

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 01 409 ПЗ

Студент

А. А. Кашевский

Руководитель

И. В. Лукьянова

МИНСК 2021

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
«___» _____ 2021 г.

ЗАДАНИЕ
по курсовой работе студента
Кашевского Арсения Анатольевича

- 1 Тема работы: “Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел”
- 2 Срок сдачи студентом законченной работы: 1 июня 2021 г.
- 3 Исходные данные к работе:
 - 3.1 исходные сомножители: $M_n = 73,48$; $M_t = 49,13$.
 - 3.2 алгоритм умножения: В.
 - 3.3 метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в дополнительном коде.
 - 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: структурные схемы приведены для умножителя 1-ого типа (ОЧУ, ОЧС, аккумулятор).
 - 3.6 логический базис для реализации ОЧС: И, НЕ; метод минимизации – алгоритм Рота.
 - 3.7 логический базис для реализации ОЧУ: ИЛИ-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 Одноразрядный четвертичный умножитель. Схема электрическая функциональная.
- 5.4 Одноразрядный четвертичных сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.
- 5.5 Преобразователь множителя. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов курсовой работы	Объём этапа, %	Срок выполнения этапа	Примечания
Разработка алгоритма умножения	10	22.02-13.03	
Разработка структурной схемы сумматора-умножителя	10	14.03-27.03	С выполнением чертежа
Разработка функциональных схем основных узлов сумматора-умножителя	50	28.03-08.05	С выполнением чертежей
Синтез комбинационных схем устройств на основе мультиплексоров	10	09.05-22.05	С выполнением чертежа
Завершение оформления пояснительной записки	20	23.05-05.06	

Дата выдачи задания: 11 февраля 2021 г.

Руководитель

И. В. Лукьянова

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ _____

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ.....	6
2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ	8
3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ.....	11
3.1 Логический синтез одноразрядного четверичного умножителя.....	11
3.2 Логический синтез одноразрядного четверичного сумматора.....	15
4 СИНТЕЗ СХЕМЫ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ	26
5 ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ (ПМ)	28
6 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ.....	29
ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	31
ПРИЛОЖЕНИЕ А	32
ПРИЛОЖЕНИЕ Б.....	33
ПРИЛОЖЕНИЕ В	34
ПРИЛОЖЕНИЕ Г	35
ПРИЛОЖЕНИЕ Д	36
ПРИЛОЖЕНИЕ Е.....	37

ВВЕДЕНИЕ

Курсовое проектирование является обязательным элементом подготовки специалиста с высшим образованием и является одной из форм текущей аттестации студента по учебной дисциплине.

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

- разработать алгоритм умножения и оценить погрешности вычислений;
- разработать структурную схему сумматора-умножителя первого типа;
- разработать функциональные схемы основных узлов сумматора-умножителя в заданных логических базисах;
- разработать схему одноразрядного четверичного сумматора на основе мультиплексора;
- разработать преобразователь множителя;
- рассчитать время умножения;
- оформить документацию о проделанной работе.

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

1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

Множимое

$$\begin{array}{r}
 73 \quad | \quad 4 \\
 \underline{4} \quad 18 \quad | \quad 4 \\
 33 \quad \underline{16} \quad 4 \quad | \quad 4 \\
 \underline{32} \quad 2 \quad 0 \quad | \quad 1 \\
 1 \\
 \hline
 0,48 \\
 \underline{4} \\
 1,92 \\
 \underline{4} \\
 3,68 \\
 \underline{4} \\
 2,72
 \end{array}$$

$M_{H4} = 1021,132$
 в соответствии с заданной
 кодировкой множимого
 $M_{H2/4} = 11000111,111001$

Множитель

$$\begin{array}{r}
 49 \quad | \quad 4 \\
 \underline{4} \quad 11 \quad | \quad 4 \\
 09 \quad \underline{12} \quad 3 \\
 \underline{8} \quad 0 \\
 1 \\
 \hline
 0,13 \\
 \underline{4} \\
 0,52 \\
 \underline{4} \\
 2,08 \\
 \underline{4} \\
 0,32
 \end{array}$$

$M_{T4} = 301,020$
 В соответствии с обычной
 весомозначной кодировкой
 множителя:
 $M_{T2/4} = 110001,001000$

Запишем сомножители в форме с плавающей запятой в прямом коде:

$M_H = 0,11000111111001$ $P_{MH} = 0,1100 + 10_{10}$ закодировано по заданию
 $M_T = 0,110001001000$ $P_{MT} = 0,0011 + 03_{10}$ 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:

$$P_{MH} = 0,1100 \quad +10_4$$

$$P_{MT} = 0,0011 \quad +03_4$$

$$P_{MH \cdot MT} = 0,1110 \quad +13_4$$

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 дополнительных кодах диада $11(3_4)$ заменяется на триаду $1\bar{0}\bar{1}$, а диада $10(2_4)$ заменяется на триаду $1\bar{1}\bar{0}$

Преобразованный множитель имеет вид: $M_T^{п4} = 1\bar{1}01,1\bar{2}0$. Перемножение мантисс по алгоритму “В” приведено в таблице 1.1.

Таблица 1.1 - Перемножение мантисс

Четверичная с/с			Двоично-четверичная с/с			Комментарии
0.	000000	000000	0.	010101010101	010101010101	$\sum_0^q \cdot 2^2$
0.	000001	021132	0.	010101010101	111100111010	$\Pi_1^q = [M_H]$
0.	000000	102113	0.	010101010101	111100111010	\sum_1^q
0.	000000	313012	0.	010101010111	110011101001	$\sum_1^q \cdot 2^2$
3.	333333	231220	1.	111111111111	010110010010	$\Pi_2^q = [-M_H]$
0.	000002	321232	0.	000100100001	111000101110	\sum_2^q
0.	000003	130122	0.	010001001000	100010111001	$\sum_2^q \cdot 2^2$
0.	000000	000000	1.	000000000000	000000000000	$\Pi_3^q = [M_H]$
0.	000003	130122	0.	010101011010	101001111110	\sum_3^q
0.	000031	301220	0.	010101101010	100111111001	$\sum_3^q \cdot 2^2$
0.	000001	021132	0.	010101010101	010101010101	$\Pi_4^q = 0$
0.	000032	323012	0.	010101101010	100111111001	\sum_4^q
0.	000323	230120	0.	010110101010	011111100101	$\sum_4^q \cdot 2^2$
0.	000000	102113	0.	010101010101	111100111010	$\Pi_5^q = [M_H]$
0.	000330	311312	0.	010110101011	011100001010	\sum_5^q
0.	003303	113120	0.	011010101101	110000101001	$\sum_5^q \cdot 2^2$
3.	333333	123101	0.	010101010100	111011110001	$\Pi_6^q = [-M_H]$
0.	003301	010130	0.	011010101110	110100001101	\sum_6^q
0.	033010	101300	0.	101010111011	010000110101	$\sum_6^q \cdot 2^2$
0.	330101	013000	1.	111111111110	010001011101	$\Pi_7^q = [M_H]$
0.	330101	013000	0.	101010111000	011100111101	$\sum_7^q = M_H \cdot M_T$

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение ($M_H \cdot M_{T4} = 0,330101013$, $R_{M_H \cdot M_T} = 6$) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

$$M_H \cdot M_{T4} = 330101.013 \quad R_{M_H \cdot M_T} = 0;$$

$$M_H \cdot M_{T10} = 3857.109375$$

Результат прямого перемножения операндов дает следующее значение:

$$M_{H10} \cdot M_{T10} = 73,48 \cdot 49.13 = 3609.152.$$

Абсолютная погрешность:

$$\Delta = 3857.109375 - 3609.152375 = 247.957031.$$

Относительная погрешность:

$$\delta = \frac{\Delta}{M_H \cdot M_T} = \frac{247,957031}{3609,152375} = 0,0687023 \quad (\delta = 6,87023\%)$$

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

2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ

Структурная схема сумматора-умножителя первого типа для алгоритма умножения «В» представлена на схеме ГУИР 400201.409 Э1.

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).

Когда устройство работает как сумматор (на входе *mul/sum* – «1»), 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).

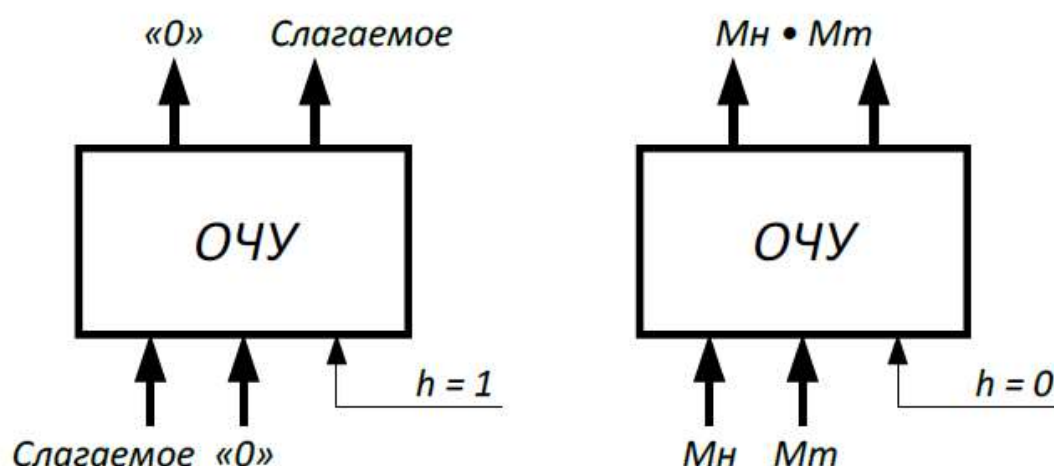


Рисунок 2.1 – Режимы работы ОЧУ

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).

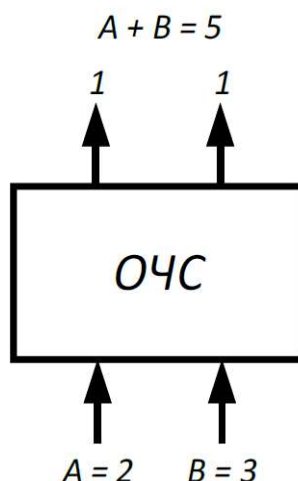


Рисунок 2.2 – Одноразрядный четверичный сумматор

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»), множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК F_2 поступает «0».

В регистре множителя в конце каждого такта умножения содержимое сдвигается на два двоичных разряда и в последнем такте умножения регистр обнуляется.

Триада множителя поступает на входы преобразователя множителя.

В регистре множителя после каждого такта умножения содержимое сдвигается на два двоичных разряда и в конце умножения регистр обнуляется.

Выход 1 ПМ переходит в единичное состояние, если текущая диада содержит отрицание ($\overline{01}$ или $\overline{10}$). В этом случае инициализируется управляющий вход F_1 формирователя дополнительного кода 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

Продолжение таблицы 2.1

0	1	Дополнительный код слагаемого
1	0	Меняется знак Мн
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.

ОЧС предназначен для суммирования результата умножения текущей диады Мн·Мт с переносом из предыдущей диады. 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 влево в конце каждого такта умножения по алгоритму «В».

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»:

$$\begin{array}{ccc} 3_4 & \cdot & 2_4 \\ \text{max} & & \text{max} \\ \text{Мн} & & \text{Мт} \end{array} = 12_4$$

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 (таблица 3.1).

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

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

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	0	0	0	0	$0*0=00$
0	0	0	0	1	0	0	0	0	Выход – код «00»
0	0	0	1	0	0	0	0	0	$0*1=00$
0	0	0	1	1	0	0	0	0	Выход – код «00»
0	0	1	0	0	0	0	0	0	$0*2=00$
0	0	1	0	1	0	0	0	0	Выход – код «00»
0	0	1	1	0	х	х	х	х	$0*3=00$
0	0	1	1	1	х	х	х	х	Выход – код «03»
0	1	0	0	0	0	0	0	0	$2*0=00$
0	1	0	0	1	0	0	0	1	Выход – код «02»
0	1	0	1	0	0	0	0	1	$2*1=02$
0	1	0	1	1	0	0	0	1	Выход – код «02»
0	1	1	0	0	1	1	0	0	$2*2=10$
0	1	1	0	1	0	0	0	1	Выход – код «02»
0	1	1	1	0	х	х	х	х	$2*3=12$
0	1	1	1	1	х	х	х	х	Выход – код «02»
1	0	0	0	0	0	0	0	0	$3*0=00$

Продолжение таблицы 3.1

1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	0	0	1	0	Выход – код «03»
1	0	0	1	0	0	0	1	0	$3*1=03$
1	0	0	1	1	0	0	1	0	Выход – код «03»
1	0	1	0	0	1	1	0	1	$3*2=12$
1	0	1	0	1	0	0	1	0	Выход – код «03»
1	0	1	1	0	х	х	х	х	$3*3=21$
1	0	1	1	1	х	х	х	х	Выход – код «03»
1	1	0	0	0	0	0	0	0	$1*0=00$
1	1	0	0	1	0	0	1	1	Выход – код «01»
1	1	0	1	0	0	0	1	1	$1*1=01$
1	1	0	1	1	0	0	1	1	Выход – код «01»
1	1	1	0	0	0	0	0	1	$1*2=02$
1	1	1	0	1	0	0	1	1	Выход – код «01»
1	1	1	1	0	х	х	х	х	$1*3=03$
1	1	1	1	1	х	х	х	х	Выход – код «01»

Минимизацию функций проведём с помощью карт Карно. Для функции P_1 заполненная карта приведена на рисунке 3.1.1. В рисунках 3.1.1 – 3.1.3 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

$y_1 y_2 h$ $x_1 x_2$									
		000	001	011	010	110	111	101	100
00		0	0	0	0	х	х	0	0
		0	0	0	0	х	х	0	1
01		0	0	0	0	х	х	0	0
		0	0	0	0	х	х	0	1
10		0	0	0	0	х	х	0	1
		0	0	0	0	х	х	0	1

Рисунок 3.1.1 – Минимизация функции P_1 при помощи карты Карно

Следовательно:

$$P_1 = \bar{x}_1 x_2 y_1 h + x_1 \bar{x}_2 y_1 h$$

Запишем результат минимизации в логическом базисе ИЛИ-НЕ:

$$P_1 = \overline{x_1 + \bar{x}_2 + \bar{y}_1 + \bar{h}} + \overline{\bar{x}_1 + x_2 + \bar{y}_1 + \bar{h}}$$

Функция P_2 идентична P_1

Для функции P_3 заполненная карта приведена на рисунке 3.1.2.

$y_1 y_2 h$ $x_1 x_2$	000	001	011	010	110	111	101	100
00	0	0	0	0	x	x	0	0
01	0	0	0	0	x	x	0	0
11	0	1	1	1	x	x	1	0
10	0	1	1	1	x	x	1	0

Рисунок 3.1.2 – Минимизация функции P_3 при помощи карт Карно

Следовательно:

$$P_3 = x_1 h + x_1 y_2$$

Запишем результат минимизации в логическом базисе ИЛИ-НЕ:

$$P_3 = \overline{\bar{x}_1 + \bar{h}} + \overline{\bar{x}_1 + \bar{y}_2}$$

Для функции P_4 заполненная карта приведена на рисунке 3.1.3.

Рисунок 3.1.3 - Минимизация функции P_4 при помощи карт Карно

$y_1 y_2 h$ $x_1 x_2$	000	001	011	010	110	111	101	100
00	0	0	0	0	x	x	0	0
01	0	1	1	1	x	x	1	0
11	0	1	1	1	x	x	1	1
10	0	0	0	0	x	x	0	1

Следовательно:

$$P_4 = x_2 h + x_2 y_2 + x_1 y_1 \bar{y}_2 \bar{h}$$

Запишем результат минимизации в логическом базисе ИЛИ-НЕ:

$$P_4 = \overline{\overline{x_2 + h} + \overline{x_2 + y_2} + \overline{x_1 + y_1 + y_2 + h}}$$

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

$$K_{P_1} = \frac{8}{5} = 1,8.$$

$$K_{P_3} = \frac{10 \cdot 5 + 10 + 5}{18} = 3,61.$$

$$K_{P_4} = \frac{14 \cdot 5 + 14 + 5}{23} = 3,87.$$

3.2 Логический синтез одноразрядного четверичного сумматора

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

Принцип работы ОЧС представлен с помощью таблицы истинности (таблица 3.2). Разряды обоих слагаемых закодированы: 0 – 01; 1 – 00; 2 – 10; 3 – 11. В таблице 3.2 выделено 16 безразличных наборов, т.к. со старших выходов ОЧУ не могут прийти коды «2» и «3».

Таблица 3.2 – Таблица истинности ОЧС

a_1	a_2	b_1	b_2	p	Π	S_1	S_2	Пример операции в четверичной с/с
1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0+0+0=00
0	0	0	0	1	0	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=10
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
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 функций Π и S_1 проведём с помощью карт Вейча. Минимизацию функции S_2 проведем с помощью алгоритма Рота. Для функции

Π заполненная карта приведена на рисунке 3.2.1. В рисунках 3.2.1 и 3.2.2 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

	a_2		b_1		b_1		a_2		
a_1	0	0	x	x	0	0	x	x	
	0	0	x	x	0	1	x	x	b_2
	0	0	x	x	0	0	x	x	
	0	1	x	x	1	1	x	x	
	p								

Рисунок 3.2.1 – Минимизация функции Π при помощи карты Вейча

Следовательно:

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

Запишем результат минимизации в логическом базисе И, НЕ:

$$\Pi = \overline{a_1 \bar{a}_2 b_1 b_2} \overline{\bar{a}_1 b_1 \bar{b}_2} \overline{\bar{a}_1 \bar{b}_2 \bar{p}}$$

Минимизация функции S_1

Для функции S_1 заполненная карта приведена на рисунке 3.2.2.

	a_2		b_1		b_1		a_2		
a_1	0	1	x	x	1	0	x	x	b_2
	0	1	x	x	1	0	x	x	
	1	0	x	x	0	1	x	x	
	1	0	x	x	0	1	x	x	
	p								

Рисунок 3.2.2 – Минимизация функции Π при помощи карты Вейча

Следовательно:

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

Запишем результат минимизации в логическом базисе И, НЕ:

$$S_1 = \overline{\overline{a_1} \overline{b_1} p} \overline{a_1 b_1 p} \overline{a_1 \overline{a_2} \overline{b_1}} \overline{a_1 b_1 \overline{p}}$$

Минимизация функции S_2 с помощью алгоритма Рота:

Определим множество единичных кубов

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

и множество безразличных кубов

$$N = \left\{ \begin{array}{l} 00010, 00011, 00100, 00101 \\ 01010, 01011, 01100, 01101 \\ 10010, 10011, 10100, 10101 \\ 11010, 11011, 11100, 11101 \end{array} \right\}.$$

Минимизацию безразличных кубов проведём с помощью карты Карно. Для безразличных кубов заполненная карта приведена на рисунке 3.2.3, где символом «х» отмечены наборы, на которых функция не определена.

$b_1 b_2 p$ $a_1 a_2$	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

Рисунок 3.2.3 – Минимизация безразличных кубов с помощью карты Карно

Множество безразличных наборов после минимизации:

$$N = \{xx01x, xx10x\}.$$

Сформируем множество $C_0 = L \cup N$:

$$C_0 = \left\{ \begin{array}{l} 00001, 00110, 00111, 01000, \\ 10111, 11000, 11001, 11110, \\ \quad \quad \quad xx01x, xx10x \end{array} \right\}.$$

Первым этапом алгоритма Рота является нахождение множества простых импликант.

В результате этой операции образуется множество кубов:

$$A_2 = \begin{pmatrix} 00x0x, x0x00, \\ 00xx1, 0xx01, \\ x0x11, 10xx0, \\ 10x1x, 1xx10 \end{pmatrix}.$$

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

C_I^* C_I	000 x1	00x 01	001 1x	00x 10	001 x0	x01 11	00x 11	001 x1	x10 00	010 x0	01x 00	10x 11	101 x1	110 0x	110 x0	11x 00	110 x1	11x 01	11x 10	111 x0
000 x1	-																			
00x 01	000 01	-																		
001 1x	00y 11	001 y1	-																	
00x 10	000 1y	00x yy	001 10	-																
001 x0	00y xy	001 0y	001 10	001 10	-															
x01 11	00y 11	001 y1	001 11	001 1y	001 1y	-														
00x 11	000 11	00x y1	001 11	00x 1y	001 1y	001 11	-													
001 x1	00y x1	001 01	001 11	001 1y	001 xy	001 11	001 11	-												
x10 00	0y0 0y	0y0 0y	0yy y0	0y0 y0	0yy 00	xyy yy	0y0 yy	0yy 0y	-											
010 x0	0y0 xy	0y0 0y	0yy 10	0y0 10	0yy x0	0yy 1y	0y0 1y	0yy xy	010 00	-										
01x 00	0y0 0y	0yx 0y	0y1 y0	0yx y0	0y1 00	0y1 yy	0yx yy	0y1 0y	010 00	010 00	-									
10x 11	y00 11	y0x y1	y01 11	y0x 1y	y01 1y	101 11	y0x 11	y01 11	1y0 yy	yy0 1y	yyx yy	-								
101 x1	y0y x1	y01 01	y01 11	y01 1y	y01 xy	101 11	y01 11	y01 x1	1yy 0y	yyy xy	yy1 0y	101 11	-							
110 0x	yy0 01	yy0 01	yyy yx	yy0 y0	yyy 00	1yy y1	yy0 y1	yyy 01	110 00	y10 00	y10 00	1y0 y1	1yy 01	-						
110 x0	yy0 xy	yy0 0y	yyy 10	yy0 10	yyy x0	1yy 1y	yy0 1y	yyy xy	110 00	y10 x0	y10 00	1y0 1y	1yy xy	110 00	-					
11x 00	yy0 0y	yyx 0y	yy1 y0	yyx y0	yy1 00	1y1 yy	yyx yy	yy1 0y	110 00	y10 00	y1x 00	1yx yy	1y1 0y	110 00	110 00	-				
110 x1	yy0 x1	yy0 01	yyy 11	yy0 1y	yyy xy	1yy 11	yy0 11	yyy x1	110 0y	y10 xy	y10 0y	1y0 11	1yy x1	110 01	110 xy	110 0y	-			
11x 01	yy0 01	yyx 01	yy1 y1	yyx yy	yy1 0y	1y1 y1	yyx y1	yy1 01	110 0y	y10 0y	y1x 0y	1yx y1	1y1 01	110 01	110 0y	11x 0y	110 01	-		
11x 10	yy0 1y	yyx yy	yy1 10	yyx 10	yy1 10	1y1 1y	yyx 1y	yy1 1y	110 y0	y10 10	y1x y0	1yx 1y	1y1 1y	110 y0	110 10	11x y0	110 1y	11x yy	-	
111 x0	yyy xy	yy1 0y	yy1 10	yy1 10	yy1 x0	1y1 1y	yy1 1y	yy1 xy	11y 00	y1y x0	y11 00	1y1 1y	1y1 xy	11y 00	11y x0	111 00	11y xy	111 0y	111 10	-
xx0 1x	000 11	000 y1	00y 1x	000 10	00y 10	x0y 11	000 11	00y 11	x10 y0	010 10	010 y0	100 11	10y 11	110 yx	110 10	110 y0	110 11	110 y1	110 10	11y 10
xx1 0x	00y 01	001 01	001 yx	001 y0	001 00	x01 y1	001 y1	001 01	x1y 00	01y 00	011 00	101 y1	101 01	11y 0x	11y 00	111 00	11y 01	111 01	111 y0	111 00

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

Также формируется множество $B_2 = C_1 - Z_1$.

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

$$C_2 = \left\{ \begin{array}{l} 00xx1, 00x1x, 001xx, x0x11 \\ x01x1, x10x0, x1x00, 110xx \\ 11x0x, 11xx0, xx01x, xx10x \end{array} \right\}.$$

В таблице 3.5 приведён следующий шаг поиска простых импликант – операция $C_2 * C_2$.

Таблица 3.5 – Поиск простых импликант $C_2 * C_2$

$C_2 * C_2$	00xx1	00x1x	001xx	x0x11	x01x1	x10x0	x1x00	110xx	11x0x	11xx0
00xx1	-									
00x1x	00x11	-								
001xx	001x1	0011x	-							
x0x11	00x11	00x11	00111	-						
x01x1	001x1	00111	001x1	x0111	-					
x10x0	0y0xy	0y010	0yux0	xy01y	xyuxy	-				
x1x00	0yx0y	0yxxy	0y100	xyxyy	xy10y	x1000	-			
110xx	yy0x1	yy01x	yyuxx	ly011	lyux1	110x0	11000	-		
11x0x	yux01	yuxxy	yy10x	lyxy1	ly101	11000	11x00	1100x	-	
11xx0	yuxxy	yux10	yy1x0	lyx1y	ly1xy	110x0	11x00	110x0	11x00	-
xx01x	00011	0001x	00y1x	x0011	x0y11	x1010	x10y0	1101x	110yx	11010
xx10x	00101	001yx	0010x	x01y1	x0101	x1y00	x1100	11y0x	1110x	11100

Новых кубов (третьей размерности) не образовалось.

$$\text{Получено множество } Z_2 = \left\{ \begin{array}{l} 00xx1, 00x1x, 001xx, x0x11 \\ x01x1, x10x0, x1x00, 110xx \\ 11x0x, 11xx0, xx01x, xx10x \end{array} \right\}.$$

Поскольку $|C_3| \leq 1$, поиск простых импликант заканчивается. Множество простых импликант:

$$Z = Z_0 \cup Z_1 \cup Z_2 = \left\{ \begin{array}{l} 00xx1, 00x1x, 001xx, x0x11 \\ x01x1, x10x0, x1x00, 110xx \\ 11x0x, 11xx0, xx01x, xx10x \end{array} \right\}.$$

Следующий этап – поиск L -экстремалей на множестве простых импликант (таблица 3.6). Для этого используется операция $\#$ (решётчатое вычитание).

Таблица 3.6 – Поиск L -экстремалей

$z\#(Z-z)$	00xx1	00x1x	001xx	x0x11	x01x1	x10x0	x1x00	110xx	11x0x	11xx0	xx01x	xx10x
00xx1	-	00x10	001x0	10x11	101x1	x10x0	x1x00	110xx	11x0x	11xx0	1x01x x101x xx010	1x10x x110x xx100
00x1x	00x01	-	00100	10x11	101x1	x10x0	x1x00	110xx	11x0x	11xx0	1x01x x101x 1x010 x1010	1x10x x110x xx100
001xx	00001	00010	-	10x11	101x1	x10x0	x1x00	110xx	11x0x	11xx0	1x01x x101x 1x010 x1010	1x10x x110x 1x100 x1100
x0x11	00001	00010	00100	-	10101	x10x0	x1x00	110xx	11x0x	11xx0	1101x 1x010 x101x 1x010 x1010	1x10x x110x 1x100 x1100
x01x1	00001	00010	00100	10011	-	x10x0	x1x00	110xx	11x0x	11xx0	1101x 1x010 x101x 1x010 x1010	1110x 1x100 x110x 1x100 x1100
x10x0	00001	00010	00100	10011	10101	-	x1100	110x1	1110x 11x01	111x0	11011 10010 x1011 10010	1110x 1x100 x110x 1x100 x1100
x1x00	00001	00010	00100	10011	10101	x1010	-	110x1	11101 11x01	11110	11011 10010 x1011 10010	11101 10100 x1101 10100
110xx	00001	00010	00100	10011	10101	01010	x1100	-	11101 11101	11110	10010 01011 10010	11101 10100 x1101 10100
11x0x	00001	00010	00100	10011	10101	01010	01100	11011	-	11110	10010 01011 10010	10100 01101 10100
11xx0	00001	00010	00100	10011	10101	01010	01100	11011	11101 11101	-	10010 01011 10010	10100 01101 10100
xx01x	00001	∅	00100	∅	10101	∅	01100	∅	11101 11101	11110	-	10100 01101 10100
xx10x	00001	∅	∅	∅	∅	∅	∅	∅	∅	11110	10010 01011 10010	-

В таблице 3.6 из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты $Z \# (Z - z)$.

Получили кубы, “подозрительные” на L -экстремальность. Проверяем в таблице 3.7.

Таблица 3.7 – Проверка на L -экстремальность

$z \# (Z - z) \cap L$	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

По результатам таблицы 3.7, L -экстремальными стали кубы E . Эти кубы обязательно должны войти в минимальное покрытие.

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

Далее необходимо проанализировать, какие из исходных единичных кубов не покрыты найденной L -экстремалью. Анализ осуществляется с помощью таблицы 3.8.

Таблица 3.8 – Поиск непокрытых наборов

$L \# E$	00001	00110	00111	01000	10111	11000	11001	11110
00xx1	Ø	00110	Ø	01000	10111	11000	11001	11110
11xx0	Ø	00110	Ø	01000	10111	Ø	11001	Ø

Из таблицы 3.8 видно, что L -экстремальными не покрыты четыре единичных куба (00110, 01000, 10111, 11001). Чтобы покрыть их, воспользуемся множеством простых импликант, не являющихся L -экстремальными. (Таблица 3.9)

Таблица 3.9 – Покрытие оставшихся кубов

$\check{Z} \cap L$	00110	01000	10111	11001
00x1x	00110	\emptyset	\emptyset	\emptyset
001xx	00110	\emptyset	\emptyset	\emptyset
x0x11	\emptyset	\emptyset	10111	\emptyset
x01x1	\emptyset	\emptyset	10111	\emptyset
x10x0	\emptyset	01000	\emptyset	\emptyset
x1x00	\emptyset	01000	\emptyset	\emptyset
110xx	\emptyset	\emptyset	\emptyset	11001
11x0x	\emptyset	\emptyset	\emptyset	11001
xx01x	\emptyset	\emptyset	\emptyset	\emptyset
xx10x	\emptyset	\emptyset	\emptyset	\emptyset

Из таблицы 3.9 видно, что кубы x01x1, 00x1x, 110xx, x1x00 максимально пересекаются с кубами из множества L^I и, следовательно, их покрывают (реализуют).

Эти четыре куба добавляются в минимальное покрытие.

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

Запишем результат минимизации в логическом базисе И-НЕ:

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

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

$$K_{\Pi} = \frac{4 \cdot 5 + 4 + 4}{13} = 2,15.$$

$$K_{S_1} = \frac{8 \cdot 5 + 8 + 5}{20} = 2,65.$$

$$K_{S_2} = \frac{8 \cdot 5 + 8 + 5}{19} = 2,79.$$

4 СИНТЕЗ СХЕМЫ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ

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.400201.425 Э2.3

Таблица 4.1 – таблица истинности ОЧС на базе мультиплексора

a_1	a_2	b_2	b_1	p	Π		S_1		S_2	
1	2	4	3	5	6		7		8	
0	0	0	0	0	0	b_1	0	p	0	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	1	0	0	b_1	1	\bar{p}	1	\bar{b}_1
0	0	1	1	1	0		0		1	
0	0	1	0	0	x		x		x	
0	0	1	0	1	x		x		x	
0	1	0	0	0	0	p	0	p	1	\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	1	0	0	p	1	\bar{p}	0	b_1
0	1	1	1	1	1		0		0	
0	1	1	0	0	x		x		x	
0	1	1	0	1	x		x		x	
1	0	0	0	0	0	p	1	\bar{p}	0	b_1
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	1	0	1	\bar{b}_1	0	p	0	p
1	0	1	1	1	1		1		1	
1	0	1	0	0	x		x		x	

Продолжение таблицы 4.1

1	0	1	0	1	x		x		x	
1	1	0	0	0	0	b_1	1	\bar{p}	1	\bar{b}_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	1	0	0	b_1	0	p	1	\bar{p}
1	1	1	1	1	0		1		0	
1	1	1	0	0	x		x		x	
1	1	1	0	1	x		x		x	

5 ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ (ПМ)

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11 и 10, заменяя их на триады $1\bar{0}1$ и $11\bar{0}$, соответственно.

Функциональная схема ПМ приведена на чертеже ГУИР.400201.409 Э2.4

Таблица 5.1 Таблица истинности ПМ.

Входная диада		Младший бит	Знак	Выходная 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	1	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	0

Проведём минимизацию P при помощи карты Карно:

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

Рисунок 5.1 – Минимизация функции P при помощи карты Карно

$$P = Q_n$$

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

Проведём минимизацию S_2 при помощи карты Карно:

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

Рисунок 5.2 – Минимизация функции S_2 при помощи карты Карно

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

6 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ

Процесс умножения можно разбить на несколько этапов:

- преобразование разряда множителя;
- формирование дополнительного кода множимого в ФДК, если это необходимо;
- умножение дополнительного кода множимого в ОЧУ;
- добавление полученного произведения в аккумулятор посредством ОЧС;
- сдвиг в аккумуляторе и регистре множителя.

Процессы умножения в ОЧУ происходят параллельно. Все ОЧУ отрабатывают одновременно. После этого результат передаётся в блок ОЧС для обработки. В этот момент начинает работу первый ОЧС. После того, как отработает первый ОЧС, он может незамедлительно передать значение переноса во второй ОЧС, который сможет приступить к сложению. Таким образом, все ОЧС отработают последовательно. Формула для расчёта временных затрат на умножение имеет следующий вид:

$$T_{\text{умн}} = n \cdot (T_{\text{сдвига}} + T_{\text{ПМ}} + T_{\text{ФДК}} + T_{\text{ОЧУ}} + (m + 1) \cdot T_{\text{ОЧС}}), \text{ где}$$

n – количество разрядов множителя;

m – количество разрядов множимого;

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

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

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

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

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

Подсчитать конкретное время выполнения умножения можно зная характеристики основных узлов конкретного сумматора-умножителя.

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. – 178 с.

[2] Луцик Ю.А., Лукьянова И.В. – Методические указания к курсовому проекту по курсу "Арифметические и логические основы вычислительной техники". – Мн.: БГУИР, 2004 г.

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

[4] Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. Мн.: Высшая школа, 1980.

ПРИЛОЖЕНИЕ А

(обязательное)

Сумматор-умножитель первого типа. Схема электрическая
структурная

ПРИЛОЖЕНИЕ Б

(обязательное)

Одноразрядный четверичный сумматор. Схема электрическая
функциональная

ПРИЛОЖЕНИЕ В
(обязательное)

Одноразрядный четверичный умножитель.
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