Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет информатики и радиоэлектроники”

Факультет компьютерных систем и сетей

Кафедра Электронных вычислительных машин

Дисциплина: Арифметические и логические основы

цифровых устройств

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_\_\_\_Е. И. Ковшер

ПОЯСНИТЕЛНАЯ ЗАПИСКА

к курсовой работе

на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-

УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 6-05-0611-05 623 ПЗ

Студент Рутковский В.К.

Руководитель Ковшер Е.И.

МИНСК 2024

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы

    цифровых устройств

 УТВЕРЖДАЮ

         Заведующий кафедрой ЭВМ

         \_\_\_\_\_\_\_\_\_ Б. В. Никульшин

         «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024г.

 ЗАДАНИЕ

по курсовой работе студента

Рутковского Вадима Кирилловича

**1.**Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-четвертичных чисел».

**2.**Срок сдачи студентом законченной работы: до 10 мая 2024г.

**3.**Исходные данные к работе:

* 1. Исходные сомножители: Мн = 25,75; Мт = 31,75;
  2. Алгоритм умножения: Г;

**3.3** Метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;

* 1. Коды четверичных цифр множимого для перехода к двоично- четверичной системе кодирования: 04 – 00, 14 – 01, 24 – 11, 34 – 10;
  2. Тип синтезируемого умножителя: 1;
  3. Логический базис для реализации ОЧС: А6; метод минимизации-алгоритм Рота
  4. Логический базис для реализации ОЧУ: А2; метод минимизации-карты Карно

**4.**Содержание пояснительной записки (перечень подлежащих разработке вопросов):

Введение. 1. Разработка алгоритма умножения. 2. Разработка структурной схемы сумматора-умножителя. 3. Разработка функциональных схем основных узлов сумматора-умножителя. 4. Синтез комбинационных схем устройств на основе мультиплексоров. 5. Оценка результатов разработки. Заключение. Список литературы.

**5.**Перечень графического материала:

**5.1** Сумматор-умножитель первого типа. Схема электрическая структурная. **5.2** Одноразрядный четверичный сумматор. Схема электрическая структурная.

**5.3** Одноразрядный четверичный умножитель-сумматор. Схема электрическая структурная

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

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов  курсовой работы | Объём этапа,  % | Срок  выполнения  этапа | Примечания |
| Разработка алгоритма умножения | 10 | 10.02-29.02 |  |
| Разработка структурной схемы сумматора-умножителя | 10 | 29.02-29.03 | С выполнением  чертежа |
| Разработка функциональных схем основных узлов сумматора-умножителя | 50 | 29.03-29.04 | С выполнением  чертежа |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 | 29.03-29.04 | С выполнением  чертежа |
| Завершение оформления  Пояснительной записки | 20 | 29.04-10.05 |  |

Дата выдачи задания: 9 февраля 2024 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_Е.И. Ковшер

ЗАДАНИЯ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_\_\_\_В.К. Рутковский

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ 5

1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ 6

1.1 Перевод сомножителей из десятичной системы счисления в четверичную 6

1.2 Нахождение произведения множителя и множимого 6

1.3 Перемножение мантисс по алгоритму Г 7

2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ 9

3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ 12

3.1 Логический синтез одноразрядного четверичного сумматора-умножителя 12

3.2 Логический синтез одноразрядного четверичного сумматора 17

3.3 Логический синтез преобразователя множителя 24

4 СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ 26

5 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ 28

ЗАКЛЮЧЕНИЕ 29

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30

ПРИЛОЖЕНИЕ А 31

ПРИЛОЖЕНИЕ Б 32

ПРИЛОЖЕНИЕ В 33

ПРИЛОЖЕНИЕ Г 34

ПРИЛОЖЕНИЕ Д 35

ПРИЛОЖЕНИЕ Е 36

**ВВЕДЕНИЕ**

В мире, где цифровые технологии играют все более важную роль в нашей повседневной жизни, понимание основ арифметики и логики цифровых устройств становится критически важным. Разработка и синтез сумматора-умножителя для двоично-четверичных чисел представляет собой значимую задачу в области цифровой электроники. Это устройство способно выполнять операции сложения и умножения двоичных чисел. В данной работе мы исследуем основы проектирования и логического синтеза такого устройства.

Кроме того, курсовая работа затрагивает не только создание самого

сумматора-умножителя, но и решение разнообразных задач, включая перевод чисел из одной системы счисления в другую, применение алгоритмов умножения, разработку функциональных схем, анализ времени работы, минимизацию логических функций и оценку преимуществ и недостатков различных методов оптимизации. В пояснительной записке представлено краткое описание процесса проектирования, а также графическая документация по структурной и функциональным схемам устройства.

Таким образом, данная тема открывает широкие перспективы для изучения основ арифметики и логики цифровых систем, а также их применения в различных областях обработки цифровой информации.

# 

# **1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ**

* 1. **Перевод сомножителей из десятичной системы счисления в четверичную**

**Множимое:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| — | 25 |  | 4 |  |  |
| 24 | — | 6 |  | 4 |
|  | **1** | 4 |  | 1 |
|  |  |  | **2** |  |

|  |  |
| --- | --- |
|  | 0,75 |
| \* | 4 |
|  | 3,00 |

В соответствии с заданной кодировкой множимого:

**Множитель:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| — | 31 |  | 4 |  |  |
| 28 | — | 7 |  | 4 |
|  | **3** | 4 |  | 1 |
|  |  |  | **3** |  |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
|  | 0,75 |
| \* | 4 |
|  | 3,00 |

В соответствии с обычной весомозначной кодировкой множителя:

**1.2 Нахождение произведения множителя и множимого**

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

3. 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 следующим:

Результат закодирован в соответствии с заданием на кодировку множимого.

Знак произведения определяется суммой по модулю два знаков сомножителей, т. е.:

**1.3 Перемножение мантисс по алгоритму Г**

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в прямых кодах диада 11() заменяется на триаду . Преобразованный множитель имеет вид Преобразованный множитель имеет вид или . Перемножение мантисс по алгоритму «Г» приведено в таблице 1.1.

Таблица 1.1-Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Четверичная С/С | | Двоично-четверичная С/C | | Комментарии |
| 1 | | 2 | | 3 |
| 0. | 00000 | 0. | 0000000000 | ∑0ч |
| 0. | 00303 2 | 0. | 000010001011 | П1ч=Мн\*2\* |
| 0. | 00303 2 | 0. | 000010001011 | ∑1ч |
| 0. | 00000 00 | 0. | 00000000000000 | П2ч=Мн\*0\*4-2 |
| 0. | 00303 2 | 0. | 000010001011 | ∑2ч |
| 0. | 00000 000 | 0. | 0000000000000000 | П3ч=Мн\*0\*4-3 |
| 0. | 00303 2 | 0. | 000010001011 | ∑3ч |
| 3. | 33333 2120 | 1. | 101010101011011100 | П4ч=Мн\*(-1)\*4-4 |
| 0. | 00303 0120 | 0. | 000010001000011100 | ∑4ч |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (, ) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

= 0;

.

Результат прямого перемножения операндов даёт следующее значение:

.

Абсолютная погрешность:

Относительная погрешность:

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 первого типа строится на базе заданных узлов ОЧУ, ОЧС, формирователя дополнительного кода и регистра результата. 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). Также из-за того, что при формировании дополнительного кода диада «11» преобразуется в «101», то на вход ОЧУ эта диада никогда не поступит, следовательно, в таблице истинности работы ОЧУ будут содержаться 8 безразличных входных наборов.

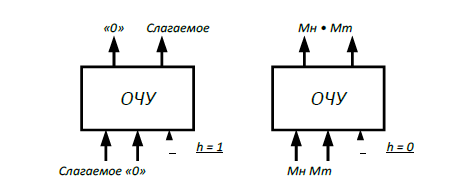


Рисунок 2.1 – Режимы работы ОЧУ

При необходимости выравнивания порядков в регистре-аккумуляторе может выполняться сдвиг мантиссы первого слагаемого. Если на вход *h* поступает «0», то ОЧУ перемножает разряды Мн и Мт.

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

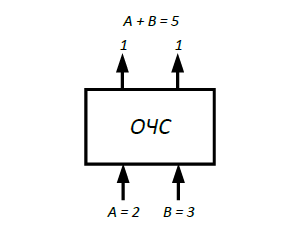


Рисунок 2.2 – Одноразрядный четверичный сумматор

В ОЧС первое слагаемое складывается с нулем, 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.

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** |

*Продолжение таблицы 2.1*

|  |  |  |
| --- | --- | --- |
| ***F1*** | ***F2*** |  |
| 0 | 0 | Дополнительный код множимого |
| 0 | 1 | Дополнительный код слагаемого |
| 1 | 0 | Меняется знак Мн |
| 1 | 1 | Меняется знак слагаемого |

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

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.

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»:

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. Количество тактов умножения определяется разрядностью Мт.

Структурная схема сумматора-умножителя приведена в приложении А.

**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 – 00; 1 – 01; 2 – 11; 3 – 10.

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 2.x 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 «00» |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1·0=00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Выход – код «01» |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1·1=01 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Выход – код «01» |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1·2=02 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход – код «01» |
| 0 | 1 | 1 | 1 | 0 | х | х | х | х | 1·3=03 |
| 0 | 1 | 1 | 1 | 1 | х | х | х | х | Выход – код «01» |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3·0=00 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Выход – код «03» |

*Продолжение таблицы 3.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 0 | 1 | 1 | 1 | 3·2=12 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Выход – код «03» |
| 1 | 0 | 1 | 1 | 0 | x | x | x | x | 3·3=21 |
| 1 | 0 | 1 | 1 | 1 | x | x | x | x | Выход – код «02» |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2·0=00 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Выход – код «02» |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2·1=02 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Выход – код «02» |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2·2=10 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Выход – код «02» |
| 1 | 1 | 1 | 1 | 0 | x | x | x | x | 2·3=12 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | x | Выход – код «02» |

Так как функция не принимает единичные значения, то минимизацию этой функции не проводим.

Минимизацию переключательных функций проведём с помощью карт Вейча. Для функции заполненная карта приведена на рисунке 3.1, где симво- лом «x» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 01 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 11 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |
| 10 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |

Рисунок 3.1 – Минимизация функции при помощи карты Вейча

Следовательно СДНФ для :

Запишем результаты обеих функций в базисе И - Константная единица - Сумма по модулю 2:

Эффективность минимизации можно оценить:

Минимизация для КНФ представлена на рисунке 3.2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 01 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 11 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |
| 10 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |

Рисунок 3.2 – Минимизация функции при помощи карты Вейча

СКНФ для :

Эффективность минимизации можно оценить:

Минимизация функции при помощи карты Вейча представлена на рисунке 3.3. Символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 01 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |
| 11 | 0 | 1 | 1 | 1 | \* | \* | 1 | 0 |  |
|  |
| 10 | 0 | 1 | 1 | 1 | \* | \* | 1 | 1 |  |
|  |

Рисунок 3.3 – Минимизация функции при помощи карты Вейча

Эффективность минимизации можно оценить:

Запишем результаты обеих функций в базисе И - Константная единица - Сумма по модулю 2:

Минимизация для КНФ представлена на рисунке 3.4

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 01 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |
| 11 | 0 | 1 | 1 | 1 | \* | \* | 1 | 0 |  |
|  |
| 10 | 0 | 1 | 1 | 1 | \* | \* | 1 | 1 |  |
|  |

Рисунок 3.4 – Минимизация функции при помощи карты Вейча

СКНФ для :

Эффективность минимизации можно оценить:

Минимизация функции представлена на рисунке 3.5. Символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 01 | 0 | 1 | 1 | 1 | \* | \* | 1 | 1 |  |
|  |
| 11 | 0 | 1 | 1 | 1 | \* | \* | 1 | 0 |  |
|  |
| 10 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |

Рисунок 3.5 – Минимизация функции при помощи карты Вейча

Следовательно:

Эффективность минимизации можно оценить:

Запишем результаты обеих функций в базисе И - Константная единица - Сумма по модулю 2:

Минимизация для КНФ представлена на рисунке 3.6

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 0 | 0 | 0 | \* | \* | 0 | 0 |  |
|  |
| 01 | 0 | 1 | 1 | 1 | \* | \* | 1 | 1 |  |
|  |
| 11 | 0 | 1 | 1 | 1 | \* | \* | 1 | 0 |  |
|  |
| 10 | 0 | 0 | 0 | 0 | \* | \* | 0 | 1 |  |
|  |

Рисунок 3.6 – Минимизация функции при помощи карты Вейча

СКНФ для :

Эффективность минимизации можно оценить:

Схема электрическая функциональная представлена в приложении Б

**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 2.x).

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

ОЧС синтезируется для схемы первого типа, поэтому в таблице истинности необходимо выделить 16 безразличных наборов, т. к. со старших выходов ОЧУ не могут прийти коды «2» и «3».

Таблица 3.2-Таблица истинности ОЧС

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | b1 | b2 | P | П | S1 | S2 | Результат в  четверичной СС |
| 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 | 0 | 1 | 0+0+1=01 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0+1+0=01 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0+1+1=02 |
| 0 | 0 | 1 | 0 | 0 | x | x | x | 0+3+0=03 |
| 0 | 0 | 1 | 1 | 0 | x | x | x | 0+2+0=02 |
| 0 | 0 | 1 | 1 | 1 | x | x | x | 0+2+1=03 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1+0+0=01 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1+0+1=02 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1+1+0=02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1+1+1=03 |
| 0 | 1 | 1 | 0 | 0 | x | x | x | 1+3+0=10 |
| 0 | 1 | 1 | 0 | 1 | x | x | x | 1+3+1=11 |
| 0 | 1 | 1 | 1 | 0 | x | x | x | 1+2+0=03 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | 1+2+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 | 1 | 0 | 0 | 3+1+0=10 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 3+1+1=11 |
| 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 | x | x | x | 3+2+0=11 |
| 1 | 0 | 1 | 1 | 1 | x | x | x | 3+2+1=12 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 2+0+0=02 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 2+0+1=03 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2+1+0=03 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 2+1+1=10 |
| 1 | 1 | 1 | 0 | 0 | x | x | x | 2+3+0=11 |
| 1 | 1 | 1 | 0 | 1 | x | x | x | 2+3+1=12 |
| 1 | 1 | 1 | 1 | 0 | x | x | x | 2+2+0=10 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | 2+2+1=11 |

Определим множество единичных кубов:

L=

и множество безразличных кубов:

N=

Склеим множества L и N для упрощения минимизации:

=

=

Сформируем множество :

Первым этапом алгоритма Рота является нахождение множества простых импликант Z. Для его реализации используется операция умножения «\*» над множествами С0, С1 и так до Сn. Операция умножения прекращается, когда мощность очередного множества Ci становится меньше двух.

Первый шаг умножения () представлен в таблице 3.3:)

Таблица 3.3-Умножения \*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \* | 10010 | 11011 | 100x1 | xx1xx |
| 10010 | - |  |  |  |
| 11011 |  | - |  |  |
|  |  |  |  |  |
| 100x1 | 1001y | 1y011 | - |  |
| xx1xx | 10y10 | 11y11 | 10yx1 | - |
|  | 1001x  10x10 | 1x011  11x11 | 10xx1 |  |

= {1001x,10x10,1x011,11x11,10xx1}

= {Ø}

= {10010,11011,100x1,xx1xx}

= {1001x,10x10,1x011,11x11,10xx1, xx1xx}

Второй шаг умножения \* представлен в таблице 3.4.

Таблица 3.4-Умножения \*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \* | 1001x | 10x10 | 1x011 | 11x11 | 10xx1 |
| 1001x | - |  |  |  |  |
| 10x10 |  | - |  |  |  |
| 1x011 |  |  | - |  |  |
| 11x11 |  |  |  | - |  |
| 10xx1 |  | 10x1y |  | 1yx11 | - |
| xx1xx | 10y1x |  | 1xy11 |  |  |
|  | 10x1x | 10x1x | 1xx11 | 1xx11 |  |

= {10x1x,1xx11}

= {Ø}

= {1001x,10x10,1x011,11x11,10xx1,xx1xx}

= {10x1x,1xx11,10xx1,xx1xx}

Третий шаг умножения \* представлен в таблице 3.4

Таблица 3.5-Умножения \*

|  |  |  |
| --- | --- | --- |
| \* | 10x1x | 1xx11 |
| 10x1x | *-* |  |
| 1xx11 |  | *-* |
| 10xx1 |  |  |
| *xx1xx* |  |  |
|  |  |  |

={Ø}

В результате всех трех операций получаем следующие простые импликанты:

Z = {10x1x,1xx11,10xx1,xx1xx}

Следующим этапом алгоритма Рота является поиск L-экстремалей на множестве простых импликант. Для этого с помощью операции вычитания «#» находим кубы, которые могут являться L-экстремалями, и проверяем принадлежность полученных кубов к множеству L с помощью операции пересечения «∩».

Операция z#(Z-z) для поиска таких кубов представлена в таблице 3.6, а операция L∩(z#(Z-z)) для проверки принадлежности этих кубов к множеству L представлена в таблице

Таблица 3.6- Операция z#(Z-z) для поиска таких кубов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z#(Z-z) | 10x1x | 1xx11 | 10xx1 | xx1xx |
| 10x1x | - | 11x11 | 10x01 | 0x1xx  x11xx  xx10x |
| 1xx11 |  | - |  | 0x1xx  011xx  x110x  x11x0  xx10x |

*Продолжение таблицы 3.6*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10xx1 | 10x10 | 11x11 | - | 0x1xx  011xx  x110x  x11x0  0x10x  x110x  xx100 |
| xx1xx | 10x10 | 11x11 | 10x01 | - |
| Остаток | 10x10 | 11x11 | 10x01 | 0x1xx  011xx  x110x  x11x0  0x10x  x110x  xx100 |

Нахождение L-экстремалей представлено в таблице 3.7

Таблица 3.7- Нахождение L-экстремалей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z#(Z-z) | 10x1x | 1xx11 | 10xx1 | xx1xx |
| 10x1x | - |  | 10x01 | 0x1xx x11xx xx10x |
| 1xx11 | 10x10 | - | 10x01 | 0x1xx 011xx x110x x11x0 xx10x |
| 10xx1 | 10x10 | 11x11 | - | 0x1xx 011xx x110x x11x0 0x10x x110x xx100 |
| xx1xx | 10x10 | 11x11 | 10x01 | - |

*Продолжение таблицы 3.7*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Остаток | 10x10 | 11x11 | 10x01 | 0x1xx 011xx x110x x11x0 0x10x x110x xx100 |

E={10x1x,1xx11,10xx1}.

После выявления L-экстремалей следует выяснить, не являются ли некоторые из них простыми импликантами, остатки которых покрывают только некоторое подмножество кубов комплекса N, которое нет необходимости покрывать, вводя в минимальное покрытие соответствующие наборы. Для этого необходимо выполнить операцию пересечения остатков, полученных при выполнении операции z#(Z\z) с кубами из комплекса L. Во множестве E необходимо оставить только те кубы, остатки от которых пересекаются с кубами из комплекса L.

Таблица 3.8- Операция пересечения остатков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z#(Z-z) ⋂ L | 10001 | 10010 | 10011 | 11011 |
| 10x10 | 100yy  Ø | 10010 | 1001y  Ø | 1y01y  Ø |
| 11x11 | 1y0y1  Ø | 1y01y  Ø | 1y011  Ø | 11011 |
| 10x01 | 10001 | 100yy  Ø | 100y1 Ø | 1y0y1 Ø |
| 0x1xx | y0y01 Ø | y0y10 Ø | y0y11 Ø | y1y11 Ø |
| 011xx | yyy01 Ø | yyy10 Ø | yyy11 Ø | y1y11 Ø |
| x110x | 1yy01 Ø | 1yyy0 Ø | 1yyy1 Ø | 11yy1 Ø |
| x11x0 | 1yy0y Ø | 1yy10 Ø | 1yy1y Ø | 11y1y Ø |
| 0x10x | y0y01 Ø | y0yy0 Ø | y0yy1 Ø | y1yy1 Ø |
| x110x | 1yy01 Ø | 1yyy0 Ø | 1yyy1 Ø | 11yy1 Ø |
| xx100 | 10y0y Ø | 10yy0 Ø | 10yyy Ø | 11yyy Ø |

Далее выясним, какие из вершин комплекса L не покрываются L-экстремалями. Для этого из каждого куба комплекса L вычтем (#) элементы множества Е

Таблица 3.9- Вычитание L из множества E

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L#E | 10001 | 10010 | 10011 | 11011 |
| 10x1x | 10001 | Ø | Ø | 11011 |
| 1xx11 | 10001 | Ø | Ø | Ø |
| 10xx1 | Ø | Ø | Ø | Ø |
| Остаток | Ø | Ø | Ø | Ø |

Из таблицы видно, что L-экстремалями покрыты все единичные кубы.

Следовательно, существует одна тупиковая (минимальная) форма:

Эффективность минимизации можно оценить:

***Минимизация функции S1:***

Минимизацию функции S1 произведём с помощью карт Карно-Вейча, которые представлены на рисунке 3.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 0 | 1 | 0 | \* | \* | \* | \* |  |
|  |
| 01 | 0 | 1 | 1 | 1 | \* | \* | \* | \* |  |
|  |
| 11 | 1 | 1 | 0 | 1 | \* | \* | \* | \* |  |
|  |
| 10 | 1 | 0 | 0 | 0 | \* | \* | \* | \* |  |
|  |

Рисунок 3.1 – Минимизация функции при помощи карты Вейча

Следовательно СДНФ для S1:

Эффективность минимизации можно оценить:

***Минимизация функции S2:***

Минимизацию функции S2 произведём с помощью карт Карно-Вейча, которые представлены на рисунке 3.2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |  |
| 00 | 0 | 1 | 1 | 1 | \* | \* | \* | \* |  |
|  |
| 01 | 1 | 1 | 0 | 1 | \* | \* | \* | \* |  |
|  |
| 11 | 1 | 0 | 0 | 0 | \* | \* | \* | \* |  |
|  |
| 10 | 0 | 0 | 1 | 0 | \* | \* | \* | \* |  |
|  |

Рисунок 3.2 – Минимизация функции при помощи карты Вейча

Следовательно СДНФ для S2:

Эффективность минимизации можно оценить:

Функциональная схема ОЧС приведена в приложении В.

## **3.3 Логический синтез преобразователя множителя**

Преобразователь множителя (ПМ) – это устройство, которое преобразовывает диады множителя в соответствии с методом умножения.

При умножении в дополнительных кодах ПМ заменяет диады 11 (34) и 10 (24) на триады и соответственно.

Принцип работы ПМ представлен с помощью таблицы истинности (таблица 3.10)

Таблица 3.10- Таблица истинности ПМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Входная диада** | | **Младший разряд** | **Знак** | **Выходная диада** | |
| **Q1** | **Q2** | **p** | **Z** | **S1** | **S2** |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |

*Продолжение таблицы 3.10*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | \* | 0 | 0 |

Минимизацию переключательных функций произведём с помощью карт Карно-Вейча. (См. рисунки 3.3-3.5)

Минимизация функции Z:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | p | |  |
|  |  | Q2 | |  |  |
|  |  |  |  |  |  |
| Q1 |  | 1 | \* | 1 | 1 |
|  |  |  |  |  |  |

Рисунок 3.3 - Минимизация функции Z картой Вейча

Следовательно:

Z = Q1

***Минимизация функции S1:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | p | |  |
|  |  | Q2 | |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  | 1 |
|  |  |  | 1 |  |  |

Рисунок 3.4 - Минимизация функции S1 картой Вейча

Следовательно:

=

***Минимизация функции S2:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Q2p |  |  |  |  |
| Q1 |  | 00 | 01 | 11 | 10 |
|  | 0 |  | 1 |  | 1 |
|  | 1 |  | 1 |  | 1 |

Рисунок 3.5 - Минимизация функции S2 картой Карно

Следовательно:

=

Функциональная схема ПМ приведена в приложении Д

# **4. ЛОГИЧЕСКИЙ СИНТЕЗ ОДНОРАЗРЯДНОГО ЧЕТВЕРИЧНОГО СУММАТОРА НА ОСНОВЕ МУЛЬТИПЛЕКСОРА**

*Мультиплексор* – это логическая схема, которая имеет *n* информационных входов, *m* управляющих входов и один выход. При этом должно выполнятся условие .

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | | b1 | | b2 | | P | | П | | Функция | | | S1 | | Функция | | S2 | | Функция |
| 1 | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | | 8 | | 9 | | 10 | | 11 |
| 0 | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | | 0 | |  | | 0 | |  |
| 0 | 0 | | 0 | | 0 | | 1 | | 0 | | 0 | | 1 | |
| 0 | 0 | | 0 | | 1 | | 0 | | 0 | | 0 | | 1 | |
| 0 | 0 | | 0 | | 1 | | 1 | | 0 | | 1 | | 1 | |
| 0 | 0 | | 1 | | 0 | | 0 | | x | |  | | | x | |  | | x | |  |
| 0 | 0 | | 1 | | 0 | | 1 | | x | | x | | x | |
| 0 | 0 | | 1 | | 1 | | 0 | | x | | x | | x | |
| 0 | 0 | | 1 | | 1 | | 1 | | x | | x | | x | |
| 0 | | 1 | | 0 | | 0 | | 0 | | 0 | | 0 | 0 | |  | | 1 | |  | |
| 0 | | 1 | | 0 | | 0 | | 1 | | 0 | | 1 | | 1 | |
| 0 | | 1 | | 0 | | 1 | | 0 | | 0 | | 1 | | 1 | |
| 0 | | 1 | | 0 | | 1 | | 1 | | 0 | | 1 | | 0 | |
| 0 | | 1 | | 1 | | 0 | | 0 | | x | |  | x | |  | | x | |  | |
| 0 | | 1 | | 1 | | 0 | | 1 | | x | | x | | x | |
| 0 | | 1 | | 1 | | 1 | | 0 | | x | | x | | x | |
| 0 | | 1 | | 1 | | 1 | | 1 | | x | | x | | x | |
| 1 | | 0 | | 0 | | 0 | | 0 | | 0 | |  | 1 | |  | | 0 | |  | |
| 1 | | 0 | | 0 | | 0 | | 1 | | 1 | | 0 | | 0 | |
| 1 | | 0 | | 0 | | 1 | | 0 | | 1 | | 0 | | 0 | |

*Продолжение таблицы 4.1*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 1 | 1 |  | 0 |  | 1 |  |
| 1 | 0 | 1 | 0 | 0 | x |  | x |  | x | - |
| 1 | 0 | 1 | 0 | 1 | x | x | x |
| 1 | 0 | 1 | 1 | 0 | x | x | x |
| 1 | 0 | 1 | 1 | 1 | x | x | x |
| 1 | 1 | 0 | 0 | 0 | 0 |  | 1 |  | 1 |  |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | x |  | x |  | x |  |
| 1 | 1 | 1 | 0 | 1 | x | x | x |
| 1 | 1 | 1 | 1 | 0 | x | x | x |
| 1 | 1 | 1 | 1 | 1 | x | x | x |

Перестроим таблицу убрав адресный вход построим мультиплексор на 4 адресных входа, так как из полученной таблицы истинности можно заметить, что поведение выходов для =1 не определено. Следовательно, вход можно упразднить, полагая =0.Результат представлен в таблице 4.2

Таблица 4.2\_Таблица ПФ ОЧС на 2 адресных входа

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | b2 | P | П | **Функция** | S1 | **Функция** | S2 | **Функция** |
| 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 1 |  |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |  | 1 |  | 0 |  |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |  | 1 |  | 1 |  |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Схема электрическая функциональная на основе мультиплексоров представлена в приложении Г

# **5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ**

Формула расчёта временных затрат на умножение:

Т=(𝑛 + 1) ∗ (𝑇ПМ + 𝑇ФДК + 𝑇ОЧУ + (m + n + 1) \* 𝑇ОЧС + 𝑇сдвига), где

𝑇ПМ – время преобразования множителя;

𝑇ФДК – время формирования дополнительного кода множимого;

𝑇ОЧУC – время умножения на ОЧУ;

𝑇ОЧС – время формирования единицы переноса в ОЧС;

𝑇сдвига – 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 первого типа, а также функциональные схемы основных узлов данного устройства. Для снижения стоимости логических схем были проведены минимизации переключательных функций с использованием различных методов. Этот подход позволил выявить преимущества и недостатки данных алгоритмов.

Одним из главных преимуществ минимизации с помощью карт Карно-Вейча является их простота и минимальное время выполнения. Однако применение этого метода для функций с большим количеством переменных может быть затруднительным. Для минимизации таких функций удобно применять алгоритм Рота, который формализует процесс минимизации и делает его доступным для выполнения с использованием компьютерных программ.

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

Реализация переключательных функций на основе мультиплексоров упростила процесс их минимизации и упростила функциональную схему одноразрядного четверичного сумматора.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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

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

[3] Луцик, Ю. А. Методические указания к курсовому проекту по курсу “Арифметические и логические основы вычислительной техники” / Ю. А. Луцик, И. В. Лукьянова – Минск : БГУИР, 2004 г – 35 с.

# **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**)**

Однозарядный четверичный сумматор. Схема электрическая функциональная на основе мультиплексоров

# **ПРИЛОЖЕНИЕ Д**

**(**обязательное**)**

Преобразователь множителя. Схема электрическая функциональная

# **ПРИЛОЖЕНИЕ Е**

**(**обязательное**)**

Ведомость документов