Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы вычислительной техники

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_\_\_\_\_\_\_\_\_Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе   
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 120 ПЗ

Студент

Руководитель

И.В. Никанов

Ю. А. Луцик

МИНСК 2017

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_\_\_ Д. И. Самаль

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2017 г.

ЗАДАНИЕ

по курсовой работе студента  
Никанова Ивана Владимировича

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-четвертичных чисел».
2. Срок сдачи студентом законченной работы: 1 июня 2017 г.
3. Исходные данные к работе:
   1. исходные сомножители: Мн = 57,65; Мт = 37,83;
   2. алгоритм умножения: Г;
   3. метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;
   4. коды четверичных цифр множимого для перехода к двоично-четвертичной системе кодирования; 04 – 10, 14 – 01, 24 – 00, 34 – 11;
   5. тип синтезируемого умножителя: 2;
   6. логический базис для реализации ОЧС: И, mod2, const1; метод минимизации – алгоритм Рота;
   7. логический базис для реализации ОЧУС: ИЛИ-НЕ; метод минимизации – карты Карно- Вейча.
4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

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 первого типа. Схема электрическая структурная.
   2. Одноразрядный четвертичный умножитель-сумматор. Схема электрическая функциональная.
   3. Одноразрядный четвертичный сумматор. Схема электрическая функциональная.
   4. Регистр-аккумулятор. Схема электрическая функциональная.
   5. Одноразрядный четвертичный 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 | 15.05-21.05 |  |

Дата выдачи задания: 10 февраля 2017 г.

Руководитель

Ю. А. Луцик

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

Оглавление

[Введение 5](#_Toc482605517)

[1 Разработка алгоритма умножения 6](#_Toc482605518)

[2 Разработка структурной схемы сумматора-умножителя 9](#_Toc482605519)

[5 Логический синтез преобразователя множителя (ПМ) 34](#_Toc482605520)

[6 Оценка результатов разработки 35](#_Toc482605521)

[Заключение 36](#_Toc482605522)

[Приложение А 38](#_Toc482605523)

[Приложение Б 39](#_Toc482605524)

[Приложение В 40](#_Toc482605525)

[Приложение Г 41](#_Toc482605526)

[Приложение Д 42](#_Toc482605527)

# **Введение**

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

Данная курсовая работа посвящена разработке сумматора-умножителя второго типа для алгоритма умножения Г в прямом коде на два разряда одновременно. Это и является целью данной работы.

Для достижения цели необходимо решить ряд задач:

Разработать алгоритм умножения и оценить погрешности вычислений.

Разработать структурную схему сумматора-умножителя второго типа.

Разработать функциональные схемы основных узлов сумматора-умножителя в заданных логических базисах.

Разработать комбинационную схему на основе мультиплексора.

Рассчитать время умножения.

# **разработка алгоритма умножения**

1. Перевод сомножителей из десятичной системы счисления в четвертичную (мантисса должна содержать шесть четвертичных разрядов).

**Множимое**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \_57  4 | 4 | | | | |
| \_14  12 | 4 | | |  |
| \_17  16  1 | 3 |  | |
| 2 |  |  |

|  |  |
| --- | --- |
| \* | 0,65 |
| 4 |
| \* | 2,60 |
| 4 |
| \* | 2,40 |
| 4 |
| \* | 1,60 |
| 4 |
|  | 2,40 |

Мн4 =321,2212.

В соответствии с заданной кодировкой множимого:

Мн2/4 =110001,00000100.

**Множитель**

|  |  |
| --- | --- |
| \* | 0,83 |
| 4 |
| \* | 3,32 |
| 4 |
| \* | 1,28 |
| 4 |
| \* | 1,12 |
| 4 |
|  | 0,48 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \_37  36 | 4 | | | | |
| \_ 9  8 | 4 | | |  |
| 1 | 2 |  | |
| 1 |  |  |
|  |



Мт4=211,3110.

В соответствии с обычной весомозначной кодировкой множителя:

Мт2/4 =100101,11010100.

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

Мн = 0, 11000100000100

Мт = 0, 10010111010100

РМн =0.0011 +034

РМт = 0.0011 +034

-закодировано по заданию,

-закодировано традиционно.

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:

РМн  = 0.0011 034

РМт = 0.0011 034

РМн·Мт = 0.0100 124

Результат закодирован в соответствии с заданием на кодировку множимого.

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

знМн знМт = 0 0 = 0.

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в прямых кодах диада 11(34) заменяется на триаду (). Преобразованный множитель имеет вид или Перемножение мантисс по алгоритму «Б» приведено в таблице 1.1.

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в дополнительных кодах диада 11(34) заменяется на триаду (), диада 10(24) заменяется на триаду 11̅0(). 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/с** | | | **Комментарии** |
| **1** | | | **2** | | | **3** |
| 0.  0.  0.  3.  0.  0.  0.  0.  0.  3.  0.  0.  0.  0.  0. | 000000  032122  032122  320302  013031  000321  020012  000130  020203  333330  020133  000000  020200  000000  020201 | 0000000  1000000  1000000  3200000  0200000  2210000  3010000  3102000  2112000  1211300  3303300  3212210  3202110  0321221  0123331 | 0.  0.  0.  1.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0. | 10 10 10 10 10 10  10 11 00 01 00 00  10 11 00 01 00 00  11 00 10 11 10 00  10 01 11 10 11 01  10 10 10 11 00 01  10 00 10 10 01 00  10 10 10 01 11 10  10 00 10 00 10 11  11 11 11 11 11 10  10 00 10 01 11 11  10 10 10 10 10 10  10 00 10 00 10 10  10 10 10 10 10 10  10 00 10 00 10 01 | 10 10 10 10 10 10 10  01 10 10 10 10 10 10  01 10 10 10 10 10 10  11 00 10 10 10 10 10  10 00 10 10 10 10 10  00 00 01 10 10 10 10  11 10 01 10 10 10 10  11 01 10 00 10 10 10  00 01 01 00 10 10 10  01 00 01 01 11 10 10  11 11 10 11 11 10 10  11 00 01 00 00 01 10  11 00 10 00 01 01 10  10 11 00 01 00 00 01  10 01 00 11 11 11 01 | = Мн  -2Мн  Мн  2Мн  -1Мн  Мн  Мн |

После окончания умножения необходимо оценить погрешность вычислений.

Для этого полученное произведение (Мн ∙ Мт4 = 0,202010123331 , РМн ∙ Мт =6) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн ∙ Мт4 = 202010,123331, РМн ∙ Мт = 0;

Мн ∙ Мт10 = 2180,4368.

Результат прямого перемножения операндов даёт следующее значение:

Мн10 ∙ Мт10 = 2180,8995.

Абсолютная погрешность:

.

Относительная погрешность:

.

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

# **разработка структурной схемы сумматора-умножителя**

Структурная схема сумматора-умножителя второго типа для алгоритма умножения «Г» приведена на рисунке приложения А.

Если устройство работает как сумматор, то оба слагаемых последовательно заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода поступает «1». 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.

Если устройство работает как умножитель, то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК поступает «0». Диада множителя поступает на входы преобразователя множителя. Единица переноса в следующую диаду, если она возникает, должна быть добавлена к следующей диаде множителя.

В регистре множителя после каждого такта умножения содержимое сдвигается на два двоичных разряда, и в конце умножения регистр обнуляется. В случае когда выход 2ПМ переходит в единичное состояние инициализируется управляющий вход F1 формирователя дополнительного кода, и на выходах ФДК формируется дополнительный код множимого с обратным знаком. 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 Мн•Мт плюс перенос из предыдущего ОЧУС. Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому перенос, формируемый ОЧУС, может быть только двоичным («0» или «1»). Частичные произведения, получаемые на выходах ОЧУС, складываются с накапливаемой частичной суммой из регистра результата с помощью цепочки. Количество тактов умножения определяется разрядностью Мт.

1. **РАЗАРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ**

**УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ**

# 

* 1. **Логический синтез одноразрядного четвертичного умножителя-сумматора.**

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 – 10, 1 – 01, 2 – 00, 3 – 11.

Управляющий вход h определяет тип операции:

«0» – умножение закодированных цифр, поступивших на информационные входы;

«1» – вывод на выходы без изменения значения разрядов, поступивших из регистра множимого.

Таблица 3.1

| **Пер.** | **Мн** | | **Мт** | | **Упр.** | **Перенос** | **Результат** | | **Результаты вычисления в четверичной с/с** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Р1*** | ***x1*** | ***x2*** | ***y1*** | ***y2*** | ***h*** | ***P*** | ***Q1*** | ***Q2*** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2\*0+0=00 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Выход 02 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2\*1+0=02 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Выход 02 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 2\*2+0=10 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | Выход 02 |
| 0 | 0 | 0 | 1 | 1 | 0 | X | X | Х | 2\*3+0=12 |
| 0 | 0 | 0 | 1 | 1 | 1 | X | X | Х | Выход 02 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1\*0+0=00 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Выход 01 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1\*1+0=01 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | Выход 01 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1\*2+0=02 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Выход 01 |
| 0 | 0 | 1 | 1 | 1 | 0 | X | X | Х | 1\*3+0=21 |
| 0 | 0 | 1 | 1 | 1 | 1 | X | X | Х | Выход 01 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0\*0+0=00 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | Выход 00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0\*1+0=00 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Выход 00 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0\*2+0=00 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Выход 00 |
| 0 | 1 | 0 | 1 | 1 | 0 | X | X | Х | 0\*3+0=02 |
| 0 | 1 | 0 | 1 | 1 | 1 | X | X | Х | Выход 00 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 3\*0+0=00 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Выход 03 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 3\*1+0=03 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | Выход 03 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 3\*2+0=12 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | Выход 03 |
| 0 | 1 | 1 | 1 | 1 | 0 | X | X | Х | 3\*3+0=21 |
| 0 | 1 | 1 | 1 | 1 | 1 | X | X | Х | Выход 03 |
| 1 | 0 | 0 | 0 | 0 | 0 | Х | Х | Х | 2\*0+1=01 |
| 1 | 0 | 0 | 0 | 0 | 1 | Х | Х | Х | Выход 00 |
| 1 | 0 | 0 | 0 | 1 | 0 | Х | Х | Х | 2\*1+1=03 |
| 1 | 0 | 0 | 0 | 1 | 1 | Х | Х | Х | Выход 00 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 2\*2+1=11 |
| 1 | 0 | 0 | 1 | 0 | 1 | Х | Х | Х | Выход 00 |
| 1 | 0 | 0 | 1 | 1 | 0 | X | X | Х | 2\*3+1=13 |
| 1 | 0 | 0 | 1 | 1 | 1 | X | X | Х | Выход 00 |
| 1 | 0 | 1 | 0 | 0 | 0 | Х | Х | Х | 1\*0+1=01 |
| 1 | 0 | 1 | 0 | 0 | 1 | Х | Х | Х | Выход 01 |
| 1 | 0 | 1 | 0 | 1 | 0 | Х | Х | Х | 1\*1+1=02 |
| 1 | 0 | 1 | 0 | 1 | 1 | Х | Х | Х | Выход 01 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1\*2+1=03 |
| 1 | 0 | 1 | 1 | 0 | 1 | Х | Х | Х | Выход 01 |
| 1 | 0 | 1 | 1 | 1 | 0 | X | X | Х | 1\*3+1=10 |
| 1 | 0 | 1 | 1 | 1 | 1 | X | X | Х | Выход 00 |
| 1 | 1 | 0 | 0 | 0 | 0 | Х | Х | Х | 0\*0+1=01 |
| 1 | 1 | 0 | 0 | 0 | 1 | Х | Х | Х | Выход 00 |
| 1 | 1 | 0 | 0 | 1 | 0 | Х | Х | Х | 0\*1+1=03 |
| 1 | 1 | 0 | 0 | 1 | 1 | Х | Х | Х | Выход 00 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0\*2+1=01 |
| 1 | 1 | 0 | 1 | 0 | 1 | Х | Х | Х | Выход 02 |
| 1 | 1 | 0 | 1 | 1 | 0 | X | X | Х | 0\*3+1=01 |
| 1 | 1 | 0 | 1 | 1 | 1 | X | X | Х | Выход 00 |
| 1 | 1 | 1 | 0 | 0 | 0 | Х | Х | Х | 3\*0+1=01 |
| 1 | 1 | 1 | 0 | 0 | 1 | Х | Х | Х | Выход 03 |
| 1 | 1 | 1 | 0 | 1 | 0 | Х | Х | Х | 3\*1+1=10 |
| 1 | 1 | 1 | 0 | 1 | 1 | Х | Х | Х | Выход 03 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 3\*2+1=13 |
| 1 | 1 | 1 | 1 | 0 | 1 | Х | Х | Х | Выход 03 |
| 1 | 1 | 1 | 1 | 1 | 0 | X | X | Х | 3\*3+1=22 |
| 1 | 1 | 1 | 1 | 1 | 1 | X | X | Х | Выход 03 |

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.

Минимизацию функций P,Q1 и Q2  проведем при помощи карт Вейча/

Для функции P :

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

h h

Для функции Q1 :

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

h h

Для функции Q2 :

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

h h

Следовательно .

.

.

Функциональная схема ОЧУC в заданном базисе представлена в приложении Б.

* 1. **Логический синтез одноразрядного четвертичного сумматора**

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

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

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

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

*Таблица 3.2*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***a1*** | ***a2*** | ***b1*** | ***b2*** | ***p*** | **П** | ***S1*** | ***S2*** | **Операции**  **в четверичной с/с** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2+2+0=10 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 2+2+1=11 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2+1+0=03 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 2+1+1=10 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2+0+0=02 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 2+0+1=03 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 2+3+0=11 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2+3+1=12 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1+2+0=03 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1+2+1=10 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1+1+0=02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1+1+1=03 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1+0+0=01 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1+0+1=02 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1+3+0=10 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1+3+1=11 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0+2+0=02 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0+2+1=03 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0+1+0=01 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0+1+1=02 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0+0+0=00 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0+0+1=01 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0+3+0=03 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0+3+1=10 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 3+2+0=11 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 3+2+1=12 |

*Продолжение таблицы 3.2*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 3+1+0=10 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 3+1+1=11 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 3+0+0=03 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 3+0+1=10 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 3+3+0=12 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3+3+1=13 |

*.*

Минимизацию переключательных функций *П* и *S2* проведём с помощью алгоритма Рота, а функции *S1* с помощью карты Вейча.

Для функции :

определим множество единичных кубов:

множество безразличных кубов:

N=

Сформируем множество :

Первый шаг умножения () приведён в таблице 3.2

*Таблица 3.2*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 00000 | 00001 | 00011 | 00110 | 00111 | 01001 | 01110 | 01111 | 10111 | 11000 | 11001 | 11010 | 11011 | 11101 | 11110 | 11111 |
| 00000 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00001 | 0000y | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00011 |  | 000y1 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00110 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 00111 |  |  | 00y11 | 0011y | - |  |  |  |  |  |  |  |  |  |  |  |
| 01001 |  | 0y001 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 01110 |  |  |  | 0y110 |  |  | - |  |  |  |  |  |  |  |  |  |
| 01111 |  |  |  |  | 0y111 |  | 0111y | - |  |  |  |  |  |  |  |  |
| 10111 |  |  |  |  | y0111 |  |  |  | - |  |  |  |  |  |  |  |
| 11000 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 11001 |  |  |  |  |  | y1001 |  |  |  | 1100y | - |  |  |  |  |  |
| 11010 |  |  |  |  |  |  |  |  |  | 110y0 |  | - |  |  |  |  |
| 11011 |  |  |  |  |  |  |  |  |  |  | 110y1 | 1101y | - |  |  |  |
| 11101 |  |  |  |  |  |  |  |  |  |  | 11y01 |  |  | - |  |  |
| 11110 |  |  |  |  |  |  | y1110 |  |  |  |  | 11y10 |  |  | - |  |
| 11111 |  |  |  |  |  |  |  | y1111 | 1y111 |  |  |  | 11y11 | 111y1 | 1111y | - |
| A1 | 0000x | 000x1 0x001 | 00x11 | 0011x 0x110 | 0x111 x0111 | x1001 | 0111x x1110 | x1111 | 1x111 | 1100x 110x0 | 110x1 11x01 | 1101x 11x10 | 11x11 | 111x1 | 1111x | Ø |

В результате этой операции сформируется новое множество кубов:

##### 0000x; 000x1; 0x001; 00x11; 0011x; 0x110; 0x111; x0111; x1001; 0111x; x1110; x1111; 1x111; 1100x; 110x0; 110x1; 11x01; 1101x; 11x10; 11x11; 111x1; 1111x

Множество *Z0* пустое.

В таблице 3.3 приведён следующий шаг поиска простых импликант с помощью операции .

*Таблица 3.3*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0000x | 000x1 | 0x001 | 00x11 | 0011x | 0x110 | 0x111 | x0111 | x1001 | 0111x | x1110 | x1111 | 1x111 | 1100x | 110x0 | 110x1 | 11x01 | 1101x | 11x10 | 11x11 | 111x1 | 1111x |
| 0000x | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 000x1 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x001 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x11 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0011x |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x110 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x111 |  |  |  |  |  | 0x11y | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0111 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x1001 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0111x |  |  |  |  | 0y11x |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| x1110 |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| x1111 |  |  |  |  |  |  |  | xy111 |  |  | x111y | - |  |  |  |  |  |  |  |  |  |  |
| 1x111 |  |  |  |  |  |  | yx111 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |
| 1100x |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 110x0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| 110x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 110xy | - |  |  |  |  |  |  |
| 11x01 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 1101x |  |  |  |  |  |  |  |  |  |  |  |  |  | 110yx |  |  |  | - |  |  |  |  |
| 11x10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |
| 11x11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 11xy1 |  | 11x1y | - |  |  |
| 111x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 11yx1 |  |  |  |  | - |  |
| 1111x |  |  |  |  |  |  |  |  |  | y111x |  |  |  |  |  |  |  | 11y1x |  |  |  | - |
| A2 | Ø | Ø | Ø | Ø | 0x11x | 0x11x | xx111 | xx111 | Ø | x111x | x111x | Ø | Ø | 110xx | 110xx | 11xx1 | 11xx1 | 11x1x | 11x1x | Ø | Ø | Ø |

В результате этой операции сформируется новое множество кубов:

##### = {0x11x; xx111; x111x; 110xx; 11xx1; 11x1x}.

##### Множество = {0000x; 000x1; 0x001; 00x11; x1001}.

В таблице 3.4 приведён следующий шаг поиска простых импликант с помощью операции .

*Таблица 3.4*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0x11x | xx111 | x111x | 110xx | 11xx1 | 11x1x |
| 0x11x | - |  |  |  |  |  |
| xx111 |  | - |  |  |  |  |
| x111x |  |  | - |  |  |  |
| 110xx |  |  |  | - |  |  |
| 11xx1 |  |  |  |  | - |  |
| 11x1x |  |  |  |  |  | - |
| A3 | Ø | Ø | Ø | Ø | Ø | Ø |

На этом поиск простых импликант заканчивается, кубы из множества С2 являются кубами множества Z2.

##### Конечное множество простых импликант:

##### Z =  {0000x; 000x1; 0x001; 00x11; x1001; 0x11x; xx111; x111x; 110xx; 11xx1; 11x1x}.

Следующий этап – поиск *L-*экстремалей на множестве простых импликант (таблица 3.5). Для этого используется операция # (решетчатое вычитание).

*Таблица 3.5 – Поиск L-экстремалей*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 0000x | 000x1 | 0x001 | 00x11 | x1001 | 0x11x | xx111 | x111x | 110xx | 11xx1 | 11x1x |
| 0000x | - | zzz1z  00011 | z1zzz  01001 | zz1yz  00x11 | 1yzzz  x1001 | z1yyz  0x11x | 11yyz  xx111 | 1yyyz  x111x | yyz1z  110xx | yy11z  11xx1 | yy1yz  11x1x |
| 000x1 | zzzz0  00000 | - | zyzzz  01001 | zz1zz  00111 | yyzzz  x1001 | z1yz0  0x11x | 11yzy  xx111 | 1yyz0  x111x | yyzz0  110xx | yy1zz  11xx1 | yy1y0  11x1x |
| 0x001 | zzzzy  00000 | zzzyz  00011 | - | zzyyz  00111 | yzzzz  11001 | zzyyz  0x11x | 1zyyz  xx111 | 1zyy0  x111x | yzz10  110xx | yz11z  11xx1 | yz1y0  11x1x |
| 00x11 | zzzyy  00000 | Ø | zyzyz  01001 | - | yyzyz  11001 | z1zz0  0111x 0x110 | 11zzz  1x111 x1111 | 1yzz0  x111x | yyz00  110xx | yyz0z  11xx1 | yyzz0  11x1x |
| x1001 | zyzzy  00000 | Ø | Ø | zyyyz  00111 | - | zzyy0 z0yy0  0111x 0x110 | z0yyz zzyyz  1x111 x1111 | zzyy0  x111x | zzz10  1101x 110x0 | zz11z  111x1 11x11 | zz1y0  11x1x |
| 0x11x | zzyyz  00000 | Ø | Ø | Ø | yzyyz  11001 | - | zyyzz 1zzzz  1x111 11111 | 1zzzz  1111x | zzzyz zyyyz  1101x 110x0 | zzyyz yyzzz  111x1 11x11 | yz0zz  11x1x |
| xx111 | zzyyy  00000 | Ø | Ø | Ø | zzyyz  11001 | zzzz0 zzzzy  01110 0x110 | - | zzzz0  11110 | zzyyy zzyzz  1101x 110x0 | zzz0z zz0zz  11101 11011 | zz0z0  1101x 11x10 |
| x111x | zyyyz  00000 | Ø | Ø | Ø | zzyyz  11001 | z0zzz  00110 | z0zzz  10111 | - | zzzyy yyyzz  1101x 110x0 | zzyyy yyyzz  11101 11011 | zzzyz zz0zz  1101x 11010 |
| 110xx | yyzzz  00000 | Ø | Ø | Ø | Ø | yyyzz  00110 | zyyzz  10111 | zzyzz  11110 | - | zzyzz  11101 | Ø |
| 11xx1 | yyzzy  00000 | Ø | Ø | Ø | Ø | yyzzy  00110 | zyzzz  10111 | zzzzy  11110 | zzzz0 zzzzy  11010 110x0 | - | Ø |
| 11x1x | yyzyz  00000 | Ø | Ø | Ø | Ø | yyzzz  00110 | zyzzz  10111 | Ø | zzzz0z  11000 | zzzyz  11101 | - |
| Остаток | 00000 | Ø | Ø | Ø | Ø | 00110 | 10111 | Ø | 11000 | 11101 | Ø |

*L*-экстремалями стали следующие простые импликанты:

##### E = {0000x; 0x11x; xx111; 110xx; 11xx1}

Проверка *L*-экстремалей.

Находим множество кубов, не покрываемых L-экстремалями (Таблица 3.6). Для этого используется операция # (решетчатое вычитание).

*Таблица 3.6*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00000 | 00001 | 00011 | 00110 | 00111 | 01001 | 01110 | 01111 | 10111 | 11000 | 11001 | 11010 | 11011 | 11101 | 11110 | 11111 |
| 0000x | zzzzz  Ø | zzzzz  Ø | zzzyz  00011 | zzyyz  00110 | zzyyz  00111 | zyzzz  01001 | zyyyz  01110 | zyyyz  01111 | yzyyz  10111 | yyzzz  11000 | yyzzz  11001 | yyzyz  11010 | yyzyz  11011 | yyyzz  11101 | yyyyz  11110 | yyyyz  11111 |
| 0x11x | Ø | Ø | zzyzz  00011 | zzzzz  Ø | zzzzz  Ø | zzyyz  01001 | zzzzz  Ø | zzzzz  Ø | yzzzz  10111 | yzyyz  11000 | yzyyz  11001 | yzyzz  11010 | yzyzz  11011 | yzzyz  11101 | yzzzz  11110 | yzzzz  11111 |
| xx111 | Ø | Ø | zzyzz  00011 | Ø | Ø | zzyyz  01001 | Ø | Ø | zzzzz  Ø | zzyyy  11000 | zzyyz  11001 | zzyzy  11010 | zzyzz  11011 | zzzyz  11101 | zzzzy  11110 | zzzzz  Ø |
| 110xx | Ø | Ø | yyzzz  00011 | Ø | Ø | zyzzz  01001 | Ø | Ø | Ø | zzzzz  Ø | zzzzz  Ø | zzzzz  Ø | zzzzz  Ø | zzyzz  11101 | zzyzz  11110 | Ø |
| 11xx1 | Ø | Ø | yyzzz  00011 | Ø | Ø | yzzzz  01001 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | zzzzz  Ø | zzzzy  11110 | Ø |
| Остаток | Ø | Ø | 00011 | Ø | Ø | 01001 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11110 | Ø |

Следовательно множество кубов, непокрываемых L-экстремалями,

L':

##### L´ = L # E = {00011; 01001; 11110}

Теперь ищем кубы, которые при минимальных затратах максимально реализуют функцию.

Множество

кубов *L.*

##### = Z\E = {000x1; 0x001; 00x11; x1001; x111x; 11x1x}

*Таблица 3.7*

|  |  |  |  |
| --- | --- | --- | --- |
|  | 00011 | 01001 | 11110 |
| 000x1 | 00011 | Ø | Ø |
| 0x001 | Ø | 01001 | Ø |
| 00x11 | 00011 | Ø | Ø |
| x1001 | Ø | 01001 | Ø |
| x111x | Ø | Ø | 11110 |
| 11x1x | Ø | Ø | 11110 |

##### В результате могут быть получены следующие тупиковые формы:

##### Fmin1 = {000x1; 0x001; x111x; 0000x; 0x11x; xx111; 110xx; 11xx1}

##### Fmin2 = {0x001; 00x11; x111x; 0000x; 0x11x; xx111; 110xx; 11xx1}

##### Fmin3 = {000x1; x1001; x111x; 0000x; 0x11x; xx111; 110xx; 11xx1}

##### Fmin4 = {00x11; x1001; x111x; 0000x; 0x11x; xx111; 110xx; 11xx1}

##### Fmin5 = {000x1; 0x001; 11x1x; 0000x; 0x11x; xx111; 110xx; 11xx1}

##### Fmin6 = {0x001; 00x11; 11x1x; 0000x; 0x11x; xx111; 110xx; 11xx1}

##### Fmin7 = {000x1; x1001; 11x1x; 0000x; 0x11x; xx111; 110xx; 11xx1}

##### Fmin8 = {00x11; x1001; 11x1x; 0000x; 0x11x; xx111; 110xx; 11xx1}

Для функции :

определим множество единичных кубов:

множество безразличных кубов:

N=

Сформируем множество :

Первый шаг умножения () приведён в таблице 3.8

*Таблица 3.8*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C0\*C0 | 00001 | 00010 | 00101 | 00110 | 01000 | 01011 | 01100 | 01111 | 10001 | 10010 | 10101 | 10110 | 11000 | 11011 | 11100 | 11111 |
| 00001 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00010 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00101 | 00y01 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00110 |  | 00y10 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 01000 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 01011 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 01100 |  |  |  |  | 01y00 |  | - |  |  |  |  |  |  |  |  |  |
| 01111 |  |  |  |  |  | 01y11 |  | - |  |  |  |  |  |  |  |  |
| 10001 | y0001 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| 10010 |  | y0010 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 10101 |  |  | y0101 |  |  |  |  |  | 10y01 |  | - |  |  |  |  |  |
| 10110 |  |  |  | y0110 |  |  |  |  |  | 10y10 |  | - |  |  |  |  |
| 11000 |  |  |  |  | y1000 |  |  |  |  |  |  |  | - |  |  |  |
| 11011 |  |  |  |  |  | y1011 |  |  |  |  |  |  |  | - |  |  |
| 11100 |  |  |  |  |  |  | y1100 |  |  |  |  |  | 11y00 |  | - |  |
| 11111 |  |  |  |  |  |  |  | y1111 |  |  |  |  |  | 11y11 |  | - |
| A1 | 00x01 x0001 | 00x10 x0010 | x0101 | x0110 | 01x00 x1000 | 01x11 x1011 | x1100 | x1111 | 10x01 | 10x10 | Ø | Ø | 11x00 | 11x11 | Ø | Ø |

В результате этой операции сформируется новое множество кубов:

##### 00x01; x0001; 00x10; x0010; x0101; x0110; 01x00; x1000; 01x11; x1011; x1100; x1111; 10x01; 10x10; 11x00; 11x11

Множество *Z0* пустое.

В таблице 3.9 приведён следующий шаг поиска простых импликант с помощью операции .

*Таблица 3.9*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 00x01 | x0001 | 00x10 | x0010 | x0101 | x0110 | 01x00 | x1000 | 01x11 | x1011 | x1100 | x1111 | 10x01 | 10x10 | 11x00 | 11x11 |
| 00x01 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0001 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x10 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0010 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| x0101 |  | x0y01 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| x0110 |  |  |  | x0y10 |  | - |  |  |  |  |  |  |  |  |  |  |
| 01x00 |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |
| x1000 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 01x11 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| x1011 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| x1100 |  |  |  |  |  |  |  | x1y00 |  |  | - |  |  |  |  |  |
| x1111 |  |  |  |  |  |  |  |  |  | x1y11 |  | - |  |  |  |  |
| 10x01 | y0x01 |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |
| 10x10 |  |  | y0x10 |  |  |  |  |  |  |  |  |  |  | - |  |  |
| 11x00 |  |  |  |  |  |  | y1x00 |  |  |  |  |  |  |  | - |  |
| 11x11 |  |  |  |  |  |  |  |  | y1x11 |  |  |  |  |  |  | - |
| A2 | x0x01 | x0x01 | x0x10 | x0x10 | Ø | Ø | x1x00 | x1x00 | x1x11 | x1x11 | Ø | Ø | Ø | Ø | Ø | Ø |

В результате этой операции сформируется новое множество кубов:

##### = {x0x01; x0x10; x1x00; x1x1}.

##### Множество пустое.

В таблице 3.10 приведён следующий шаг поиска простых импликант с помощью операции .

*Таблица 3.10*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| . | x0x01 | x0x10 | x1x00 | x1x11 |
| x0x01 | - |  |  |  |
| x0x10 | x0xyy | - |  |  |
| x1x00 | xyx0y | xyxy0 | - |  |
| x1x11 | xyxy1 | xyx1y | x1xyy | - |
| A3 | Ø | Ø | Ø | Ø |

На этом поиск простых импликант заканчивается, кубы из множества С2 являются кубами множества Z2.

##### Конечное множество простых импликант:

##### Z =  {x0x01; x0x10; x1x00; x1x11}.

Следующий этап – поиск *L-*экстремалей на множестве простых импликант (таблица 3.11). Для этого используется операция # (решетчатое вычитание).

*Таблица 3.11*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z#(Z-z) | x0x01 | x0x10 | x1x00 | x1x11 |
| x0x01 | - | zzyy  x0x10 | zyzzy  x1x00 | zyzyz  x1x11 |
| x0x10 | zzzyy  x0x01 | - | zyzyz  x1x00 | zyzzy  x1x11 |
| x1x00 | zyzzy  x0x01 | zyzyz  x0x10 | - | zzzyy  x1x11 |
| x1x11 | zyzyz  x0x01 | zyzzy  x0x10 | zzzyy  x1x00 | - |
| Остаток | x0x01 | x0x10 | x1x00 | x1x11 |

*L*-экстремалями стали следующие простые импликанты:

##### E = {x0x01; x0x10; x1x00; x1x11}

Проверка *L*-экстремалей.

*Таблица 3.12*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) n L | 00001 | 00010 | 00101 | 00110 | 01000 | 01011 | 01100 | 01111 | 10001 | 10010 | 10101 | 10110 | 11000 | 11011 | 11100 | 11111 |
| x0x01 | 00001 | Ø | 00101 | Ø | Ø | Ø | Ø | Ø | 10001 | Ø | 10101 | Ø | Ø | Ø | Ø | Ø |
| x0x10 | Ø | 00010 | Ø | 00110 | Ø | Ø | Ø | Ø | Ø | 10010 | Ø | 10110 | Ø | Ø | Ø | Ø |
| x1x00 | Ø | Ø | Ø | Ø | 01000 | Ø | 01100 | Ø | Ø | Ø | Ø | Ø | 11000 | Ø | 11100 | Ø |
| x1x11 | Ø | Ø | Ø | Ø | Ø | 01011 | Ø | 01111 | Ø | Ø | Ø | Ø | Ø | 11011 | Ø | 11111 |

##### = Z - E = {Ø}.

##### Из таблицы 3.12 видно, что все кубы пересекаются с кубами множества L. Таким образом можно сделать вывод, что минимальное покрытие – это и есть множество L – экстремалей E = {x0x01; x0x10; x1x00; x1x11}.

Для функции S1:

a1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  | 1 | 1 |  |  | 1 |  |
| 1 | 1 | 1 |  | 1 |  |  |  |
|  |  |  | 1 |  | 1 | 1 | 1 |
|  | 1 |  |  | 1 | 1 |  | 1 |

b2

b1

p

p

a2

Следовательно

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

𝐾1=; 𝐾2=;

𝐾3=.

Функциональная схема ОЧC в заданном базисе представлена в приложении В.

1. **СИНТЕЗ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРА**

Мультиплексор – это логическая схема, имеющая n входов,m управляющих входов и один выход. При этом должно выполняться равенство . На выход мультиплексора может быть пропущен без изменений любой (один) логический сигнал, поступающий на информационные входы. Порядковый номер информационного входа, значение с которого в данный момент должно быть передано на выход, должно быть передано на выход, определяется двоичным кодам на управляющих входах. Для синтеза ОЧС будем использовать мультиплексор “один из восьми” (1 из 8).

*Таблица 4.1 – таблица истинности для синтеза ПФ ОЧС*

| ***a1*** | ***a2*** | ***b1*** | ***b2*** | ***p*** | **П** | **П** | ***S1*** | ***S1*** | ***S2*** | ***S2*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** |  | **7** |  | **8** |  |
| 0 | 0 | 0 | 0 | 0 | 1 |  | 1 |  | 0 |  |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |  | 0 |  | 0 |  |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |  | 1 |  | 1 |  |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |  | 0 |  | 1 |  |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | “0” | 0 |  | 0 |  |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |  | 1 |  | 0 |  |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

*Продолжение таблицы 4.1*

| 1 | 1 | 0 | 0 | 0 | 1 | “1” | 0 |  | 1 |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |  | 1 |  | 1 |  |
| 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 мультиплексорах представлена в приложении Г.

# **Логический синтез преобразователя множителя (ПМ)**

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11, заменяя их на триады .

*Таблица истинности ПМ. Таблица 5*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вх. диада | | Tригер | Зн. | Вых. диада | |
| Qn | Qn-1 | T | 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 |

Проведём минимизацию **P** при помощи карты Карно:

Qn-1T

Qn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P** | 00 | 01 | 11 | 10 |
| 0 |  |  |  |  |
| 1 | 1 | 1 | 1 | 1 |

P = Qn

Очевидно, что S1 не минимизируется, поэтому

Проведём минимизацию **S2** при помощи карты Карно:

Qn-1T

Qn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S2** | 00 | 01 | 11 | 10 |
| 0 |  | 1 |  | 1 |
| 1 |  | 1 |  | 1 |

# **оценка результатов разработки**

Формула расчёта временных затрат на умножение:

, где

– время преобразования множителя;

– время формирования дополнительного кода множимого;

– время умножения на ОЧУС;

– время формирования единицы переноса в ОЧС;

– время сдвига частичной суммы;

n – количество разрядов на множителе.

**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 (Вейче) можно выделить простоту и минимальные затраты времени. Однако применение данного способа для функций многих переменных будет затруднительно. Функциональные схемы были построены в различных логических базисах. Это позволило закрепить теоретические знания основных законов булевой алгебры, например, правило де Моргана. Также можно отметить, что необходимо сократить количество уровней в логической схеме для уменьшения времени работы данного устройства.

Реализация переключательных функций на основе мультиплексоров позволила облегчить процесс минимизации этих функций.

1. **. Список использованных источников**

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

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.

Лысиков, Б. Г. Арифметические и логические основы цифровых авто- матов / Б. Г. Лысиков. – Минск : Выш. шк., 1980. – 342 с.

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

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

Усатенко, С. Т. Выполнение электрических схем по ЕСКД : справочник / С. Т. Усатенко, Т. К. Каченюк, М. В. Терехова. – М. : Изд-во стандартов, 1989. – 325 с.

**Приложение А**

*(обязательное)*

Сумматор-умножитель второго типа. Схема электрическая структурная

**Приложение Б**

*(обязательное)*

Одноразрядный четверичный умножитель-сумматор. 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