

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

\_\_\_\_\_И.В. Лукьянова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-  
УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 312ПЗ

Студент

А. М. Гнездилов

Руководитель

И. В. Лукьянова

МИНСК 2022

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

Гнездилова Александра

Максимовича

- 1 Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-десятичных чисел»
- 2 Срок сдачи студентом законченной работы: до 20 мая 2022 г.
- 3 Исходные данные к работе:
  - 3.1 исходные сомножители:  $M_n = 12,39$ ;  $M_t = 95,91$ ;
  - 3.2 алгоритм умножения: Б;
  - 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:  $0_4 - 00$ ,  $1_4 - 11$ ,  $2_4 - 01$ ,  $3_4 - 10$ ;
  - 3.5 тип синтезируемого умножителя: 1;
  - 3.6 логический базис для реализации ОЧС: ИЛИ-НЕ; метод минимизации – карты Карно – Вейча;
  - 3.7 логический базис для реализации ОЧУ: ИЛИ; исключающее ИЛИ; метод минимизации – алгоритм Рота.

**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 первого типа. Схема электрическая структурная.

**5.2** Одноразрядный четверичный сумматор. Схема электрическая функциональная.

**5.3** Одноразрядный четверичный умножитель. Схема электрическая функциональная.

**5.4** Регистр-аккумулятор. Схема электрическая функциональная.

**5.5** Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.

### КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов курсовой работы	Объём 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 2022 г.

Руководитель \_\_\_\_\_/И. В. Лукьянова /

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_/ /

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ.....	6
2. СИНТЕЗ СТРУКТУРЫ СУММАТОРА-УМНОЖИТЕЛЯ ПЕРВОГО ТИПА .....	9
3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ.....	12
3.1. Логический синтез ОЧС.....	12
3.2. Логический синтез ОЧУ.....	20
4. ЭФФЕКТИВНОСТЬ МИНИМИЗАЦИИ .....	24
5. СИНТЕЗ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРА.....	25
6. ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ .....	27
7. ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ .....	28
ЗАКЛЮЧЕНИЕ .....	29
ЛИТЕРАТУРА .....	30
ПРИЛОЖЕНИЕ А .....	31
ПРИЛОЖЕНИЕ Б.....	32
ПРИЛОЖЕНИЕ В .....	33
ПРИЛОЖЕНИЕ Г .....	34
ПРИЛОЖЕНИЕ Д .....	35
ПРИЛОЖЕНИЕ Е.....	36

## **ВВЕДЕНИЕ**

Целью данной работы является разработка сумматора-умножителя первого типа для алгоритма умножения «Б» в дополнительном коде на два разряда одновременно. Чтобы это осуществить нам нужно решить ряд задач:

1. Разработать алгоритм умножения и оценить погрешности вычислений.
2. Разработать структурную схему сумматора-умножителя первого типа.
3. Разработать функциональные схемы основных узлов сумматора-умножителя в заданных логических базисах.
4. Разработать комбинационную схему на основе мультиплексора.
5. Рассчитать временные затраты на умножение.

# 1. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ

Перевод данных сомножителей из десятичной системы счисления в четверичную систему счисления:

Множимое:

12   4	*	0,91
12 3   4		<u>4</u>
0 0 0		1,56
3	*	<u>4</u>
		2,24
	*	<u>4</u>
		0,96
	*	<u>4</u>
		3,84

$M_{H4}=30,1203$   
 $M_{H2/4}=1000,11010010$

Множитель:

95   4	*	0,91
92 23   4		<u>4</u>
3 20 5   4		3,64
3 4 1	*	<u>4</u>
1		2,56

$M_{T4}=1133,32$   
 $M_{T2/4}=01011111,1110$

Сомножители в форме с плавающей запятой в прямом коде:

$M_H = 0, 100011010010$        $P_{MH} = 0.0001$  (закодирован согласно заданию)

$M_T = 0, 010111111110$        $P_{MT} = 0.0100$  (закодирован традиционно)

Сложении порядков:

$P_{MH} = 0.0001$	+02
$P_{MT} = 0.0100$	+10
$P = 0.1101$	+12

Результат операции закодирован согласно задания на кодировку множимого.  
 Знак произведения определяем суммой по модулю двух знаков  
 сомножителей:

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

Для умножения мантисс предварительно преобразуем множитель, чтобы  
 исключить диаду 11 ( $3_4$ ), заменив ее на триаду 101.

Преобразованный множитель имеет вид:  $M_T^p = 011000000\overline{1}10$

$$M_H = 301203$$

$$M_{T_4} = 113332$$

$$M_T^p = 1200\overline{1}2$$

$$M_T^p = 011000000\overline{0}110$$

$$[-M_H]_d = 3.032131$$

$$[2M_H]_{\text{пр}} = 0,1203012$$

Умножение мантисс по алгоритму «Б» приведено в таблице 1.1.

Таблица 1.1 - Умножение мантисс

Четверичная с/с		Двоично-четверичная с/с		Комментарии
1		2		3
0.	000000000000	0.	000000000000000000000000	$\Sigma_0^q = 0$
<u>0.</u>	<u>000001203012</u>	<u>0.</u>	<u>000000000011010010001101</u>	$\Pi_1^q = M_H \cdot 2 \cdot 2^0$
0.	000001203012	0.	000000000011010010001101	$\Sigma_1^q$
<u>3.</u>	<u>333330321310</u>	<u>1.</u>	<u>101010101000100111101100</u>	$\Pi_2^q = M_H \cdot (-1) \cdot 2^1$
3.	333332130322	1.	101010101001111000100101	$\Sigma_2^q$
<u>0.</u>	<u>000000000000</u>	<u>0.</u>	<u>000000000000000000000000</u>	$\Pi_3^q = M_H \cdot 0 \cdot 2^2$
3.	333332130322	1.	101010101001111000100101	$\Sigma_3^q$
<u>0.</u>	<u>000000000000</u>	<u>0.</u>	<u>000000000000000000000000</u>	$\Pi_4^q = M_H \cdot 0 \cdot 2^3$
3.	333332130322	1.	101010101001111000100101	$\Sigma_4^q$
<u>0.</u>	<u>012030120000</u>	<u>0.</u>	<u>001101001000110100000000</u>	$\Pi_5^q = M_H \cdot 2 \cdot 2^4$
0.	012022310322	0.	001101000101101100100101	$\Sigma_5^q$
<u>0.</u>	<u>030120300000</u>	<u>0.</u>	<u>001000110100100000000000</u>	$\Pi_6^q = M_H \cdot 1 \cdot 2^5$
0.	102203210322	0.	110001010010011100100101	$\Sigma_6^q$

Окончив умножение, необходимо оценить погрешность вычислений. Для этого полученное произведение  $(M_H * M_T)_4 = 0,102203210322$  ( $P_{M_H} * P_{M_T} = 12_4$ ) приводим к нулевому порядку, а затем переводим в десятичную систему счисления:

$$(M_H * M_T)_4 = 102203,210322 \quad (P_{M_H} * P_{M_T} = 0)$$

$$(M_H * M_T)_{10} = 1187,5766$$

$$M_{H10} * M_{T10} = 1188,3249$$

Абсолютная погрешность:

$$\Delta = 1188,3249 - 1187,5766 = 0,7483$$

$$\delta = \frac{\Delta}{(M_H * M_T)_{10}} = \frac{0,7483}{1188,3249} = 0,0006297$$

$$\delta = 0,06297\%$$



## 2.РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ

Структурная схема сумматора-умножителя 1-го типа алгоритма умножения “Б” приведена в приложении А.

Если устройство работает как сумматор (на входе mul/sum – “1”), то оба слагаемых последовательно записываются в регистр множимого, а на управляющий вход формирователя дополнительного кода (ФДК) F2 поступает «1». Нужно помнить, что числа представлены в форме с плавающей запятой. Поэтому, перед тем как складывать мантиссы, необходимо выравнивать порядки. В блоке порядков нужно обеспечить сравнение порядков, используя сумматор порядков, и в зависимости от знака результата сдвигать первое или второе слагаемое. Реализация сдвига мантиссы числа с меньшим порядком будет зависеть от алгоритма умножения. Этим будет определяться порядок подачи слагаемых на операцию и то, где будет сдвигаться мантисса (в регистре множимого или в регистре результата). На выходах ФДК формируется дополнительный код одного слагаемого с учетом знака. Это слагаемое может быть записано в регистр множимого, при этом управляющие сигналы, поступающие на входы «h» всех ОЧУ, дают возможность переписать на выходы ОЧУ разряды слагаемого без изменений. Если есть необходимость выравнивания порядков, то в регистре-аккумуляторе может выполняться сдвиг мантиссы первого слагаемого. Если на вход «h» поступает «0», то ОЧУ перемножает разряды  $M_n$  и  $M_t$ .

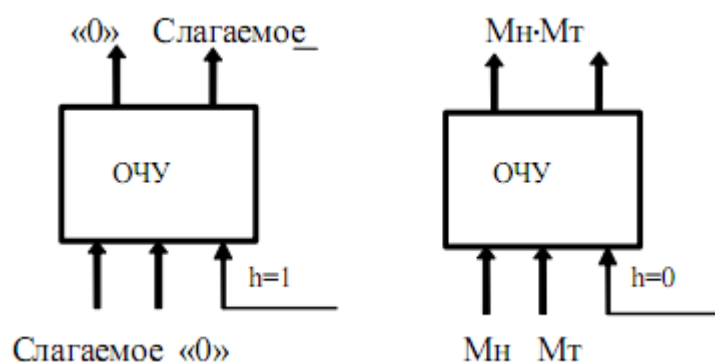


Рисунок 2.1 – Режимы работы ОЧУ

Одноразрядный четверичный сумматор предназначен для сложения двух двоично-четверичных цифр, подаваемых на его входы.

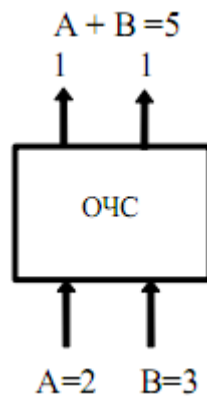


Рисунок 2.2 – Однозарядный четверичный сумматор

В ОЧС первое слагаемое складывается с нулем, так как на старших выходах ОЧУ будут формироваться только коды нуля. После первое слагаемое попадает в регистр-аккумулятор, который изначально обнулен.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУ и ОЧС попадает в аккумулятор, где складывается с первым слагаемым. Таким образом, аккумулятор (накапливающий сумматор) складывает операнды и хранит результат. Разрядность аккумулятора должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения при суммировании переноса.

Если устройство работает как умножитель (на входе mul/sum – “0”), то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК F2 поступает «0». Диада множителя поступает на входы преобразователя множителя (ПМ). Задачей ПМ является преобразование диады множителя в соответствии с алгоритмом преобразования. При этом в случае образования единицы переноса в старшую диаду множителя она должна быть учтена при преобразовании этой старшей диады (выход 1 ПМ). В регистре множителя в конце каждого такта умножения содержимое сдвигается на 2 двоичных разряда, и в последнем такте умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения при умножении по алгоритму “Б” (регистр множителя служит как бы “продолжением” регистра результата). Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание (01). В этом случае инициализируется управляющий вход F1 формирователя дополнительного кода (ФДК), и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на -1). Принцип работы ФДК в зависимости от управляющих сигналов приведен в таблице 2.1

Таблица 2.1 - Режимы работы формирователя дополнительного кода.

Сигналы на входах ФДК		Результат на выходах ФДК
F1	F2	
0	0	Дополнительный код множимого
0	1	Дополнительный код слагаемого
1	0	Меняется знак Мн
1	1	Меняется знак слагаемого

На выходах 3,4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУ вместе с диадами множимого.

ОЧУ предназначен только для умножения двух четверичных цифр. Если в процессе умножения возникает перенос в следующий разряд, необходимо предусмотреть возможность его прибавления. Для суммирования результата умножения текущей диады  $M_n \cdot M_t$  с переносом из предыдущей диады предназначены ОЧС. Значит, чтобы полностью сформировать частичное произведение четверичных сомножителей, нужна комбинация цепочек ОЧУ и ОЧС. Частичные суммы формируются в аккумуляторе. На первом этапе он обнулен, и первая частичная сумма получается за счет сложения первого частичного произведения (сформированного на выходах ОЧС) и нулевой частичной суммы (хранящейся в аккумуляторе). Потом в аккумуляторе происходит сложение  $i$ -й частичной суммы с  $(i+1)$ -м частичным произведением, результат сложения сохраняется. Содержимое аккумулятора сдвигается на один четверичный разряд влево в конце каждого такта умножения по алгоритму «Б».

На четырех выходах ОЧУ формируется результат умножения диад  $M_n \cdot M_t$ . Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому в старшем разряде произведения максимальной цифрой может оказаться только «1» :

$$\begin{array}{ccc} 3 & \cdot & 2 \\ \text{max} & & \text{max} \\ M_n & & M_t \end{array} = 12.$$

Отсюда следует, что на младшие входы ОЧС никогда не поступят диады цифр, соответствующие кодам «2» и «3», следовательно, в таблице истинности работы ОЧС будут содержаться 16 безразличных входных наборов. Частичные суммы хранятся в аккумуляторе и регистре множителя, так как алгоритм умножения «Б» предполагает возможность синхронного сдвига этих устройств. Количество тактов умножения определяется разрядностью  $M_t$ .

### 3.РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ

#### 3.1 Логический синтез одноразрядного четверичного сумматора

ОЧС - это комбинационное устройство (5 входов и 3 выхода):

- 2 разряда одного слагаемого (множимого);
- 2 разряда второго слагаемого (регистр результата);
- вход переноса из младшего ОЧС.

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

Принцип работы ОЧС представлен с помощью таблицы 3.1.

Таблица 3.1 - Таблица истинности ОЧС

$a_1$	$a_2$	$b_1$	$b_2$	$p$	$\Pi$	$S_1$	$S_2$	Пример
0	0	0	0	0	0	0	0	0+0+0=00
0	0	0	0	1	0	1	1	0+0+1=01
0	0	0	1	0	x	x	x	0+2+0=02
0	0	0	1	1	x	x	x	0+2+1=03
0	0	1	0	0	x	x	x	0+3+0=03
0	0	1	0	1	x	x	x	0+3+1=10
0	0	1	1	0	0	1	1	0+1+0=01
0	0	1	1	1	0	0	1	0+1+1=02
0	1	0	0	0	0	0	1	2+0+0=02
0	1	0	0	1	0	1	0	2+0+1=03
0	1	0	1	0	x	x	x	2+2+0=10
0	1	0	1	1	x	x	x	2+2+1=11
0	1	1	0	0	x	x	x	2+3+0=11
0	1	1	0	1	x	x	x	2+3+1=12
0	1	1	1	0	0	1	0	2+1+0=03
0	1	1	1	1	1	0	0	2+1+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	x	x	x	3+2+0=11
1	0	0	1	1	x	x	x	3+2+1=12
1	0	1	0	0	x	x	x	3+3+0=12
1	0	1	0	1	x	x	x	3+3+1=13
1	0	1	1	0	1	0	0	3+1+0=10
1	0	1	1	1	1	1	1	3+1+1=11
1	1	0	0	0	0	1	1	1+0+0=01
1	1	0	0	1	0	0	1	1+0+1=02
1	1	0	1	0	x	x	x	1+2+0=03
1	1	0	1	1	x	x	x	1+2+1=10
1	1	1	0	0	x	x	x	1+3+0=10

Продолжение таблицы 3.1

1	1	1	0	1	x	x	x	1+3+1=11
1	1	1	1	0	0	0	1	1+1+0=02
1	1	1	1	1	0	1	0	1+1+1=03

Минимизацию выходов ОЧС  $S_2$  проведем с помощью карты Карно и Алгоритма Рота, минимизацию  $S_1$  и  $П$  проведем с помощью карт Карно.

**Минимизация функции  $П$  картами Карно:**

$b_1b_2p$ $a_1a_2$	000	001	011	010	110	111	101	100
00			X	X			X	X
01			X	X		1	X	X
11			X	X			X	X
10		1	X	X	1	1	X	X

Минимизировав функцию, получим:

$$f_{min_{ДНФ}} = a_1\bar{a}_2p + \bar{a}_1a_2b_1p + a_1\bar{a}_2b_1$$

**Минимизация функции  $S_1$  картами Карно:**

$b_1b_2p$ $a_1a_2$	000	001	011	010	110	111	101	100
00		1	X	X	1		X	X
01		1	X	X	1		X	X
11	1		X	X		1	X	X
10	1		X	X		1	X	X

Минимизировав функцию, получим:

$$f_{min_{СДНФ}} = a_1\bar{b}_2\bar{p} + \bar{a}_1\bar{b}_1p + \bar{a}_1b_2\bar{p} + a_1b_1p$$

### Минимизация функции $S_2$ картами Карно:

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

Минимизировав функцию, получим:

$$f_{\min_{\text{СДНФ}}} = \bar{a}_1 \bar{a}_2 p + \bar{a}_1 \bar{a}_2 b_2 + a_2 \bar{b}_2 \bar{p} + a_1 a_2 \bar{b}_2 + a_1 a_2 \bar{p} + \bar{a}_2 b_1 p$$

### Минимизация функции $S_2$ алгоритмом Рота:

Определим количество единичных кубов:

$$L = \{ 00001; 00110; 00111; 01000; 10111; 11000; 11001; 11110 \}$$

и множество безразличных кубов:

$$N = \{ 00010; 00011; 00100; 00101; 01010; 01011; 01100; 01101; 10010; 10011; 10100; 10101; 11010; 11011; 11100; 11101 \}$$

Таблица 3.2 – Минимизация N

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

$$N = \{ xx01x; xx10x \}$$

$C_0 = \{ 00001; 00110; 00111; 01000; 10111; 11000; 11001; 11110, xx01x; xx10x \}$

Первым этапом алгоритма Рота является нахождение множества простых импликант. Для реализации этого этапа будем использовать операцию умножения (\*) над множествами  $C_0$ ,  $C_1$  и т.д., пока в результате операции будут образовываться новые кубы большей размерности. Первый шаг умножения ( $C_0 * C_0$ ) приведен в табл. 3.3. В результате этой операции сформируем новое множество кубов:

Таблица 3.3 – Поиск простых импликант ( $C_0 * C_0$ )

$C_0 * C_0$	00001	00110	00111	01000	10111	11000	11001	11110	xx01x	xx10x
00001	-									
00110		-								
00111		0011y	-							
01000				-						
10111			y0111		-					
11000				y1000		-				
11001						1100y	-			
11110								-		
xx01x	000y1	00y10	00y11	010y0	10y11	110y0	110y1	11y10	-	
xx10x	00y01	001y0	001y1	01y00	101y1	11y00	11y01	111y0		-
A1	000x1 00x01	0011x 00x10 001x0	x0111 00x11 001x1	x1000 010x0 01x00	10x11 101x1	1100x 110x0 11x00	110x1 11x01	11x10 111x0	$\emptyset$	$\emptyset$

$A1 = \{ 000x1; 00x01; 0011x; 00x10; 001x0; x0111; 00x11; 001x1; x1000; 010x0; 01x00; 10x11; 101x1; 1100x; 110x0; 11x00; 110x1; 11x01; 11x10; 111x0 \}$

$Z0 = \{ \emptyset \}$

$B1 = \{ 00001; 00110; 00111; 01000; 10111; 11000; 11001; 11110; xx01x; xx10x \}$

$C1 = \{ 000x1; 00x01; 0011x; 00x10; 001x0; x0111; 00x11; 001x1; x1000; 010x0; 01x00; 10x11; 101x1; 1100x; 110x0; 11x00; 110x1; 11x01; 11x10; 111x0; xx01x; xx10x \}$

Таблица 3.4 – Поиск простых импликант ( $C_1 * C_1$ )

$C_1 * C_1$	000x1	00x01	0011x	00x10	001x0	x0111	00x11	001x1	x1000	010x0	01x00	10x11	101x1	1100x	110x0	11x00	110x1	11x01	11x10	111x0	xx01x	xx10x
000x1	-																					
00x01		-																				
0011x			-																			
00x10				-																		
001x0					-																	
x0111						-																
00x11		00xy1		00x1y			-															
001x1	00yx1				001xy			-														
x1000									-													
010x0										-												
01x00											-											
10x11							y0x11					-										
101x1								y01x1					-									
1100x														-								
110x0										y10x0					-							
11x00											y1x00					-						
110x1															110xy		-					
11x01																11x0y		-				
11x10																11xy0			-			
111x0																11yx0				-		
xx01x			00y1x			x0y11			x10y0					110yx							-	
xx10x			001yx			x01y1			x1y00					11y0x								-
A2	00xx1	00xx1	00x1x 001xx	00x1x	001xx	x0x11 x01x1	x0x11	x01x1	x10x0 x1x00	x10x0	x1x00	∅	∅	110xx 11x0x	110xx 11xx0	11x0x 11xx0	∅	∅	∅	∅	∅	∅

$$A2 = \{ 00xx1; 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; 11xx0 \}$$

$$Z1 = \{ \emptyset \}$$

$$B2 = \{ 000x1; 00x01; 0011x; 00x10; 001x0; x0111; 00x11; 001x1; x1000; 010x0; 01x00; 10x11; 101x1; 1100x; 110x0; 11x00; 110x1; 11x01; 11x10; 111x0; xx01x; xx10x \}$$

$$C2 = \{ 00xx1; 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; 11xx0; xx01x; xx10x \}$$



Таблица 3.5 – Поиск простых импликант ( $C_2 * C_2$ )

$C_2 * C_2$	00xx1	00x1x	001xx	x0x11	x01x1	x10x0	x1x00	110xx	11x0x	11xx0	xx01x	xx10x
00xx1	-											
00x1x		-										
001xx			-									
x0x11				-								
x01x1					-							
x10x0						-						
x1x00							-					
110xx								-				
11x0x									-			
11xx0										-		
xx01x											-	
xx10x												-
A3	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅

$$A3 = \{ \emptyset \}$$

Новых кубов (третьей размерности) не образовалось. На этом заканчивается этап поиска простых импликант, так как  $|C3| \leq 1$ . Конечное множество простых импликант  $Z = \{ 00xx1; 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; 11xx0; xx01x; xx10x \}$

Следующий этап - поиск L-экстремалей на множестве простых импликант. Для этого используется операция # (решетчатое вычитание). В табл. 3.6 из каждой простой импликанты поочередно вычитаются все остальные простые импликанты  $Z \# (Z \setminus z)$ , результат операции (последняя строка таблицы) указывает на то, что L-экстремалиями стали следующие простые импликанты:

$$E = \{ 00xx1; 11xx0 \}.$$

### Таблица 3.6 – Поиск L-экстремалей

[illegible]

Проверка L-экстремалий показана в таблице 3.7

Таблица 3.7

z#(Z-z) n L	00001	00110	00111	01000	10111	11000	11001	11110
00001	00001 Ø	00ууу Ø	00уу1 Ø	0у00у Ø	у0уу1 Ø	уу00у Ø	уу001 Ø	ууууу Ø
11110	ууууу Ø	уу110 Ø	уу11у Ø	у1уу0 Ø	1у11у Ø	11уу0 Ø	11ууу Ø	11110
10010	у00уу Ø	у0у10 Ø	у0у1у Ø	уу0у0 Ø	10у1у Ø	1у0у0 Ø	1у0уу Ø	1уу10 Ø
01011	0у0у1 Ø	0уу1у Ø	0уу11 Ø	010уу Ø	ууу11 Ø	у10уу Ø	у10у1 Ø	у1у1у Ø
10010	у00уу Ø	у0у10 Ø	у0у1у Ø	уу0у0 Ø	10у1у Ø	1у0у0 Ø	1у0уу Ø	1уу10 Ø
10100	у0у0у Ø	у01у0 Ø	у01уу Ø	ууу00 Ø	101уу Ø	1уу00 Ø	1уу0у Ø	1у1у0 Ø
01101	0уу01 Ø	0у1уу Ø	0у1у1 Ø	01у0у Ø	уу1у1 Ø	у1у0у Ø	у1у01 Ø	у11уу Ø
10100	у0у0у Ø	у01у0 Ø	у01уу Ø	ууу00 Ø	101уу Ø	1уу00 Ø	1уу0у Ø	1у1у0 Ø

$$E = \{ 00xx1; 11xx0 \}$$

$$Z' = Z - E = \{ 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; xx01x; xx10x \}$$

Таблица 3.8 – Множество L-экстремалей

L#E	00001	00110	00111	01000	10111	11000	11001	11110
00xx1	zzzzz Ø	zzzzy 00110	zzzzz Ø	zyzzy 01000	yzzzz 10111	уузzy 11000	уузzy 11001	уузzy 11110
11xx0	Ø	уузzy 00110	Ø	yzzzz 01000	zyzzy 10111	zzzzz Ø	zzzzy 11001	zzzzz Ø
Остаток	Ø	00110	Ø	01000	10111	Ø	11001	Ø

Множество кубов, непокрываемых L-экстремалиями,  $L' = L \# E = \{ 00110; 01000; 10111; 11001 \}$

Таблица 3.9– Множество кубов, непокрываемых L-экстремалями

Z' in L	00110	01000	10111	11001
00x1x	00110	∅	∅	∅
001xx	00110	∅	∅	∅
x0x11	∅	∅	10111	∅
x01x1	∅	∅	10111	∅
x10x0	∅	01000	∅	∅
x1x00	∅	01000	∅	∅
110xx	∅	∅	∅	11001
11x0x	∅	∅	∅	11001
xx01x	∅	∅	∅	∅
xx10x	∅	∅	∅	∅

Могут быть получены следующие тупиковые формы:

$$F_{min1} = \{00x1x; x01x1; x1x00; 11x0x; 00xx1; 11xx0\} =$$

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

Преобразуем функции  $\Pi$ ,  $S_1$ ,  $S_2$  к заданному логическому базису:

$$\begin{aligned} \Pi &= \overline{\bar{a}_1 + a_2 + \bar{p}} + \overline{a_1 + \bar{a}_2 + \bar{b}_1 + \bar{p}} + \overline{\bar{a}_1 + a_2 + \bar{b}_1} \\ S_1 &= \overline{\bar{a}_1 + b_2 + p} + \overline{a_1 + b_1 + \bar{p}} + \overline{a_1 + \bar{b}_2 + p} + \overline{\bar{a}_1 + \bar{b}_1 + \bar{p}} \\ S_2 &= \overline{a_1 + a_2 + \bar{p}} + \overline{a_1 + a_2 + \bar{b}_2} + \overline{\bar{a}_2 + b_2 + p} + \overline{\bar{a}_1 + \bar{a}_2 + b_2} + \\ &\quad \overline{\bar{a}_1 + \bar{a}_2 + p} + \overline{a_2 + \bar{b}_1 + \bar{p}} \end{aligned}$$

Функциональная схема в заданном логическом базисе ОЧС представлена в приложении Б.

### 3.2. Логический синтез одноразрядного четверичного умножителя

ОЧУ - это комбинационное устройство, имеющее 5 входов и 4 выхода.

Принцип работы ОЧУ представлен с помощью таблицы истинности (таблица 3.2).

Таблица 3.2 – Таблица истинности ОЧУ

$x_1$	$x_2$	$y_1$	$y_2$	$h$	$P_1$	$P_2$	$P_3$	$P_4$	Пример
0	0	0	0	0	0	0	0	0	$0*0=00$
0	0	0	0	1	0	0	0	0	Выход «00»
0	0	0	1	0	0	0	0	0	$0*0=00$
0	0	0	1	1	0	0	0	0	Выход «00»
0	0	1	0	0	0	0	0	0	$0*2=00$
0	0	1	0	1	0	0	0	0	Выход «00»
0	0	1	1	0	х	х	х	х	$0*3=00$
0	0	1	1	1	х	х	х	х	Выход «00»
0	1	0	0	0	0	0	0	0	$2*0=00$
0	1	0	0	1	0	0	0	1	Выход «02»
0	1	0	1	0	0	0	0	1	$2*1=02$
0	1	0	1	1	1	1	0	0	Выход «02»
0	1	1	0	0	0	1	0	1	$2*2=10$
0	1	1	0	1	0	0	0	1	Выход «02»
0	1	1	1	0	х	х	х	х	$2*3=12$
0	1	1	1	1	х	х	х	х	Выход «02»
1	0	0	0	0	0	0	0	0	$3*0=00$
1	0	0	0	1	0	0	1	0	Выход «03»
1	0	0	1	0	0	0	1	0	$3*1=03$
1	0	0	1	1	0	0	1	0	Выход «03»
1	0	1	0	0	1	1	0	1	$3*2=12$
1	0	1	0	1	0	0	1	0	Выход «03»
1	0	1	1	0	х	х	х	х	$3*3=21$
1	0	1	1	1	х	х	х	х	Выход «03»
1	1	0	0	0	0	0	0	0	$1*0=00$
1	1	0	0	1	0	0	1	1	Выход «01»
1	1	0	1	0	0	0	1	1	$1*1=01$
1	1	0	1	1	0	0	1	1	Выход «01»
1	1	1	0	0	0	0	0	1	$1*2=02$
1	1	1	0	1	0	0	1	1	Выход «01»
1	1	1	1	0	х	х	х	х	$1*3=03$
1	1	1	1	1	х	х	х	х	Выход «01»

Минимизацию выходов ОЧУ  $P_1, P_2, P_3$  и  $P_4$  проведем с помощью карт Вейча.

### Минимизация функции $P_1$ картами Вейча:

Diagram illustrating a 2D convolution operation on a 4x8 grid. The grid is labeled with  $x_1$  (top),  $x_2$  (bottom),  $y_1$  (left), and  $y_2$  (right). The grid contains the following values:

1	*	*	*	1	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*

The value 1 in the top-left cell is highlighted with a green box, and the value 1 in the top-right cell is highlighted with a red box. The grid is divided into two horizontal sections by a line, with the left section labeled  $h$  and the right section labeled  $h$ .

Минимизировав функцию, получим:

$$f_{min_{\text{сДНФ}}} = x_1 \bar{x}_2 y_1 \bar{h} + \bar{x}_1 x_2 y_1 \bar{h}$$

### Минимизация функции $P_2$ картами Вейча:

[illegible]

Минимизировав функцию, получим:  $f_{min_{CDHF}} = x_1 \bar{x}_2 y_1 \bar{h} + \bar{x}_1 x_2 y_1 \bar{h}$

### Минимизация функции $P_3$ картами Вейча:

Diagram illustrating a 2D convolution operation on a grid. The grid is labeled with  $x_1$  (horizontal axis) and  $y_1$  (vertical axis) on the left, and  $x_2$  (horizontal axis) and  $y_2$  (vertical axis) on the right. The grid contains values 1 and \*. The values 1 are highlighted with colored boxes: a blue box around the bottom-left 1, a green box around the top-middle 1s, and a red box around the middle-right 1s. The values \* are in the top-left, top-middle, top-right, and middle-left positions.

Минимизировав функцию, получим:

$$f_{min_{\text{ДНФ}}} = x_1 h + x_1 x_2 y_2 + x_1 \bar{x}_2 \bar{y}_1 \bar{y}_2$$

**Минимизация функции  $P_4$  картами Вейча:**

$x_1$							
$y_1$	1		1	1		1	
	*	*	*	*	*	*	*
			1	1	1	1	
			1			1	
$x_2$							
$h$				$h$			
$y_2$							

Минимизировав функцию, получим:

$$f_{min_{\text{ДНФ}}} = x_1 y_1 \bar{h} + x_2 h + x_1 x_2 y_1 \bar{y}_2 + x_2 y_2$$

Преобразуем функции ,  $S_1$ ,  $S_2$  к заданному логическому базису:

$$P_1/P_2 = (((x_1 \oplus 1) + x_2 + (y_1 \oplus 1) + h) \oplus 1) + ((x_1 + (x_2 \oplus 1) + (y_1 \oplus 1) + h) \oplus 1)$$

$$P_3 = (((x_1 \oplus 1) + (h \oplus 1)) \oplus 1) + (((x_1 \oplus 1) + (x_2 \oplus 1) + (y_2 \oplus 1)) \oplus 1) + (((x_1 \oplus 1) + x_2 + y_1 + y_2) \oplus 1)$$

$$P_4 = (((x_1 \oplus 1) + (y_1 \oplus 1) + h) \oplus 1) + (((x_2 \oplus 1) + (h \oplus 1)) \oplus 1) + ((x_1 \oplus 1) + (x_2 \oplus 1) + (y_1 \oplus 1) + y_2) \oplus 1 + (((x_2 \oplus 1) + (y_2 \oplus 1)) \oplus 1)$$

Функциональная схема в заданном логическом базисе ОЧУ представлена в приложении В.

#### 4.ЭФФЕКТИВНОСТЬ МИНИМИЗАЦИИ

Для проведения оценки эффективности минимизации переключательных функций необходимо посчитать цену схемы до минимизации и цену схемы после минимизации.

Все рассчитанные данные сведены в таблицу 4.1 и 4.2

Таблица 4.1 - Эффективность минимизации ОЧС

Выход схемы	Количество входов до минимизации	Количество входов после минимизации	Эффективность минимизации
$\Pi$	$c=5+4*5+4=29$	$c=10+2+3=15$	1,93
$S_1$	$c=5+8*5+8=53$	$c=12+4+4=20$	2,65
$S_2$	$c=5+8*5+8=53$	$c=18+4+6=28$	1,89

Таблица 4.2 - Эффективность минимизации ОЧУ

Выход схемы	Количество входов до минимизации	Количество входов после минимизации	Эффективность минимизации
$P_1/P_2$	$c=5+2*5+2=17$	$c=8+3+2=13$	1,3
$P_3$	$c=5+9*5+9=59$	$c=9+3+3=15$	3,93
$P_4$	$c=5+10*5+10=65$	$c=11+2+4=17$	3,82



## 5.СИНТЕЗ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРА

Мультиплексор – это логическая схема, имеющая  $n$  входов,  $m$  управляющих входов и один выход. При этом должно выполняться равенство  $n = 2^m$ . На выход мультиплексора может быть пропущен без изменений любой (один) логический сигнал, поступающий на информационные входы. Порядковый номер информационного входа, значение с которого в данный момент должно быть передано на выход, должно быть передано на выход, определяется двоичным кодом на управляющих входах. Для синтеза ОЧС будем использовать мультиплексор «один из восьми» (1 из 8-ми). Входы  $D_0, D_1, \dots, D_7$  – это информационные входы мультиплексора. Входы  $A_0, A_1, A_2$  – управляющие входы.

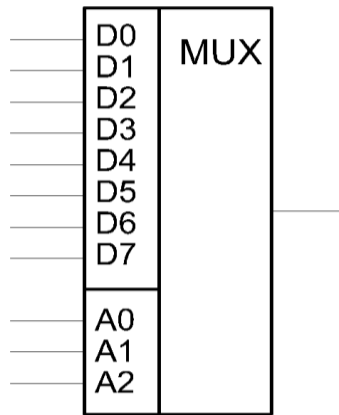


Рисунок 5.1 - Мультиплексор «один из восьми»

Используя таблицу истинности ОЧС, составим таблицу истинности для построения ОЧС на мультиплексорах (таблица 5.1).

Управление мультиплексором осуществляется тремя переменными:  $a_1, a_2, b_1$ , а вход соответствующих значений функций на информационные входы обеспечивается реализацией этих функций на дополнительных логических элементах.

$a_1$	$a_2$	$b_1$	$b_2$	$p$	$\Pi$		$S_1$		$S_2$	
0	0	0	0	0	0	“0”	0	$\bar{b}_2 * P$	0	$\bar{b}_2 * P$
0	0	0	0	1	0		1		1	
0	0	0	1	0	x		x		x	
0	0	0	1	1	x		x		x	
0	0	1	0	0	x	“0”	x	$b_2 * \bar{p}$	x	“1”
0	0	1	0	1	x		x		x	
0	0	1	1	0	0		1		1	
0	0	1	1	1	0		0		1	
0	1	0	0	0	0	“0”	0	$\bar{b}_2 * P$	1	$\bar{b}_2 * \bar{p}$
0	1	0	0	1	0		1		0	
0	1	0	1	0	x		x		x	
0	1	0	1	1	x		x		x	
0	1	1	0	0	x	$b_2 * P$	x	$b_2 * \bar{p}$	x	“0”
0	1	1	0	1	x		x		x	
0	1	1	1	0	0		1		0	
0	1	1	1	1	1		0		0	
1	0	0	0	0	0	$\bar{b}_2 * P$	1	$\bar{b}_2 * \bar{p}$	0	“0”
1	0	0	0	1	1		0		0	
1	0	0	1	0	x		x		x	
1	0	0	1	1	x		x		x	
1	0	1	0	0	x	“1”	x	$b_2 * P$	x	$b_2 * P$
1	0	1	0	1	x		x		x	
1	0	1	1	0	1		0		0	
1	0	1	1	1	1		1		1	
1	1	0	0	0	0	“0”	1	$\bar{b}_2 * \bar{p}$	1	“1”
1	1	0	0	1	0		0		1	
1	1	0	1	0	x		x		x	
1	1	0	1	1	x		x		x	
1	1	1	0	0	x	“0”	x	$b_2 * P$	x	$b_2 * \bar{p}$
1	1	1	0	1	x		x		x	
1	1	1	1	0	0		0		1	
1	1	1	1	1	0		1		0	

Функциональная схема ОЧС на основе мультиплексора представлена в приложении Г.

## 6. ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11, заменяя их на триады  $1\overline{0}1$ .

Таблица 6 - таблица истинности ПМ

Вх. диада		Триггер	Зн.	Вых. диада	
$Q_1$	$Q_2$	T	П	$S_1$	$S_2$
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	0	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	0

Проведём минимизацию П при помощи карты Карно:

		$Q_2T$				
		$Q_1$	00	01	11	10
	0					
	1		1	1	1	

$$P = Q_1 T + Q_1 Q_2$$

Очевидно, что  $S_1$  не минимизируется, поэтому  $S_1 = \overline{Q_1} Q_2 T + Q_1 \overline{Q_2} T$

Проведём минимизацию  $S_2$  при помощи карты Карно:

		$Q_2T$			
		$Q_1$	00	01	11
	0		1		1
	1		1		1

$$S_2 = \overline{Q_2} T + Q_2 \overline{T}$$

Реализация ПМ представлена в приложении Д.

## 7.ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ

Временные затраты на умножение сомножителей определяются в основном затратами на образование частичных произведений, получаемых на выходах ОЧС.

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

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

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

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

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

$T_{\text{сдвига}}$  – время сдвига частичной суммы;

$n$  – количество разрядов на множителе.

## ЗАКЛЮЧЕНИЕ

При выполнении курсового проекта были разработаны структурная схема сумматора-умножителя первого типа и функциональные схемы ОЧС и ОЧУ.

Одноразрядный четверичный умножитель – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда и регистра Мн, 2 разряда из регистра Мт и управляющий вход h) и 4 двоичных выхода.

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.

Мультиплексор-это логическая схема, имеющая n информационных входов, m управляющих входов и один выход. При этом должно выполняться условие  $n=2^m$ .

Принцип работы мультиплексора состоит в следующем. 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.

Также удалось усвоить навыки работы с различными типами минимизации переключательных функций, с проектированием функциональных схем в различных базисах и с расчетом времени работы схем.

## ЛИТЕРАТУРА

- [1] Савельев А.Я. Прикладная теория цифровых автоматов. М.: Высшая школа, 1985.
- [2] Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. Мн.: Высшая школа, 1980.
- [3] Лысиков Б.Г. Цифровая вычислительная техника. Мн.: , 2003 г.
- [4] Луцик Ю.А., Лукьянова И.В., Ожигина М.П. – Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". -Мн.: ротاپринт МРТИ ,2001 г.
- [5] Луцик Ю.А., Лукьянова И.В.– Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". - Мн.:ротاپринт МРТИ ,2004 г.