

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
вычислительной техники

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 01 520 ПЗ

Студент

Е. В. Максимчик

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
вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Б.В.Никульшин
«__» _____ 2022 г.

ЗАДАНИЕ

По курсовой работе студента
Максимчика Егора Валерьевича

- 1 Тема работы: Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел.
- 2 Срок сдачи студентом законченной работы: до 22 мая 2022 г.
- 3 Исходные данные к работе:
 - 3.1 Исходные сомножители: $M_n = 98,27$; $M_t = 12,23$.
 - 3.2 Алгоритм умножения: А.
 - 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; $0_4 - 11$, $1_4 - 10$, $2_4 - 00$, $3_4 - 01$.
 - 3.5 Тип синтезируемого умножителя: структурные схемы приведены для умножителя 1-ого типа (ОЧУ, ОЧС, аккумулятор).
 - 3.6 Логический базис для реализации ОЧС: И, ИЛИ, НЕ; метод минимизации – алгоритм Рота.
 - 3.7 логический базис для реализации ОЧУ: И, НЕ; 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 первого типа. Схема электрическая структурная.

5.2 Одноразрядный четвертичный сумматор. Схема электрическая функциональная.

5.3 Регистр-аккумулятор. Схема электрическая функциональная.

5.4 Одноразрядный четвертичных 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

_____ Е. В. Максимчик

СОДЕРЖАНИЕ

Введение	5
1. Разработка алгоритма умножения	6
2. Разработка структурной схемы сумматора-умножителя	9
3. Логический синтез одноразрядного четверичного сумматора	12
4. Логический синтез одноразрядного четверичного умножителя.....	16
5. Логический синтез одноразрядного четверичного сумматора на основе мультиплексора	24
6. Логический синтез преобразователя множителя	26
7. Временные затраты на умножение	27
Заключение	28
Список использованных источников	29
Приложение А	30
Приложение Б	31
Приложение В	32
Приложение Г	33
Приложение Д	34
Приложение Е	35

ВВЕДЕНИЕ

Данная курсовая работа посвящена разработке алгоритмов выполнения операций умножения и сложения. На основе полученных алгоритмов требуется разработать и синтезировать следующие устройства: одноразрядный четвертичный сумматор (ОЧС), одноразрядный четвертичный умножитель (ОЧУ), а также переключательные функции ОЧС на мультиплексорах. Минимизация перечисленных устройств осуществляется с помощью карт Карно-Вейча и алгоритма извлечения Рота. На основе полученных данных требуется построить схемы этих устройств и проанализировать результаты (эффективность минимизации и время выполнения операций).

1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

1.1 Перевод сомножителей из десятичной системы счисления в четверичную

Множимое

98 4	0,27
<u>96</u> 24 4	<u>4</u>
2 <u>24</u> 6 4	1,08
0 4 1	<u>4</u>
2	0,32

$M_{H4} = 1202,10$

в соответствии с заданной кодировкой множимого

$M_{H2/4} = 10001100,1011$

Множитель

12 4	0,23
<u>12</u> 3	<u>4</u>
0	0,92
	<u>4</u>
	3,68
	<u>4</u>
	2,72
	<u>4</u>
	2,88

$M_{T4} = 30,0322$

$M_{T2/4} = 1100,00111010$

множитель представляется
обычным весомозначным кодом:

0₄ - 01, 1₄ - 11, 2₄ - 00, 3₄ - 10

для всех вариантов

Запишем сомножители в форме с плавающей запятой в прямом коде:

$M_H = 0,100011001011$ $P_{M_H} = 0,1011$ + 10₄ - закодировано по условию

$M_T = 0,110000111010$ $P_{M_T} = 0,1011$ + 02₄ - закодировано традиционно

Умножение двух чисел с плавающей запятой на два разряда множителя одновременно в прямых кодах. Это сводится к сложению порядков, формированию знака произведения, преобразованию разрядов множителя согласно алгоритму, и перемножению мантисс сомножителей.

Порядок произведения будет равен:

$P_{M_H} =$	0.1011	10
$P_{M_T} =$	<u>0.0010</u>	<u>02</u>
$P_{M_H \cdot M_T} =$	0.1000	12

Результат закодирован в соответствии с заданием на кодировку множимого.

Знак произведения определяется суммой по модулю “два” знаков сомножителей:

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

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в прямых кодах диада $11_2(3_4)$ заменяется на триаду $10\bar{1}_2(\bar{1}\bar{1}_4)$. Преобразованный множитель имеет вид:

$$M_T^{\text{п}} = 1\bar{1}01\bar{1}22.$$

Перемножение мантисс по алгоритму “А” приведено в табл. 1.1.

Таблица 1.1 - Перемножение мантисс

Четверичная с/с			Двоично-четверичная с/с			Комментарии
1			2			3
0.	000000		11.	111111111111		$\sum_0^{\text{ч}}$
<u>0.</u>	<u>301020</u>		<u>11.</u>	<u>011110110011</u>		$\Pi_1^{\text{ч}} = M_H \cdot 2$
0.	301020		11.	011110110011		$\sum_1^{\text{ч}}$
0.	030102	0	11.	110111101100	11	$\sum_1^{\text{ч}} \cdot 2^{-2}$
<u>0.</u>	<u>301020</u>		<u>11.</u>	<u>011110110011</u>		$\Pi_2^{\text{ч}} = M_H \cdot 2$
0.	331122	0	11.	010110100000	11	$\sum_2^{\text{ч}}$
0.	<u>033112</u>	20	11.	110101101000	0011	$\sum_2^{\text{ч}} \cdot 2^{-2}$
<u>3.</u>	<u>213130</u>		<u>01.</u>	<u>001001100111</u>		$\Pi_3^{\text{ч}} = M_H \cdot (-1)$
3.	312302	20	01.	011000011100	0011	$\sum_3^{\text{ч}}$
3.	331230	220	01.	010110000111	000011	$\sum_3^{\text{ч}} \cdot 2^{-2}$
<u>0.</u>	<u>120210</u>		<u>11.</u>	<u>100011001011</u>		$\Pi_4^{\text{ч}} = M_H \cdot 1$
0.	112100	220	11.	101000101111	000011	$\sum_4^{\text{ч}}$
0.	011210	0220	11.	111010001011	11000011	$\sum_4^{\text{ч}} \cdot 2^{-2}$
<u>0.</u>	<u>000000</u>		<u>11.</u>	<u>111111111111</u>		$\Pi_5^{\text{ч}} = 0$
0.	011210	0220	11.	111010001011	11000011	$\sum_5^{\text{ч}}$
0.	001121	00220	11.	111110100010	1111000011	$\sum_5^{\text{ч}} \cdot 2^{-2}$
<u>3.</u>	<u>213130</u>		<u>01.</u>	<u>001001100111</u>		$\Pi_6^{\text{ч}} = M_H \cdot (-1)$
3.	220311	00220	01.	000011011010	1111000011	$\sum_6^{\text{ч}}$
3.	322031	100220	01.	010000110110	101111000011	$\sum_6^{\text{ч}} \cdot 2^{-2}$
<u>0.</u>	<u>120210</u>		<u>11.</u>	<u>100011001011</u>		$\Pi_7^{\text{ч}} = M_H \cdot 1$
0.	102301	100220	11.	101100011110	101111000011	$\sum_7^{\text{ч}}$

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение ($M_H \cdot M_{T4} = 0,102301100220$, $P_{M_H \cdot M_T} = 6$) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

$$\begin{aligned} M_H \cdot M_{T4} &= 102301,110022 & P_{M_H \cdot M_T} &= 0; \\ M_H \cdot M_{T10} &= 1201,3149. \end{aligned}$$

Результат прямого перемножения операндов даёт следующее значение:

$$M_H \cdot M_{T10} = 1201,8421.$$

Абсолютная погрешность:

$$\Delta = |1201,8421 - 1201,3149| = 0,5272$$

Относительная погрешность:

$$\delta = \frac{\Delta}{M_H \cdot M_T} = \frac{0,5272}{1201,8421} = 0,00043866 \quad (\delta = 0,043866\%)$$

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 первого типа строится на базе заданных узлов ОЧУ, ОЧС, формирователя дополнительного кода, преобразователя множителя и аккумулятора. Управление режимами работы схемы осуществляется внешним сигналом *mul/sum*, который определяет вид текущей арифметической операции (умножение или суммирование)

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

На выходах ФДК формируется дополнительный код одного из слагаемых с учётом знака. Это слагаемое может быть записано в регистр результата, при этом управляющие сигналы, поступающие на входы h всех ОЧУ, дают возможность переписать на выходы ОЧУ разряды слагаемого без изменений (рисунок 2.1)

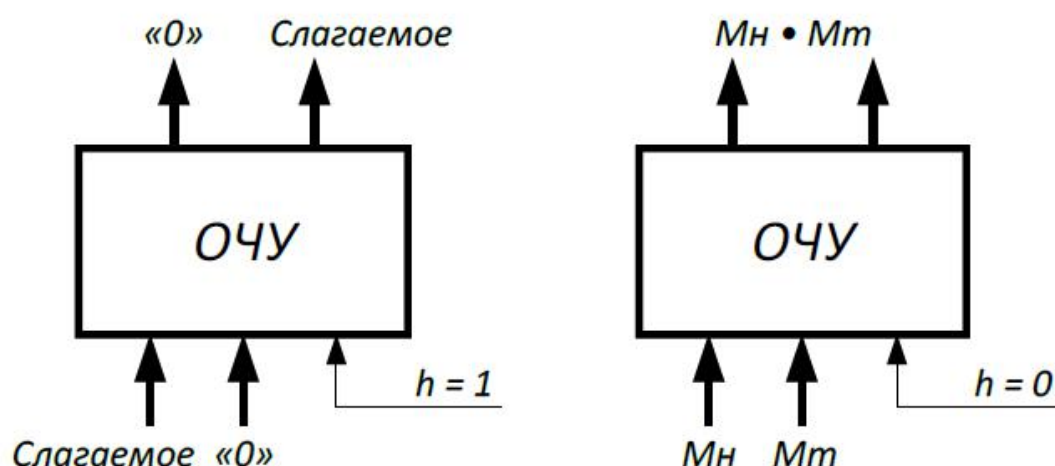


Рисунок 2.1 – Режимы работы ОЧУ

Если на вход h поступает «0», то ОЧУ перемножает разряды M_n и M_t .

Одноразрядный четверичный сумматор предназначен для сложения двух двоично-четверичных цифр, подаваемых на его входы (рисунок 2.2).

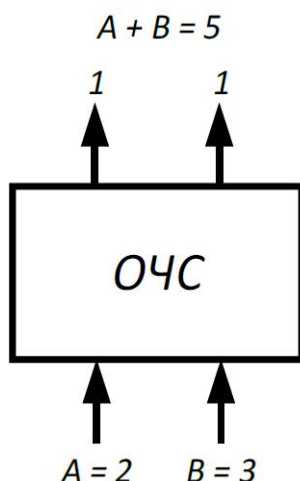


Рисунок 2.2 – Одноразрядный четверичный сумматор

В ОЧС первое слагаемое складывается с нулём, т.к. на старших выходах ОЧУ будут формироваться только коды нуля. Затем первое слагаемое попадает в регистр-аккумулятор, который изначально обнулён.

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.

Когда устройство работает как умножитель (на входе *Mul/sum* - «0»), то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК F_2 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 следующей старшей диады (выход 1 ПМ), т.е. сохраняться до следующего такта на триггер.

В регистре множителя в конце каждого такта умножения содержимое сдвигается на два двоичных разряда и в последнем такте умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения.

Выход 1 ПМ переходит в единичное состояние, если текущая диада содержит отрицание ($0\bar{1}$). В этом случае инициализируется управляющий вход F_7 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, 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 множимого
1	1	Меняется знак слагаемого

На выходах 2 и 3 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУ вместе с диадами множимого.

ОЧУ предназначен лишь 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 Мн·Мт с переносом из предыдущей диады предназначен ОЧС. Следовательно, чтобы полностью сформировать частичное произведение четверичных сомножителей, необходима комбинация цепочек ОЧУ и ОЧС.

Частичные суммы формируются в аккумуляторе. На первом этапе он обнулён и первая частичная сумма получается за счёт сложения первого частичного произведения (сформированного на выходах ОЧС) и нулевой частичной суммы (хранящейся в аккумуляторе).

В аккумуляторе происходит сложение i -й частичной суммы с $(i+1)$ -м частичным произведением, результат сложения сохраняется. Содержимое аккумулятора сдвигается на один четверичный разряд вправо в конце каждого такта умножения по алгоритму «А».

На четырёх выходах ОЧУ формируется результат умножения диад Мн·Мт. Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому в старшем разряде произведения максимальной цифрой может оказаться только «1»:

$$\begin{array}{ccccc} 3 & \cdot & 2 & = & 12 \\ \text{max} & & \text{max} & & \\ \text{Мн} & & \text{Мт} & & \end{array}$$

Это означает, 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.

Количество тактов умножения определяется разрядностью Мт.

3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ

3.1. Логический синтез одноразрядного четверичного умножителя

Одноразрядный четверичный умножитель – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда из регистра Мн, 2 разряда из регистра Мт и управляющий вход h) и 4 двоичных выхода. Принцип работы ОЧУ представлен с помощью таблицы истинности.

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

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

Управляющий вход h определяет тип операции:

- «0» – умножение закодированных цифр, поступивших на информационные входы;
- «1» – вывод на выходы без изменения значения разрядов, поступивших из регистра множимого.

В таблице 3.1 выделено восемь безразличных наборов, т.к. на входы ОЧУ из разрядов множителя не может поступить код «11».

Таблица 3.1 – Таблица истинности ОЧУ

Мн		Мт		Упр.	Старшие разряды		Младшие разряды		Пример операции в четверичной с/с
x_1	x_2	y_1	y_2	h	P_1	P_2	P_3	P_4	
1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	1	1	1	1	2*0=00
0	0	0	0	1	1	1	0	0	Выход – код «02»
0	0	0	1	0	1	1	0	0	2*1=02
0	0	0	1	1	1	1	0	0	Выход – код «02»
0	0	1	0	0	1	0	1	1	2*2=10
0	0	1	0	1	1	1	0	0	Выход – код «02»
0	0	1	1	0	х	х	х	х	2*3=12
0	0	1	1	1	х	х	х	х	Выход – код «02»
0	1	0	0	0	1	1	1	1	3*0=00
0	1	0	0	1	1	1	0	1	Выход – код «03»
0	1	0	1	0	1	1	0	1	3*1=03
0	1	0	1	1	1	1	0	1	Выход – код «03»
0	1	1	0	0	1	0	0	0	3*2=12
0	1	1	0	1	1	1	0	1	Выход – код «03»
0	1	1	1	0	х	х	х	х	3*3=21
0	1	1	1	1	х	х	х	х	Выход – код «03»
1	0	0	0	0	1	1	1	1	1*0=00

Продолжение таблицы 3.1

1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	1	1	1	0	Выход – код «01»
1	0	0	1	0	1	1	1	0	$1*1=01$
1	0	0	1	1	1	1	1	0	Выход – код «01»
1	0	1	0	0	1	1	0	0	$1*2=10$
1	0	1	0	1	1	1	1	0	Выход – код «01»
1	0	1	1	0	х	х	х	х	$1*3=03$
1	0	1	1	1	х	х	х	х	Выход – код «01»
1	1	0	0	0	1	1	1	1	$0*0=00$
1	1	0	0	1	1	1	1	1	Выход – код «00»
1	1	0	1	0	1	1	1	1	$0*1=00$
1	1	0	1	1	1	1	1	1	Выход – код «00»
1	1	1	0	0	1	1	1	1	$0*2=00$
1	1	1	0	1	1	1	1	1	Выход – код «00»
1	1	1	1	0	х	х	х	х	$0*3=00$
1	1	1	1	1	х	х	х	х	Выход – код «00»

Минимизацию функций P_1 и P_2 проведем с помощью карт Карно. Для функции P_1 заполненная карта изображена на рисунке 3.1.1. В рисунках под номерами 3.1.1 - 3.1.4 символом «х» отмечены наборы, на которых функция может принимать произвольные значения(безразличные наборы).

Рисунок 3.1.1 – Минимизация функции P_1 при помощи карты Карно

y_1y_2h x_1x_2	000	001	011	010	110	111	101	100
00	1	1	1	1	х	х	1	1
01	1	1	1	1	х	х	1	1
11	1	1	1	1	х	х	1	1
10	1	1	1	1	х	х	1	1

Следовательно:

$$P_1 = 1.$$

Для функции P_2 заполненная карта Карно приведена на рисунке 3.1.2

Рисунок 3.1.2 – Минимизация функции P_2 при помощи карты Карно

$\begin{matrix} y_1 y_2 h \\ x_1 x_2 \end{matrix}$	000	001	011	010	110	111	101	100
00	1	1	1	1	x	x	1	0
01	1	1	1	1	x	x	1	0
11	1	1	1	1	x	x	1	1
10	1	1	1	1	x	x	1	1

После выделения контуров получается следующее:

$$P_2 = x_1 + y_1 + h = x_1 y_1 h$$

Минимизацию функций P_3 и P_4 проведем при помощи карт Вейча.
Для функции P_3 заполненная карта приведена под рисунком 3.1.3.

Рисунок 3.1.3 – Минимизация функции P_3 при помощи карты Вейча

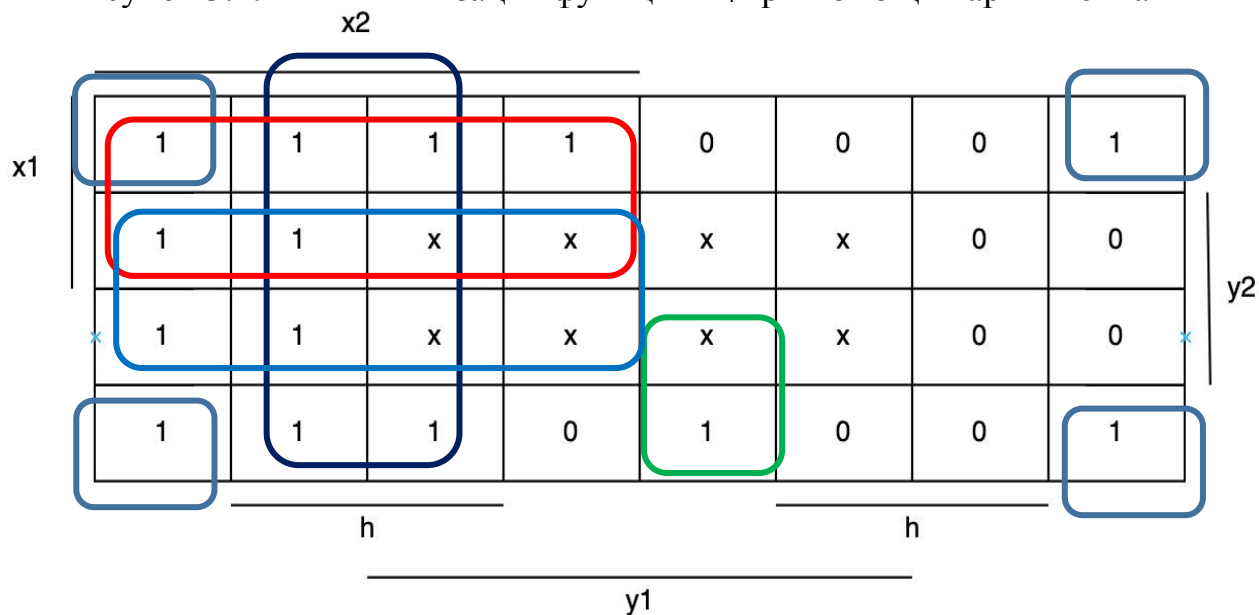
	x_2								
x_1	1	1	1	1	0	1	1	1	y_2
	1	1	x	x	x	x	1	1	
	0	0	x	x	x	x	0	0	
	1	0	0	0	1	0	0	1	
	h				h				
	y_1								

В итоге получается следующее уравнение:

$$P_3 = x_1 x_2 + x_1 y_1 + x_1 h + y_1 y_2 h + x_1 x_2 y_1 h = x_1 x_2 * x_1 y_1 * x_1 h * y_1 y_2 h * x_1 x_2 y_1 h$$

Результат упрощения функции P_4 с помощью карты Вейча приведён на рисунке 3.1.4

Рисунок 3.1.4 – Минимизация функции P_4 при помощи карты Вейча



После проведения минимизации получаем функцию:

$$P_4 = x_1x_2 + x_2y_2 + x_2h + y_1y_2h + x_1x_2y_1h = x_1x_2 * x_2y_2 * x_2h * y_1y_2h * x_1x_2y_1h$$

3.1 Логический синтез одноразрядного четверичного умножителя

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 3.1).

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

В таблица 3.1 выделено 16 безразличных наборов, т.к со страших выходов ОЧУ не могут прийти коды «2» и «3».

Таблица 3.1.5 – Таблица истинности ОЧС:

a1	a2	b1	b2	P	П	S1	S2	Пример в четверичной с/с
1	2	3	4	5	6	7	8	9
0	0	0	0	0	x	x	x	$2 + 2 + 0 = 10$
0	0	0	0	1	x	x	x	$2 + 2 + 1 = 11$
0	0	0	1	0	x	x	x	$2 + 3 + 0 = 11$
0	0	0	1	1	x	x	x	$2 + 3 + 1 = 12$
0	0	1	0	0	0	0	1	$2 + 1 + 0 = 03$
0	0	1	0	1	1	1	1	$2 + 1 + 1 = 10$
0	0	1	1	0	0	0	0	$2 + 0 + 0 = 02$
0	0	1	1	1	0	0	1	$2 + 0 + 1 = 03$
0	1	0	0	0	x	x	x	$3 + 2 + 0 = 11$
0	1	0	0	1	x	x	x	$3 + 2 + 1 = 12$
0	1	0	1	0	x	x	x	$3 + 3 + 0 = 12$
0	1	0	1	1	x	x	x	$3 + 3 + 1 = 13$
0	1	1	0	0	1	1	1	$3 + 1 + 0 = 10$
0	1	1	0	1	1	1	0	$3 + 1 + 1 = 11$
0	1	1	1	0	0	0	1	$3 + 0 + 0 = 03$
0	1	1	1	1	1	1	1	$3 + 0 + 1 = 10$
1	0	0	0	0	x	x	x	$1 + 2 + 0 = 03$
1	0	0	0	1	x	x	x	$1 + 2 + 1 = 10$
1	0	0	1	0	x	x	x	$1 + 3 + 0 = 10$
1	0	0	1	1	x	x	x	$1 + 3 + 1 = 11$
1	0	1	0	0	0	0	0	$1 + 1 + 0 = 02$
1	0	1	0	1	0	0	1	$1 + 1 + 1 = 03$
1	0	1	1	0	0	1	0	$1 + 0 + 0 = 01$
1	0	1	1	1	0	0	0	$1 + 0 + 1 = 02$
1	1	0	0	0	x	x	x	$0 + 2 + 0 = 02$

1	1	0	0	1	x	x	x	$0 + 2 + 1 = 03$
1	1	0	1	0	x	x	x	$0 + 3 + 0 = 03$
1	1	0	1	1	x	x	x	$0 + 3 + 1 = 10$
1	1	1	0	0	0	1	0	$0 + 1 + 0 = 01$
1	1	1	0	1	0	0	0	$0 + 1 + 1 = 02$
1	1	1	1	0	0	1	1	$0 + 0 + 0 = 00$
1	1	1	1	1	0	1	0	$0 + 0 + 1 = 01$

Минимизацию функций Π , S_2 проведем при помощи карт Вейча-Карно.
Для функции Π заполненная карта Карно показана на рисунке 3.1.6.

Рисунок 3.1.6 – Минимизация функции Π при помощи карты Карно.

		x2							
x1		x	x					x	x
		x	x					x	x
		x	x	1				x	x
		x	x	1	1		1	x	x
		h				h			
		y1							
		y2							

В результате минимизации получена функция:

$$\Pi = a_1 a_2 p + a_1 a_2 b_2 + a_1 a_2 b_2 p$$

Для минимизации функции S_2 используется карта Вейча заполненная на рисунке 3.1.7.

Рисунок 3.1.7 – Минимизация функции S_2 при помощи карты Вейча.

$b_1 b_2 p$ $a_1 a_2$	000	001	011	010	110	111	101	100
00	x	x	x	x	0	1	1	1
01	x	x	x	x	1	1	0	1
11	x	x	x	x	0	0	0	0
10	x	x	x	x	1	0	1	0

В результате минимизации получим функцию:

$$S_2 = a_1 a_2 b_2 + a_1 a_2 p + a_1 b_2 p + a_2 b_2 p + a_1 a_2 b_2 p$$

Минимизацию функции S_1 проведем с помощью алгоритма Рота. Для функции Π заполненная карта Вейча приведена на рисунке 3.2.1. В рисунках 3.2.1 - 3.2.2 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

Минимизация функции S_1 с помощью алгоритма Рота

Исходное покрытие функции задано множествами кубов на которых функция принимает единичное значение – L (единичных) и множество кубов на которых функция не определена – N (безразличных).

$$L = \{00101, 01100, 01101, 01111, 10110, 11100, 11101, 11111\}$$

$$N = \{00000, 00001, 00010, 00011, 01000, 01001, 01010, 01011, 10000, 10001, 10010, 10011, 11000, 11001, 11010, 11011\}$$

Минимизацию безразличных кубов проведем с помощью карты Карно. Для безразличных наборов заполненная карта приведена на рисунке 3.2.5, где символом «х» обозначены наборы, на которых функция не определена.

Рисунок 3.2.5 – Минимизация безразличных кубов.

	000	001	011	010	110	111	101	101
00	x	x	x	x				
01	x	x	x	x				
11	x	x	x	x				
10	x	x	x	x				

В результате минимизации безразличных кубов получим:

$$N = \{xx0xx\}$$

Сформируем множество $C_0 = L \cup N$

$$C_0 = \{00101, 01100, 01101, 01111, 10110, 11000, 11101, 11111, xx0xx\}$$

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., до тех пор пока будут образовываться новые кубы большей размерности.

Первый шаг умножения ($C_0 * C_0$) приведён в таблице 3.2.6

Таблица 3.2.6 – Поиск простых импликант ($C_0 * C_0$).

$C_0 * C_0$	00101	01100	01101	01111	10110	11100	11101	11111
00101	-							
01100		-						
01101	0y101	0110y	-					
01111			011y1	-				
10110					-			
11100		y1100				-		
11101			y1101			1110y	-	
11111				y1111			111y1	-
xx0xx	00y01	01y00	01y01	01y11	10y10	11y00	11y01	11y11

В результате сформируем новое множество кубов:

$$A_1 = \{x1100, x1101, x1111, 0x101, 00x01, \\ 01x00, 01x01, 01x11, 10x10, 11x00, 11x01, \\ 11x11, 011x1, 111x1, 0110x, 1110x\}$$

Множество Z_0 кубов, не участвовавших в образовании новых кубов, пустое. Так же формируется множество $B_1 = C_0 - Z_0$.

Для следующего шага получения множества Z формируется множество $C_1 = A_1 \cup B_1$.

$$C_1 = \{x1100, x1101, x1111, 0x101, 00x01, \\ 01x00, 01x01, 01x11, 10x10, 11x00, 11x01, \\ 11x11, 011x1, 111x1, 0110x, 1110x\}$$

В таблице 3.2.7 приведен следующий этап поиска простых импликант – операция умножения $C_1 * C_1$.

Таблица 3.2.7 – Поиск простых импликант ($C_1 * C_1$)

$C_1 * C_1$	x1100	x1101	x1111	0x101	00x01	01x00	01x01	01x11	10x10	11x00	11x01	11x11	011x1	111x1	0110x	1110x
x1100	–															
x1101	x110y	–														
x1111		x11y1	–													
0x101				–												
00x01					–											
01x00						–										
01x01					0yx01	01x0y	–									
01x11							01xy1	–								
10x10									–							
11x00						y1x00				–						
11x01							y1x01			11x0y	–					
11x11								y1x11			11xy1	–				
011x1													–			
111x1													y11x1	–		
0110x															–	
1110x															y110x	–
xx0xx	x1y00	x1y01	x1y11	0xy01									01yx1	11yx1	01y0x	11y0x

В результате операции $C_1 * C_1$ сформируем новое множество кубов A_2 :

$$A_2 = \{x1x00, x11x1, x1x01, x1x11, 0xx01, 01x0x, 01xx1, x1x11, 11x0x, 11xx1\}$$

Множество кубов, не участвовавших в образовании кубов новой размерности $Z_1 = \{10x10\}$.

Также формируется множество $B_2 = C_1 - Z_1$.

Для следующего шага получения множества Z формируется множество $C_2 = A_2 \cup B_2$

$$C_2 = \{x110x, x11x1, x1x01, 11x0x, 11xx1, 01x0x, 01xx1, 0xx01, x1x00, x1x11, xx0xx\}$$

В таблице 3.2.8 приведен следующий этап поиска простых импликант – операция умножения $C_2 * C_2$.

Таблица 3.2.8 – Поиск простых импликант $C_2 * C_2$

$C_2 * C_2$	x110x	x11x1	x1x01	11x0x	11xx1	01x0x	01xx1
x110x	—						
x11x1		—					
x1x01			—				
11x0x				—			
11xx1					—		
01x0x				y1x0x		—	
01xx1					y1xx1		—
0xx01							
x1x00							
x1x11			x1x0y				
x1x11			x1xy1				
xx0xx	x1y0x	x1yx1					

В результате очередной операции получим новое множество кубов A_3 :
 $A_3 = \{x1x0x, x1xx1\}$

Множество кубов, не участвовавших в образовании кубов новой размерности $Z_2 = \{10x10, 0xx01\}$.

Также формируется множество $B_3 = C_2 - Z_2$.

Для следующего шага получения множества Z формируется множество $C_3 = A_3 \cup B_3$.

В таблица 3.2.9 приведён следующий этап поиска простых импликант – операция умножения $C_3 * C_3$

Таблица 3.2.9 – Поиск простых импликант $C_3 * C_3$

$C_3 * C_3$	x1x0x	x1xx1
x1x0x	—	
x1xx1		—
xx0xx		

Новых кубов (четвертой размерности) в результате операции $C_3 * C_3$ не образовалось.

Получено множество $Z_3 = \{x1x0x, x1xx1, xx0xx\}$

Поскольку $|C_4| \leq 1$, поиск простых импликант заканчивается. Множество простых импликант:

$$Z = Z_0 \cup Z_1 \cup Z_2 \cup Z_3 = \{10x10, 0xx01, x1x0x, x1xx1, xx0xx\}$$

Следующий этап алгоритма – поиск L-экстремалей на множестве простых импликант (таблица 3.2.10). Для этого используется операция # (решёточное вычитание).

Таблица 3.2.10 – Поиск L-экстремалей

$z\#(Z-z)$	10x10	0xx01	x1x0x	x1xx1
10x10	–	0xx01	x1x0x	x1xx1
0xx01	10x10	–	11x0x x1x00	11x11 x1x11
x1x0x	10x10	00x01	–	11x11 x1x11
x1xx1	10x10	00x01	11x00 x1x00	–
xx0xx	10110	00101	11100 x1100	11100 x1100
Остаток	10110	00101	11100 x1100	11100 x1100

В таблице 3.2.10 из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты $Z\#(Z-z)$.

Получили кубы, претендующие на L-экстремальность. Проверяем в таблице 3.2.11.

Таблица 3.2.11 – Проверка на L-экстремальность

$z\#(Z-z) \cap L$	00101	01100	01101	01111	10110	11100	11101	11111
10110	\emptyset	\emptyset	\emptyset	\emptyset	10110	\emptyset	\emptyset	\emptyset
00101	00101	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
11100	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	11100	\emptyset	\emptyset
x1100	\emptyset	01100	\emptyset	\emptyset	\emptyset	11100	\emptyset	\emptyset
11111	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	11111
x1111	\emptyset	\emptyset	\emptyset	01111	\emptyset	\emptyset	\emptyset	11111
0001x	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

0x010	∅	∅	∅	∅	∅	∅	∅	∅
000x0	∅	∅	∅	∅	∅	∅	∅	∅
0x010	∅	∅	∅	∅	∅	∅	∅	∅
11010	∅	∅	∅	∅	∅	∅	∅	∅
x1010	∅	∅	∅	∅	∅	∅	∅	∅
1000x	∅	∅	∅	∅	∅	∅	∅	∅
x0000	∅	∅	∅	∅	∅	∅	∅	∅
100x1	∅	∅	∅	∅	∅	∅	∅	∅
100x1	∅	∅	∅	∅	∅	∅	∅	∅
10011	∅	∅	∅	∅	∅	∅	∅	∅
x0011	∅	∅	∅	∅	∅	∅	∅	∅

По результатам таблицы 3.2.11, L-экстремальями стали кубы E. Эти кубы обязательно должны войти в минимальное покрытие.

$$E = \{ 10x10, 0xx01, x1x0x, x1xx1 \}$$

Далее необходимо проанализировать, какие из исходных единичных кубов не покрыты найденной L-экстремалью. Анализ осуществляется с помощью таблицы 3.2.12.

Таблица 3.2.12 – Поиск непокрытых кубов

L#E	00101	01100	01101	01111	10110	11100	11101	11111
10x10	00101	01100	01101	01111	∅	11100	11101	11111
0xx01	∅	01100	∅	01111	∅	11100	11101	11111
x1x0x	∅	∅	∅	01111	∅	∅	∅	11111
x1xx1	∅	∅	∅	∅	∅	∅	∅	∅
Остаток	∅	∅	∅	∅	∅	∅	∅	∅

Из таблицы 3.2.12 видно, что все наборы покрыты.

Следовательно, минимальное покрытие будет выглядеть следующим образом:

$$S_1 = a_1a_2b_2p + a_1b_2p + a_2b_2 + a_2p.$$

4. ЛОГИЧЕСКИЙ СИНТЕЗ ОДНОРАЗРЯДНОГО ЧЕТВЕРИЧНОГО СУММАТОРА НА ОСНОВЕ МУЛЬТИПЛЕКСОРА

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.

Функции ОЧС зависят от пяти переменных. Удобно взять мультиплексор с тремя управляющими входами, это позволит упростить одну нашу большую функцию от пяти аргументов до восьми функций от одной переменной.

Функциональная схема ОЧС на базе мультиплексоров приведена на чертеже ГУИР.400201.520 Э2.3

a ₁	a ₂	b ₁	b ₂	p	Π	Функция	S ₁	Функция	S ₂	Функция
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	x		x		x	
0	0	0	0	1	x		x		x	
0	0	0	1	0	x		x		x	
0	0	0	1	1	x		x		x	
0	0	1	0	0	0	$\overline{b_2 + \overline{p}}$	0	$\overline{b_2 + \overline{p}}$	1	$\overline{b_2 + p}$
0	0	1	0	1	1		1		1	
0	0	1	1	0	0		0		0	
0	0	1	1	1	0		0		1	
0	1	0	0	0	x		x		x	
0	1	0	0	1	x		x		x	
0	1	0	1	0	x		x		x	
0	1	0	1	1	x		x		x	
0	1	1	0	0	1	$\overline{b_2 + p}$	1	$\overline{b_2 + p}$	1	$b_2 + \overline{p}$
0	1	1	0	1	1		1		0	
0	1	1	1	0	0		0		1	
0	1	1	1	1	1		1		1	
1	0	0	0	0	x		x		x	
1	0	0	0	1	x		x		x	
1	0	0	1	0	x		x		x	
1	0	0	1	1	x		x		x	
1	0	1	0	0	0	“0”	0	$\overline{\overline{b_2 + p}}$	0	$\overline{\overline{b_2 + p}}$
1	0	1	0	1	0		0		1	
1	0	1	1	0	0		1		0	
1	0	1	1	1	0		0		0	
1	1	0	0	0	x		x		x	
1	1	0	0	1	x		x		x	
1	1	0	1	0	x		x		x	
1	1	0	1	1	x		x		x	
1	1	1	0	0	0	“0”	1	$b_2 + \overline{p}$	0	$\overline{\overline{b_2 + p}}$
1	1	1	0	1	0		0		0	
1	1	1	1	0	0		1		1	
1	1	1	1	1	0		1		0	

5 ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ (ПМ)

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11, заменяя их на триады $10\bar{1}$.

Таблица 5.1 - Таблица истинности ПМ.

Вх. диада		Мл. бит	Зн.	Вых. диада	
Q_n	Q_{n-1}	Q_{n-2}	P	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

Минимизируем выходные функции S_1, S_2 и P картами Карно

Таблица 5.2 – Минимизация функции P

$Q_{n-1}Q_{n-2}$		00	01	11	10
Q_n					
0					
1			1	1	1

$$f_{\min_{\text{ДНХ}}} = Q_n Q_{n-1} + Q_n Q_{n-2}$$

Видно, что S_1 не минимизируется, поэтому $S_1 = Q_n \overline{Q_{n-1}} \overline{Q_{n-2}} + \overline{Q_n} Q_{n-1} Q_{n-2}$

Проведём минимизацию S_2 при помощи карт Карно:

Таблица 5.3 – Минимизация функции S_2

$Q_{n-1}Q_{n-2}$		00	01	11	10
Q_n					
0			1		1
1			1		1

$$S_2 = \overline{Q_{n-1}} Q_{n-2} + Q_{n-1} \overline{Q_{n-2}} = Q_{n-1} \oplus Q_{n-2}$$

6 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ

6.1 ОЦЕНКА ЭФФЕКТИВНОСТИ МИНИМИЗАЦИИ ПЕРЕКЛЮЧАТЕЛЬНЫХ ФУНКЦИЙ

Для проведения оценки эффективности минимизации переключательных функций необходимо посчитать цену схемы до минимизации и цену схемы после минимизации. Эффективность минимизации k определяется как:

$$k = \frac{C_{до_мин}}{C_{после_мин}}$$

Таблица 6.1 – Эффективность минимизации ОЧУ

Вых. схемы	Рассчитанная цена схемы		Эфф. мин. k
	До минимизации	После минимизации	
P ₁	c = 24*5+5+24=149	c = 1	149
P ₂	c = 22*5+5+22=137	c = 4	34,25
P ₃	c = 14*5+5+14=89	c = 6 + 13 + 5 = 24	3,7
P ₄	c = 14*5+5+14=89	c = 6 + 13 + 5 = 24	3,7

Таблица 6.2 – Эффективность минимизации ОЧС

Вых. схемы	Рассчитанная цена схемы		Эфф. мин. k
	До минимизации	После минимизации	
П	c = 4*5+5+4=29	c = 10 + 3 = 13	2,23
S ₁	c = 8*5+5+8=53	c = 11 + 4 = 15	3,53
S ₂	c = 8*5+5+8=53	c = 16 + 5 = 21	2,52

6.2 Временные затраты на умножение

Формула расчёта временных затрат на умножение:

$$T_y = 6 \cdot (T_{сдвига} + T_{ПМ} + T_{ХДК} + T_{ОЧУ} + 6T_{ОЧС}), \text{ где}$$

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

$T_{ОЧУ}$ – время умножения на ОЧУ;

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

$T_{ПМ}$ – время преобразования множителя;

$T_{ХДК}$ – время формирования дополнительного кода множимого.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы были выполнены первоначально заданные цели, а именно разработана структурная схема сумматора-умножителя первого типа и функциональные схемы основных узлов данного устройства. Для уменьшения стоимости логических схем, переключательные функции были минимизированы 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. В то же время, минимизация алгоритмом Рота вручную может быть очень времязатратной, если функция принимает большое количество единичных и безразличных наборов.

Функциональные схемы были построены в различных логических базисах. Это позволило закрепить теоретические знания основных законов булевой алгебры, например, правило де Моргана. Также можно отметить, что необходимо сократить количество уровней в логической схеме для уменьшения времени работы данного устройства.

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

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Луцик Ю.А., Лукьянова И.В. – Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". – Минск: БГУИР, 2014 г. – 174с.
- [2] Луцик Ю.А., Лукьянова И.В. – Методические указания к курсовому проекту по курсу "Арифметические и логические основы вычислительной техники". – Мн.: БГУИР, 2004 г. – 77с.
- [3] Искра, Н. А. Арифметические и логические основы вычислительной техники: пособие / Н. А. Искра, И. В. Лукьянова, Ю. А. Луцик. – Минск: БГУИР, 2016г. – 75с.
- [4] Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. Мн.: Высшая школа, 1980. – 342с.
- [5] Лысиков Б.Г. Цифровая вычислительная техника / Б.Г. Лысиков – Минск: Высш. шк., 2003. – 242с.
- [6] Единая система конструкторской документации (ЕСКД): справ. пособие / С.С. Борушек [и др.]. – М.: Изд-во стандартов, 1989. – 352с.

ПРИЛОЖЕНИЕ А
(обязательное)

Сумматор-умножитель первого типа.
Схема электрическая структурная.

ПРИЛОЖЕНИЕ Б

(обязательное)

Одноразрядный четверичный умножитель.

Схема электрическая функциональная.

ПРИЛОЖЕНИЕ В

(обязательное)

Одноразрядный четверичный сумматор.

Схема электрическая функциональная

ПРИЛОЖЕНИЕ Г

(обязательное)

Одноразрядный четверичный сумматор.

Реализация на мультиплексорах.

Схема электрическая функциональная

ПРИЛОЖЕНИЕ Д
(обязательное)

Преобразователь множителя.
Схема электрическая функциональная

ПРИЛОЖЕНИЕ Е

(обязательное)

Ведомость документов