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

\_\_\_\_\_\_\_\_\_\_ Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 210 ПЗ

Студент Е. А. Катковская

Руководитель Ю. А. Луцик

Минск 2022

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
 информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы цифровых устройств

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

Б. В. Никульшин

« » 20 г.

ЗАДАНИЕ

по курсовой работе студента

Катковской Елизаветы Александровны

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-десятичных чисел»
2. Срок сдачи студентом законченной работы: до 20 мая 2022 г.
3. Исходные данные к работе:
   1. исходные сомножители: Мн = 37,67; Мт = 34,92;
   2. алгоритм умножения: Г;
   3. метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в дополнительных кодах;
   4. коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования: 04 – 00, 14 – 01, 24 – 10, 34 – 11;
   5. тип синтезируемого умножителя: 2;
   6. логический базис для реализации ОЧС: ИЛИ, генератор «1», сложение по модулю 2; метод минимизации – карты Карно – Вейча;
   7. логический базис для реализации ОЧУС: ИЛИ, НЕ; метод минимизации – алгоритм Рота.
4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

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

1. Перечень графического материала:
   1. Сумматор-умножитель второго типа. Схема электрическая структурная.
   2. Одноразрядный четверичный сумматор. Схема электрическая функциональная.
   3. Одноразрядный четверичный умножитель. Схема электрическая функциональная.
   4. Регистр-аккумулятор. Схема электрическая функциональная.
   5. Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объём этапа,  % | Срок выполнения  этапа | Примечания |
| Разработка алгоритма умножения | 10 | 10.02–20.02 |  |
| Разработка структурной схемы  сумматора-умножителя | 10 | 21.02–09.03 | С выполнением  чертежа |
| Разработка функциональных схем основных узлов сумматора- умножителя | 50 | 10.03–30.04 | С выполнением чертежей |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 | 01.05–15.05 | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 | 15.05–20.05 |  |

Дата выдачи задания: 10 февраля 2022 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_/Ю. А. Луцик /

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

**СОДЕРЖАНИЕ**

[Введение 5](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303619)

[1 Разработка алгоритма умножения 6](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303620)

[2 Разработка структурной схемы сумматора-умножителя](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303621) 9

[3 Разработка функциональных схем основных узлов сумматора-умножителя 1](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303622)0

[3.1 Логический синтез одноразрядного четверичного сумматора-умножителя 1](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303624)0

[3.2 Логический синтез одноразрядного четверичного сумматора 23](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303625)

[3.2 Логический синтез Преобразователя множителя 2](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303625)7

[4 Синтез схемЫ ОЧС на основе мультиплексора](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303626) 29

[5 Оценка результатов разработки](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303627) 31

[Заключение](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303628) 32

[Список использованных источников](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303629) 33

[Приложение А Сумматор-умножитель второго типа. Схема электрическая структурная](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303630) 34

[Приложение Б Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303631) 35

[Приложение В Одноразрядный четверичный сумматор. Схема электрическая функциональная](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303632) 36

[Приложение Г Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303633) 37

[Приложение Д Преобразователь множителя.Схема электрическая функциональная](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303634) 38

[Приложение Е Ведомость документов](file:///D:\1ФЛЕШКА\Курсовая\Варианты%20курсачей\Сахарук\Мое(О%20Господи(Чё))детище(да)(Содержание).docx#_Toc450303634) 39

**ВЕДЕНИЕ**

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 цифровых устройств”, 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.

Мн = 37,67; Мт = 34,92.

|  |
| --- |
| 0.67  \* 4 |
| 2.68 |
| \* 4 |
| 2.72  \* 4 |
| 2.88 |

**Множимое**

|  |  |  |
| --- | --- | --- |
| \_37  36 | 4 | |
| \_9  8 | 4 |
| \_ 1 | 2 |
|  | 1 |
|  | | |

Мн4 = 211,222.

В соответствии с кодировкой множимого:

Мн2/4 =100101,101010.

**Множитель**

|  |
| --- |
| 0.92  \* 4 |
| 3.68  \* 4 |
| 2.72  \* 4 |
| 2.88 |

|  |  |  |  |
| --- | --- | --- | --- |
| \_34  32 | 4 | |  |
| \_8  8 | 4 |
| 2 | 2 |
| 0 | |
|  | |

Мт4 = 202,322.

В соответствии с обычной весомозначной кодировкой множителя:

Мт2/4 = 100010,111010.

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

Мн = 0.00100101101010 Рмн = 0.0100 + 104  – закодировано по заданию

Мт = 0.00100010111010 Рмт = 0.0100 + 104 – закодировано традиционно

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

Порядок произведения будет следующим:

Рмн = 0.0100 +104

Рмт = 0.0100 +104

=0.1000 +204

Результат закодирован в соответствии с заданием на кодировку множимого.

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

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в дополнительных кодах диады 10(24) и11(34) заменяются на триады 10 10. Преобразованный множитель имеет вид: Мтп4 = 110 или Мтп2 =0100100000. Перемножение мантисс по алгоритму “Г” приведено в табл. 1.1.

Таблица 1.1 - Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Четверичная с/c** | | **Двоично-четверичная с/с** | | **Комментарий** |
| 0.  0.  0.  3.  0.  0.  0.  3.  0.  0.  0.  3.  0.  3.  0. | 00000000000000  00211222000000  00211222000000  33231023000000  00102311000000  00002112220000  00111023220000  33333122112000  00110211332000  00000000000000  00110211332000  33333331221120  00110203213120  33333332310230  00110202130010 | 00.  00.  00.  11.  00.  00.  00.  11.  00.  00.  00.  11.  00.  11.  00. | 00 00 00 00 00 00 00 00 00 00 00 00 00 00  00 00 10 01 01 10 10 10 00 00 00 00 00 00  00 00 10 01 01 10 10 10 00 00 00 00 00 00  11 11 10 11 01 00 10 11 00 00 00 00 00 00  00 00 01 00 10 11 01 01 00 00 00 00 00 00  00 00 00 00 10 01 01 10 10 10 00 00 00 00  00 00 01 01 01 00 10 11 10 10 00 00 00 00  11 11 11 11 11 01 10 10 01 01 10 00 00 00  00 00 01 01 00 10 01 01 11 11 10 00 00 00  00 00 00 00 00 00 00 00 00 00 00 00 00 00  00 00 01 01 00 10 01 01 11 11 10 00 00 00  11 11 11 11 11 11 11 01 10 10 01 01 10 00  00 00 01 01 00 10 00 11 10 01 11 01 10 00  11 11 11 11 11 11 11 10 11 01 00 10 11 00  00 00 01 01 00 10 00 10 01 11 00 00 01 00 |  |

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

;

,4385

Результат прямого перемножения операндов дает результат:

Абсолютная погрешность:

1315,4364 – 1314,4385= 0,9979.

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 второго типа**

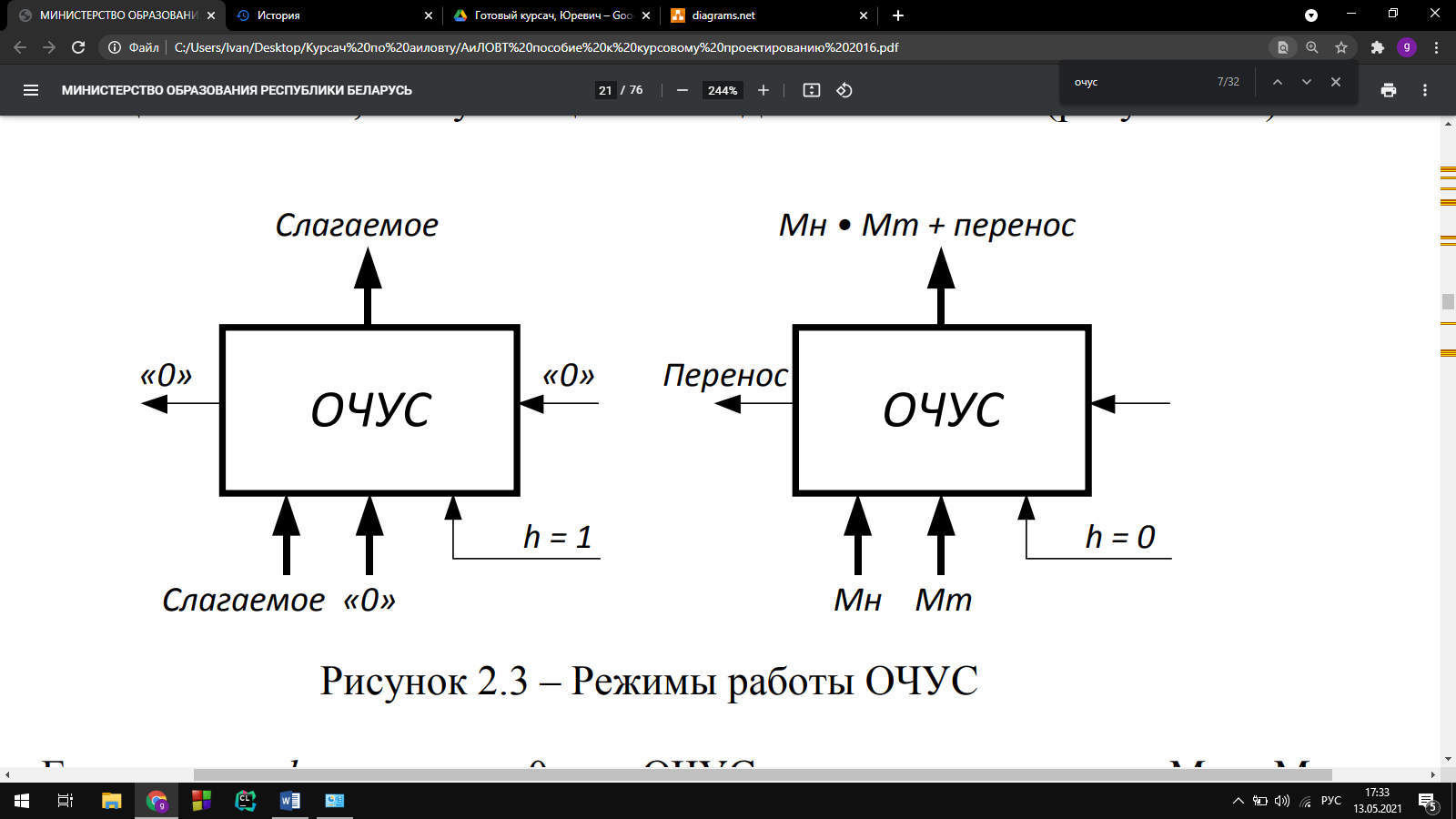
*Если устройство работает как сумматор*, то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода F2 поступает «1». Первое слагаемое переписывается в регистр результата под действием управляющих сигналов, поступающих на входы h всех ОЧУС (рисунок 2.1).

Рисунок 2.1 – Режимы работы ОЧУС

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

В ОЧС 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 ОЧС 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.

Схема структурная сумматора-умножителя представлена в ПРИЛОЖЕНИИ A**.**

**3 Разработка ФУНКЦИОНАЛЬНЫХ схем ОСНОвных узлов СУММАТОРА-УМНОЖИТЕЛЯ**

**3.1. Логический синтез одноразрядного четверичного сумматора-умножителя**

ОЧУС – это комбинационное устройство, имеющее шесть входов (два

разряда из регистра множимого, два разряда из регистра множителя, вход переноса и управляющий вход h) и три выхода.

Принцип работы ОЧУС 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 – 10; 3 – 11.

Управляющий вход h определяет тип операции:

0 – умножение закодированных цифр, поступивших на информационные входы, и добавление переноса;

1 – вывод на выходы без изменения значений разрядов, поступивших из регистра множимого.

В таблице 3.1 выделено 36 безразличных наборов, т. к. на входы ОЧУС из разрядов множителя не может поступить код «11», при работе ОЧУС как сумматора на вход переноса не может поступить единица, а при умножении на ноль или единицу на вход переноса также не может поступить единица.

Таблица 3.1 – Таблица истинности ОЧУС

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **P1** | **x1** | **x2** | **y1** | **y2** | **Упр.**  **h** | **П** | **Q1** | **Q2** | **Результат операции в четверичной с/с** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0\*0+0=00 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Выход код «00» |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0\*1+0=00 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Выход код «00» |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0\*2+0=00 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | Выход код «00» |
| 0 | 0 | 0 | 1 | 1 | 0 | x | x | x | 0\*3+0=00 |
| 0 | 0 | 0 | 1 | 1 | 1 | x | x | x | Выход код «00» |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1\*0+0=00 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Выход код «01» |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1\*1+0=01 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | Выход код «01» |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1\*2+0=02 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Выход код «01» |
| 0 | 0 | 1 | 1 | 1 | 0 | x | x | x | 1\*3+0=03 |

*Продолжение таблицы 3.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 1 | 1 | 1 | 1 | x | x | x | Выход код «01» |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2\*0+0=00 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | Выход код «02» |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2\*1+0=02 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Выход код «02» |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2\*2+0=10 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Выход код «02» |
| 0 | 1 | 0 | 1 | 1 | 0 | x | x | x | 2\*3+0=12 |
| 0 | 1 | 0 | 1 | 1 | 1 | x | x | x | Выход код «02» |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | 1 | 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 | 0\*0+1=01 |
| 1 | 0 | 0 | 0 | 0 | 1 | x | x | x | Выход код «00» |
| 1 | 0 | 0 | 0 | 1 | 0 | x | x | x | 0\*1+1=01 |
| 1 | 0 | 0 | 0 | 1 | 1 | x | x | x | Выход код «00» |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0\*2+1=01 |
| 1 | 0 | 0 | 1 | 0 | 1 | x | x | x | Выход код «00» |
| 1 | 0 | 0 | 1 | 1 | 0 | x | x | x | 0\*3+1=01 |
| 1 | 0 | 0 | 1 | 1 | 1 | x | x | x | Выход код «00» |
| 1 | 0 | 1 | 0 | 0 | 0 | x | x | x | 1\*0+1=01 |
| 1 | 0 | 1 | 0 | 0 | 1 | x | x | x | Выход код «01» |
| 1 | 0 | 1 | 0 | 1 | 0 | x | x | x | 1\*1+1=02 |
| 1 | 0 | 1 | 0 | 1 | 1 | x | x | x | Выход код «01» |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1\*2+1=03 |
| 1 | 0 | 1 | 1 | 0 | 1 | x | x | x | Выход код «01» |
| 1 | 0 | 1 | 1 | 1 | 0 | x | x | x | 1\*3+1=10 |
| 1 | 0 | 1 | 1 | 1 | 1 | x | x | x | Выход код «01» |
| 1 | 1 | 0 | 0 | 0 | 0 | x | x | x | 2\*0+1=01 |
| 1 | 1 | 0 | 0 | 0 | 1 | x | x | x | Выход код «02» |
| 1 | 1 | 0 | 0 | 1 | 0 | x | x | x | 2\*1+1=03 |
| 1 | 1 | 0 | 0 | 1 | 1 | x | x | x | Выход код «02» |

*Продолжение таблицы 3.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 2\*2+1=11 |
| 1 | 1 | 0 | 1 | 0 | 1 | x | x | x | Выход код «02» |
| 1 | 1 | 0 | 1 | 1 | 0 | x | x | x | 2\*3+1=13 |
| 1 | 1 | 0 | 1 | 1 | 1 | x | x | x | Выход код «02» |
| 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 |
| 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» |

Минимизацию переключательный функций проведём с помощью карт Карно и Вейча и Алгоритма Рота. Для функции *П*заполненная карта приведена на рисунке 3.1.1. В рисунках 3.1.1 – 3.1.3 символом «х» отмечены безразличные наборы.



Рисунок 3.1.1 – Минимизация функции *П*при помощи карты Вейча

Следовательно:

Запишем результат минимизации в логическом базисе ИЛИ, НЕ:

## Для функции Q1 определим множество единичных кубов:

## L=

## и множество безразличных кубов:

## N=

##### В исходных множествах произвели склеивание кубов и получили:

## L=

## и множество безразличных кубов:

N=

## Первым этапом алгоритма Рота является нахождение множества простых импликант. Сформируем множество *С0* = *L* ⋃ *N*:

## Для реализации этого этапа будем использовать операцию умножения (\*) над множествами *С*0, *С*1 и т. Д.

## Первый шаг умножения (*С*0*\*С*0) приведён в таблице 3.1.1.

## Таблица 3.1.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| C0\*C0 | 01x010 | 01x101 | xx1100 | 01x0x1 | 1xx101 | xxx11x | 1xx0xx |
| 01x010 | - |  |  |  |  |  |  |
| 01x101 |  | - |  |  |  |  |  |
| xx1100 |  | 01110y | - |  |  |  |  |
| 01x0x1 | 01x01y | 01xy01 |  | - |  |  |  |

*Продолжение таблицы 3.1.1*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1xx101 |  | y1x101 | 1x110y |  | - |  |  |
| xxx11x | 01xy10 | 01x1y1 | xx11y0 | 01xy11 | 1xx1y1 | - |  |
| 1xx0xx | y1x010 |  | 1x1y00 | y1x0x1 | 1xxy01 | 1xxy1x | - |
| A1 | 01x01x 01xx10 x1x010 | 01110x 01xx01 x1x101 01x1x1 | 1x110x xx11x0 1x1x00 | 01xx11 x1x0x1 | 1xx1x1 1xxx01 | 1xxx1x | Ø |

## В результате этой операции сформируется новое множество кубов:

## А1=

## Множество *Z*0 кубов, не участвовавших в образовании новых кубов имеет вид:

## Z0=

## Также формируется множество В1 = С0 − Z0. Для следующего шага получения множества Z формируется множество С1 = А1 U В1.

## В1={01x010, 01x101, xx1100, 01x0x1, 1xx101, xxx11x, 1xx0xx}

## C1=

В таблице 3.1.2 приведён следующий шаг поиска простых импликант с помощью операции *С*1*\*С*1.

## В результате образовалось множество *А*2 кубов:

## A2 =

## Множество *Z*1 кубов, не участвовавших в образовании новых кубов имеет вид:

## Z1 = {Ø}

## Также формируется множество В2 = С1 – Z1. Для следующего шага получения множества Z формируется множество С2 = А2 U В2.

## B2=

## C2=

В таблице 3.1.3 приведён следующий шаг поиска простых импликант с помощью операции *С*2*\*С*2.

## В результате образовалось множество *А*3 кубов:

## A3 =

## Множество *Z*2 кубов, не участвовавших в образовании новых кубов:

## Z2 = {Ø}

## Также формируется множество В3 = С2 – Z2. Для следующего шага получения множества Z формируется множество С3 = А3 U В3.

## B3=

## C3=

В таблице 3.1.4 приведён следующий шаг поиска простых импликант с помощью операции *С*3*\*С*3.

## Таблица 3.1.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C3\*C3 | x1xx1x | x111xx | x1xxx1 | 1x1xxx |
| x1xx1x | - |  |  |  |
| x111xx |  | - |  |  |
| x1xxx1 |  |  | - |  |
| 1x1xxx |  |  |  | - |
| 1xxxx1 |  |  |  |  |
| xx11x0 |  |  |  |  |
| 1xxx1x |  |  |  |  |
| xxx11x |  |  |  |  |
| 1xx0xx |  |  |  |  |
| A4 | Ø | Ø | Ø | Ø |

## В результате образовалось множество А4 кубов:

## А4 = {Ø}

## Множество *Z*4 кубов, не участвовавших в образовании новых кубов имеет вид:

## Z3={x1xx1x,x111xx,x1xxx1,1x1xxx,1xxxx1,xx11x0,1xxx1x, xxx11x, 1xx0xx}

## Кубов новой размерности не образовалось. На этом процесс выявления простых импликант окончен. Таким образом сформировано множество простых импликант:

## Z=Z0 U Z1 U Z2 U Z3=

## Следующий этап – поиск *L*-экстремалей на множестве простых импликант (таблица 3.1.5). Для этого используется операция # (решётчатое вычитание).

Таким образом, из таблицы получено кубы, “подозрительные” на *L*-экстремальность. Проверяем их в таблице 3.1.6.

## E=

## Выясним далее, какие из вершин комплекса L не покрываются L-экстремалями. Для этого из каждого куба комплекса L вычтем (#) элементы множества Е (табл. 3.1.7). В результате вычитания получим L1 = L#Е.

## Таким образом получена минимальная тупиковая форма:

##### FminДНФ =

## Таблица 3.1.5

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | x1xx1x | x111xx | x1xxx1 | 1x1xxx | 1xxxx1 | xx11x0 | 1xxx1x | xxx11x | 1xx0xx |
| x1xx1x | - | zzzz0z  x1110x | zzzz0z  x1xx01 | z0zz0z  101xxx 1x1x0x | z0zz0z  10xxx1 1xxx01 | z0zz0z  x011x0 xx1100 | z0zzzz  10xx1x | z0zzzz  x0x11x | z0zz0z  10x0xx 1xx00x |
| x111xx | zz00zz  x10x1x x1x01x | - | zz00zz  x10x01 x1x001 | 101xxx  z0z0zz 101x0x 1x100x | 10xxx1  z000zz 10xx01 1x0x01 1xx001 | x011x0  z0zzzz x01100 | 10xx1x | x0x11x | 10x0xx 1xx00x |
| x1xxx1 | zzzzz0  x10x10  zzzzz0 x1x010 | zzzzz0  x11100 | - | 101xxx 101x0x  z0zzz0 10100x 1x1000 | 10xxx1 10xx01  z0zzzz 100x01  z0zzzz 10x001 | x011x0 x01100 | 10xx1x | x0x11x | 10x0xx  z0zzz0 10x00x 1xx000 |
| 1x1xxx | 0zyzzz  x10x10  0z0zzz 01x010 x10010 | 0zzzzz  011100 | x10x01  0z0zzz 01x001 x10001 | - | zz0zzz  100xx1  zz0zzz 100x01  zz0zzz 100001 | 0zzzzz  0011x0  0zzzzz 001100 | zz0zzz  100x1x | 0z0zzz  00x11x x0011x | zz0zzz  1000xx  zz0zzz 10000x  zz0zzz 1x0000 |
| 1xxxx1 | x10x10  01x010  x10010 | 011100 | 010x01 01x001  0zzzzz 010001 | zzzzz0  101xx0  zzzzz0 101x00  zzzzz0 101000 1x1000 | - | 0011x0 001100 | zzzzz0  100x10 | 00x11x  0zzzz0 00011x x00110 | zzzzz0  1000x0  zzzzz0 100000 1x0000 |
| xx11x0 | x10x10 01x010 x10010 | zzzzzz  Ø | 010x01 01x001 010001 | zzz0zz  1010x0  zzz000 101000 101000 1x1000 | 100xx1 100x01 100x01 100001 | - | 100x10 | zz0zz1  00011x 00x111 00011x x00110 | 1000x0 100000 1x0000 |
| 1xxx1x | 010x10 01x010  0zzzzz 010010 | Ø | 010x01 01x001 010001 | zzzz0z  101000 101000 101000 1x1000 | zzzz0z  100x01 100x01 100x01 100001 | 0011x0 001100 | - | 00011x 00x111 00011x  0zzzzz 000110 | zzzz0z  100000 100000 1x0000 |
| xxx11x | zzz0zz  010010 01x010 010010 | Ø | 010x01 01x001 010001 | 101000 101000 101000 1x1000 | 100x01 100x01 100x01 100001 | zzzz0z  001100 001100 | zzz0zz  100010 | - | 100000 100000 1x0000 |
| 1xx0xx | 010010 01x010 010010 | Ø | 010x01 01x001 010001 | zzzzzz  Ø | zzz1zz  100101  zzz1zz 100101  zzz1zz 100101 | 001100 001100 | zzzzzz  Ø | 00011x 00x111 00011x 000110 | - |
| Остаток | 010010 01x010 010010 | Ø | 010x01 01x001 010001 | Ø | 100101 100101 100101 | 001100 001100 | Ø | 00011x 00x111 00011x 000110 | 100000 100000 1x0000 |

Проверим результаты минимизации картой Карно. Для функции *Q1* заполненная карта приведена на рисунке 3.1.2.



Рисунок 3.1.2 - Минимизация функции *Q1* при помощи карты Карно

Следовательно:

*Q1* =x1y2+x1h+x2y1.

Запишем результат минимизации в логическом базисе ИЛИ, НЕ:

Для функции Q*2* заполненная карта приведена на рисунке 3.1.3.



Рисунок 3.1.3 – Минимизация функции *Q2* при помощи карт Карно

Следовательно:

Q2 = P+x2y2+x2h

Запишем результат минимизации в логическом базисе ИЛИ, НЕ:

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

*=*

*=*

*=*

Схема электрическая функциональная ОЧУС представлена в ПРИЛОЖЕНИИ Б**.**

**3.2. Логический синтез одноразрядного четверичного сумматора**

Одноразрядный четверичный сумматор – это комбинационное устройство, имеющее 5 двоичных входов и 3 двоичных выхода.

Принцип работы ОЧС представлен с помощью таблицы истинности (таблица 3.2)

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

Так как ОЧС синтезируется для схемы второго типа, то безразличные наборы в таблице истинности отсутствуют.

Таблица 3.2 – Таблица истинности ОЧС

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a1** | **a2** | **b1** | **b2** | **p** | **П** | **S1** | **S2** | **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 | 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 | 0 | 0+1+1=02 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0+2+0=02 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0+2+1=03 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0+3+0=03 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0+3+1=10 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1+0+0=01 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1+0+1=02 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1+1+0=02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1+1+1=03 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1+2+0=03 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1+2+1=10 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1+3+0=10 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1+3+1=11 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2+0+0=02 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 2+0+1=03 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2+1+0=03 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2+1+1=10 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2+2+0=10 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 2+2+1=11 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 2+3+0=11 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2+3+1=12 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 3+0+0=03 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 3+0+1=10 |

*Продолжение таблицы 3.2*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 3+1+0=10 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 3+1+1=11 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 3+2+0=11 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 3+2+1=12 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 3+3+0=12 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3+3+1=13 |

Минимизацию переключательный функций проведём с помощью карт Карно и Вейча. Для функции *П*заполненная карта приведена на рисунке 3.2.1. В рисунках 3.2.1 – 3.2.3 символом «х» отмечены безразличные наборы.



Рисунок 3.2.1 – Минимизация функции *П*при помощи карты Вейча

Следовательно:

П=a1b1+ a1b1b2+a2b1p+b1b2p+a1a2b1+a1a2p+a1b2p

Запишем результат минимизации в логическом базисе И, логическое ИЛИ, генератор «1»:

Для функции S1заполненная карта приведена на рисунке 3.2.2.



Рисунок 3.2.2 – Минимизация функции *S1* при помощи карт Карно

Следовательно:

S1 =

Запишем результат минимизации в логическом базисе И, логическое ИЛИ, генератор «1»:

Для функции S2заполненная карта приведена на рисунке 3.2.3.



Рисунок 3.2.3 – Минимизация функции *S2* при помощи карт Карно

Следовательно:

S2 =

Запишем результат минимизации в логическом базисе И, логическое ИЛИ, генератор «1»:

S2 =

)

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

*=*

*=*

*=*

Схема электрическая функциональная ОЧС представлена 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 диаду 11 (34) на триаду и диаду 10 (24) на триаду .

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

Таблица 3.3.1 – Таблица истинности ПМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Qn | Qn-1 | Qn-2 | P | 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 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

Следовательно:

Минимизацию функции S2 проведём с помощью карты Карно.



Рисунок 3.3 – Минимизация функцииS2при помощи карты Карно

Следовательно:

=

Эффективность минимизации по количеству входов логических элементов:

*=*

Схема электрическая функциональная ПМ представлена в ПРИЛОЖЕНИИ Д**.**

**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 = 2m.

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.

Таблица 4– таблица истинности ОЧС на базе мультиплексора

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

*Продолжение таблицы 4*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 1 | 1 | 0 | 0 | 0 | 0 |  | 1 |  | 1 |  |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |  | 1 |  |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Функциональная схема ОЧС на базе мультиплексоров в ПРИЛОЖЕНИИ Г**.**

**5 оценка результатов разработки**

Формула расчёта временных затрат на умножение:

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

𝑇ПМ – время преобразования множителя;

𝑇ФДК – время формирования дополнительного кода множимого;

𝑇ОЧУC – время умножения на ОЧУC;

𝑇ОЧС – время формирования единицы переноса в ОЧС;

𝑇сдвига – время сдвига частичной суммы;

n – количество разрядов множителя.

m – количество разрядов множимого

**Заключение**

В процессе выполнения курсовой работы была разработана структурная схема сумматора-умножителя первого 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. Метод Квайна-Мак-Класки совместил в себе одновременно простоту минимизации и её эффективность. Однако проведение минимизации функции данным методом будет очень долгим, если эта функция принимает единичные и безразличные значения на большом количестве наборов переменных.

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**

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

Луцик, Ю. А. Учебное пособие по курсу «Арифметические и логические основы вычислительной техники» / Ю. А. Луцик, И. В. Лукьянова, М. П. Ожигина. – Минск : МРТИ, 2001. – 77 с.

**Приложение А**

*(обязательное)*

Сумматор-умножитель второго типа. Схема электрическая структурная

**Приложение Б**

*(обязательное)*

Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная

**Приложение В**

*(обязательное)*

Одноразрядный четверичный сумматор. Схема электрическая функциональная

**Приложение Г**

*(обязательное)*

Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная

**Приложение Д**

*(обязательное)*

Преобразователь множителя.

Схема электрическая функциональная

**ПРиложение Е**

*(обязательное)*

*Ведомость документов*