline{b_2 \oplus p}$
0	0	0	0	1	1		0		0	
0	0	0	1	0	1		1		0	
0	0	0	1	1	1		0		1	
0	0	1	0	0	1	$\bar{b}_2 + p$	1	$\bar{b}_2 + p$	1	$\overline{b_2 \oplus p}$
0	0	1	0	1	1		1		0	
0	0	1	1	0	0		0		0	
0	0	1	1	1	1		1		1	
0	1	0	0	0	1	1	1	$b_2 + \bar{p}$	0	$b_2 \oplus p$
0	1	0	0	1	1		0		1	
0	1	0	1	0	1		1		1	
0	1	0	1	1	1		1		0	
0	1	1	0	0	0	$\bar{b}_2 * p$	0	$\bar{b}_2 * p$	0	$b_2 \oplus p$
0	1	1	0	1	1		1		1	
0	1	1	1	0	0		0		1	
0	1	1	1	1	0		0		0	
1	0	0	0	0	1	$\bar{b}_2 + p$	1	$\bar{b}_2 + p$	1	$\overline{b_2 \oplus p}$
1	0	0	0	1	1		1		0	
1	0	0	1	0	0		0		0	
1	0	0	1	1	1		1		1	

Продолжение таблицы 4.1

1	0	1	0	0	0	0	0	$b_2 * \bar{p}$	1	$\overline{b_2 \oplus p}$
1	0	1	0	1	0		0		0	
1	0	1	1	0	0		1		0	
1	0	1	1	1	0		0		1	
1	1	0	0	0	0	$\bar{b}_2 * p$	0	$\bar{b}_2 * p$	0	$b_2 \oplus p$
1	1	0	0	1	1		1		1	
1	1	0	1	0	0		0		1	
1	1	0	1	1	0		0		0	
1	1	1	0	0	0	0	1	$b_2 + \bar{p}$	0	$b_2 \oplus p$
1	1	1	0	1	0		0		1	
1	1	1	1	0	0		1		1	
1	1	1	1	1	0		1		0	

При синтезе ОЧС на основе мультиплексоров схема получается более эффективной и менее затратной.