Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы цифровых устройств

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_\_\_\_\_\_И. В. Лукьянова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 529 ПЗ

Студент

Руководитель

М. А. Тюшкевич

И. В. Лукьянова

МИНСК 2022

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы цифровых устройств

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_ Б. В. Никульшин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2022 г.

ЗАДАНИЕ

по курсовой работе студента  
 Тюшкевича Максима Андреевича

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел»
2. Срок сдачи студентом законченной работы: 20 мая 2022 г.
3. Исходные данные к работе:

**3.1** исходные сомножители: Мн = 2,66; Мт = 88,47;

* 1. алгоритм умножения: Б;

**3.3** метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямом коде;

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

четверичной системе кодирования: 0– 11, 1 – 01, 2 – 10, 3– 11;

**3.5** Тип синтезируемого умножителя: 2;

**3.6** Логический базис для синтеза ОЧС: ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ, КОНСТАНТНАЯ ЕДИНИЦА; метод минимизации – карты Карно, алгоритм Рота;

**3.7** Логический базис для синтеза ОЧУС: И-НЕ; метод минимизации – карты Вейча;

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

вопросов):

Введение. 1. Разработка алгоритма умножения. 2. Разработка структурной

схемы сумматора-умножителя. 3. Разработка функциональных схем

основных узлов сумматора-умножителя. 4. Логический синтез одноразрядного четверичного сумматора на основе мультиплексоров. 5. Логический синтез преобразователя множителя. 6. Оценка результатов разработки. Заключение. Список использованных источников.

**5** Перечень графического материала:

**5.1** Сумматор-умножитель. Схема структурная.

**5.2** ОЧС. Схема электрическая функциональная.

**5.3** ОЧУС. Схема электрическая функциональная.

**5.4** ОЧС на мультиплексорах. Схема электрическая функциональная.

**5.5** Преобразователь множителя. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объем 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 чертежа |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 | 01.05-15.05 | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 | 15.05-20.05 |  |

Дата выдачи задания: 10 февраля 2022 г.

Руководитель И. В. Лукьянова

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

**СОДЕРЖАНИЕ**

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

1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ …………………………… 6

1.1. Перевод сомножителей из десятичной системы счисления в четверичную …………………………………………………………………… 6

2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА­- УМНОЖИТЕЛЯ …………………………………………..……………………. 9

3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ ……………………………………..…….. 12

3.1. Логический синтез одноразрядного четверичного сумматора .…. 12

3.2. Логический синтез одноразрядного четверичного умножителя-сумматора ……………………………………………………………………… 20

4. ЛОГИЧЕСКИЙ СИНТЕЗ ОДНОРАЗРЯДНОГО ЧЕТВЕРИЧНОГО СУММАТОРА НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ …………………… 24

5. ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ………………………………………………………………. 26

6. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ .………….…………………. 29

6.1. Расчет времени умножения на один разряд множителя ….……. 29

6.2. Расчет времени умножения на n разрядов множителя ………… 29

6.3. Расчет эффективности минимизации …………………………… 30

ЗАКЛЮЧЕНИЕ …………………………………………………………….. 31

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ ...………………….… 32

ПРИЛОЖЕНИЕ А ………………………………...………………………… 33

ПРИЛОЖЕНИЕ Б ………………………………..………………………… 34

ПРИЛОЖЕНИЕ В ………………………………..………………………… 35

ПРИЛОЖЕНИЕ Г …………………………………..……………………… 36

ПРИЛОЖЕНИЕ Д ………………………………..………………………… 37

ПРИЛОЖЕНИЕ Е ……………………….………..………………………… 38

# **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

2 | 4 0.66 Мн4 = 2,22203

2 0 4 в соответствии с заданной

0 2.64 кодировкой множимого

4 Мн2/4 = 10,1010101100

2.56

4

2.24

4

0.96

4

3.64

Множитель

88 | 4 0.47 Мт4 = 1120,13

89 22 | 4 4

0 20 5 | 4 1.88 Мт2/4 = 01011000,0111

2 4 1 4 *множитель представляется*

1 3.52 *обычным весомозначным*

*кодом.*

Запишем сомножители в форме с плавающей запятой в прямом коде:

Мн = 0,101010101100 РМн = 0.1101 + 014 – закодировано по заданию

Мт = 0,010110000111 РМт = 0.0100 + 104 – 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 = 0.1101 01

РМт = 0.0100 10

РМн∙Мт = 0.0101 11

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

Таблица 1.1 - Перемножение мантисс

|  |  |  |
| --- | --- | --- |
| **Четверичная с/c** | **Двоично-четверичная с/с** | **Комментарии** |
| **1** | **2** | **3** |
| 0.000000000000 | 11.11 11 11 11 11 11 11 11 11 11 11 11 |  |
| 3.333333111131 | 00.00 00 00 00 00 00 01 01 01 01 11 01 |  |
| 3.333333111131 | 00.00 00 00 00 00 00 01 01 01 01 11 01 |  |
| 0.000011110120 | 11.11 11 11 11 01 01 01 01 11 01 10 11 |  |
| 0.000010221311 | 11.11 11 11 11 01 11 10 10 01 00 01 01 |  |
| 0.000000000000 | 11.11 11 11 11 11 11 11 11 11 11 11 11 |  |
| 0.000010221311 | 11.11 11 11 11 01 11 10 10 01 00 01 01 |  |
| 0.001111012000 | 11.11 11 01 01 01 01 11 01 10 11 11 11 |  |
| 0.001121233311 | 11.11 11 01 01 10 01 10 00 00 00 01 01 |  |
| 0.002222030000 | 11.11 11 10 10 10 10 11 00 11 11 11 11 |  |
| 0.010003323311 | 11.11 01 11 11 11 00 00 10 00 00 01 01 |  |
| 0.022220300000 | 11.11 10 10 10 10 11 00 11 11 11 11 11 |  |
| 0.032230223311 | 11.11 00 10 10 00 11 10 10 00 00 01 01 |  |

Результат произведения денормализован вправо. После нормализации порядок результата равен РМн∙Мт =

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн∙Мт4 = 0,32230223311, РМн∙Мт = 4) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн∙Мт4 = 3223,0223311 РМн∙Мт = 0;

Мн∙Мт10 = 235,1712

Результат прямого перемножения операндов дает следующее значение:

Мн10 · Мт10 = 52,26 · 83,31 = 235,3302.

Абсолютная погрешность:

Δ = 235,3302 – 235,1712= 0,209.

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**

Структурная схема сумматора-умножителя второго типа для алгоритма умножения «Б» представлена на чертеже ГУИР.400201.529 Э1 (Приложение А).

Структурная схема сумматора-умножителя строится на базе заданных узлов ОЧУС, ОЧС, формирователя дополнительного кода (ФДК), преобразователя множителя (ПМ) и регистра результата. Управление режимами работы устройства осуществляется с помощью внешнего сигнала Mul/sum, который определяет вид текущей арифметической операции (умножение или сложение).

Если устройство работает как сумматор, то есть на вход Mul/sum поступает «1», то оба слагаемых последовательно заносятся в регистр множимого, а на управляющий вход ФДК F2 поступает «1». Необходимо обеспечить выполнение алгоритма сложения чисел, представленных в форме с плавающей запятой, базируясь на схеме умножителя, реализующего заданный алгоритм умножения.

На выходах ФДК формируется дополнительный код одного из слагаемых с учетом знака. Это слагаемое может быть записано в регистр результата, при этом управляющие сигналы, поступающие на входы h всех ОЧУС, дают возможность переписать на выходы ОЧУС разряды слагаемого без изменений (см. рисунок 2.1).

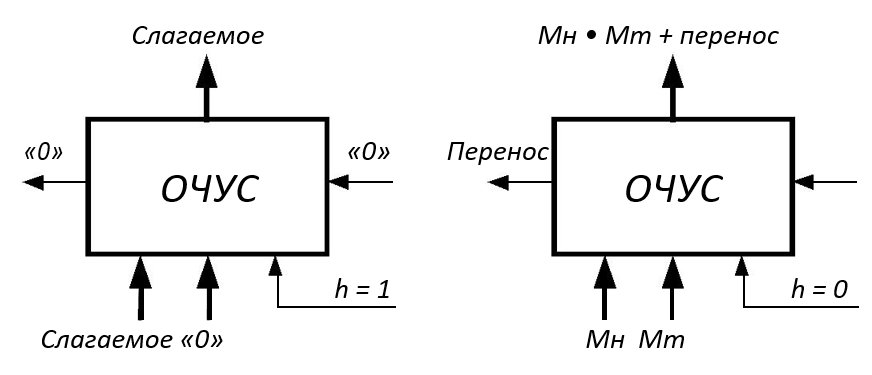


Рисунок 2.1 – Режимы работы ОЧУС

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

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

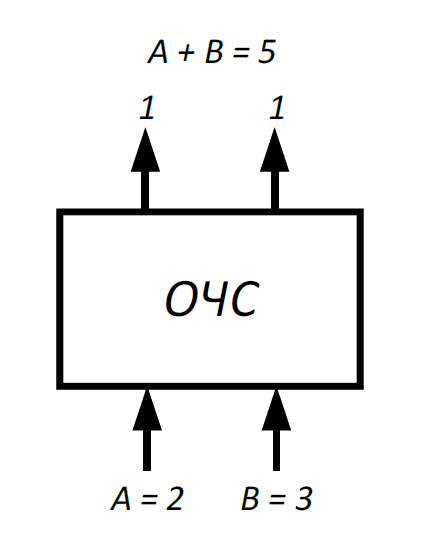


Рисунок 2.2 – Одноразрядный четверичный сумматор

В ОЧС первое слагаемое складывается с нулем, записанным в регистре результата и, соответственно, без изменений заносится в регистр результата.

Сумма хранится в регистре результата. Разрядность регистра результата должна быть на единицу больше, чем количество разрядов исходных слагаемых. Это объясняется возможным переносом при сложении.

Если устройство работает как умножитель, то есть на вход Mul/sum поступает «0», то множимое и множитель заносятся в соответствующие регистры, а на управляющий вход ФДК F2 поступает «0».

Диада множителя поступает на входы преобразователя множителя. Единица переноса в следующую диаду, если она возникает, должна быть добавлена к следующей диаде множителя (выход 1 ПМ) в следующем такте, т. е. должна храниться на триггере до следующего такта.

В регистре множимого после каждого такта умножения содержимое сдвигается на два двоичных разряда и в конце умножения регистр обнуляется.

Таблица 2.1 – Режимы работы формирователя дополнительного кода

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |  |

Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание (). В этом случае инициализируется управляющий вход ФДК, и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на -1).

На выходах 3 и 4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУС вместе с диадами множимого. На трех 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»):

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | ∙ 2 = | 1 2 (+1 в случае переноса из | |
| max | max | max | предыдущего ОЧУС) |
| 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 нулем).

Частичные суммы хранятся в регистре результата. Количество тактов умножения определяется разрядностью Мт.

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

**3.1. Логический синтез одноразрядного четверичного сумматора**

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

Принцип работы ОЧС представлен с помощью таблицы истинности.

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

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

Таблица 3.1 – Таблица истинности ОЧС

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **P** | **П** |  |  | **Пример в четверичной с/c** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 3+3+0=12 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 3+3+1=13 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 3+1+0=10 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 3+1+1=11 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 3+2+0=11 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 3+2+1=12 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 3+0+0=03 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3+0+1=10 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1+3+0=10 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1+3+1=11 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1+1+0=02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1+1+1=03 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1+2+0=03 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1+2+1=10 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1+0+0=01 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1+0+1=02 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2+3+0=11 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2+3+1=12 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2+1+0=03 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2+1+1=10 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 2+2+0=10 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 2+2+1=11 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 2+0+0=02 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 2+0+1=03 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0+3+0=03 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0+3+1=10 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0+1+0=01 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0+1+1=02 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0+2+0=02 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 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 |

Функции П, , минимизируем с помощью карт Карно.

Функцию П минимизируем с помощью алгоритма Рота.

Таблица 3.1.1 – Минимизация функции П

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

В заданном базисе:

Таблица 3.1.2 – Минимизация функции

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

В заданном базисе:

Таблица 3.1.3 – Минимизация функции

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

В заданном базисе:

Функцию П минимизируем алгоритмом Рота:

Определим множество единичных кубов

L = .

Поскольку ОЧС синтезируется для схемы 2-го типа, то безразличные кубы отсутствуют.

Сформируем множество :

= .

Первым этапом алгоритма Рота является нахождение множества простых импликант. Для этого будем использовать операцию умножения над множествами *С0*, *С1* и т. д., пока в результате операции будут образовываться новые кубы большей размерности.

Первый шаг умножения (С0\*С0) показан в таблице 3.1.4.

Таблица 3.1.4 – Поиск простых импликант С0\*С0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С0\*С0 | 00000 | 00001 | 00010 | 00011 | 00100 | 00101 | 00111 | 01000 | 01001 | 01101 | 10000 | 10001 | 10011 | 10100 | 10101 | 11001 |
| 00000 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00001 | 0000y | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00010 | 000y0 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00011 |  | 000y1 | 0001y | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 00100 | 00y00 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 00101 |  | 00y01 |  |  | 0010y | - |  |  |  |  |  |  |  |  |  |  |
| 00111 |  |  |  | 00y11 |  | 001y1 | - |  |  |  |  |  |  |  |  |  |
| 01000 | 0y000 |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 01001 |  | 0y001 |  |  |  |  |  | 0100y | - |  |  |  |  |  |  |  |
| 01101 |  |  |  |  |  | 0y101 |  |  | 01y01 | - |  |  |  |  |  |  |
| 10000 | y0000 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 10001 |  | y0001 |  |  |  |  |  |  |  |  | 1000y | - |  |  |  |  |
| 10011 |  |  |  | y0011 |  |  |  |  |  |  |  | 100y1 | - |  |  |  |
| 10100 |  |  |  |  | y0100 |  |  |  |  |  | 10y00 |  |  | - |  |  |
| 10101 |  |  |  |  |  | y0101 |  |  |  |  |  | 10y01 |  | 1010y | - |  |
| 11001 |  |  |  |  |  |  |  |  | y1001 |  |  | 1y001 |  |  |  | - |

В результате умножения сформировано множество :

=.

После умножения простые импликанты не выявлены, следовательно .

Сформируем множества = /= и .

=.

Второй шаг умножения (С1\*С1) показан в таблице 3.1.5.

В результате умножения сформировано множество :

=.

После умножения простые импликанты не выявлены, следовательно .

Сформируем множества = /=и .

=.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С1\*С1 | 0000x | 000x0 | 00x00 | 0x000 | x0000 | 000x1 | 00x01 | 0x001 | x0001 | 0001x | 00x11 | x0011 | 0010x | x0100 | 001x1 | 0x101 | x0101 | 0100x | 01x01 | x1001 | 1000x | 10x00 | 100x1 | 10x01 | 1x001 | 1010x |
| 0000x | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 000x0 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x00 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x000 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0000 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 000x1 |  | 000xy |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x01 |  |  | 00x0y |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x001 |  |  |  | 0x00y |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0001 |  |  |  |  | x000y |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0001x | 000yx |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x11 |  |  |  |  |  |  | 00xy1 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0011 |  |  |  |  |  |  |  |  | x00y1 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0010x | 00y0x |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0100 |  |  |  |  | x0y00 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 001x1 |  |  |  |  |  | 00yx1 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 0x101 |  |  |  |  |  |  |  | 0xy01 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| x0101 |  |  |  |  |  |  |  |  | x0y01 |  |  |  |  | x010y |  |  | - |  |  |  |  |  |  |  |  |  |
| 0100x | 0y00x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 01x01 |  |  |  |  |  |  | 0yx01 |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| x1001 |  |  |  |  |  |  |  |  | xy001 |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 1000x | y000x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 10x00 |  |  | y0x00 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 100x1 |  |  |  |  |  | y00x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |
| 10x01 |  |  |  |  |  |  | y0x01 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 10x0y |  | - |  |  |
| 1x001 |  |  |  |  |  |  |  | yx001 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |
| 1010x |  |  |  |  |  |  |  |  |  |  |  |  | y010x |  |  |  |  |  |  |  | 10y0x |  |  |  |  | - |

Таблица 3.1.5 – Поиск простых импликант С1\*С1

Таблица 3.1.6 – Поиск простых импликант С2\*С2

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С1\*С1 | 000xx | 00x0x | 0x00x | x000x | x0x00 | 00xx1 | x00x1 | 0xx01 | x0x01 | xx001 | x010x | 10x0x |
| 000xx | - |  |  |  |  |  |  |  |  |  |  |  |
| 00x0x |  | - |  |  |  |  |  |  |  |  |  |  |
| 0x00x |  |  | - |  |  |  |  |  |  |  |  |  |
| x000x |  |  |  | - |  |  |  |  |  |  |  |  |
| x0x00 |  |  |  |  | - |  |  |  |  |  |  |  |
| 00xx1 |  |  |  |  |  | - |  |  |  |  |  |  |
| x00x1 |  |  |  |  |  |  | - |  |  |  |  |  |
| 0xx01 |  |  |  |  |  |  |  | - |  |  |  |  |
| x0x01 |  |  |  |  | x0x0y |  |  |  | - |  |  |  |
| xx001 |  |  |  |  |  |  |  |  |  | - |  |  |
| x010x |  |  |  | x0y0x |  |  |  |  |  |  | - |  |
| 10x0x |  | y0x0x |  |  |  |  |  |  |  |  |  | - |

Третий шаг умножения (С2\*С2) показан в таблице 3.1.6.

В результате умножения сформировано множество :

=.

В результате умножения сформировалось множество простых импликант :

= .

Сформируем множества = / и .

= .

= .

Третий шаг умножения (С3\*С3) показан в таблице 3.1.7.

Таблица 3.1.7 – Поиск простых импликант С3\*С3

|  |  |
| --- | --- |
| С3\*С3 | x0x0x |
| x0x0x | - |

Кубов новой размерности не образовалось.

.

.

Образуем множество простых импликант Z:

Z==.

Таблица 3.2.8 Поиск L-экстремалей.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 000xx | 0x00x | 00xx1 | x00x1 | 0xx01 | xx001 | x0x0x |
| 000xx | - | z1zzz  0100x | zz1zz  001x1 | 1zzzz  100x1 | z11zz  01x01  0x101 | 11zzz  1x001  x1001 | 1z1zz  10x0x  x010x |
| 0x00x | zzz1z  0001x | - | zzy1z  001x1 | yzz1z  100x1 | zz1zz  01101  zzyzz  0z101 | yzzzz  1x001  1zzzz  11001 | yz1zz  10x0x  1zyzz  x010x |
| 00xx1 | zzzz0  00010 | zyzz0  0100x | - | yzzzz  100x1 | zyzzz  01101  z1zzz  01101 | y1zzz  1x001  yyzzz  11001 | yzzz0  10x0x  1zzz0  1010x  x0100 |
| x00x1 | zzzzy  00010 | zyzzz  0100x | zzyzz  001x1 | - | zyyzz  01101 | z1zzz  11001  zyzzz  11001 | zz1zz  1010x  zzyz0  1010x  z0yzy  x0100 |
| 0xx01 | zzzyz  00010 | zzzz0  01000 | zzz1z  00111 | yzz1z  100x1 | - | yzzzz  11001 | yzzz0  1010x  yzzz0  1010x  1zzzy  x0100 |
| xx001 | zzzyy  00010 | zzzzy  01000 | zzyyz  00111 | zzz1z  10011 | zzyzz  01101 | - | xxyz0  1010x  zzyzy  x0100 |
| x0x0x | zzzyz  00010 | zyzzz  01000 | zzzyz  00111 | zzzyz  10011 | z0zzz  01101 | zyzzz  11001 | - |
| Остаток | 00010 | 01000 | 00111 | 10011 | 01101 | 11001 | 1010x  x0100 |

L-экстремали не были найдены, следовательно множество простых импликант Z является конечным и минимальная ДНФ будет выглядеть так:

Функциональная схема ОЧС представлена на чертеже ГУИР.400201.529 Э2.1 (Приложение Б).

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

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 ОЧУС описывается с помощью таблицы истинности (таблица 3.2).

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

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

Таблица 3.2 – Таблица истинности ОЧУС

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Пер.** | **Мн** | | **Мт** | | **Упр.**  **h** | **Пер.**  **p** | **Результат** | | **Пример в четверичной с/c** |
|  |  |  |  |  |  |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 30+0=00 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 31+0=03 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 32+0=12 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 0 | 1 | 1 | 0 | X | X | X | 33+0=21 |
| 0 | 0 | 0 | 1 | 1 | 1 | X | X | X | Выход 03 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 10+0=01 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Выход 01 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 11+0=01 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | Выход 01 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 12+0=02 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Выход 01 |
| 0 | 0 | 1 | 1 | 1 | 0 | X | X | X | 13+0=03 |
| 0 | 0 | 1 | 1 | 1 | 1 | X | X | X | Выход 01 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 20+0=02 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | Выход 02 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 21+0=02 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Выход 02 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 22+0=10 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Выход 02 |
| 0 | 1 | 0 | 1 | 1 | 0 | X | X | X | 23+0=12 |
| 0 | 1 | 0 | 1 | 1 | 1 | X | X | X | Выход 02 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 00+0=00 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Выход 00 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 01+0=00 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | Выход 00 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 02+0=00 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | Выход 00 |
| 0 | 1 | 1 | 1 | 1 | 0 | X | X | X | 03+0=00 |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 1 | 1 | 1 | 1 | 1 | X | X | X | Выход 00 |
| 1 | 0 | 0 | 0 | 0 | 0 | X | X | X | 30+1=00 |
| 1 | 0 | 0 | 0 | 0 | 1 | X | X | X | Выход 03 |
| 1 | 0 | 0 | 0 | 1 | 0 | X | X | X | 31+1=10 |
| 1 | 0 | 0 | 0 | 1 | 1 | X | X | X | Выход 03 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 32+1=12 |
| 1 | 0 | 0 | 1 | 0 | 1 | X | X | X | Выход 03 |
| 1 | 0 | 0 | 1 | 1 | 0 | X | X | X | 33+1=22 |
| 1 | 0 | 0 | 1 | 1 | 1 | X | X | X | Выход 03 |
| 1 | 0 | 1 | 0 | 0 | 0 | X | X | X | 10+1=01 |
| 1 | 0 | 1 | 0 | 0 | 1 | X | X | X | Выход 01 |
| 1 | 0 | 1 | 0 | 1 | 0 | X | X | X | 11+1=02 |
| 1 | 0 | 1 | 0 | 1 | 1 | X | X | X | Выход 01 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1+1=03 |
| 1 | 0 | 1 | 1 | 0 | 1 | X | X | X | Выход 01 |
| 1 | 0 | 1 | 1 | 1 | 0 | X | X | X | 13+1=10 |
| 1 | 0 | 1 | 1 | 1 | 1 | X | X | X | Выход 01 |
| 1 | 1 | 0 | 0 | 0 | 0 | X | X | X | 20+1=01 |
| 1 | 1 | 0 | 0 | 0 | 1 | X | X | X | Выход 02 |
| 1 | 1 | 0 | 0 | 1 | 0 | X | X | X | 2+1=03 |
| 1 | 1 | 0 | 0 | 1 | 1 | X | X | X | Выход 02 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 22+1=11 |
| 1 | 1 | 0 | 1 | 0 | 1 | X | X | X | Выход 02 |
| 1 | 1 | 0 | 1 | 1 | 0 | X | X | X | 23+1=13 |
| 1 | 1 | 0 | 1 | 1 | 1 | X | X | X | Выход 02 |
| 1 | 1 | 1 | 0 | 0 | 0 | X | X | X | 00+1=01 |
| 1 | 1 | 1 | 0 | 0 | 1 | X | X | X | Выход 00 |
| 1 | 1 | 1 | 0 | 1 | 0 | X | X | X | 01+1=01 |
| 1 | 1 | 1 | 0 | 1 | 1 | X | X | X | Выход 00 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 02+1=01 |
| 1 | 1 | 1 | 1 | 0 | 1 | X | X | X | Выход 00 |
| 1 | 1 | 1 | 1 | 1 | 0 | X | X | X | 03+1=01 |
| 1 | 1 | 1 | 1 | 1 | 1 | X | X | X | Выход 00 |

Функции p, , минимизируем с помощью карт Вейча.

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

Рисунок 3.2.1 – Минимизация функции р

В заданном базисе:

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

Рисунок 3.2.2 – Минимизация функции

В заданном базисе:

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

Рисунок 3.2.3 – Минимизация функции

В заданном базисе:

Функциональная схема ОЧУС представлена на чертеже ГУИР.400201.529 Э2.2 (Приложение В).

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

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 (см. таблицу 4.1):

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 управляющими 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 в самостоятельно заметить их минимальную форму.

Таблица 4.1 – Таблица истинности ОЧС на базе мультиплексоров

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **P** | **П** | **Функция** |  | **Функция** |  | **Функция** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 0 | 0 | 0 | 0 | 0 | 1 | «1» | 1 |  | 0 |  |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |  | 0 | p | 1 |  |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |  | 1 |  | 1 |  |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |  | 0 | p | 0 |  |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |  | 0 | p | 1 |  |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |  | 1 |  | 1 |  |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 1 | 1 | 0 | 0 | 0 | 0 |  | 0 | p | 0 |  |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | «0» | 1 |  | 0 |  |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Функциональная схема ОЧС на базе мультиплексоров представлена на чертеже ГУИР.400201.529 Э2.3(Приложение Г).

# **5. ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ (ПМ)**

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11, заменяя их на триады и диад 10, заменяя их на триады (см. таблицу 5.1).

Таблица 5.1 – Таблица истинности ПМ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Входная диада** | | **Перенос из предыдущей диады** | **Перенос в следующую диаду** | **Знак выходной диады** | **Выходная диада** | |
|  |  |  | **Q** | **h** |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Минимизируем выходные функции Q, h, S1, S2 картами Карно.

Таблица 5.2 *–* Минимизация функции Q

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

Таблица 5.3 *–* Минимизация функции S1

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

Таблица 5.4 *–* Минимизация функции S2

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

Таблица 5.4 *–* Минимизация функции h

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

Функциональная схема преобразователя множителя представлена на чертеже ГУИР.400201.529 Э2.4 (Приложение Д).

# **6. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ**

## **6.1. Расчет времени умножения на один разряд множителя**

Время умножения на один разряд множителя является суммой временных затрат на различных этапах умножения. Для лучшего понимания следует использовать структурную схему разрабатываемого устройства (см. чертеж ГУИР 400201.529 Э1).

Процесс умножения можно разбить на несколько этапов:

1. Преобразование разряда множителя.
2. Если необходимо, формирование дополнительного кода множимого в ФДК.
3. Умножение дополнительного кода множимого в ОЧУС.
4. Добавление полученного произведения в регистр результата с помощью ОЧС.

Процессы умножения в ОЧУС и ОЧС могут происходить параллельно. После того, как первый ОЧУС получит свой результат, он может быть сразу передан в блок ОЧС для обработки. В этот момент независимо друг от друга могут начать работу первый ОЧС и второй ОЧУС. После того, как отработает второй ОЧУС, он может незамедлительно передать свой результат во второй ОЧС, который сможет приступить к сложению только после того, как получит перенос из первого ОЧС. В результате возможного большего времени работы ОЧС может возникать задержка. Выполнение блока ОЧС, который работает с (n-1) старшими разрядами регистра результата, займет всегда (n-1)\*tОЧС. Формула для расчета времени имеет следующий вид:

TУмн = tПр + tФДК + tОЧУС + max(m\*tОЧУС,(m+1)\*tОЧС) + (n-1)\*tОЧС

В формуле TУмн – общее время умножения на один разряд множителя; tПр  – время преобразования разряда множителя; tФДК – время формирования дополнительного кода; tОЧУС и tОЧС – время работы ОЧУС и ОЧС.

После того, как получен результат с первого ОЧУС, он передается на первый ОЧС и второй ОЧУС. Чтобы блок из (n-1) старших ОЧС мог начать работать, должны отработать еще m+1 ОЧС и m ОЧУС, но так как они могут работать параллельно, мы берем максимальное время их работы.

## **6.2. Расчет времени умножения на n разрядов множителя**

Расчет времени умножения на n разрядов вычисляется аналогично, описанная выше формула целиком умножается на n разрядов множителя.

Формула расчета времени умножения на n разрядов имеет следующий вид:

TУмн = n\*(tПр + tФДК + tОЧУС + max(m\*tОЧУС,(m+1)\*tОЧС) + (n-1)\*tОЧС)

В формуле TУмн – общее время умножения на один разряд множителя; tПр  – время преобразования разряда множителя; tФДК – время формирования дополнительного кода; tОЧУС и tОЧС – время работы ОЧУС и ОЧС.

## **6.3. Расчет эффективности минимизации**

Для проведения оценки эффективности минимизации переключательных функций необходимо посчитать цену схемы до минимизации и цену схемы после минимизации. Эффективность минимизации k определяется как:

Таблица 6.3.1 – Эффективность минимизации ОЧУС

|  |  |  |  |
| --- | --- | --- | --- |
| Вых.  схемы | Рассчитанная цена схемы | | Эфф.  мин. k |
| До минимизации | После минимизации |
| P | C=4\*6+4+6=34 | C=1+3+2=6 | 5,66 |
| Q1 | C=12\*6+12+6=90 | C=3+8+4=15 | 6 |
| Q2 | C=12\*6+12+6=90 | C=4+10+3=17 | 5,29 |

Таблица 6.3.2 – Эффективность минимизации ОЧС

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вых.  схемы | Рассчитанная цена схемы | | | Эфф.  мин. k |
| До минимизации | | После минимизации |
|  | C=16\*5+16+5=101 |  | C=20+7+4=31 | 3,25 |
|  | C=16\*5+16+5=101 |  | C=4+12+4=20 | 5,05 |
|  | C=16\*5+16+5=101 |  | C=12+48+5=65 | 1,55 |

# **ЗАКЛЮЧЕНИЕ**

В процессе выполнения курсовой работы были выполнены первоначальные цели, а именно разработаны структурная схема сумматора-умножителя второго типа и функциональные схемы основных узлов данного устройства. Для уменьшения стоимости логических схем, переключательные функции были минимизированы картами Карно-Вейча. 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. В то же время, минимизация алгоритмом Рота вручную может быть много времени, если функция принимает большое количество единичных и безразличных наборов.

Функциональные схемы были построены в различных логических базисах. Это помогло закрепить теоретические знания основных законов булевой алгебры, например, правило де Моргана. 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 и упростить функциональную схему одноразрядного четверичного сумматора.

# **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.

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

**ПРИЛОЖЕНИЕ А**

(обязательное)

Сумматор-умножитель. Схема структурная

**ПРИЛОЖЕНИЕ Б**

(обязательное)

ОЧС. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ В**

(обязательное)

ОЧУС. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Г**

(обязательное)

ОЧС на мультиплексорах.

Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Д**

(обязательное)

Преобразователь множителя.

Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Е**

(обязательное)

Ведомость документов