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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

РАЗРАБОТКА СУММАТОРА-УМНОЖИТЕЛЯ

БГУИР КР 1-40 02 01 027 ПЗ

Студент: группы 950503, Сякачёв П.В.

Руководитель: ассистент каф. ЭВМ,

Куприянова Д.В.

Минск 2020

Учреждение образования

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  
 Сякачёва Павла Владимировича

1. Тема работы: Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел.
2. Срок сдачи студентом законченной работы: с 20 по 25 мая 2020 г.
3. Исходные данные к работе:
   1. исходные сомножители: Мн = 47,54; Мт = 53,35.
   2. алгоритм умножения: А.
   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-четверичной системе кодирования; 04 – 01, 14 – 00, 24 – 11, 34 – 10.
   5. тип синтезируемого умножителя: 2.
   6. логический базис для реализации ОЧС: А6; метод минимизации – алгоритм Рота.
   7. логический базис для реализации ОЧУС: А7; 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 четвертичного сумматора на основе мультиплексора. 6. Логический синтез преобразователя множителя. 7. Временные затраты на умножение. 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-сумматор. Схема электрическая функциональная.
   4. Преобразователь множителя.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. Разработка структурной схемы сумматора-умножителя | 20 | 15.02-01.03 | С выполнением чертежа |
| Разработка функциональных схем основных узлов сумматора-умножителя | 50 | 10.03-27.03 | С выполнением чертежей |
| Синтез комбинационных схем устройств на основе мультиплексоров. Синтез преобразователя множителя. | 10 | 27.03-11.04 | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 | 11.04-16.04 |  |

Дата выдачи задания: 14 февраля 2020 г.

Руководитель Д. В. Куприянова

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_\_\_ П. В. Сякачёв

**СОДЕРЖАНИЕ**

Введение ……………………………………………………………………… 5

1. Разработка алгоритма умножения ………………………………………… 6

2. Разработка структурной схемы сумматора-умножителя …………………. 9

3. Логический синтез одноразрядного четверичного умножителя-сумматора ……………………………………………………………………… 12

4. Логический синтез одноразрядного четверичного сумматора ………….. 25

5. Логический синтез одноразрядного четвертичного сумматора на

основе мультиплексора …………………..………………………………… 29

6. Логический синтез преобразователя множителя ………………………. 31

7. Временные затраты на умножение .……………………………………... 33

Заключение ………………………………………………………………….. 34

Список использованных источников ...………………………………….… 35

Приложение А ………………………………………………………………. 36

Приложение Б ……………………………………………………………….. 37

Приложение В ………………………………………………………………. 38

Приложение Г ………………………………………………………………. 39

Приложение Д ………………………………………………………………. 40

Приложение Е ………………………………………………………………. 41

Приложение Ж………………………………………………………………. 42

Приложение З .………………………………………………………………. 43

Приложение И ………………………………………………………………. 44

Приложение К ………………………………………………………………. 45

**ВВЕДЕНИЕ**

Данная курсовая работа посвящена разработке алгоритмов выполнения операций умножения и сложения. На основе полученных алгоритмов требуется разработать и синтезировать следующие устройства: одноразрядный четвертичный сумматор (ОЧС), одноразрядный четвертичный умножитель-сумматор (ОЧУС), а также переключательные 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. **Перевод сомножителей из десятичной системы счисления в четверичную.**

Множимое

47 | 4 0.54 Мн4 = 233,202

4 11| 4 4 в соответствии с заданной

7 8 2 2.16 кодировкой множимого

4 3 4 Мн2/4 =111010,110111

3 0.64

4

2.56

Множитель

53 | 4 0.35 Мт4 = 311,112

4 13| 4 4

13 12 3 1.40 Мт2/4 = 110101,010110

12 1 4 *множитель представляется*

1 1.60 *обычным весомозначным*

4 *кодом:* 04 - 00, 14 - 01, 24 - 10, 34 - 11

2.40

Запишем сомножители в форме с плавающей запятой в прямом коде:

Мн = 0,111010110111 РМн = 0.0110 + 0310 – закодировано по заданию

Мт = 0,110101010110 РМт = 0.0011 + 0310 – 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:

РМн = 0.0110 034

РМт = 0.0011 034

РМн∙Мт = 0.0011 124

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 1. Преобразованный множитель имеет вид: Мтп4 = 111112 или Мтп2= 010101010110.

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

Таблица 1.1 - Перемножение мантисс

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Четверичная с/c | | | Двоично-четверичная с/с | | | Комментарии |
| **1** | | | **2** | | | **3** |
| 0. | 0000000 |  | 01. | 01 01 01 01 01 01 01 |  | ∑0 ч |
| 0. | 1133010 |  | 01. | 00 00 10 10 01 00 01 |  | П1Ч= Мн ∙ 2 |
| 0. | 1133010 |  | 01. | 00 00 10 10 01 00 01 |  | ∑1 ч |
| 0. | 0113301 | 0 | 01. | 01 00 00 10 10 01 00 | 01 | ∑1 ч∙ 2-2 |
| 0. | 0233202 |  | 01. | 01 11 10 10 11 01 11 |  | П2Ч = Мн ∙ 1 |
| 0. | 1013103 | 0 | 01. | 00 01 00 10 00 01 10 | 01 | ∑2 ч |
| 0. | 0101310 | 30 | 01. | 01 00 01 00 10 00 01 | 10 01 | ∑2 ч∙ 2-2 |
| 0. | 0233202 |  | 01. | 01 11 10 10 11 01 11 |  | П3Ч= Мн ∙ 1 |
| 0. | 1001112 | 30 | 01. | 00 01 01 00 00 00 11 | 10 01 | ∑3 ч |
| 0. | 0100111 | 230 | 01. | 01 00 01 01 00 00 00 | 11 10 01 | ∑3 ч∙ 2-2 |
| 0. | 0233202 |  | 01. | 01 11 10 10 11 01 11 |  | П4Ч= Мн ∙ 1 |
| 0. | 0333313 | 230 | 01. | 01 10 10 10 10 00 10 | 11 10 01 | ∑4 ч |
| 0. | 0033331 | 3230 | 01. | 01 01 10 10 10 10 00 | 10 11 10 01 | ∑4 ч∙ 2-2 |
| 0. | 0233202 |  | 01. | 01 11 10 10 11 01 11 |  | П5Ч= Мн ∙ 1 |
| 0. | 0333133 | 3230 | 01. | 01 10 10 10 00 10 10 | 10 11 10 01 | ∑5 ч |
| 0. | 0033313 | 33230 | 01. | 01 01 10 10 10 00 10 | 10 10 11 10 01 | ∑5 ч ∙ 2-2 |
| 3. | 3100132 |  | 10. | 10 00 01 01 00 10 11 |  | П6Ч= [-Мн]д |
| 3. | 3200111 | 33230 | 10. | 10 11 01 01 00 00 00 | 10 10 11 10 01 | ∑6 ч |
| 3. | 3320011 | 133230 | 10. | 10 10 11 01 01 00 00 | 00 10 10 11 10 01 | ∑6 ч∙ 2-2 |
| 0. | 0233202 |  | 01. | 01 11 10 10 11 01 11 |  | П7Ч= Мн ∙ 1 |
| 0. | 0213213 | 133230 | 01. | 01 11 00 10 11 00 10 | 00 10 10 11 10 01 | ∑7 ч |
| 0. | 0021321 | 3133230 | 01. | 01 01 11 00 10 11 00 | 10 00 10 10 11 10 01 | ∑7 ч∙ 2-2 |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн∙Мт4 = 0,213213133230, РМн∙Мт = 6) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн∙Мт4 = 213213,133230 РМн∙Мт = 0;

Мн∙Мт10 = 2535,5941

Результат прямого перемножения операндов дает следующее значение:

Мн10 · Мт10 = 47,54 · 53,35 = 2 536,259‬.

Абсолютная погрешность:

Δ = 2536,259 – 2535,5941 = 0,6649.

Относительная погрешность:

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

**2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ**

**СУММАТОРА-УМНОЖИТЕЛЯ**

Структура второго типа строится на базе заданных узлов ОЧУС 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 второго типа для алгоритма

умножения «А» приведена в приложении А.

*Если устройство работает как сумматор*,то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода *F*2 поступает «1». Необходимо обеспечить выполнение алгоритма сложения чисел, представленных в форме с плавающей запятой, базируясь на схеме умножителя, реализующего заданный алгоритм умножения.

Первое слагаемое переписывается в регистр результата под действием управляющих сигналов, поступающих на входы *h* всех ОЧУС (рисунок 1).

Если на вход *h* поступает «0», то ОЧУС перемножает разряды Мн и Мт и добавляет к полученному результату перенос из предыдущего ОЧУС (см. рисунок 2.1).

Мн ∙ Мт + перенос

Слагаемое

Перенос

“0”

ОЧУС

“0”

ОЧУС

h = 0

h = 1

Слагаемое “0”

Мн Мт

Рис. 2.1. Режимы работы ОЧУС

В ОЧС первое слагаемое складывается с нулём, записанным в регистре

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

Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание (01̅̅̅̅). В этом случае инициализируется управляющий вход *F*1 формирователя дополнительного кода и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на «– 1»).

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

Режимы работы формирователя дополнительного кода - Таблица 2.1.

|  |  |  |
| --- | --- | --- |
| Сигналы 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 |

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:

3 ∙ 2 = 1 2

max max max

Мн Мт перенос

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

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

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

**3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ**

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

Одноразрядный четверичный умножитель-сумматор — это комбинационное устройство, имеющее 6 входов (2 разряда из регистра Мн, 2 разряда из регистра Мт, вход переноса и управляющий вход h) и 3 выхода. Принцип работы ОЧУС описывается с помощью таблицы истинности (табл.3.1.1).

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

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

Таблица 3.1.1 - Таблица истинности ОЧУС

| Пер. | Мн  Мт | | Мт  Перенос | | Упр. | Перенос | Результат  Мт | | Результат операции |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Р1* | *x1* | *x2* | *y1* | *y2* | *h* | *P* | *Q1* | *Q2* | в четверичной с/c |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 ∙ 0 + 0 = 00 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Выход 01 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 ∙ 1 + 0 = 01 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Выход 01 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 ∙ 2 + 0 = 02 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | Выход 01 |
| 0 | 0 | 0 | 1 | 1 | 0 | x | x | x | 1 ∙ 3 + 0 = 03 |
| 0 | 0 | 0 | 1 | 1 | 1 | x | x | x | Выход 01 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 ∙ 0 + 0 = 00 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Выход 00 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 ∙ 1 + 0 = 00 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | Выход 00 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 ∙ 2 + 0 = 00 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Выход 00 |
| 0 | 0 | 1 | 1 | 1 | 0 | x | x | x | 0 ∙ 3 + 0 = 00 |
| 0 | 0 | 1 | 1 | 1 | 1 | x | x | x | Выход 00 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 ∙ 0 + 0 = 00 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | Выход 03 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 3 ∙ 1 + 0 = 03 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Выход 03 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 3 ∙ 2 + 0 = 12 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Выход 03 |
| 0 | 1 | 0 | 1 | 1 | 0 | х | х | х | 3 ∙ 3 + 0 = 21 |

*Продолжение таблицы 3.1.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 1 | 0 | 1 | 1 | 1 | х | х | х | Выход 03 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 2 ∙ 0 + 0 = 00 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Выход 02 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 2 ∙ 1 + 0 = 02 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | Выход 02 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 2 ∙ 2 + 0 = 10 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | Выход 02 |
| 0 | 1 | 1 | 1 | 1 | 0 | х | х | х | 2 ∙ 3 + 0 = 12 |
| 0 | 1 | 1 | 1 | 1 | 1 | х | х | х | Выход 02 |
| 1 | 0 | 0 | 0 | 0 | 0 | х | х | х | 1 ∙ 0 + 1 = 01 |
| 1 | 0 | 0 | 0 | 0 | 1 | х | х | х | Выход 01 |
| 1 | 0 | 0 | 0 | 1 | 0 | х | х | х | 1 ∙ 1 + 1 = 10 |
| 1 | 0 | 0 | 0 | 1 | 1 | х | х | х | Выход 01 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 ∙ 2 + 1 = 03 |
| 1 | 0 | 0 | 1 | 0 | 1 | х | х | х | Выход 01 |
| 1 | 0 | 0 | 1 | 1 | 0 | х | х | х | 1 ∙ 3 + 1 = 10 |
| 1 | 0 | 0 | 1 | 1 | 1 | х | х | х | Выход 01 |
| 1 | 0 | 1 | 0 | 0 | 0 | х | х | х | 0 ∙ 0 + 1 = 01 |
| 1 | 0 | 1 | 0 | 0 | 1 | х | х | х | Выход 00 |
| 1 | 0 | 1 | 0 | 1 | 0 | х | х | х | 0 ∙ 1 + 1 = 01 |
| 1 | 0 | 1 | 0 | 1 | 1 | х | х | х | Выход 00 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 ∙ 2 + 1 = 01 |
| 1 | 0 | 1 | 1 | 0 | 1 | х | х | х | Выход 00 |
| 1 | 0 | 1 | 1 | 1 | 0 | х | х | х | 0 ∙ 3 + 1 = 01 |
| 1 | 0 | 1 | 1 | 1 | 1 | х | х | х | Выход 00 |
| 1 | 1 | 0 | 0 | 0 | 0 | х | х | х | 3 ∙ 0 + 1 = 01 |
| 1 | 1 | 0 | 0 | 0 | 1 | х | х | х | Выход 03 |
| 1 | 1 | 0 | 0 | 1 | 0 | х | х | х | 3 ∙ 1 + 1 = 10 |
| 1 | 1 | 0 | 0 | 1 | 1 | х | х | х | Выход 03 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 3 ∙ 2 + 1 = 13 |
| 1 | 1 | 0 | 1 | 0 | 1 | х | х | х | Выход 03 |
| 1 | 1 | 0 | 1 | 1 | 0 | х | х | х | 3 ∙ 3 + 1 = 22 |
| 1 | 1 | 0 | 1 | 1 | 1 | х | х | х | Выход 03 |
| 1 | 1 | 1 | 0 | 0 | 0 | х | х | х | 2 ∙ 0 + 1 = 01 |
| 1 | 1 | 1 | 0 | 0 | 1 | х | х | х | Выход 02 |
| 1 | 1 | 1 | 0 | 1 | 0 | х | х | х | 2 ∙ 1 + 1 = 03 |
| 1 | 1 | 1 | 0 | 1 | 1 | х | х | х | Выход 02 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 2 ∙ 2 + 1 = 11 |
| 1 | 1 | 1 | 1 | 0 | 1 | х | х | х | Выход 02 |
| 1 | 1 | 1 | 1 | 1 | 0 | х | х | х | 2 ∙ 3 + 1 = 13 |
| 1 | 1 | 1 | 1 | 1 | 1 | х | х | х | Выход 02 |

Управляющий вход *h* определяет тип операции: 0 – умножение закодированных цифр, поступивших на информационные входы, и добавление переноса; 1 – вывод на выходы без изменения значения разрядов, поступивших из регистра множимого.

В табл. 3.1.1 выделено 36 безразличных наборов, так как 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 функции *Q1* проведем с помощью алгоритма Рота. Минимизацию функций *P* и *Q2*проведем при помощи карт Вейча.

Для функции *P*:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 |  |  | 1 |  |  |  |  |  |  |  |
|  | 1 | x | x | 1 |  | x | x |  |  |  |  |
|  | x | x | x | x | x | x | x | x |  |  |
|  | x | x | x | x | x | x | x | x |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | x | x | x | x | x | x | x | x |  |  |  |
|  |  | x | x | x | x | x | x | x | x |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  | *h* | |  |  | *h* | |  |  |  |  |

Рисунок 3.1.1 – Минимизация функции при помощи карты Вейча

Функция для реализации в заданном базисе (А7) будет иметь вид:

Для функции *Q2*:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 0 |  |  |  |  | 0 |  |  |  |  |
|  | 0 | x | x |  |  | x | x | 0 |  |  |  |
|  | x | x | x | x | x | x | x | x |  |  |
|  | x | x | x | x | x | x | x | x |  |  |  |
|  |  | 0 | 0 |  |  |  |  | 0 | 0 |  |  |
|  |  | x | x | x | x | x | x | x | x |  |  |  |
|  |  | x | x | x | x | x | x | x | x |  |  |
|  |  |  | 0 |  |  |  |  | 0 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  | *h* | |  |  | *h* | |  |  |  |  |

Рисунок 3.1.2 – Минимизация функции *Q2* при помощи карты Вейча

Функция для реализации в заданном базисе (А7) будет иметь вид:

Для минимизации функции *П* воспользуемся алгоритмом Рота.

Определим множество единичных кубов:

Далее определим множество безразличных кубов:

Склеим до минимума множество N:

}

Сформируем множество :

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C0\*C0 | 000100 | 010001 | 010010 | 010011 | 010100 | 010101 | 011001 | 011010 | 011011 | 011101 | 100100 | 101100 | 110100 | 1xx110 | 0xx11x | 1xx1x1 | 1xx0xx |
| 000100 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 010001 | 0y0y0y | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 010010 | 0y0yy0 | 0100yy | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 010011 | 0y0yyy | 0100y1 | 01001y | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 010100 | 0y0100 | 010y0y | 010yy0 | 010yyy | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 010101 | 0y010y | 010y01 | 010yyy | 010yy1 | 01010y | - |  |  |  |  |  |  |  |  |  |  |  |
| 011001 | 0yyy0y | 01y001 | 01y0yy | 01y0y1 | 01yy0y | 01yy01 | - |  |  |  |  |  |  |  |  |  |  |
| 011010 | 0yyyy0 | 01y0yy | 01y010 | 01y01y | 01yyy0 | 01yyyy | 0110yy | - |  |  |  |  |  |  |  |  |  |
| 011011 | 0yyyyy | 01y0y1 | 01y01y | 01y011 | 01yyyy | 01yyy1 | 0110y1 | 01101y | - |  |  |  |  |  |  |  |  |
| 011101 | 0yy10y | 01yy01 | 01yyyy | 01yyy1 | 01y10y | 01y101 | 011y01 | 011yyy | 011yy1 | - |  |  |  |  |  |  |  |
| 100100 | y00100 | yy0y0y | yy0yy0 | yy0yyy | yy0100 | yy010y | yyyy0y | yyyyy0 | yyyyyy | yyy10y | - |  |  |  |  |  |  |
| 101100 | y0y100 | yyyy0y | yyyyy0 | yyyyyy | yyy100 | yyy10y | yy1y0y | yy1yy0 | yy1yyy | yy110y | 10y100 | - |  |  |  |  |  |
| 110100 | yy0100 | y10y0y | y10yy0 | y10yyy | y10100 | y1010y | y1yy0y | y1yyy0 | y1yyyy | y1y10y | 1y0100 | 1yy100 | - |  |  |  |  |
| 1xx110 | y001y0 | y10yyy | y10y10 | y10y1y | y101y0 | y101yy | y11yyy | y11y10 | y11y1y | y111yy | 1001y0 | 1011y0 | 1101y0 | - |  |  |  |
| 0xx11x | 0001y0 | 010yy1 | 010y10 | 010y11 | 0101y0 | 0101y1 | 011yy1 | 011y10 | 011y11 | 0111y1 | y001y0 | y011y0 | y101y0 | yxx110 | - |  |  |
| 1xx1x1 | y0010y | y10y01 | y10y1y | y10y11 | y1010y | y10101 | y11y01 | y11y1y | y11y11 | y11101 | 10010y | 10110y | 11010y | 1xx11y | yxx111 | - |  |
| 1xx0xx | y00y00 | y10001 | y10010 | y10011 | y10y00 | y10y01 | y11001 | y11010 | y11011 | y11y01 | 100y00 | 101y00 | 110y00 | 1xxy10 | yxxy1x | 1xxyx1 | - |
| A1 | 0x0100 x00100 0001x0 | 0100x1 010x01 01x001 x10001 | 01001x 01x010 010x10 x10010 | 01x011 010x11 x10011 | 01010x x10100 0101x0 | 01x101 0101x1 x10101 | 0110x1 011x01 x11001 | 01101x 011x10 x11010 | 011x11 x11011 | 0111x1 x11101 | 10x100 1x0100 1001x0 10010x 100x00 | 1011x0 10110x 101x00 | 1101x0 11010x 110x00 | xxx110 1xx11x 1xxx10 | xxx111 | 1xxxx1 | Ø |

Таблица 3.1.2 - Поиск простых импликант ()

Из таблицы 3.2.2 – Поиск простых импликант () следует:

Множество кубов, не участвовавших в образовании новых кубов, пустое.

Сформируем множество :

После этой операции сформируется новое множество кубов .

Таблица 3.1.3 поиска простых импликант С1 \*С1 показана в приложении Б.

Из таблицы поиска простых импликант () следует:

Множество кубов, не участвовавших в образовании новых кубов, пустое.

Сформируем множество :

После этой операции сформируется новое множество кубов .

Таблица 3.1.4 поиска простых импликант С2 \*С2 показана в приложении В.

Из таблицы поиска простых импликант () следует:

Множество кубов, не участвовавших в образовании новых кубов, пустое.

Сформируем множество :

После этой операции сформируется новое множество кубов .

Таблица 3.1.4 поиска простых импликант С3 \*С3 показана в приложении Г.

Из таблицы поиска простых импликант () следует:

После этой операции сформируется новое множество кубов :

Таблица 3.1.5 - Поиск простых импликант ()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C4\*C4 | x1xxx1 | x1xx1x | 10xxxx | 1x0xxx |
| x1xxx1 | - |  |  |  |
| x1xx1x | x1xx11 | - |  |  |
| 10xxxx | 1yxxx1 | 1yxx1x | - |  |
| 1x0xxx | 110xx1 | 110x1x | 100xxx | - |
| 1xxx1x | 11xx11 | 11xx1x | 10xx1x | 1x0x1x |
| xxx11x | x1x111 | x1x11x | 10x11x | 1x011x |
| 1xxxx1 | 11xxx1 | 11xx11 | 10xxx1 | 1x0xx1 |
| 1xx0xx | 11x0x1 | 11x01x | 10x0xx | 1x00xx |
| A5 | Ø | Ø | Ø | Ø |

Из таблицы 3.1.5 – Поиск простых импликант () следует:

На этом этап поиска простых импликант заканчивается т.к. .

Конечное множество простых импликант:

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

В таблице 3.2.6 из каждой простой импликанты поочередно вычитаются все остальные простые импликанты Z#(Z\z), результат операции (последняя строка таблицы) указывает на то, что *L*-экстремалями стали следующие простые импликанты:

Таблица 3.2.6 поиска *L*-экстремалей показана в приложении Д.

Необходимо проверить, нет ли среди полученных *L*-экстремалей таких, которые стали *L*-экстремалями за счет 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)).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) n L | 000100 | 010001 | 010010 | 010011 | 010100 | 010101 | 011001 | 011010 | 011011 | 011101 | 100100 | 101100 | 110100 |
| 000100 | 000100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 011x01 | Ø | Ø | Ø | Ø | Ø | Ø | 011001 | Ø | Ø | 011101 | Ø | Ø | Ø |
| 01x001 | Ø | 010001 | Ø | Ø | Ø | Ø | 011001 | Ø | Ø | Ø | Ø | Ø | Ø |
| 011001 | Ø | Ø | Ø | Ø | Ø | Ø | 011001 | Ø | Ø | Ø | Ø | Ø | Ø |
| 011010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 011010 | Ø | Ø | Ø | Ø | Ø |
| 01x010 | Ø | Ø | 010010 | Ø | Ø | Ø | Ø | 011010 | Ø | Ø | Ø | Ø | Ø |
| 011010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 011010 | Ø | Ø | Ø | Ø | Ø |
| 101100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 101100 | Ø |
| 00111x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 001110 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 00x111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 001111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 111000 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Таблица 3.2.8 – Поиск непокрытых исходных наборов.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 000100 | 010001 | 010010 | 010011 | 010100 | 010101 | 011001 | 011010 | 011011 | 011101 | 100100 | 101100 | 110100 |
| xx01x0 | Ø | 010001 | 010010 | 010011 | Ø | 010101 | 011001 | 011010 | 011011 | 011101 | Ø | 101100 | Ø |
| x1xxx1 | Ø | Ø | 010010 | Ø | Ø | Ø | Ø | 011010 | Ø | Ø | Ø | 101100 | Ø |
| x1xx1x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 101100 | Ø |
| 10xxxx | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| Остаток | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Таблица 3.2.7 - Проверка L-экстремалей

Далее необходимо проанализировать, какие из исходных единичных кубов (множество L) не покрыты найденной L-экстремалью. Этот анализ осуществляется с помощью таблицы 3.2.8.

Таким образом, минимальное покрытие - множество L-экстремалей *E*:

Функция для реализации в заданном базисе (А7) будет иметь вид:

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

Функциональная схема ОЧУC в заданном базисе показана в приложении Е.

**3.2. Логический синтез одноразрядного четверичного сумматора**

Одноразрядный четверичный сумматор – это комбинационное устройство, имеющее 5 входов (2 разряда одного слагаемого, 2 разряда второго слагаемого и вход переноса) и 3 выхода. Принцип работы ОЧС представлен с помощью таблицы истинности (табл. 3.2.1).

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

В таблице 3.2.1 ОЧС синтезируется для схемы 2-го типа, поэтому безразличные наборы в таблице истинности отсутствуют.

Таблица 3.2.1 – 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 | 1 | 1 | 1+1+0=02 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1+1+1=03 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1+0+0=01 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1+0+1=02 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1+3+0=10 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1+3+1=11 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1+2+0=03 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1+2+1=10 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0+1+0=01 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0+1+1=02 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0+0+0=00 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0+0+1=01 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0+3+0=03 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0+3+1=10 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0+2+0=02 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0+2+1=03 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 3+1+0=10 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 3+1+1=11 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 3+0+0=03 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 3+0+1=10 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 3+3+0=12 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 3+3+1=13 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 3+2+0=11 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3+2+1=12 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2+1+0=03 |

*Продолжение таблицы 3.2.1*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 2+1+1=10 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 2+0+0=02 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 2+0+1=03 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 2+3+0=11 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 2+3+1=12 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 2+2+0=10 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 2+2+1=11 |

Минимизацию функций П, S*1* и S*2* проведём с помощью карт Карно. Заполненные карты для функций П, S*1* и S*2* приведены на рисунках 3.2.1, 3.2.2 и 3.2.3 соответственно.

Для функции *П*:

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

Рисунок 3.2.1 – Минимизация функции *П*при помощи карты Карно.

Функция для реализации в заданном базисе (А6) будет иметь вид:

Для функции *S1*:

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

Рисунок 3.2.2 – Минимизация функции *S1* при помощи карты Карно.

Функция для реализации в заданном базисе (А6) будет иметь вид:

Для функции *S2*:

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

Рисунок 3.2.3 – Минимизация функции *S2* при помощи карты Карно.

Функция для реализации в заданном базисе (А6) будет иметь вид:

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

Функциональная схема ОЧУC в заданном базисе показана в приложении Ж.

**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 должно выполняться условие *n* = .

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 ОЧC приведен 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 | 0 | “0” | 1 | +p | 1 |  |
| 000 | 0 | 1 | 0 | 1 | 0 |
| 000 | 1 | 0 | 0 | 0 | 0 |
| 000 | 1 | 1 | 0 | 1 | 1 |
| 001 | 0 | 0 | 1 | +p | 0 | b2 | 1 |  |
| 001 | 0 | 1 | 1 | 0 | 0 |
| 001 | 1 | 0 | 0 | 1 | 0 |
| 001 | 1 | 1 | 1 | 0 | 1 |
| 010 | 0 | 0 | 0 | “0” | 0 | p | 0 | p+b2 |
| 010 | 0 | 1 | 0 | 1 | 1 |
| 010 | 1 | 0 | 0 | 0 | 1 |
| 010 | 1 | 1 | 0 | 0 | 0 |
| 011 | 0 | 0 | 0 | p | 1 | b2+ | 0 | p+b2 |
| 011 | 0 | 1 | 1 | 0 | 1 |
| 011 | 1 | 0 | 0 | 1 | 1 |
| 011 | 1 | 1 | 0 | 1 | 0 |

*Продолжение таблицы 4.1*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 0 | 0 | 1 | +p | 0 | b2 | 1 |  |
| 100 | 0 | 1 | 1 | 0 | 0 |
| 100 | 1 | 0 | 0 | 1 | 0 |
| 100 | 1 | 1 | 1 | 0 | 1 |
| 101 | 0 | 0 | 1 | “1” | 1 | +p | 1 |  |
| 101 | 0 | 1 | 1 | 1 | 0 |
| 101 | 1 | 0 | 1 | 0 | 0 |
| 101 | 1 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 0 | p | 1 | b2+ | 0 | p+b2 |
| 110 | 0 | 1 | 1 | 0 | 1 |
| 110 | 1 | 0 | 0 | 1 | 1 |
| 110 | 1 | 1 | 0 | 1 | 0 |
| 111 | 0 | 0 | 1 | “1” | 0 | p | 0 | p+b2 |
| 111 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 0 | 1 | 0 | 1 |
| 111 | 1 | 1 | 1 | 0 | 0 |

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 10.

*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 выходные функции картами Карно.

*Таблица 5.2 – Минимизация функции P*

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

*Таблица 5.3 – Минимизация функции*

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

*Таблица 5.4 – Минимизация функции*

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

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 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-Вейча 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.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Луцик Ю.А., Лукьянова И.В. – Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". – Минск: БГУИР, 2014 г.

Луцик Ю.А., Лукьянова И.В. – Методические указания к курсовому проекту по курсу “Арифметические и логические основы вычислительной техники”. – Мн.: БГУИР, 2004 г.

**ПРИЛОЖЕНИЕ А**

*(обязательное)*

Сумматор-умножитель второго типа. Схема электрическая структурная

**ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Поиск простых импликант *C1 \* C1*

**ПРИЛОЖЕНИЕ В**

*(обязательное)*

Поиск простых импликант *C2 \* C2*

**ПРИЛОЖЕНИЕ Г**

*(обязательное)*

Поиск простых импликант *C3 \* C3*

**ПРИЛОЖЕНИЕ Д**

*(обязательное)*

Поиск L-экстремалей

**ПРИЛОЖЕНИЕ Е**

*(обязательное)*

Одноразрядный четверичный умножитель-сумматор.

Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Ж**

*(обязательное)*

Одноразрядный четверичный сумматор. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ З**

*(обязательное)*

Одноразрядный четверичный сумматор.

Реализация на мультиплексорах.

Схема электрическая функциональная

**ПРИЛОЖЕНИЕ И**

*(обязательное)*

Преобразователь множителя. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ К**

*(обязательное)*

Ведомость документов