

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра вычислительных машин, систем и сетей
Дисциплина: Арифметические и логические основы
цифровых устройств

К ЗАЩИТЕ ДОПУСТИТЬ
_____ Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-
УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 116 ПЗ

Студент
Руководитель

Е.О. Лукьянов
Ю.А. Луцик

МИНСК 2023

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Арифметические и логические основы
цифровых устройств

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Б. В. Никульшин
«_____» _____ 20__ г.

ЗАДАНИЕ
по курсовой работе студента
Лукьянова Евгения Олеговича

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-десятичных чисел»
2. Срок сдачи студентом законченной работы: до 20 мая 2023г.
3. Исходные данные к работе:
 - 3.1. Исходные сомножители: $M_n = 52,33$; $M_t = 37,62$;
 - 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: $0_4 - 01$, $1_4 - 10$, $2_4 - 00$, $3_4 - 11$;
 - 3.5. Тип синтезируемого умножителя: 2;
 - 3.6. Логический базис для реализации ОЧС: И, НЕ; метод минимизации – алгоритм Рота.
 - 3.7. Логический базис для реализации ОЧУС: И, Исключающее ИЛИ, Генератор единицы; 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. Умножитель-сумматор 2 типа. Схема электрическая структурная.

- 5.2.** Однозарядный четверичный сумматор. Схема электрическая функциональная.
- 5.3.** Однозарядный четверичный умножитель. Схема электрическая функциональная. Однозарядный четверичный умножитель. Схема электрическая функциональная.
- 5.4.** Однозарядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.
- 5.5.** Преобразователь множителя. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов курсовой работы	Объём этапа, %	Срок выполнения этапа	Примечания
Разработка алгоритма умножения	10	14.02-20.02	
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: 14 февраля 2023г.

Руководитель

_____Ю.А. Луцик

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ

_____Е.О. Лукьянов

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	
1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ	
2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ	
3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ	
3.1. Логический синтез одноразрядного четверичного сумматора.....	
3.2. Логический синтез одноразрядного четверичного сумматора-умножителя	
3.3. Логический синтез преобразователя множителя	
4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ	
5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ	
ЗАКЛЮЧЕНИЕ	
СПИСОК ЛИТЕРАТУРЫ	
ПРИЛОЖЕНИЕ А Сумматор-умножитель первого типа. Схема электрическая структурная	
ПРИЛОЖЕНИЕ Б Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная.....	
ПРИЛОЖЕНИЕ В Одноразрядный четверичный сумматор. Схема электрическая функциональная.....	
ПРИЛОЖЕНИЕ Г Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная	
ПРИЛОЖЕНИЕ Д Преобразователь множителя. Схема электрическая функциональная	
ПРИЛОЖЕНИЕ Е Ведомость документов	

ВВЕДЕНИЕ

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

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

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

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

1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

1. Перевод сомножителей из десятичной системы счисления в четверичную.

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

$$\begin{array}{r} 0,33 \\ * \\ \hline 4 \\ \hline 1,32 \\ * \\ \hline 4 \\ \hline 1,28 \\ * \\ \hline 4 \\ \hline 1,12 \\ * \\ \hline 4 \\ \hline 0,48 \end{array}$$

$$M_{H4} = 310,1110.$$

В соответствии с заданной кодировкой множимого:

$$M_{H2/4} = 111001,10101001$$

$$\begin{array}{r|l} - & 37 \\ \hline & 36 \\ \hline & 1 \end{array} \begin{array}{l} 4 \\ - 9 \\ \hline 8 \end{array} \begin{array}{l} 4 \\ 2 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0,62 \\ * \\ \hline 4 \\ \hline 2,48 \\ * \\ \hline 4 \\ \hline 1,92 \\ * \\ \hline 4 \\ \hline 3,68 \\ * \\ \hline 4 \\ \hline 2,72 \end{array}$$

$$M_{T4} = 211,2132.$$

В соответствии с заданной кодировкой множителя:

$$M_{T2/4} = 100101,10011110$$

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

$$M_H = 0,11100110101001 \quad P_{MH} = 0.0111 + 03_{10}$$

$$M_T = 0,10010110011110 \quad P_{MT} = 0.0011 + 03_{10}$$

Порядок произведения:

$$P_{MH} = 0.0111 \quad 03_4$$

$$\begin{aligned} P_{M_T} &= \frac{0.0011 \ 03_4}{0.1000 \ 12_4} \\ P_{M_H \cdot M_T} &= \end{aligned}$$

Знак произведения определяется суммой по модулю два знаков сомножителей:

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

При умножении чисел в прямых кодах диада $11(3_4)$ заменяется на триаду $10\bar{1}$. Преобразованный множитель имеет вид $M_T^H = 21122\bar{1}2$. Перемножение мантисс по алгоритму «А» представлено в таблице 1.1

Таблица 1.1 — Перемножение мантисс

Четверичная с/с			Двоично-четверичная с/с			Комментарии
1			2			3
0.	00000000		0.	0101010101010101		\sum_0^q
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_1^q = M_H * 2$
0.	12202220		0.	1000000100000001		\sum_1^q
0.	01220222	0	0.	0110000001000000	01	$\sum_1^q * 4^{-1}$
3.	<u>30232230</u>		1.	<u>1101001100001101</u>		$\Pi_2^q = -1 * M_H$
3.	32113112	0	1.	1100101011101000	01	\sum_2^q
3.	33211311	20	1.	1111001010111010	0001	$\sum_2^q * 4^{-1}$
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_3^q = M_H * 2$
0.	12020131	20	0.	1000010001101110	0001	\sum_3^q
0.	01202013	120	0.	0110000100011011	100001	$\sum_3^q * 4^{-1}$
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_4^q = M_H * 2$
0.	20010233	120	0.	0001011001001111	100001	\sum_4^q
0.	02001023	3120	0.	0100010110010011	11100001	$\sum_4^q * 4^{-1}$
0.	<u>03101110</u>		0.	<u>0111100110101001</u>		$\Pi_5^q = M_H * 1$
0.	11102133	3120	0.	1010100100101111	11100001	\sum_5^q
0.	01110213	33120	0.	0110101001001011	1111100001	$\sum_5^q * 4^{-1}$
0.	<u>03101110</u>		0.	<u>0111100110101001</u>		$\Pi_6^q = M_H * 1$
0.	10211323	33120	0.	1001001010110011	1111100001	\sum_6^q
0.	01021132	333120	0.	0110010010101100	111111100001	$\sum_6^q * 4^{-1}$
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_7^q = M_H * 2$
0.	13230012	333120	0.	1011001101011000	111111100001	\sum_7^q
0.	01323001	2333120	0.	0110110011010110	001111111100001	$\sum_7^q * 4^{-1}$

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение ($M_{H4} * M_{T4} = 0.013230012$

333120, $P_{M_H} * P_{M_T} = 7$) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

$$\begin{aligned}(M_H * M_T)_4 &= 132300,12333120 & P_{M_H * M_T} &= 0; \\ (M_H * M_T)_{10} &= 1968,4368.\end{aligned}$$

Результат прямого перемножения операндов дает следующее:

$$M_{H10} * M_{T10} = 52,33 * 37,62 = 1968,6546.$$

Абсолютная погрешность:

$$\Delta = 1968,6546 - 1968,4368 = 0,2178.$$

Относительная погрешность:

$$\delta = \frac{\Delta}{M_{H10} * M_{T10}} = \frac{0,2178}{1968,6546} = 0,00011 \quad (\delta = 0,011\%)$$

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

Если устройство работает как сумматор, то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода F_2 поступает «1». Необходимо обеспечить выполнение алгоритма сложения чисел, представленных в форме с плавающей запятой, базируясь на схеме умножителя, реализующего заданный алгоритм умножения.

Первое слагаемое переписывается в регистр результата под действием управляющих сигналов, поступающих на входы h всех ОЧУС (рисунок 2.1).

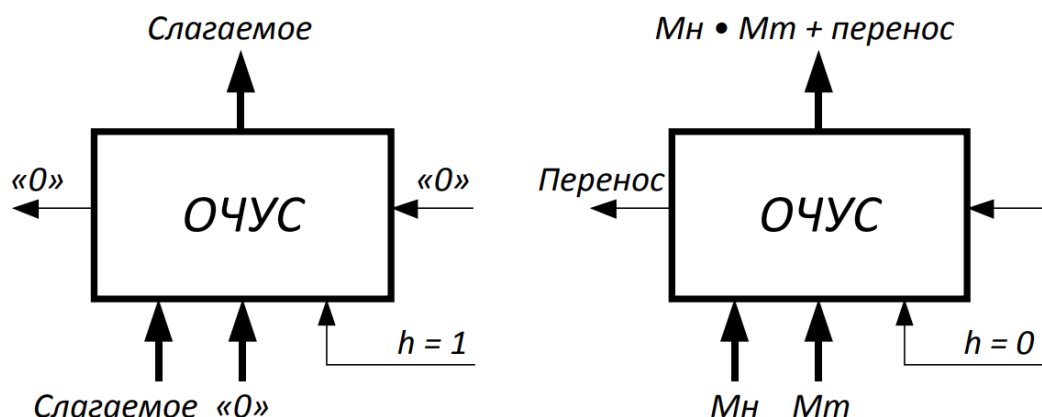


Рисунок 2.1 – Режимы работы ОЧУС

Если на вход h поступает «0», то ОЧУС перемножает разряды M_n и M_t и добавляет к полученному результату перенос из предыдущего ОЧУС.

В ОЧУС первое слагаемое складывается с нулём, записанным в регистре результата, и переписывается без изменений в регистр результата.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУС попадает на входы ОЧУС и складывается с первым слагаемым, хранящимся в регистре результата.

Сумма хранится в регистре результата. Разрядность регистра результата должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения при суммировании переноса.

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

Диада множителя поступает на входы преобразователя множителя. Единица переноса в следующую диаду, если она возникает, должна быть добавлена к следующей диаде множителя (выход 1 ПМ) в следующем такте, т. е. должна храниться на триггере до следующего такта.

В регистре множителя после каждого такта умножения содержимое сдвигается на два двоичных разряда, и в конце умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения при умножении по алгоритму «А».

Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание ($\overline{01}$). 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 отражён в таблице 2.1.

На выходах 3 и 4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУС вместе с диадами множимого. На трёх выходах ОЧУС формируется результат умножения диад $M_n \cdot M_t$ плюс перенос из предыдущего ОЧУС. Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому перенос, формируемый ОЧУС, может быть только двоичным («0» или «1»):

$$\begin{array}{ccccc} 3 & \cdot & 2 & = & 12 \quad (+1 \text{ в случае переноса из} \\ & & & & \text{предыдущего ОЧУС)} \\ \text{max} & & \text{max} & & \text{max} \\ M_n & & M_t & & \text{Перенос} \end{array}$$

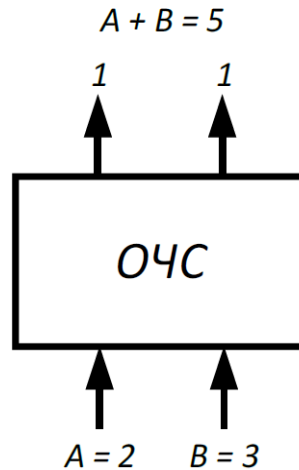
Так как на входы ОЧУС из регистра M_t не могут поступить коды «3», в таблице истинности работы ОЧУС будут содержаться 16 безразличных входных наборов.

Частичные произведения, получаемые на выходах ОЧУС, складываются с накапливаемой частичной суммой из регистра результата с помощью цепочки ОЧС (на первом такте выполняется сложение с нулём).

Частичные суммы хранятся в регистре результата и регистре множителя, т. к. алгоритм умножения «А» предполагает возможность синхронного сдвига этих регистров. Количество тактов умножения определяется разрядностью 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.

Разрядность аккумулятора должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения переноса при суммировании.

Если 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 при умножении по алгоритму «А» (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 F_1 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 Мн
1	1	Меняется знак слагаемого

На выходах 3 и 4 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»:

$$\begin{array}{c} 3 \\ \text{max} \\ \text{Мн} \end{array} \cdot \begin{array}{c} 2 \\ \text{max} \\ \text{Мт} \end{array} = \begin{array}{c} 1 \ 2 \end{array}$$

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.

Частичные суммы хранятся в аккумуляторе и регистре множителя, т.к. алгоритм умножения «А» предполагает возможность синхронного сдвига этих устройств. Количество тактов умножения определяется разрядностью Мт.

3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ

3.1 Логический синтез одноразрядного четверичного сумматора-умножителя

ОЧУС – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда из регистра Мн, 2 разряда из регистра Мт и управляющий вход h) и 3 двоичных выхода.

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

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

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

Управляющий вход h определяет тип операции:

«0» – вывод результата умножения закодированных цифр с добавлением переноса из предыдущего ОЧУС, перенос в следующий ОЧУС.

«1» – вывод без изменения значения разрядов, поступивших из регистра множимого, перенос *из* и *в* ОЧУС равны нулю.

В таблице 3.1.1 выделены безразличные наборы, т.к. на входы ОЧУС из разрядов множителя не может поступить код «11».

Таблица 3.1.1 — Таблица истинности ОЧУС

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

Продолжение таблицы 3.1.1

0	1	0	0	1	0	0	1	0	$1 * 1 + 0 = 01$
0	1	0	0	1	1	0	1	0	Выход – код «01»
0	1	0	1	0	0	0	0	0	$1 * 2 + 0 = 02$
0	1	0	1	0	1	0	1	0	Выход – код «01»
0	1	0	1	1	0	x	x	x	$1 * 3 + 0 = 03$
0	1	0	1	1	1	x	x	x	Выход – код «01»
0	1	1	0	0	0	0	0	1	$3 * 0 + 0 = 00$
0	1	1	0	0	1	0	1	1	Выход – код «03»
0	1	1	0	1	0	0	1	1	$3 * 1 + 0 = 03$
0	1	1	0	1	1	0	1	1	Выход – код «03»
0	1	1	1	0	0	1	0	0	$3 * 2 + 0 = 12$
0	1	1	1	0	1	0	1	1	Выход – код «03»
0	1	1	1	1	0	x	x	x	$3 * 3 + 0 = 21$
0	1	1	1	1	1	x	x	x	Выход – код «03»
1	0	0	0	0	0	x	x	x	$2 * 0 + 1 = 02$
1	0	0	0	0	1	x	x	x	Выход – код «02»
1	0	0	0	1	0	x	x	x	$2 * 1 + 1 = 03$
1	0	0	0	1	1	x	x	x	Выход – код «02»
1	0	0	1	0	0	1	1	0	$2 * 2 + 1 = 11$
1	0	0	1	0	1	x	x	x	Выход – код «02»
1	0	0	1	1	0	x	x	x	$2 * 3 + 1 = 13$
1	0	0	1	1	1	x	x	x	Выход – код «02»
1	0	1	0	0	0	x	x	x	$0 * 0 + 1 = 01$
1	0	1	0	0	1	x	x	x	Выход – код «00»
1	0	1	0	1	0	x	x	x	$0 * 1 + 1 = 01$
1	0	1	0	1	1	x	x	x	Выход – код «00»
1	0	1	1	0	0	0	1	0	$0 * 2 + 1 = 01$
1	0	1	1	0	1	x	x	x	Выход – код «00»
1	0	1	1	1	0	x	x	x	$0 * 3 + 1 = 01$
1	0	1	1	1	1	x	x	x	Выход – код «00»
1	1	0	0	0	0	x	x	x	$1 * 0 + 1 = 01$
1	1	0	0	0	1	x	x	x	Выход – код «01»
1	1	0	0	1	0	x	x	x	$1 * 1 + 1 = 02$
1	1	0	0	1	1	x	x	x	Выход – код «01»
1	1	0	1	0	0	0	1	1	$1 * 2 + 1 = 03$
1	1	0	1	0	1	x	x	x	Выход – код «01»
1	1	0	1	1	0	x	x	x	$1 * 3 + 1 = 10$
1	1	0	1	1	1	x	x	x	Выход – код «01»
1	1	1	0	0	0	x	x	x	$3 * 0 + 1 = 01$
1	1	1	0	0	1	x	x	x	Выход – код «03»
1	1	1	0	1	0	x	x	x	$3 * 1 + 1 = 10$

Продолжение таблицы 3.1.1

1	1	1	0	1	1	x	x	x	Выход – код «03»
1	1	1	1	0	0	1	1	1	$3 * 2 + 1 = 13$
1	1	1	1	0	1	x	x	x	Выход – код «03»
1	1	1	1	1	0	x	x	x	$3 * 3 + 1 = 22$
1	1	1	1	1	1	x	x	x	Выход – код «03»

Минимизация функции Р:

Минимизацию функции Р проведем с помощью карт Вейча. Для функции Р 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).

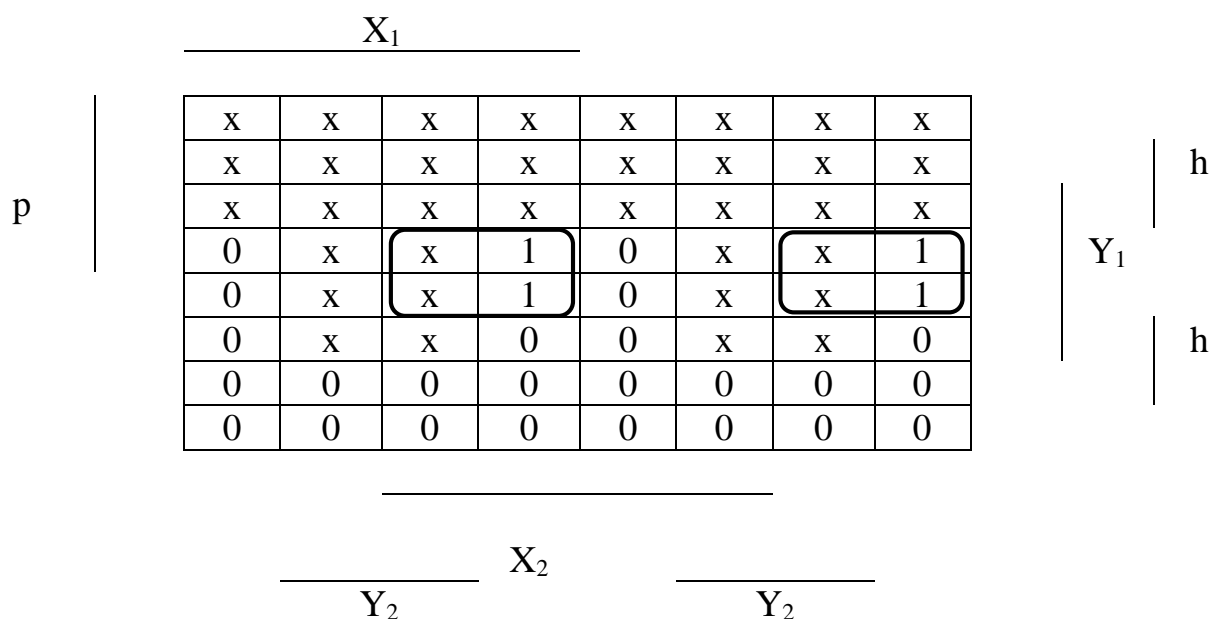


Рисунок 3.1.1 — Минимизация функции Р картой Вейча

Следовательно:

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

Запишем результат в базисе И-Константная единица-Сумма по модулю:

$$P = y_1 \cdot (h \oplus 1) \cdot (x_1 \oplus x_2)$$

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

$$K = \frac{6 \cdot 4 + 4 + 4}{13} = 2,62$$

Минимизация функции Q_1 :

X_1								
p	<div><div><div>x</div><div>x</div><div>x</div><div>x</div></div><div><div>x</div><div>x</div><div>x</div><div>x</div></div><div><div>x</div><div>x</div><div>x</div><div>x</div></div><div><div>1</div><div>x</div><div>x</div><div>1</div></div></div>	<div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div></div>	<div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div></div>	<div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div></div>	<div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div></div>	<div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div></div>	<div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div></div>	<div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div><div><div>x</div><div>x</div><div>x</div><div>1</div></div></div>
	<div><div><div>0</div><div>x</div><div>x</div><div>0</div></div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>1</div><div>x</div><div>x</div><div>0</div></div><div><div>1</div><div>1</div><div>1</div><div>1</div></div></div>	<div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div>	<div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div>	<div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div>	<div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div>	<div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div>	<div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div>	<div><div><div>0</div><div>0</div><div>x</div><div>x</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div><div><div>0</div><div>0</div><div>0</div><div>0</div></div></div>
	X_2							
	Y_2				Y_2			

Рисунок 3.1.2 — Минимизация функции Q_1 картой Вейча

Следовательно:

$$Q_1 = p + x_1 y_2 + x_1 h$$

Запишем результат в базисе И-Константная единица-Сумма по модулю: $Q_1 = (((p \oplus 1) \cdot (x_1 y_2 \oplus 1) \cdot (x_1 h \oplus 1)) \oplus 1)$

Эффективность минимизации:

$$K = \frac{12 \cdot 6 + 12 + 6}{8} = 11,25$$

Минимизация функции Q_2 :

X_1							
p	x	x	x	x	x	x	x
	x	x	x	x	x	x	x
	x	x	x	x	x	x	x
	1	x	x	1	0	x	0
	0	x	x	0	1	x	1
	0	x	x	1	1	x	0
	0	0	1	1	1	1	0
	1	0	1	1	1	1	0
X_2							
Y_2				Y_2			

Рисунок 3.1.3 — Минимизация функции Q_2 картой Вейча

Следовательно:

$$Q_2 = \bar{y}_1 \bar{y}_2 \bar{h} + x_2 h + x_2 y_1 + p x_1 + \bar{p} \bar{x}_1 y_1 \bar{h}$$

Запишем результат в базисе И-Константная единица-Сумма по модулю: $Q_2 = (((((y_1 \oplus 1) \cdot (y_2 \oplus 1) \cdot (h \oplus 1)) \oplus 1) \cdot (x_2 h \oplus 1) \cdot (x_2 y_1 \oplus 1) \cdot (p x_1 \oplus 1) \cdot (((p \oplus 1) \cdot (x_1 \oplus 1) \cdot y_1 \cdot (h \oplus 1)) \oplus 1)) \oplus 1)$

Эффективность минимизации:

$$K = \frac{16 \cdot 6 + 16 + 6}{23} = 5,13$$

Функциональная схема ОЧУС приведена в приложении В.

3.2 Логический синтез одноразрядного четверичного сумматора

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

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

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

Таблица 3.2.1 — 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	1	0	1	$2 + 2 + 0 = 10$
0	0	0	0	1	1	1	0	$2 + 2 + 1 = 11$
0	0	0	1	0	0	0	0	$2 + 0 + 0 = 02$
0	0	0	1	1	0	1	1	$2 + 0 + 1 = 03$
0	0	1	0	0	0	1	1	$2 + 1 + 0 = 03$
0	0	1	0	1	1	0	1	$2 + 1 + 1 = 10$
0	0	1	1	0	1	1	1	$2 + 3 + 0 = 11$
0	0	1	1	1	1	0	0	$2 + 3 + 1 = 12$
0	1	0	0	0	0	0	0	$0 + 2 + 0 = 02$
0	1	0	0	1	0	1	1	$0 + 2 + 1 = 03$
0	1	0	1	0	0	0	1	$0 + 0 + 0 = 00$
0	1	0	1	1	0	1	0	$0 + 0 + 1 = 01$
0	1	1	0	0	0	1	0	$0 + 1 + 0 = 01$
0	1	1	0	1	0	0	0	$0 + 1 + 1 = 02$
0	1	1	1	0	0	1	1	$0 + 3 + 0 = 03$
0	1	1	1	1	1	0	1	$0 + 3 + 1 = 10$
1	0	0	0	0	0	1	1	$1 + 2 + 0 = 03$
1	0	0	0	1	1	0	1	$1 + 2 + 1 = 10$
1	0	0	1	0	0	1	0	$1 + 0 + 0 = 01$

Продолжение таблицы 3.2.1

1	0	0	1	1	0	0	0	$1 + 0 + 1 = 02$
1	0	1	0	0	0	0	0	$1 + 1 + 0 = 02$
1	0	1	0	1	0	1	1	$1 + 1 + 1 = 03$
1	0	1	1	0	1	0	1	$1 + 3 + 0 = 10$
1	0	1	1	1	1	1	0	$1 + 3 + 1 = 11$
1	1	0	0	0	1	1	0	$3 + 2 + 0 = 11$
1	1	0	0	1	1	0	0	$3 + 2 + 1 = 12$
1	1	0	1	0	0	1	1	$3 + 0 + 0 = 03$
1	1	0	1	1	1	0	1	$3 + 0 + 1 = 10$
1	1	1	0	0	1	0	1	$3 + 1 + 0 = 10$
1	1	1	0	1	1	1	0	$3 + 1 + 1 = 11$
1	1	1	1	0	1	0	0	$3 + 3 + 0 = 12$
1	1	1	1	1	1	1	1	$3 + 3 + 1 = 13$

Минимизация функции П:

Минимизацию функции П проведем с помощью карт Карно. Для функции П заполненная карта приведена на рисунке 3.2.2.

$\begin{matrix} b_1 b_2 p \\ a_1 a_2 \end{matrix}$	000	001	011	010	110	111	101	100
00	1	1	0	0	1	1	1	0
01	0	0	0	0	0	1	0	0
11	1	1	1	0	1	1	1	1
10	0	1	0	0	1	1	0	0

Рисунок 3.2.2 — Минимизация функции П картой Карно

Следовательно:

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

Запишем результат в базисе И-НЕ:

$$\Pi = \overline{(\bar{a}_1 \bar{a}_2 \bar{b}_1 \bar{b}_2)} \cdot \overline{(a_1 a_2 \bar{b}_2)} \cdot \overline{(a_1 a_2 \bar{b}_1 p)} \cdot \overline{(a_1 \bar{b}_1 \bar{b}_2 \bar{p})} \cdot \overline{(\bar{a}_2 b_1 b_2)} \cdot \overline{(a_1 b_1 b_2)} \cdot \overline{(b_1 b_2 p)} \cdot \overline{(\bar{a}_1 \bar{a}_2 b_1 p)}$$

Эффективность минимизации:

$$K = \frac{16 \cdot 5 + 16 + 5}{41} = 2,46$$

Минимизация функции S_1

Минимизацию функции S_1 проведем с помощью карт Карно. Для функции S_1 заполненная карта приведена на рисунке 3.2.3.

$\begin{matrix} b_1b_2p \\ a_1a_2 \end{matrix}$	000	001	011	010	110	111	101	100
00	0	1	1	0	1	0	0	1
01	0	1	1	0	1	0	0	1
11	1	0	0	1	0	1	1	0
10	1	0	0	1	0	1	1	0

Рисунок 3.2.3 — Минимизация функции S_1 картой Карно

Следовательно: $S_1 = \bar{a}_1\bar{b}_1p + a_1b_1p + a_1\bar{b}_1\bar{p} + \bar{a}_1b_1\bar{p}$

Запишем результат в базисе И-НЕ:

$$S_1 = \overline{(\bar{a}_1\bar{b}_1p)} \cdot \overline{(a_1b_1p)} \cdot \overline{(a_1\bar{b}_1\bar{p})} \cdot \overline{(\bar{a}_1b_1\bar{p})}$$

Эффективность минимизации:

$$K = \frac{16 \cdot 5 + 16 + 5}{19} = 5,32$$

Минимизация функции S_2

Определим множество единичных кубов:

$$L = \left\{ \begin{array}{l} 00000, 00011, 00100, \\ 00101, 00110, 01001, \\ 01010, 01110, 01111, \\ 10000, 10001, 10101, \\ 10110, 11010, 11011, \\ 11100, 11111 \end{array} \right\}$$

Множество безразличных кубов пустое.

Сформируем множество $C_0 = L \cup N$:

$$C_0 = \{00000, 00011, 00100, 00101, 00110, 01001, 01010, 01110, 01111, 10000, 10001, 10101, 10110, 11010, 11011, 11100, 11111\}$$

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 ($C_0 * C_0$) приведён в таблице 3.2.4.

Таблица 3.2.4 – Поиск простых импликант ($C_0 * C_0$)

$C_0 * C_0$	00000	00011	00100	00101	00110	01001	01010	01110	01111	10000	10001	10101	10110	11010	11011
00000	-														
00011		-													
00100	00x00		-												
00101			0010x	-											
00110			001x0		-										
01001						-									
01010							-								
01110					0x110		01x10	-							
01111								0111x	-						
10000	x0000									-					
10001										1000x	-				
10101				x0101							10x01	-			
10110					x0110								-		
11010							x1010							-	
11011														1101x	-
11100															
11111									x1111						11x11

Продолжение таблицы 3.2.4

$C_0 * C_0$	11100	11111
00000		
00011		
00100		
00101		
00110		
01001		
01010		
01110		
01111		
10000		
10001		
10101		
10110		
11010		
11011		
11100	-	
11111		-

В результате этой операции сформируется новое множество кубов:

$$C_1 = \{00x00, x0000, 0010x, 001x0, x0101, 0x110, x0110, 01x10, x1010, 0111x, x1111, 1000x, 10x01, 1101x, 11x11\}$$

$$Z_0 = \{00011, 01001, 11100\}$$

В таблице 3.2.5 приведён следующий шаг поиска простых импликант с помощью операции $C_1 * C_1$.

Таблица 3.2.5 – Поиск простых импликант $C_1 * C_1$

$C_1 * C_1$	0001x	x0000	0010x	001x0	x0101	0x110	x0110	01x10	x1010	0111x	x1111	1000x	10x01	1101x	11x11
0001x	-														
x0000		-													
0010x			-												
001x0				-											
x0101					-										
0x110						-									
x0110							-								
01x10								-							
x1010									-						
0111x										-					
x1111											-				
1000x												-			
10x01													-		
1101x														-	
11x11															-

Новых кубов (второй размерности) не образовалось.

Конечное множество простых импликант $Z = \{00011; 01001; 11100; x0110; 1101x; 11x11; 0010x; 001x0; x0000; 10x01; 01x10; 0111x; 00x00; x0101; 0x110; x1010; x1111; 1000x\}$

На этом заканчивается этап поиска простых импликант.

Следующий этап – поиск L-экстремалей на множестве простых импликант (таблица 3.2.6). Для этого используется операция # (вычитание).

Таблица 3.2.6 – Поиск L-экстремалей

z#(Z-z)	00011	01001	11100	x0110	1101x	11x11	0010x	001x0	x0000	10x01	01x10	0111x	00x00	x0101
00011	-	zyzyz 01001	zyzyz 11100	1zyzy x0110	yyzz0 1101x	yy1zz 11x11	zzyy0 0010x	zzy0y 001x0	1zzyy x0000	yz1yz 10x01	zylzy 01x10	zyyz0 0111x	zz1yy 00x00	1zyyz x0101
01001	zyzyz 00011	-	zyzyz 11100	1yyyy x0110	yyzz0 1101x	yz1yz 11x11	zyyz0 0010x	zyy1y 001x0	1yzzz x0000	yy1zz 10x01	zz1yy 01x10	zzyy0 0111x	zylzy 00x00	1yyzz x0101
11100	yyyyy 00011	zyzyz 01001	-	0zyyz x0110	zzyy1 1101x	zz0yy 11x11	yyzz1 0010x	yyz1z 001x0	0yyzz x0000	zy0zy 10x01	yz0yz 01x10	yyzz1 0111x	yy0zz 00x00	0yzzz x0101
x0110	zzyzy 00011	zyyyy 01001	zyzyz 11100	-	zyyz1 1101x	zy0zy 11x11	zzzy1 0010x	zzz0z 00100	zzyyz x0000	zz0yy 10x01	zy0zz 01x10	zyzz1 0111x	zz0yz 00x00	zzzyy x0101
1101x	yyzzz 00011	yzzyz 01001	zzyyz 11100	0yyzz x0110	-	zz1zz 11111	yyyyz 0010x	yyyyz 00100	0yzyz x0000	zylzy 10x01	yz1zz 01x10	yzyzz 0111x	yy1yz 00x00	0yyyz x0101
11x11	yyzzz 00011	yzzyz 01001	zzzyy 11100	0yzzz x0110	zzzz0 11010	-	yyzy0 0010x	yyzyy 00100	0yzyy x0000	zyzyz 10x01	yzzzy 01x10	yyzz0 0111x	yyzyy 00x00	0yzyz x0101
0010x	zzyyz 00011	zyyzz 01001	yyzzz 11100	1zzyz x0110	yyyyz 11010	yyzyz 11111	-	zzzzz Ø	1zyzz x0000	yz0zz 10x01	zy0yz 01x10	zyzyz 0111x	zz0zz 00000	1zzzz 10101
001x0	zzyzy 00011	zyyzy 01001	yyzzz 11100	1zzzz 10110	yyyzz 11010	yyzzy 11111	zzzz1 00101	-	1zyzz x0000	yz0zy 10x01	zy0zz 01x10	zyzz1 0111x	zzyzz 00000	yzzzy 10101
x0000	zzzyy 00011	zyzzy 01001	zyyzz 11100	zzyyz 10110	zyzyz 11010	zyyyy 11111	zzyzy 00101		-	zz1zy 10x01	zylzy 01x10	zyyy1 0111x	zzzzz Ø	zzyzy 10101
10x01	yzzyz 00011	yyzzz 01001	zyzzy 11100	zzzyy 10110	zyzyy 11010	zyzyz 11111	yzzzz 00101		0zzzy x0000	-	yyzyy 01x10	yyzy0 0111x		zzzzz Ø
01x10	zyzzy 00011	zzzyy 01001	yzzyz 11100	yyzzz 10110	yzzzz 11010	yzzzy 11111	zyzyy 00101		1yzyz x0000	yyzyy 10x01	-	zzzz1 01111		
0111x	zyyzz 00011	zzyyz 01001	yzzyz 11100	yyzzz 10110	yzyzz 11010	yzzzz 11111	zyzyz 00101		1yyyz x0000	yy0yz 10x01	zz0zz 01010	-		

Продолжение таблицы 3.2.6

00x00	zzzyy 00011	zyzzz 01001	yyzzz 11100	yzzyz 10110	yyzyz 11010	yyzyy 11111	zzzzz 00101		1zzzz 10000	yzzzy 10x01	zyzyz 01010	zyzyy 01111	-	
x0101	zzyyz 00011	zyyzz 01001	zyzzz 11100	zzzyy 10110	zyyyy 11010	zyzyz 11111	zzzzz Ø		zzyzy 10000	zz0zz 10001	zyyyy 01010	zyzyz 01111		-
0x110	zzyzy 00011	zzyyy 01001	yzzyz 11100	yzzzz 10110	yzyzz 11010	yzzzy 11111			yzyyz 10000	yzyyy 10001	zzyzz 01010	zzzzz 01111		
x1010	zyzzy 00011	zzzyy 01001	zzyyz 11100	zyyzz 10110	zzzzz Ø	zzyzy 11111			zyzyz 10000	zyzyy 10001	zzzzz Ø	zzyzy 01111		
x1111	zyyzz 00011	zzyyz 01001	zzzyy 11100	zyzzz 10110		zzzzz Ø			zyyyy 10000	zyyyz 10001		zzzzz Ø		
1000x	yzzyz 00011	yyzzz 01001	zyyzz 11100	zzyyz 10110					zzzzz Ø	zzzzz Ø				
Остаток	00011	01001	11100	10110	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø

Продолжение таблицы 3.2.6

$z\#(Z-z)$	0x110	x1010	x1111	1000x
00011	z1yzy 0x110	1yzzzy x1010	1yyzzz x1111	yzzy0 1000x
01001	z0yyy 0x110	1zzyy x1010	1zyyz x1111	yyzz0 1000x
11100	y0zyz 0x110	0zyyz x1010	0zzyy x1111	zyyz1 1000x
x0110	z1zzz 01110	zyyzz x1010	zyzzzy x1111	zzyy1 1000x
1101x	yzyzz 01110	0zzzz 01010	0zyzz x1111	zyzyz 1000x
11x11	yzzzy 01110	yzzzy 01010	0zzzz 01111	zyzy0 1000x
0010x	zyzyz 01110	zyyyz 01010	zyzyz 01111	yzyzz 1000x
001x0	zyzzz 01110	zyyzz 01010	zyzzzy 01111	yzyz1 1000x
x0000	zyyyz 01110	zyzyz 01010	zyyyy 01111	zzzz1 10001
10x01	yyzyy 01110	yyzyy 01010	yyzyz 01111	zzzzz Ø
01x10	zzzzz Ø	zzzzz Ø	zzzzzy 01111	
0111x			zzzzz Ø	

Продолжение таблицы 3.2.6

00x00				
x0101				
0x110	-			
x1010		-		
x1111			-	
1000x				-
Остаток	Ø	Ø	Ø	Ø

В таблице 3.2.6 из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты $Z \# (Z - z)$.

Множество L-экстремалей = $E = \{00011, 01001, 11100, x0110\}$

Следующий этап – поиск непокрытых кубов из множества L (таблица 3.2.7). Для этого используется операция # (вычитание).

Таблица 3.2.7 – Поиск непокрытых кубов из множества L

L # E	00000	00011	00100	00101	00110	01001	01010	01110	01111	10000	10001	10101	10110	11010
00011	00000	∅	00100	00101	00110	01001	01010	01110	01111	10000	10001	10101	10110	11010
01001	00000	∅	00100	00101	00110	∅	01010	01110	01111	10000	10001	10101	10110	11010
11100	00000	∅	00100	00101	00110	∅	01010	01110	01111	10000	10001	10101	10110	11010
x0110	00000	∅	00100	00101	∅	∅	01010	01110	01111	10000	10001	10101	∅	11010
Остаток	00000	∅	00100	00101	∅	∅	01010	01110	01111	10000	10001	10101	∅	11010

Продолжение таблицы 3.2.7

L # E	11011	11100	11111
00011	11011	11100	11111
01001	11011	11100	11111
11100	11011	Ø	11111
x0110	11011	Ø	11111
Остаток	11011	Ø	11111

Из таблицы 3.2.7 следует, что множество кубов, непокрытых L-экстремалиями, $L' = L \# E = \{00000; 00100; 00101; 01010; 01110; 01111; 10000; 10001; 10101; 11010; 11011; 11111\}$

Следующий этап – поиск минимального числа кубов из множества $Z' = Z \setminus E$, которыми будут покрыты оставшиеся кубы из множества L' (таблица 3.2.8). Для этого используется операция \cap (пересечение).

$Z' = Z - E = \{1101x; 11x11; 0010x; 001x0; x0000; 10x01; 01x10; 0111x; 00x00; x0101; 0x110; x1010; x1111; 1000x\}$

Из оставшихся кубов множества Z' необходимо найти кубы с минимальной ценой и одновременно с этим кубы, которые максимально покрывают кубы множества L' .

Таблица 3.2.8 – Поиск не перекрытых кубов из множества L

$L' \cap Z'$	00000	00100	00101	01010	01110	01111	10000	10001	10101	11010	11011	11111
1101x	yy0y0 Ø	yyyy0 Ø	yyyy1 Ø	y1010 Ø	y1y10 Ø	y1y11 Ø	1y0y0 Ø	1y0y1 Ø	1yyy1 Ø	11010	11011	11y11 Ø
11x11	yy0yy Ø	yy1yy Ø	yy1y1 Ø	y101y Ø	y111y Ø	y1111 Ø	1y0yy Ø	1y0y1 Ø	1y1y1 Ø	1101y Ø	11011	11111
0010x	00y00 Ø	00100	00101	0yyy0 Ø	0y1y0 Ø	0y1y1 Ø	y0y00 Ø	y0y01 Ø	y0101 Ø	yyyy0 Ø	yyyy1 Ø	yy1y1 Ø
001x0	00y00 Ø	00100	0010y Ø	0yy10 Ø	0y110 Ø	0y11y Ø	y0y00 Ø	y0y0y Ø	y010y Ø	yyy10 Ø	yyy1y Ø	yy11y Ø
x0000	00000	00y00 Ø	00y0y Ø	0y0y0 Ø	0yyy0 Ø	0yyyy Ø	10000	1000y Ø	10y0y Ø	1y0y0 Ø	1y0yy Ø	1yyyy Ø
10x01	y000y Ø	y010y Ø	y0101 Ø	yy0yy Ø	yy1yy Ø	yy1y1 Ø	1000y Ø	10001	10101	1y0yy Ø	1y0y1 Ø	1y1y1 Ø
01x10	0y0y0 Ø	0y1y0 Ø	0y1yy Ø	01010	01110	0111y Ø	yy0y0 Ø	yy0yy Ø	yy1yy Ø	y1010 Ø	y101y Ø	y111y Ø
0111x	0yyy0 Ø	0y1y0 Ø	0y1y1 Ø	01y10 Ø	01110	01111	yyyy0 Ø	yyyy1 Ø	yy1y1 Ø	y1y10 Ø	y1y11 Ø	y1111 Ø
00x00	00000	00100	0010y Ø	0y0y0 Ø	0y1y0 Ø	0y1yy Ø	y0000 Ø	y000y Ø	y010y Ø	yy0y0 Ø	yy0yy Ø	yy1yy Ø
x0101	00y0y Ø	0010y Ø	00101	0yyyy Ø	0y1yy Ø	0y1y1 Ø	10y0y Ø	10y01 Ø	10101	1yyyy Ø	1yyy1 Ø	1y1y1 Ø
0x110	00yy0 Ø	001y0 Ø	001yy Ø	01y10 Ø	01110	0111y Ø	y0yy0 Ø	y0yyy Ø	y01yy Ø	y1y10 Ø	y1y1y Ø	y111y Ø

Продолжение таблицы 3.2.8

x1010	0y0y0 Ø	0yuy0 Ø	0yuyy Ø	01010	01y10 Ø	01y1y Ø	1y0y0 Ø	1y0yy Ø	1yyyy Ø	11010	1101y Ø	11y1y Ø
x1111	0yuyy Ø	0y1yy Ø	0y1y1 Ø	01y1y Ø	0111y Ø	01111	1yyyy Ø	1yyy1 Ø	1y1y1 Ø	11y1y Ø	11y11 Ø	11111
1000x	y0000 Ø	y0y00 Ø	y0y01 Ø	yy0y0 Ø	yyyy0 Ø	yyyy1 Ø	10000	10001	10y01 Ø	1y0y0 Ø	1y0y1 Ø	1yyy1 Ø

Могут быть получены следующие тупиковые формы:

$$C_{min1} = \{11x11; 0010x; x0000; 10x01; 0111x; x1010; 00011; 01001; 11100; x0110\}$$

$$C_{min2} = \{1101x; 0010x; x0000; 10x01; 01x10; x1111; 00011; 01001; 11100; x0110\}$$

$$C_{min3} = \{11x11; 0111x; 00x00; x0101; x1010; 1000x; 00011; 01001; 11100; x0110\}$$

$$C_{min4} = \{1101x; 01x10; 00x00; x0101; x1111; 1000x; 00011; 01001; 11100; x0110\}$$

Поиск минимального покрытия завершён.

$$S_2 = a_1a_2b_2p + \bar{a}_1\bar{a}_2b_1\bar{b}_2 + \bar{a}_2\bar{b}_1\bar{b}_2\bar{p} + a_1\bar{a}_2\bar{b}_2p + \bar{a}_1a_2b_1b_2 + a_2\bar{b}_1b_2\bar{p} + \bar{a}_1\bar{a}_2\bar{b}_1b_2p + \bar{a}_1a_2\bar{b}_1\bar{b}_2p + a_1a_2b_1\bar{b}_2\bar{p} + \bar{a}_2b_1b_2\bar{p}$$

Запишем результат в базисе И-НЕ:

$$S_2 = \overline{(a_1a_2b_2p)} \cdot \overline{(\bar{a}_1\bar{a}_2b_1\bar{b}_2)} \cdot \overline{(\bar{a}_2\bar{b}_1\bar{b}_2\bar{p})} \cdot \overline{(a_1\bar{a}_2\bar{b}_2p)} \cdot \overline{(\bar{a}_1a_2b_1b_2)} \cdot \overline{(a_2\bar{b}_1b_2\bar{p})} \cdot \overline{(\bar{a}_1\bar{a}_2\bar{b}_1b_2p)} \cdot \overline{(\bar{a}_1a_2\bar{b}_1\bar{b}_2p)} \cdot \overline{(a_1a_2b_1\bar{b}_2\bar{p})} \cdot \overline{(\bar{a}_2b_1b_2\bar{p})}$$

Эффективность минимизации:

$$K = \frac{17 \cdot 5 + 17 + 5}{58} = 1,84$$

Функциональная схема ОЧС приведена в приложении Б.

3.3. Логический синтез преобразователя множителя

Преобразователь множителя (ПМ) – это устройство, которое служит для исключения из множителя диад 11, заменяя их на триады 10 $\bar{1}$. Если из Вых.2 поступает 1 в ФДК, то значит число – отрицательное.

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

Таблица 3.3.1 – Таблица истинности ПМ

Входная диада		Пер. из триггера	Знак	Сигнал в ФДК	Выходная диада	
Q_1	Q_2	T	D	F	Y_1	Y_2
1	2	3	4	5	6	7
0	0	0	0	0	0	0
0	0	1	0	0	0	1

0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	0	1	0
1	0	1	1	1	0	1
1	1	0	1	1	0	1
1	1	1	1	0	0	0

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

Функция D

Для функции D заполненная карта приведена на рисунке 3.3.2.

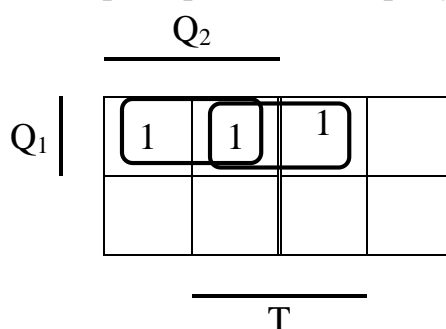


Рисунок 3.3.2 – Минимизация функции Q при помощи карты Вейча

Следовательно:

$$D = Q_1 Q_2 + Q_1 T = Q_1 (Q_2 + T).$$

Функция F

Для функции F заполненная карта приведена на рисунке 3.3.3.

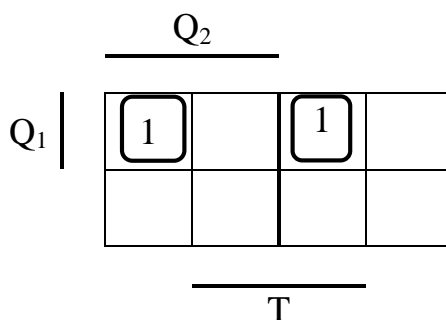


Рисунок 3.3.3 – Минимизация функции F при помощи карты Вейча

Следовательно:

$$F = Q_1 \overline{Q_2} T + Q_1 Q_2 \overline{T} = Q_1 (T \oplus Q_2).$$

Функция Y1

Для функции Y1 заполненная карта приведена на рисунке 3.3.4.

	Q_2			
Q_1				1
		1		
	T			

Рисунок 3.3.4 – Минимизация функции Y_1 при помощи карты Вейча

Следовательно:

$$Y_1 = \overline{Q_1} Q_2 T + Q_1 \overline{Q_2} \overline{T}.$$

Функция Y_2

Для функции Y_2 заполненная карта приведена на рисунке 3.3.5.

	Q_2			
Q_1	1		1	
	1		1	
	T			

Рисунок 3.3.5 – Минимизация функции Y_2 при помощи карты Вейча

Следовательно:

$$Y_2 = \overline{Q_2} T + Q_2 \overline{T} = T \oplus Q_2.$$

Функциональная схема ПМ приведена в приложении Д.

4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ

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

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

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

Управляющее поле такого мультиплексора будет определяться тремя переменными, следовательно, число групп с одинаковыми значениями этих переменных будет равно восьми. Также, реализация нескольких ПФ требует для каждой ПФ отдельного мультиплексора.

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

Таблица истинности для синтеза ПФ ОЧС приведена в таблице 4.1.

Таблица 4.1 – Таблица истинности для синтеза ПФ ОЧС

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

Продолжение таблицы 4.1

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

Функциональная схема ОЧС на основе мультиплексоров представлена в приложении Г.

5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ

Формула расчёта временных затрат на умножение:

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

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

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

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

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

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

n – количество разрядов множителя;

m – количество разрядов множимого.

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

ЗАКЛЮЧЕНИЕ

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

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

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

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

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

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

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

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

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

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

ПРИЛОЖЕНИЕ А

(обязательное)

Сумматор-умножитель второго типа. Схема электрическая структурная

ПРИЛОЖЕНИЕ Б

(обязательное)

Одноразрядный четверичный сумматор. Схема электрическая функциональная

ПРИЛОЖЕНИЕ В

(обязательное)

Одноразрядный четверичный сумматор-умножитель. Схема
электрическая функциональная

ПРИЛОЖЕНИЕ Г

(обязательное)

Однозарядный четверичный сумматор. Схема электрическая функциональная
на основе мультиплексоров

ПРИЛОЖЕНИЕ Д

(обязательное)

Преобразователь множителя. Схема электрическая функциональная

ПРИЛОЖЕНИЕ Е
(обязательное)
Ведомость документов