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 01 518 ПЗ

Студент А. А. Старикевич

Руководитель И. В. Лукьянова

Минск 2020

Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

информатики и радиоэлектроники”

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы

вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

ЭВМ

\_\_\_\_\_\_\_\_\_\_Б. В. Никульшин

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2020 г.

ЗАДАНИЕ

по курсовой работе студента

Старикевича Антона Александровича

**1** Тема работы: Проектирование и логический синтез сумматора-умножителя

двоично-четверичных чисел.

**2** Срок сдачи студентом законченной работы:

**3** Исходные данные к работе:

**3.1** исходные сомножители: Мн = 12,97; Мт = 75,39.

**3.2** алгоритм умножения: А.

**3.3** метод умножения: умножение закодированного двоично-четверичного

множимого на два разряда двоичного множителя одновременно в

прямых кодах.

**3.4** коды четверичных цифр множимого для перехода к двоично-

четверичной системе кодирования; 04 ­­– 11, 14 – 01, 24 – 00, 34 – 10.

**3.5** тип синтезируемого умножителя: 1.

**3.6** логический базис для реализации ОЧС: А5; метод минимизации –

алгоритм Рота.

**3.7** Логический базис для реализации ОЧУ: А5; метод минимизации –

карты Карно-Вейча.

**4.** Содержание пояснительной записки (перечень подлежащих разработке

вопросов):

Введение. 1. Разработка алгоритма умножения. 2. Разработка

структурной схемы сумматора-умножителя. 3. Логический синтез

одноразрядного четверичного сумматора. 4. Логический синтез

одноразрядного четверичного умножителя-сумматора. 5. Логический

синтез одноразрядного четверичного сумматора на основе

мультиплексора. 6. Логический синтез преобразователя множителя.

7. Временные затраты на умножение. Заключение. Список литературы.

**5** Перечень графического материалы:

**5.1** Сумматор-умножитель первого типа. Схема электрическая структурная.

**5.2** Одноразрядный четверичный сумматор. Схема электрическая

функциональная.

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

электрическая функциональная.

**5.4** Преобразователь множителя. Схема электрическая функциональная.

**5.5** 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 | 50 | 10.03 - 19. | С выполнением чертежей |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 |  | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 |  |  |

Дата выдачи задания: 28 февраля 2020 г.

Руководитель И. В. Лукьянова

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

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение……………………………………………………………………….......... | 5 |
| 1. Разработка алгоритма умножения………………………………………………. | 6 |
| 2. Разработка структурной схемы сумматора-умножителя……………………. | 9 |
| 3. Разработка функциональных схем основных узлов сумматора – умножителя | 10 |
| 3.1 Логический синтез одноразрядного четверичного умножителя……………. | 10 |
| 3.2 Логический синтез одноразрядного четверичного сумматора……………… | 15 |
| 4. Логический синтез одноразрядного четвертичного сумматора на основе мультиплексора…………………………………………………………………….. | 23 |
| 5. Логический синтез преобразователя множителя……………………………… | 25 |
| 6. Временные затраты на умножение……………………………………………... | 26 |
| Заключение…………………………………………………………………………. | 26 |
| Список использованных источников……………………………………………... | 27 |
| Приложение А……………………………………………………………………… | 28 |
| Приложение Б……………………………………………………………………… | 29 |
| Приложение В……………………………………………………………………… | 30 |
| Приложение Г…………………………………………………………………........ | 31 |
| Приложение Д……………………………………………………………………… | 32 |
| Приложение Е…………………………………………………………………........ | 33 |

**ВВЕДЕНИЕ**

Данная курсовая работа посвящена разработке алгоритмов выполнения операций умножения и сложения. На основе полученных алгоритмов требуется разработать требуется разработать и синтезировать следующие устройства: одноразрядный четверичный сумматор (ОЧС), одноразрядный четверичный умножитель-сумматор (ОЧУС), а также переключательные функции ОЧС на мультиплексорах. Минимизация перечисленных функции ОЧС на мультиплексорах. Минимизация перечисленных устройств осуществляется с помощью карт Карно-Вейча и алгоритма извлечения Рота. На основе полученных данных требуется построить схемы этих устройств и проанализировать результаты (эффективность минимизации и время выполнения операций).

**1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ**

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

Множимое

12|4 0.97 Мн4 = 30,3320

12|3 4 в соответствии с заданной кодировкой

0 3.88 множимого

4 Мн2/4 = 1011, 10100011

3.52

4

2.08

4

0.32

Множитель

75| 4 0.39 Мт4 = 1023,12

72| 18| 4 4

3| 16| 4| 4 1.56 Мт2/4 = 01001011, 0110

2| 4| 1 4 *множитель представляется обычным*

0 2.24 *весомозначным кодом*:

04 = 00, 14 = 01, 24 = 10, 34 = 11

для всех вариантов

Запишем сомножители в форме с плавающей запятой в прямом коде:

Мн = 0,101110100011 PМн = 0.0000 +0210 — закодировано по заданию

Мт = 0,010010110110 PМт = 0.0100 +0410 — закодировано традиционно

Умножение двух чисел с плавающей запятой на два разряда множителя

одновременно в прямых кодах. Это сводится к сложению порядков, формированию знака произведения, преобразованию разрядов множителя согласно алгоритму, и перемножению мантисс сомножителей.

Порядок произведения будет равен:

PМн = 0.0000 02

PМт = 0.0100 10

PМн\*Мт = 0.0100 12

Результат закодирован в соответствии с заданием на кодировку множимого.

Знак произведения определяется суммой по модулю “два” знаков сомножителей:

зн Мн ⊕ зн Мт = 0 ⊕ 0 = 0

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в прямых кодах диада 11(34) заменяется на триаду 10´1. Преобразованный множитель имеет вид: Мтп4 = 111´1´12 или

Мтп2 = 0100010´1´0´1´0110

Перемножение мантисс по алгоритму “Б” приведено в табл. 1.1.

Таблица 1.1 – Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Четверичная с/с | | Двоично-четверичная с/c | | Комментарии |
| 0. | 000000000000 | 11. | 11 11 11 11 11 11 11 11 11 11 11 11 | ∑ 0ч |
| 0. | 000001213300 | 11. | 11 11 11 11 11 01 00 01 10 10 11 11 | П1ч = 2⋅Мн |
| 0. | 000001213300 | 11. | 11 11 11 11 11 01 00 01 10 10 11 11 | ∑ 1ч |
| 0. | 000003033200 | 11. | 11 11 11 11 11 10 11 10 10 00 11 11 | П2ч = 1⋅Мн⋅21 |
| 0. | 000010313100 | 11. | 11 11 11 11 01 11 10 01 10 01 11 11 | ∑2ч |
| 3. | 333303001300 | 10. | 10 10 10 10 11 10 11 11 01 10 11 11 | П3ч = (-1)⋅Мн⋅22 |
| 3. | 333313321000 | 10. | 10 10 10 10 01 10 10 00 01 11 11 11 | ∑3ч |
| 3. | 333030013000 | 10. | 10 10 10 11 10 11 11 01 10 11 11 11 | П4ч = (-1)⋅Мн⋅23 |
| 3. | 333010000000 | 10. | 10 10 10 11 01 11 11 11 11 11 11 11 | ∑4ч |
| 0. | 003033200000 | 11. | 11 11 10 11 10 10 00 11 11 11 11 11 | П5ч = 1⋅Мн⋅24 |
| 0. | 002103200000 | 11. | 11 11 00 01 11 10 00 11 11 11 11 11 | ∑5ч |
| 0. | 030332000000 | 11. | 11 10 11 10 10 00 11 11 11 11 11 11 | П6ч = 1⋅Мн⋅25 |
| 0. | 033101200000 | 11. | 11 10 10 01 11 01 00 11 11 11 11 11 | ∑6ч |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн⋅Мт4 = 0,33101200000, PМн⋅Мт = 6)

Приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн⋅Мт4 = 33101,200000 PМн⋅Мт = 0;

Мн⋅Мт10 = 977.5

Результат прямого перемножения операндов даёт следующее значение:

Мн10⋅Мт10 = 977,8083.

Абсолютная погрешность:

Δ = 977,8083 – 977,5 = 0,3083.

Относительная погрешность:

δ = = = 0,00031529697 (δ = 0,031529697%)

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

**2.РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ**

Структура первого типа строится на базисе заданных узлов ОЧУ и ОЧС. Управление режимами работы обеих схем осуществляется внешним сигналом *mul/sum,* который определяет вид текущей арифметической операции (умножение или суммирование).

Структурная схема сумматора-умножителя первого типа для алгоритма умножения “Б” приведена в приложении А.

*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) *F2* поступает «1».

Следует учесть, что числа представлены в форме с плавающей запятой, поэтому, прежде чем складывать мантиссы, необходимо выровнять их порядки.

В блоке порядков необходимо обеспечить сравнение порядком, используя сумматор порядков, и в зависимости от знака результата сдвигать первое или второе слагаемое.

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

Разрядность аккумулятор 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 *F2* поступает «0».

Принцип работы ФДК, в зависимости от управляющих сигналов, приведён в таблице 2.1.

Таблица 2.1 – Режимы работы формирователя дополнительного кода

|  |  |  |
| --- | --- | --- |
| **Сигнал на входах ФДК** | | **Результат на выходах ФДК** |
| *F1* | *F2* |
| 0 | 0 | Дополнительный код множимого |
| 0 | 1 | Дополнительный код слагаемого |
| 1 | 0 | Меняется знак Мн |
| 1 | 1 | Меняется знак слагаемого |

ОЧУ предназначен лишь для умножения двух четверичных цифр. Если в процессе умножения возникает перенос в следующий разряд, необходимо предусмотреть возможность его прибавления.

Для суммирования результата умножения текущей диады Мн·Мт с переносом из предыдущей диады предназначен одноразрядный четверичный сумматор (ОЧС). Следовательно, чтобы полностью сформировать частичное произведение четверичных сомножителей, необходима комбинация цепочек ОЧУ и ОЧС.

**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 – 11; 1 – 01; 2 – 00; 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 3.1 выделено 8 безразличных наборов, т.к. на выходы 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** |
| x1 | x2 | y1 | y2 | h | p1 | p2 | p3 | p4 |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2⋅0 = 00 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | Выход – код 02 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 2⋅1 = 02 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | Выход – код 02 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 2⋅2 = 10 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | Выход – код 02 |
| 0 | 0 | 1 | 1 | 0 | X | X | X | X | 2⋅3 = 12 |
| 0 | 0 | 1 | 1 | 1 | X | X | X | X | Выход – код 02 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1⋅0 = 00 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | Выход – код 01 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1⋅1 = 01 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | Выход – код 01 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1⋅2 = 02 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | Выход – код 01 |
| 0 | 1 | 1 | 1 | 0 | X | X | X | X | 1⋅3 = 03 |
| 0 | 1 | 1 | 1 | 1 | X | X | X | X | Выход – код 01 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 3⋅0 = 00 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | Выход – код 03 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 3⋅1 = 03 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | Выход – код 03 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 3⋅2 = 12 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 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 03 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0⋅0 = 00 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | Выход – код 00 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0⋅1 = 00 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | Выход – код 00 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0⋅2 = 00 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | Выход – код 00 |
| 1 | 1 | 1 | 1 | 0 | X | X | X | X | 0⋅3 = 00 |
| 1 | 1 | 1 | 1 | 1 | X | X | X | X | Выход – код 00 |

Минимизацию переключательных функций проведём с помощью карт Вейча, функцию P3 также минимизируем с помощью алгоритма Ротта. В рисунках 3.1.1 – 3.1.4 символом “x” отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

Рисунок 3.1.1 – минимизация функции p1 при помощи карты Вейча

x2

h

x1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | X | 1 | 0 | X | 1 | 1 |
| 1 | 1 | X | 1 | 1 | X | 1 | 1 |
| 1 | 1 | X | 1 | 1 | X | 1 | 1 |
| 1 | 1 | X | 1 | 0 | X | 1 | 1 |

y1

y2

y2

p1 МДНФ = x2 + h + y1

Функция для реализации в заданном базисе(А1) будет иметь вид:

p1 МДНФ = x2 + h + y1

Рисунок 3.1.2 – минимизация функции p2 при помощи карты Вейча



h

x1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | X | 1 | 1 | X | 1 | 1 |
| 1 | 1 | X | 1 | 1 | X | 1 | 1 |
| 1 | 1 | X | 1 | 1 | X | 1 | 1 |
| 1 | 1 | X | 1 | 1 | X | 1 | 1 |

y1

y2

y2

p2 МДНФ = 1

Рисунок 3.1.3 – минимизация функции p3 при помощи карты Вейча



h

x1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | X | 1 | 0 | X | 1 | 1 |
| 1 | 1 | X | 1 | 1 | X | 1 | 1 |
| 1 | 0 | X | 1 | 0 | X | 0 | 0 |
| 1 | 1 | X | 1 | 1 | X | 0 | 1 |

y1

y2

y2

p3 МДНФ = x1h + x2h + x2y1 + y1y2h + x1x2y2h

Функция для реализации в заданном базисе(А1) будет иметь вид:

p3 МДНФ = x1h + x2h + x2y1 + y1y2h + x1x2y2h

Рисунок 3.1.4 – минимизация функции p4 при помощи карты Вейча



h

x1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | X | 1 | 0 | X | 0 | 1 |
| 1 | 1 | X | 1 | 0 | X | 0 | 0 |
| 1 | 1 | X | 1 | 0 | X | 0 | 0 |
| 1 | 1 | X | 0 | 1 | X | 0 | 1 |

y1

y2

y2

p4 МДНФ = x1x2 + x2h + x2y1 + y1y2h + x1x2y2h

Функция для реализации в заданном базисе(А1) будет иметь вид:

p4 МДНФ = x1x2 + x2h + x2y1 + y1y2h + x1x2y2h

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

KP1 = = 19,83

KP2 = = 39,6

KP3 = = 4,7

KP4 = = 3,3

Функциональная схема ОЧУ в заданном базисе показана 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 3.2)

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

В таблице 3.2 выделено 16 безразличных наборов, т.к. со старших входов ОЧУ не могут прийти коды “2” и “3”.

Таблица 3.2.1 – Таблица истинности ОЧС

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **а1** | **а2** | **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 | X | X | X | 2+2+0 = 10 |
| 0 | 0 | 0 | 0 | 1 | X | X | X | 2+2+1 = 11 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2+1+0 = 03 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2+1+1 = 10 |
| 0 | 0 | 1 | 0 | 0 | X | X | X | 2+3+0 = 11 |
| 0 | 0 | 1 | 0 | 1 | X | X | X | 2+3+1 = 12 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2+0+0 = 02 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 2+0+1 = 03 |
| 0 | 1 | 0 | 0 | 0 | X | X | X | 1+2+0 = 03 |
| 0 | 1 | 0 | 0 | 1 | X | X | X | 1+2+1 = 10 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 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 | 0 | 0 | 1 | 1+0+0 = 01 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1+0+1 = 02 |
| 1 | 0 | 0 | 0 | 0 | X | X | X | 3+2+0 = 11 |
| 1 | 0 | 0 | 0 | 1 | X | X | X | 3+2+1 = 12 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 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 | 0 | 1 | 0 | 3+0+0 = 03 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3+0+1 = 10 |
| 1 | 1 | 0 | 0 | 0 | X | X | X | 0+2+0 = 02 |
| 1 | 1 | 0 | 0 | 1 | X | X | X | 0+2+1 = 03 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0+1+0 = 01 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0+1+1 = 02 |
| 1 | 1 | 1 | 0 | 0 | X | X | X | 0+2+0 = 02 |
| 1 | 1 | 1 | 0 | 1 | X | X | X | 0+2+1 = 03 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0+0+0 = 00 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0+0+1 = 01 |

Минимизацию функций проведём с помощью карт Карно. Заполненные карты приведены на рисунках 3.2.1 – 3.2.3.

Для функции П:

x3x4x5

x1x2

100

101

111

110

000

010

011

001

00

10

11

01

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | X | 1 | 0 | 0 | 0 | X | X |
| X | X | 0 | 0 | 0 | 0 | X | X |
| X | X | 0 | 0 | 0 | 0 | X | X |
| X | X | 1 | 1 | 0 | 1 | X | X |

Рисунок 3.2.1 – Минимизация функции П при помощи карты Карно

ПМДНФ = x1x2x5 + x2x3x5 + x1x2x3

Функция для реализации в заданном базисе(А5) будет иметь вид:

ПМДНФ = x1 + x2 + x5 + x2 + x3 + x5 + x1 + x2 + x3

Для функции S1:

x3x4x5

x1x2

100

101

111

110

000

010

011

001

00

10

11

01

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | X | 1 | 1 | 0 | 1 | X | X |
| X | X | 1 | 0 | 0 | 0 | X | X |
| X | X | 0 | 0 | 1 | 0 | X | X |
| X | X | 0 | 1 | 1 | 1 | X | X |

Рисунок 3.2.2 – Минимизация функции S1 при помощи карты Карно

S1МДНФ = x2x3x5 + x1x2x3 + x1x3x5 + x1x2x5 + x1x3x5 + x1x2x5 + x2x3x5

Функция для реализации в заданном базисе(А5) будет иметь вид:

S1МДНФ = x2 + x3 + x5 + x1 + x2 + x3 + x1 + x3 + x5 + x1 + x2 + x5 + x1 + x3 + x5 +

+ x1 + x2 + x5 + x2 + x3 + x5

Для функции S2:

x3x4x5

x1x2

100

101

111

110

000

010

011

001

00

10

11

01

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | X | 1 | 0 | 0 | 0 | X | X |
| X | X | 0 | 0 | 1 | 0 | X | X |
| X | X | 0 | 1 | 1 | 1 | X | X |
| X | X | 1 | 1 | 0 | 1 | X | X |

Рисунок 3.2.3 – Минимизация функции S2 при помощи карты Карно

S2МДНФ = x1x3x5 + x1x2x3 + x2x3x5 + x1x2x5 + x2x3x5 + x1x2x3 + x1x2x5

Функция для реализации в заданном базисе(А5) будет иметь вид:

S2МДНФ = x1 + x3 + x5 + x1 + x2 + x3 + x2 + x3 + x5 + x1 + x2 + x5 + x2 + x3 + x5 +

+ x1 + x2 + x3 + x1 + x2 + x5

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

KП = = 2,1

KS1 = = 1,75

KS2 = = 1,8

Для минимизации функции p3 воспользуемся алгоритмом Рота. Определим множество единичных кубов:

L = {00000, 00100, 01000, 10000, 10001, 10010, 10011, 10101, 11000, 11001,

11010, 11011, 11100, 11101}

Далее определим множество безразличных кубов:

N = {00110, 00111, 01110, 01111, 10110, 10111, 11110, 11111}

Склеим всевозможные кубы во множествах L и N:

L = {11010, 11100, 00x00, x1000, 1x101, 110x1, 100xx}

N = {xx11x}

Сформируем множество С0 = L N:

C0 = {11010, 11100, 00x00, x1000, 1x101, 110x1, 100xx, xx11x}

Первым этапом алгоритма Рота является нахождение множества простых импликант. Для реализации этого этапа будем использовать операцию умножения (\*) над множествами. Первый шаг умножения (С0 \* С0) приведён в таблице 3.2.2.

Таблица 3.2.2 – Поиск простых импликант (С0 \* С0)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С0\*С0 | 11010 | 11100 | 00x00 | x1000 | 1x101 | 110x1 | 100xx | xx11x |
| 11010 | - |  |  |  |  |  |  |  |
| 11100 | 11yy0 | - |  |  |  |  |  |  |
| 00x00 | yy0y0 | yy100 | - |  |  |  |  |  |
| x1000 | 110y0 | 11y00 | 0y000 | - |  |  |  |  |
| 1x101 | 11yyy | 1110y | y010y | 11y0y | - |  |  |  |
| 110x1 | 1101y | 11y0y | yy00y | 1100y | 11y01 | - |  |  |
| 100xx | 1y010 | 1yy00 | y0000 | 1y000 | 10y01 | 1y0x1 | - |  |
| xx11x | 11y10 | 111y0 | 001y0 | x1yy0 | 1x1y1 | 11y11 | 10y1x | - |
| A1 | 110x0  1101x  1x010  11x10 | 11x00  1110x  111x0 | 0x000  x0000  001x0 | 1100x  1x000 | 11x01  10x01  1x1x1 | 1x0x1  11x11 | 10x1x | Ø |

Из таблицы 3.2.2 – Поиск простых импликант (C0 \* C0) следует:

A1 = {110x0, 1101x, 1x010, 11x10, 11x00, 1110x, 111x0, 0x000, x0000, 001x0,

1100x, 1x000, 11x01, 10x01, 1x1x1, 1x0x1, 11x11, 10x1x}

Z0 = {Ø}

B1 = {11010, 11100, 00x00, x1000, 1x101, 110x1, 100xx, xx11x}

C1 = {110x0, 1101x, 1x010,11x10,11x00,1110x, 111x0, 0x000, x000, 001x0,

1100x, 1x000, 11x01, 10x01, 1x1x1, 1x0x1, 11x11, 10x1x, 00x00, x1000,

100xx, xx11x}

Второй шаг умножения (С1 \* С1) приведён в таблице 3.2.3

Таблица 3.2.3 – Поиск простых импликант (С1 \* С1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С1\*С1 | 110x0 | 1101x | 1x010 | 11x10 | 11x00 | 1110x | 111x0 | 0x000 | x0000 | 001x0 | 1100x | 1x000 | 11x01 | 10x01 | 1x1x1 | 1x0x1 | 11x11 | 10x1x |
| 110x0 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1101x | 11010 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1x010 | 11010 | 11010 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11x10 | 11010 | 11010 | 11010 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11x00 | 11000 | 110y0 | 110y0 | 11xy0 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1110x | 11y00 | 11yyx | 11yy0 | 111y0 | 11100 | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 111x0 | 11yx0 | 11y10 | 11y10 | 11110 | 11100 | 11100 | - |  |  |  |  |  |  |  |  |  |  |  |
| 0x000 | y1000 | y10y0 | yx0y0 | y10y0 | y1000 | y1y00 | y1y00 | - |  |  |  |  |  |  |  |  |  |  |
| x0000 | 1y000 | 1y0y0 | 100y0 | 1y0y0 | 1y000 | 1yy00 | 1yy00 | 00000 | - |  |  |  |  |  |  |  |  |  |
| 001x0 | yyyx0 | yyy10 | y0y10 | yy110 | yy100 | yy100 | yy1x0 | 00y00 | 00y00 | - |  |  |  |  |  |  |  |  |
| 1100x | 11000 | 110yx | 110y0 | 110y0 | 11000 | 11y0x | 11y00 | y1000 | 1y000 | yyy00 | - |  |  |  |  |  |  |  |
| 1x000 | 11000 | 110y0 | 1x0y0 | 110y0 | 11000 | 11y00 | 11y00 | yx000 | 10000 | y0y00 | 11000 | - |  |  |  |  |  |  |
| 11x01 | 1100y | 110y1 | 110yy | 11xyy | 11x0y | 11101 | 1110y | y100y | 1y00y | yy10y | 11001 | 1100y | - |  |  |  |  |  |
| 10x01 | 1y00y | 1y0y1 | 100yy | 1yxyy | 1yxoy | 1y101 | 1y10y | y000y | 1000y | y010y | 1y001 | 1000y | 1yx01 | - |  |  |  |  |
| 1x1x1 | 11yxy | 11y11 | 1xy1y | 1111y | 1110y | 11101 | 111xy | yxy0y | 10y0y | y01xy | 11y01 | 1xy0y | 11101 | 10101 | - |  |  |  |
| 1x0x1 | 110xy | 11011 | 1x01y | 1101y | 1100y | 11y01 | 11yxy | yx00y | 1000y | y0yxy | 11001 | 1x00y | 11001 | 10001 | 1xyx1 | - |  |  |
| 11x11 | 1101y | 11011 | 1101y | 11x1y | 11xyy | 111y1 | 1111y | y10yy | 1y0yy | yy11y | 110y1 | 110yy | 11xy1 | 1yxy1 | 11111 | 11011 | - |  |
| 10x1x | 1y010 | 1y01x | 10010 | 1yx10 | 1yxy0 | 1y1yx | 1y110 | y00y0 | 100y0 | y0110 | 1y0yx | 100y0 | 1yxy1 | 10xy1 | 10111 | 10011 | 1yx11 | - |
| 00x00 | yy000 | yy0y0 | y00y0 | yyxy0 | yyx00 | yy100 | yy100 | 00000 | 00000 | 00100 | yy000 | y0000 | yyx0y | y0x0y | y010y | y000y | yyxyy | y0xy0 |
| x1000 | 11000 | 110y0 | 110y0 | 110y0 | 11000 | 11y00 | 11y00 | 01000 | xy000 | 0yy00 | 11000 | 11000 | 1100y | 1y00y | 11y0y | 1100y | 110yy | 1y0y0 |
| 100xx | 1y0x0 | 1y01x | 10010 | 1y010 | 1y000 | 1yy0x | 1yyx0 | y0000 | 10000 | y0yx0 | 1y00x | 10000 | 1y001 | 10001 | 10yx1 | 100x1 | 1y011 | 1001x |
| xx11x | 11y10 | 11y1x | 1xy10 | 11110 | 111y0 | 111yx | 11110 | 0xyy0 | x0yy0 | 00110 | 11yyx | 1xyy0 | 111y1 | 101y1 | 1x111 | 1xy11 | 11111 | 1011x |
| А2 | 11xx0  110xx  1x0x0 | 110xx  1x01x  1x01x  11x1x | 1x0x0  1x01x  1xx10 | 11xx0  11x1x  1xx10 | 11x0x | 11x0x  111xx | 111xx | xx000 | xx000 | Ø | 1x00x | 1x00x | 1xx01  11xx1 | 10xx1 | 1xx11  10xx1 | 1xx11 | 1xx11 | Ø |

Из таблицы 3.2.3 – Поиск простых импликант (С1\*С1) следует:

A2 = {11xx0, 110xx, 1x0x0, 1x01x, 11x1x, 1xx10, 11x0x, 111xx, xx000, 1x00x,

1xxx1}

Z1 = {001x0, 00x00}

B­­2 = {110x0, 1101x, 1x010, 11x10, 11x00, 1110x, 111x0, 0x000, x0000, 1100x,

1x000, 11x01, 10x01, 1x1x1, 1x0x1, 11x11, 10x1x, x1000, 100xx, xx11x}

C2 = {11xx0, 110xx, 1x0x0, 1x01x, 11x1x, 1xx10, 11x0x, 111xx, xx000, 1x00x,

1xxx1, 10x1x, 100xx, xx11x}

Третий шаг умножения () приведён в таблице 3.2.4

Таблица 3.2.4 – Поиск простых импликант (С2\*С2)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С2\*С2 | 11xx0 | 110xx | 1x0x0 | 1x01x | 11x1x | 1xx10 | 11x0x | 111xx | xx000 | 1x00x | 1xxx1 |
| 11xx0 | - |  |  |  |  |  |  |  |  |  |  |
| 110xx | 110x0 | - |  |  |  |  |  |  |  |  |  |
| 1x0x0 | 110x0 | 110x0 | - |  |  |  |  |  |  |  |  |
| 1x01x | 11010 | 1101x | 1x010 | - |  |  |  |  |  |  |  |
| 11x1x | 11x10 | 1101x | 11010 | 1101x | - |  |  |  |  |  |  |
| 1xx10 | 11x10 | 11010 | 1x010 | 1x010 | 11x10 | - |  |  |  |  |  |
| 11x0x | 11x00 | 1100x | 11000 | 110yx | 11xyx | 11xy0 | - |  |  |  |  |
| 111xx | 111x0 | 11yxx | 11yx0 | 11y1x | 1111x | 11110 | 1110x | - |  |  |  |
| xx000 | 11000 | 11000 | 1x000 | 1x0y0 | 110y0 | 1x0y0 | 11000 | 11y00 | - |  |  |
| 1x00x | 11000 | 1100x | 1x000 | 1x0yx | 110yx | 1x0y0 | 1100x | 11y0x | 1x000 | - |  |
| 1xxx1 | 11xxy | 110x1 | 1x0xy | 1x011 | 11x11 | 1xx1y | 11x01 | 111x1 | 1x00y | 1x001 | - |
| 10x1x | 1yx10 | 1y01x | 10010 | 1001x | 1yx1x | 10x10 | 1yxyx | 1y11x | 100y0 | 100yx | 10x11 |
| 100xx | 1y0x0 | 1y0xx | 100x0 | 1001x | 1y01x | 10010 | 1y00x | 1yyxx | 10000 | 1000x | 100x1 |
| xx11x | 11110 | 11y1x | 1xy10 | 1xy1x | 1111x | 1x110 | 111yx | 1111x | xxyy0 | 1xyyx | 1x111 |
| А3 | 11xxx | 11xxx  1x0xx | 1x0xx | 1x0xx  1xx1x | 11xxx  1xx1x | 1xx1x | Ø | Ø | Ø | Ø | Ø |

Из таблицы 3.2.4 – Поиск простых импликант () следует:

A3 = {11xxx, 1x0xx, 1xx1x}

Z2 = {001x0, 00x00, xx000}

B3 = {11xx0, 110xx, 1x0x0, 1x01x, 11x1x, 1xx10, 11x0x, 111xx, 1x00x, 1xxx1,

10x1x, 100xx, xx11x}

C3 = {11xxx, 1x0xx, 1xx1x, 1xxx1, xx11x}

Четвёртый шаг умножения (С3 \* С3) приведён в таблице 3.2.5

Таблица 3.2.5 – Поиск простых импликант (С3 \* С3)

|  |  |  |  |
| --- | --- | --- | --- |
| С3 \* С­­­­­­­3 | 11xxx | 1x0xx | 1xx1x |
| 11xxx | - |  |  |
| 1x0xx | 110xx | - |  |
| 1xx1x | 11x1x | 1x01x | - |
| 1xxx1 | 11xx1 | 1x0x1 | 1xx11 |
| xx11x | 1111x | 1xy1x | 1x11x |
| A4 | Ø | Ø | Ø |

Из таблицы 3.2.5 – Поиск простых импликант (С3 \* С3) следует:

А4 = {Ø}

На этом поиск простых импликант заканчивается т. к. С4 = Ø

Конечное множество простых импликант Z = {001x0, 00x00, xx000, 11xxx, 1x0xx,

1xx1x, 1xxx1, xx11x}

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

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

Таблица 3.2.6 – Поиск L-экстремалей

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 001x0 | 00x00 | xx000 | 11xxx | 1x0xx | 1xx1x | 1xxx1 | xx11x |
| 001x0 | - | 00x00 | xx000 | 11xxx | 1x0xx | 1xx1x | 1xxx1 | 1x11x  x111x  xx111 |
| 00x00 | 00110 | - | 1x000  x1000 | 11xxx | 1x0xx | 1xx1x | 1xxx1 | 1x11x  x111x  xx111 |
| xx000 | 00110 | 00x00 | - | 11xxx  11x1x  11xx1 | 1x01x  1x0x1 | 1xx1x | 1xxx1 | 1x11x  x111x  xx111 |
| 11xxx | 00110 | 00x00 | 10000  01000 | - | 1001x  100x1 | 10x1x | 10xx1 | 1011x  0111x  0x111  x0111 |
| 1x0xx | 00110 | 00x00 | 01000 | 111xx  1111x  11xx1 | - | 1011x | 101x1 | 1011x  0111x  0x111  x0111 |
| 1xx1x | 00110 | 00x00 | 01000 | 1110x  11x01 | 10001 | - | 10101 | 0111x  0x111  00111 |
| 1xxx1 | 00110 | 00x00 | 01000 | 11100 | Ø | 10110 | - | 0111x  0x111  00111 |
| xx11x | Ø | 00x00 | 01000 | 11100 | Ø | Ø | 10101 | - |
| Остаток | Ø | 00x00 | 01000 | 11100 | Ø | Ø | 10101 | 0111x  0x111  00111 |

Результат операции (последняя строка таблицы) указывает на то, что L-экстремалями стали следующие простые импликанты:

E = {00x00, xx000, 11xxx, 1xxx1, xx11x}

Необходимо проверить, нет ли среди полученных L-экстремалей таких, которые стали L-экстремалями за счёт безразличных кубов. Для этого в таблице 3.2.7 из кубов множества L вычитаются остатки простых импликант, полученные в таблице 3.2.6 (результат выполнения операции Z#(Z-z).

Таблица 3.2.7 – Проверка L-экстремалей

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z)∩L | 00000 | 00100 | 01000 | 10000 | 10001 | 10010 | 10011 | 10101 | 11000 | 11001 | 11010 | 11011 | 11100 | 11101 |
| 00x00 | 00000 | 00100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 01000 | Ø | Ø | 01000 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 11100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11100 | Ø |
| 10101 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10101 | Ø | Ø | Ø | Ø | Ø |
| 0111x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 0x111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 00111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Из таблицы 3.2.7 видно что куб xx11x стал L-экстремалью только за счёт множества N. Следовательно, мы исключаем его из множества L-экстремалей.

E = {00x00, xx000, 11xxx, 1xxx1}

Следующим действием нужно проверить все ли вершины из комплекса L покрываются L-экстремалями. Для этого из каждого куба L вычитаем (#) элементы множества E.

Таблица 3.2.8 – Проверка L-экстремалей

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00000 | 00100 | 01000 | 10000 | 10001 | 10010 | 10011 | 10101 | 11000 | 11001 | 11010 | 11011 | 11100 | 11101 |
| 00x00 | Ø | Ø | 01000 | 10000 | 10001 | 10010 | 10011 | 10101 | 11000 | 11001 | 11010 | 11011 | 11100 | 11101 |
| xx000 | Ø | Ø | Ø | Ø | 10001 | 10010 | 10011 | 10101 | Ø | 11001 | 11010 | 11011 | 11100 | 11101 |
| 11xxx | Ø | Ø | Ø | Ø | 10001 | 10010 | 10011 | 10101 | Ø | Ø | Ø | Ø | Ø | Ø |
| 1xxx1 | Ø | Ø | Ø | Ø | Ø | 10010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| Остаток | Ø | Ø | Ø | Ø | Ø | 10010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Из таблицы 3.2.8 видно что L-экстремали не покрывают только один куб. Таким образом, множество кубов, не покрытых L-экстремалями:

L´ = {10010}

Так же формируем множество Z´, отняв от множества Z множество E:

Z´ = {001x0, 1x0xx, 1xx1x, xx11x}

Теперь из полученного множества Z´ нужно выбрать минимальное число кубов с минимальной ценой, чтобы покрыть непокрытые L-экстремалями элементы комплекса L. Для этого выполним операцию частичного упорядочивания.

Таблица 3.2.9 – Поиск составляющих ТФ

|  |  |
| --- | --- |
| Z´ ∩ L´ | 10010 |
| 001x0 | Ø |
| 1x0xx | 10010 |
| 1xx1x | 10010 |
| xx11x | Ø |

Функциональная схема ОЧС в заданном базисе показана в приложении В.

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

Мультиплексор – это логическая схема, имеющая *n* информационных входов, *m* управляющих входов и один выход. При этом будет 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. Удобно взять мультиплексор с тремя адресными входами, это позволит упростить одну нашу большую функцию от пяти аргументов до восьми функций от двух переменных. Функции от двух переменных достаточно просты для того, чтобы самостоятельно заметить их минимальную форму.

Синтез дополнительных логических схем для ПФ ОЧС приведён в таблице 4.1.

Таблица 4.1. – Таблица истинности для ОЧС на мультиплексорах

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a2b1 | b2 | p | П | П | S1 | S1 | S2 | S2 |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 000 | 0 | 0 | X |  | X |  | X |  |
| 000 | 0 | 1 | X | p | X | “1” | X | p |
| 000 | 1 | 0 | 0 |  | 1 |  | 0 |  |
| 000 | 1 | 1 | 1 |  | 1 |  | 1 |  |
| 001 | 0 | 0 | X |  | X |  | X |  |
| 001 | 0 | 1 | X | “0” | X | p | X | “0” |
| 001 | 1 | 0 | 0 |  | 0 |  | 0 |  |
| 001 | 1 | 1 | 0 |  | 1 |  | 0 |  |
| 010 | 0 | 0 | X |  | X |  | X |  |
| 010 | 0 | 1 | X | “0” | X | p | X | “0” |
| 010 | 1 | 0 | 0 |  | 0 |  | 0 |  |
| 010 | 1 | 1 | 0 |  | 1 |  | 0 |  |
| 011 | 0 | 0 | X |  | X |  | X |  |
| 011 | 0 | 1 | X | “0” | X | “0” | X | p |
| 011 | 1 | 0 | 0 |  | 0 |  | 1 |  |
| 011 | 1 | 1 | 0 |  | 0 |  | 0 |  |
| 100 | 0 | 0 | X |  | X |  | X |  |
| 100 | 0 | 1 | X | “1” | X | p | X | “1” |
| 100 | 1 | 0 | 1 |  | 1 |  | 1 |  |
| 100 | 1 | 1 | 1 |  | 0 |  | 1 |  |
| 101 | 0 | 0 | X |  | X |  | X |  |
| 101 | 0 | 1 | X | p | X | “1” | X | p |
| 101 | 1 | 0 | 0 |  | 1 |  | 0 |  |
| 101 | 1 | 1 | 1 |  | 1 |  | 1 |  |
| 110 | 0 | 0 | X |  | X |  | X |  |
| 110 | 0 | 1 | X | “0” | X | “0” | X | p |
| 110 | 1 | 0 | 0 |  | 0 |  | 1 |  |
| 110 | 1 | 1 | 0 |  | 0 |  | 0 |  |
| 111 | 0 | 0 | X |  | X |  | X |  |
| 111 | 0 | 1 | X | “0” | X | p | X | “1” |
| 111 | 1 | 0 | 0 |  | 1 |  | 1 |  |
| 111 | 1 | 1 | 0 |  | 0 |  | 1 |  |

Функциональная схема реализации ОЧС на мультиплексорах приведена в приложении Г.

**5. ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ**

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11, заменяя их на триады 101

Таблица 5.1 – Таблица истинности ПМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вх. диада | | Мл. бит | Зн. | Вых. диада | |
| 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 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

Минимизируем функции P, S1 иS2 картами Карно.

Таблица 5.2 – Минимизация функции P

Qn-1

Qn-2

Qn

00 01 11 10

1

0

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

PМДНФ = QnQn-1 + QnQn-2

Таблица 5.3 – Минимизация функции S1

Qn-1

Qn-2

Qn

00 01 11 10

1

0

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

S1МДНФ = QnQn-1Qn-2 + QnQn-1Qn-2

Таблица 5.4 – Минимизация функции S2

Qn-1

Qn-2

Qn

00 01 11 10

1

0

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

S2МДНФ = Qn-1Qn-2 + Qn-1Qn-2

Функциональная схема ПМ приведена в приложении Д.

**6. ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ**

Формула расчёта временных затрат на умножение:

, где

– время преобразования множителя;

– время формирования дополнительного кода множимого;

– время умножения на ОЧУ;

– 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 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. 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**

1. Луцик Ю.А., Лукьянова И.В. – Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". – Минск: БГУИР, 2014 г.
2. Луцик Ю.А., Лукьянова И.В. – Методические указания к курсовому проекту по курсу “Арифметические и логические основы вычислительной техники”. – Мн.: БГУИР, 2004 г.
3. Искра, Н. А. Арифметические и логические основы вычислительной техники: пособие / Н. А. Искра, И. В. Лукьянова, Ю. А. Луцик. – Минск: БГУИР, 2016. – 75 с.
4. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. Мн.: Вышейшая школа, 1980.
5. Лысиков Б.Г. Цифровая вычислительная техника. Мн.: 2003 г.