Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Военный факультет

Кафедра вычислительных машин, систем и сетей

Дисциплина: Арифметические и логические основы

вычислительной техники

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 01 312 ПЗ

Студент Р. Ю. Феденко

Руководитель И. В. Лукьянова

МИНСК 2022

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Военный факультет

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы

вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_ Б. В. Никульшин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_г.

ЗАДАНИЕ

по курсовой работе студента

Феденко Романа Юрьевича

1. Тема работы: «Проектирование и логический синтез сумматора- умножителя двоично-десятичных чисел»
2. Срок сдачи студентом законченной работы: 20 мая 2022г.
3. Исходные данные к работе:
   1. Исходные сомножители: Мн = 96,59; Мт = 15,77;
   2. Алгоритм умножения: Б;
   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 – 00, 14 – 01, 24 – 11, 34 – 10;
   5. Тип синтезируемого умножителя: 2;
   6. Логический базис для реализации ОЧС: И-НЕ; метод минимизации – алгоритм Рота.
   7. Логический базис для реализации ОЧУС: И, Сумма по модулю, Константная единица; 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. Разработка функциональных схем основных узлов сумматора-умножителя. 4. Синтез комбинационных схем устройств на основе мультиплексоров. 5. Оценка результатов разработки. Заключение. Список литературы.

1. Перечень графического материала:
   1. Сумматор-умножитель 2 типа. Схема электрическая структурная.
   2. Однозарядный четверичный сумматор. Схема электричская

функциональная.

* 1. Однозарядный четверичный умножитель. Схема электрическая функциональная.
  2. Регистр-аккумулятор. Схема электрическая функциональная.
  3. Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объём этапа, % | Срок выполнения этапа | Примечания |
| Разработка алгоритма умножения | 10 | 10.02-20.02 |  |
| Разработка структурной схемы сумматора-умножителя | 10 | 21.02-09.03 | С выполнением чертежа |
| Разработка функциональных схем основных узлов сумматора-умножителя | 50 | 10.03-30.04 | С выполнением чертежей |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 | 01.05-15.05 | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 | 15.05-21.05 |  |

Дата выдачи задания: 10 февраля 2022 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_И. В. Лукьянова

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

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ ............................................................................................................

1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ ..............................................

2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИ-ТЕЛЯ ..................................................................................................................................

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

3.1. Логический синтез одноразрядного четверичного сумматора....................

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

3.3. Логический синтез преобразователя множителя .......................................

4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ

МУЛЬТИПЛЕКСОРОВ ...................................................................................

5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ .................................................... ЗАКЛЮЧЕНИЕ .................................................................................................... СПИСОК ЛИТЕРАТУРЫ ....................................................................................

ПРИЛОЖЕНИЕ А Сумматор-умножитель второго типа. Схема

электрическая структурная .................................................................................

ПРИЛОЖЕНИЕ Б Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная.............................................................

ПРИЛОЖЕНИЕ В Одноразрядный четверичный сумматор. Схема электрическая функциональная..........................................................................

ПРИЛОЖЕНИЕ Г Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная ..............................

ПРИЛОЖЕНИЕ Д Преобразователь множителя. Схема электрическая функциональная ………………………….............

ПРИЛОЖЕНИЕ Е Ведомость документов ...…………………………………

**ВВЕДЕНИЕ**

Курсовое проектирование является обязательным элементом подготовки специалиста с высшим образованием и одной из форм текущей аттестации студента по учебной дисциплине. Для студентов это первая работа такого рода и объёма. Она содержит результаты теоретических и экспериментальных исследований по дисциплине “Арифметические и логические основы вычислительной техники”, включает совокупность аналитических, расчётных, экспериментальных заданий и предполагает выполнение конструкторских работ и разработку графической документации.

Целью данной курсовой работы является проектирование такого цифрового устройства, как двоично-четверичный сумматор-умножитель (СУ). Сумматор является одним из центральных узлов арифметико-логического устройства (АЛУ) вычислительной машины, поэтому глубокое понимание принципов его работы критически важно для современного инженера. Для того чтобы спроектировать данное устройство, необходимо пройти несколько последовательных этапов разработки:

* Разработка алгоритма умножения чисел, по которому работает СУ
* Разработка структурной схемы СУ
* Разработка функциональной схемы основных узлов структурной схемы СУ
* Оценка результатов проделанной работы
* Оформление документации по проделанной работе

В ходе выполнения курсовой работы автором были пройдены все эти этапы. В настоящей пояснительной записке изложено краткое описание процесса проектирования и приведена разработанная автором графическая документация по структурной схеме и функциональным схемам основных её узлов.

**1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ**

1. Перевод сомножителей из десятичной системы счисления в четверичную.

96 | 4 0.59

96 24 | 4 4

0 24 6 | 4 2.36

0 4 1 4

2 1.44

Мн4 = 1200,21.

В соответствии с заданной кодировкой множимого:

Мн2/4 = 01110000,1101

15 | 4 0.77

12 3 4

3 3.08

4

0.32

4

1.28

4

1.12

Мт4 = 33,3011.

В соответствии с заданной кодировкой множителя:

Мт2/4 = 1111,11000101

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

Мн = 0,011100001101 РМн = 0.0100 +1010

Мт = 0,111111000101 РМт = 0.0011 + 0210

Порядок произведения:

|  |  |  |  |
| --- | --- | --- | --- |
| РМн | = | 0.0100 | 104 |
| РМт | = | 0.0011 | 024 |
| РМн∙Мт | = | 0.0111 | 124 |
|  |  |  |  |

Знак произведения определяется суммой по модулю два знаков

сомножителей:

зн Мн ⊕ зн Мт = 0 ⊕ 0 = 0.

При умножении чисел в прямых кодах диада 11(34) заменяется на триаду . Преобразованный множитель имеет вид М = .

Перемножение мантисс по алгоритму «Б» представлено в таблице 1.1

Таблица 1.1 — Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Четверичная с/c | | Двоично-четверичная с/с | | Комментарии |
| **1** | | **2** | | **3** |
| 0. | 0000000000000 | 00. | 00000000000000000000000000 | ∑0 ч |
| 0. | 0000000120021 | 00. | 00000000000000011100001101 | П1Ч=Мн∙40 |
| 0. | 0000000120021 | 00. | 00000000000000011100001101 | ∑1 ч |
| 0. | 0000001200210 | 00. | 00000000000001110000110100 | П2Ч=Мн∙41 |
| 0. | 0000001320231 | 00. | 00000000000001101100111001 | ∑2 ч |
| 0. | 0000000000000 | 00. | 00000000000000000000000000 | П3Ч=Мн∙0∙42 |
| 3. | 0000001320231 | 10. | 00000000000001101100111001 | ∑3 ч |
| 3. | 3333213313000 | 10. | 10101010110110100110000000 | П4Ч =Мн∙(-1) ∙ 43 |
| 3. | 3333221233231 | 10. | 10101010111101111010111001 | ∑4 ч |
| 0. | 0000000000000 | 00. | 00000000000000000000000000 | П5Ч =Мн∙0∙44 |
| 3. | 3333221233231 | 10. | 10101010111101111010111001 | ∑5 ч |
| 0. | 0000000000000 | 00. | 00000000000000000000000000 | П6Ч =Мн∙0∙45 |
| 3. | 3333221233231 | 10. | 10101010111101111010111001 | ∑6 ч |
| 0. | 0120021000000 | 00. | 00011100001101000000000000 | П7Ч =Мн∙46 |
| 0. | 0113302233231 | 00. | 00010110100011111010111001 | ∑7 ч |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн4 ∙ Мт4 = 0,0113302233231, РМн ∙ Мт = 7) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн4 ∙ Мт4 = 113302,233231 РМн ∙ Мт = 0;

Мн10 ∙ Мт10 = 1522,7453613281.

Результат прямого перемножения операндов дает следующее:

Мн10 ∙ Мт10 = 96,59 ∙ 15,77 = 1523,2243

Абсолютная погрешность:

Δ = 1523,2243– 1522,7453613281 = 0,4789386719

Относительная погрешность:

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

**2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ**

*Если устройство работает как сумматор*, то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода F2 поступает «1». 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 2.1).

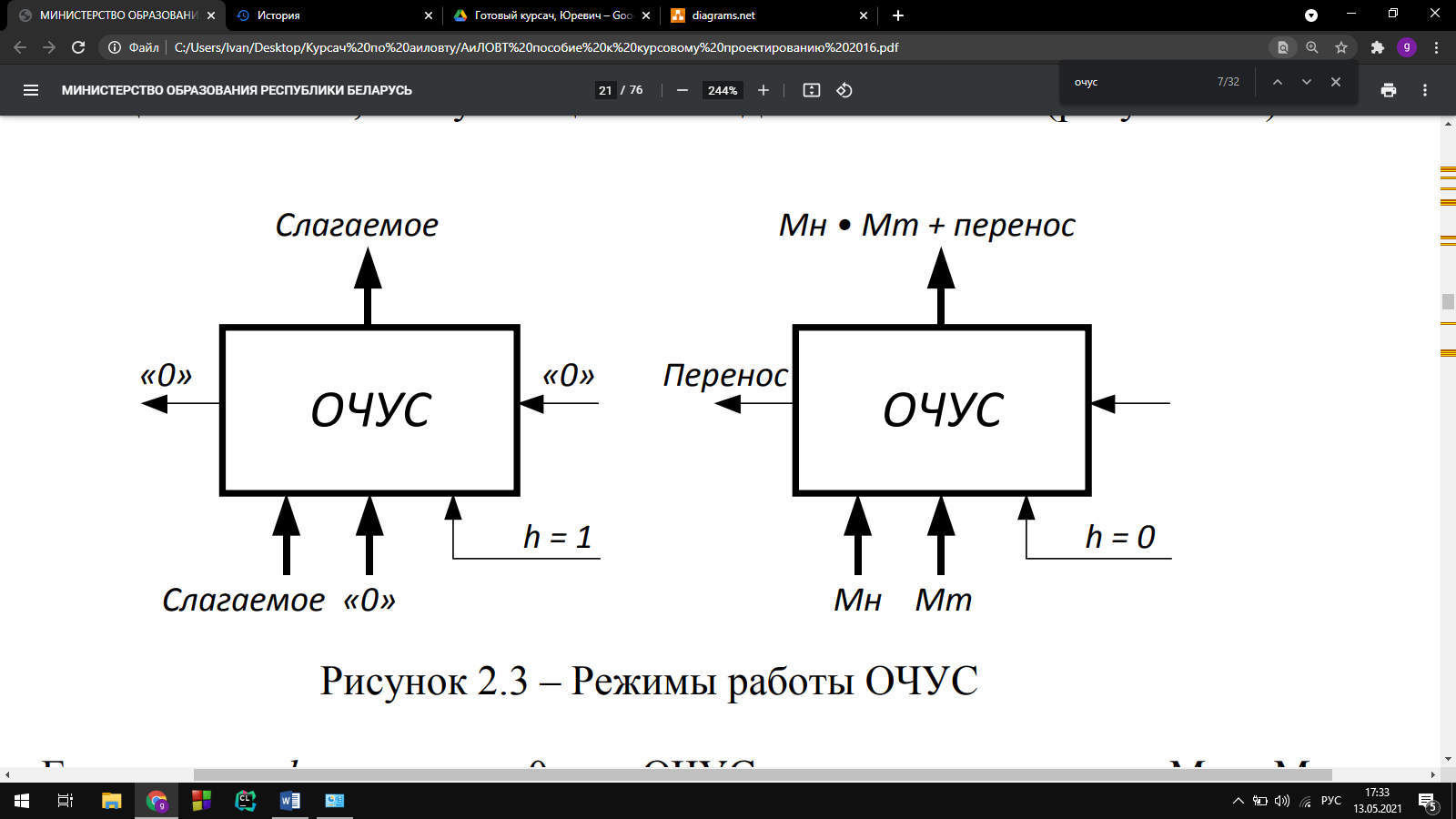


Рисунок 2.1 – Режимы работы ОЧУС

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

В ОЧС первое слагаемое складывается с нулём, записанным в регистре результата, и переписывается без изменений в регистр результата.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУС попадает на входы ОЧС и складывается с первым слагаемым, хранящимся в регистре результата.

Сумма хранится в регистре результата. Разрядность регистра результата должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения при суммировании переноса.

*Если устройство работает как умножитель*, то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК 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 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 (). В этом случае инициализируется управляющий вход F1 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.

На выходах 3 и 4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУС вместе с диадами множимого. На трёх выходах ОЧУС формируется результат умножения диад Мн·Мт плюс перенос из предыдущего ОЧУС. Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому перенос, формируемый ОЧУС, может быть только двоичным («0» или «1»):

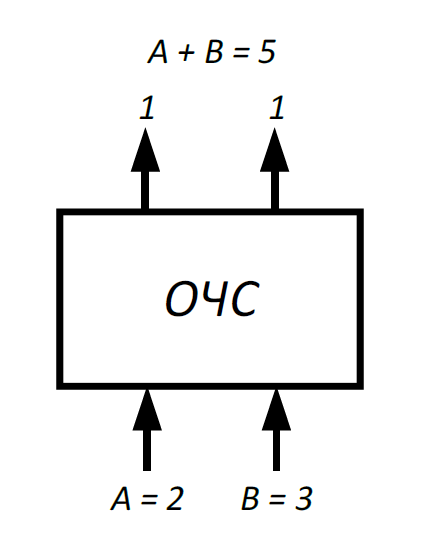
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | ∙ | 2 | = | 12 | (+1 в случае переноса из предыдущего ОЧУС) | |
| max |  | max |  | max | |  | |
| Мн |  | Мт |  | Перенос | |  | |

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. алгоритм умножения «А» предполагает возможность синхронного сдвига этих регистров. Количество тактов умножения определяется разрядностью Мт.

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 нулём, т.к. на старших выходах ОЧУ будут формироваться только коды нуля. Затем первое слагаемое попадает в регистр-аккумулятор, который изначально обнулён.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУ и ОЧС попадает в аккумулятор, где складывает с первым слагаемым. Таким образом, аккумулятор (накапливающий сумматор) складывает операнды и хранит результат.

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».

Диада множителя поступает на входы преобразователя множителя (ПМ). Задачей ПМ является преобразование диады множителя в соответствии с алгоритмом преобразования. При этом в случае образования единицы переноса в старшую диаду множителя она должна быть учтена при преобразовании следующей старшей диады (выход 1 ПМ), т.е. сохраняться до следующего такта на триггер.

В регистре множителя в конце каждого такта умножения содержимое сдвигается на два двоичных разряда и в последнем такте умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения при умножении по алгоритму «А» (регистр множителя служит как бы «продолжением» регистра результата).

Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание (01). В этом случае инициализируется управляющий вход F1 формирователя дополнительного кода (ФДК) и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на «-1»).

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

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 |

На выходах 3 и 4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУ вместе с диадами множимого.

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 ОЧС. Следовательно, чтобы полностью сформировать частичное произведение четверичных сомножителей, необходима комбинация цепочек ОЧУ и ОЧС.

Частичные суммы формируются в аккумуляторе. На первом этапе он обнулён и первая частичная сумма получается за счёт сложения первого частичного произведения (сформированного на выходах ОЧС) и нулевой частичной суммы (хранящейся в аккумуляторе).

В аккумуляторе происходит сложение *i*-й частичной суммы с (*i*+1)-м частичным произведением, результат сложения сохраняется. Содержимое аккумулятора сдвигается на один четверичный разряд вправо в конце каждого такта умножения по алгоритму «А».

На четырёх выходах ОЧУ формируется результат умножения диад Мн·Мт. Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому в старшем разряде произведения максимальной цифрой может оказаться только «1»:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | · | 2 | = | 1 2 |
| max |  | max |  |  |
| Мн |  | Мт |  |  |

Это означает, что на младшие входы ОЧС никогда не поступят диады цифр, соответствующие кодам «2» и «3», следовательно, в таблице истинности работы ОЧС будут содержаться 16 безразличных входных наборов.

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

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

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

ОЧУС – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда из регистра Мн, 2 разряда из регистра Мт и управляющий вход *h*) и 3 двоичных выхода.

Принцип работы ОЧУС представлен с помощью таблицы истинности (таблица 3.1.1).

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

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

Управляющий вход *h* определяет тип операции:

«0» – вывод результата умножения закодированных цифр с добавлением переноса из предыдущего ОЧУС, перенос в следующий ОЧУС.

«1» – вывод без изменения значения разрядов, поступивших из регистра множимого, перенос *из* и *в* ОЧУС равны нулю.

В таблице 3.1.1 выделены безразличные наборы, т.к. на входы ОЧУС из разрядов множителя не может поступить код «11».

Таблица 3.1.1 — Таблица истинности ОЧУС

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

Продолжение таблицы 3.1.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 3 \* 1 + 0 = 03 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Выход – код «03» |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 3 \* 2 + 0 = 12 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Выход – код «03» |
| 0 | 1 | 0 | 1 | 1 | 0 | x | x | x | 3 \* 3 + 0 = 13 |
| 0 | 1 | 0 | 1 | 1 | 1 | x | x | x | Выход – код «03» |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 \* 0 + 0 = 00 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Выход – код «02» |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 2 \* 1 + 0 = 02 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | Выход – код «02» |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 2 \* 2 + 0 = 10 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | Выход – код «02» |
| 0 | 1 | 1 | 1 | 1 | 0 | x | x | x | 2 \* 3 + 0 = 12 |
| 0 | 1 | 1 | 1 | 1 | 1 | x | x | x | Выход – код «02» |
| 1 | 0 | 0 | 0 | 0 | 0 | х | х | х | 0 \* 0 + 1 = 01 |
| 1 | 0 | 0 | 0 | 0 | 1 | х | х | х | Выход – код «00» |
| 1 | 0 | 0 | 0 | 1 | 0 | х | х | х | 0 \* 1 + 1 = 01 |
| 1 | 0 | 0 | 0 | 1 | 1 | х | х | х | Выход – код «00» |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 \* 2 + 1 = 01 |
| 1 | 0 | 0 | 1 | 0 | 1 | х | х | х | Выход – код «00» |
| 1 | 0 | 0 | 1 | 1 | 0 | x | x | x | 0 \* 3 + 1 = 01 |
| 1 | 0 | 0 | 1 | 1 | 1 | x | x | x | Выход – код «00» |
| 1 | 0 | 1 | 0 | 0 | 0 | х | х | х | 1 \* 0 + 1 = 01 |
| 1 | 0 | 1 | 0 | 0 | 1 | х | х | х | Выход – код «01» |
| 1 | 0 | 1 | 0 | 1 | 0 | х | х | х | 1 \* 1 + 1 = 02 |
| 1 | 0 | 1 | 0 | 1 | 1 | х | х | х | Выход – код «01» |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 \* 2 + 1 = 03 |
| 1 | 0 | 1 | 1 | 0 | 1 | х | х | х | Выход – код «01» |
| 1 | 0 | 1 | 1 | 1 | 0 | x | x | x | 1 \* 3 + 1 = 10 |
| 1 | 0 | 1 | 1 | 1 | 1 | x | x | x | Выход – код «01» |
| 1 | 1 | 0 | 0 | 0 | 0 | х | х | х | 3 \* 0 + 1 = 01 |
| 1 | 1 | 0 | 0 | 0 | 1 | х | х | х | Выход – код «03» |
| 1 | 1 | 0 | 0 | 1 | 0 | х | х | х | 3 \* 1 + 1 = 10 |
| 1 | 1 | 0 | 0 | 1 | 1 | х | х | х | Выход – код «03» |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 3 \* 2 + 1 = 13 |
| 1 | 1 | 0 | 1 | 0 | 1 | х | х | х | Выход – код «03» |
| 1 | 1 | 0 | 1 | 1 | 0 | x | x | x | 3 \* 3 + 1 = 22 |
| 1 | 1 | 0 | 1 | 1 | 1 | x | x | x | Выход – код «03» |
| 1 | 1 | 1 | 0 | 0 | 0 | х | х | х | 2 \* 0 + 1 = 01 |
| 1 | 1 | 1 | 0 | 0 | 1 | х | х | х | Выход – код «02» |
| 1 | 1 | 1 | 0 | 1 | 0 | х | х | х | 2 \* 1 + 1 = 03 |

Продолжение таблицы 3.1.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 1 | 1 | х | х | х | Выход – код «02» |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 2 \* 2 + 1 = 11 |
| 1 | 1 | 1 | 1 | 0 | 1 | х | х | х | Выход – код «02» |
| 1 | 1 | 1 | 1 | 1 | 0 | x | x | x | 2 \* 3 + 1 = 13 |
| 1 | 1 | 1 | 1 | 1 | 1 | x | x | x | Выход – код «02» |

**Минимизация функции P:**

Минимизацию функции P проведем с помощью карт Карно. Для функции Р заполненная карта приведена на рисунке 3.1.1. В рисунках 3.1.1 – 3.1.3 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 000 |  |  |  |  | \* | \* |  |  |
| 001 |  |  |  |  | \* | \* |  |  |
| 011 |  |  |  |  | \* | \* |  | 1 |
| 010 |  |  |  |  | \* | \* |  | 1 |
| 110 | \* | \* | \* | \* | \* | \* | \* | 1 |
| 111 | \* | \* | \* | \* | \* | \* | \* | 1 |
| 101 | \* | \* | \* | \* | \* | \* | \* |  |
| 100 | \* | \* | \* | \* | \* | \* | \* |  |

Рисунок 3.1.1 — Минимизация функции Р картой Карно

Следовательно:

P = x1y1

Запишем результат в базисе И-Константная единица-Сумма по модулю:

P = x1∙y1∙

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

*K = =* 8

**Минимизация функции Q1:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 000 |  |  |  |  | \* | \* |  |  |
| 001 |  |  |  |  | \* | \* |  | 1 |
| 011 |  | 1 | 1 | 1 | \* | \* | 1 |  |
| 010 |  | 1 | 1 | 1 | \* | \* | 1 | 1 |
| 110 | \* | \* | \* | \* | \* | \* | \* | 1 |
| 111 | \* | \* | \* | \* | \* | \* | \* |  |
| 101 | \* | \* | \* | \* | \* | \* | \* | 1 |
| 100 | \* | \* | \* | \* | \* | \* | \* |  |

Рисунок 3.1.2 — Минимизация функции Q1 картой Карно

Следовательно:

Q1= x1y2 + x1h+ x12y1+ 1x2y1

Запишем результат в базисе И-Константная единица-Сумма по модулю:

Q1 = (((x1 1) ∙ (y2  1)) 1) ∙ (((x1  1) ∙ (h 1)) 1) ∙

(((x11) ∙ x2 ∙(y1 1)) 1) ∙ (((x21) ∙ (y11) ∙ x1 ∙ h) 1)

Эффективность минимизации:

K = = 6,78

**Минимизация функции Q2:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 000 |  |  |  |  | \* | \* |  |  |
| 001 |  | 1 | 1 | 1 | \* | \* | 1 | 1 |
| 011 |  | 1 | 1 | 1 | \* | \* | 1 |  |
| 010 |  |  |  |  | \* | \* |  | 1 |
| 110 | \* | \* | \* | \* | \* | \* | \* |  |
| 111 | \* | \* | \* | \* | \* | \* | \* | 1 |
| 101 | \* | \* | \* | \* | \* | \* | \* |  |
| 100 | \* | \* | \* | \* | \* | \* | \* | 1 |

Рисунок 3.1.3 — Минимизация функции Q2 картой Карно

Следовательно:

Q2 = x2y2 + x2h+ 112 + 1x1x2 + 11x2 + 1x12

Запишем результат в базисе И-Константная единица-Сумма по модулю:

Q2 = (((x2 1) ∙ (y2  1)) 1) ∙ (((x2  1) ∙ (h 1)) 1) ∙

(((11) ∙ x1 ∙ x2) 1) ∙ (((11) ∙ (x11) ∙ (x21)) 1) ∙ ((1 ∙ x1∙ (x21) ∙ (y11)) 1) ∙ ((1 ∙ (x11) ∙ x2 ∙ (y11) ) ∙ h) 1)

Эффективность минимизации:

K = = 4,357

**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 3.2.1)

Кодировка слагаемых обоих разрядов: 0 – 00, 1 – 01, 2 – 11, 3 – 10;

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 | 0 | 0 | 0 | 0 + 0 + 0 = 00 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 + 0 + 1 = 01 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 + 1 + 0 = 01 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 + 1 + 1 = 02 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 + 3 + 0 = 03 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 + 3 + 1 = 10 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 + 2 + 0 =02 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 + 2 + 1 = 03 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 + 0 + 0 = 01 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 + 0 + 1 = 02 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 + 1 + 0 = 02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 + 1 + 1 = 03 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 + 3 + 0 = 10 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 + 3 + 1 = 11 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 + 2 + 0 = 03 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 + 2 + 1 = 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 + 0 + 0 = 03 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 3 + 0 + 1 = 10 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 3 + 1 + 0 = 10 |

Продолжение таблицы 3.2.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 3 + 1 + 1 = 11 |
| 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 | 1 | 3 + 2 + 0 = 11 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3 + 2 + 1 = 12 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 2 + 0 + 0 = 02 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 2 + 0 + 1 = 03 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2 + 1 + 0 = 03 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 2 + 1 + 1 = 10 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 2 + 3 + 0 = 11 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 2 + 3 + 1 = 12 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 2 + 2 + 0 = 10 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 2 + 2 + 1 =11 |

**Минимизация функции П:**

Определим множество единичных кубов:

Множество безразличных кубов пустое.

Сформируем множество С0 = L ⋃ N:

C0 = {, , , , , , , , , , , , , , , }

Первым этапом алгоритма Рота является нахождение множества простых импликант.

Для реализации этого этапа 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) приведён в таблице 3.2.2.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С0\*С0 | 00101 | 01100 | 01101 | 01111 | 10001 | 10010 | 10011 | 10100 | 10101 | 10110 | 10111 | 11011 | 11100 | 11101 | 11110 | 11111 |
| 00101 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01100 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01101 | 0y101 | 0110y | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01111 |  |  | 011y1 | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 10001 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 10010 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 10011 |  |  |  |  | 100y1 | 1001y | - |  |  |  |  |  |  |  |  |  |
| 10100 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 10101 | y0101 |  |  |  | 10y01 |  |  | 1010y | - |  |  |  |  |  |  |  |
| 10110 |  |  |  |  |  | 10y10 |  | 101y0 |  | - |  |  |  |  |  |  |
| 10111 |  |  |  |  |  |  | 10y11 |  | 101y1 | 1011y | - |  |  |  |  |  |
| 11011 |  |  |  |  |  |  | 1y011 |  |  |  |  | - |  |  |  |  |
| 11100 |  | y1100 |  |  |  |  |  | 1y100 |  |  |  |  | - |  |  |  |
| 11101 |  |  | y1101 |  |  |  |  |  | 1y101 |  |  |  | 1110y | - |  |  |

Продолжение таблицы 3.2.2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11110 |  |  |  |  |  |  |  |  |  | 1y110 |  |  | 111y0 |  | - |  |
| 11111 |  |  |  | y1111 |  |  |  |  |  |  | 1y111 | 11y11 |  | 111y1 | 1111y | - |
| А1 | 0x101 x0101 | 0110x x1100 | 011x1 x1101 | x1111 | 100x1 10x01 | 1001x 10x10 | 10x11 1x011 | 1010x 101x0 1x100 | 101x1 1x101 | 1011x 1x110 | 1x111 | 11x11 | 1110x 111x0 | 111x1 | 1111x | Ø |

В результате этой операции сформируется новое множество кубов:

##### С1 = {0x101; x0101; 0110x; x1100; 011x1; x1101; x1111; 100x1; 10x01; 1001x; 10x10; 10x11; 1x011; 1010x; 101x0; 1x100; 101x1; 1x101; 1011x; 1x110; 1x111; 11x11; 1110x; 111x0; 111x1; 1111x}

Множество Z0 кубов, не участвовавших в образовании новых кубов, пустое.

В таблице 3.2.3 приведён следующий шаг поиска простых импликант с помощью операции С1\*С1.

Таблица 3.2.3 – Поиск простых импликант С1\* С1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C1\*C1 | 0x101 | x0101 | 0110x | x1100 | 011x1 | x1101 | x1111 | 100x1 | 10x01 | 1001x | 10x10 | 10x11 | 1x011 |
| 0x101 | - |  |  |  |  |  |  |  |  |  |  |  |  |
| x0101 |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 0110x |  |  | - |  |  |  |  |  |  |  |  |  |  |
| x1100 |  |  |  | - |  |  |  |  |  |  |  |  |  |
| 011x1 |  |  |  |  | - |  |  |  |  |  |  |  |  |
| x1101 |  | xy101 |  | x110y |  | - |  |  |  |  |  |  |  |
| x1111 |  |  |  |  |  | x11y1 | - |  |  |  |  |  |  |
| 100x1 |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 10x01 |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 1001x |  |  |  |  |  |  |  |  |  | - |  |  |  |
| 10x10 |  |  |  |  |  |  |  |  |  |  | - |  |  |
| 10x11 |  |  |  |  |  |  |  |  | 10xy1 |  | 10x1y | - |  |
| 1x011 |  |  |  |  |  |  |  |  |  |  |  |  | - |
| 1010x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 101x0 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1x100 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 101x1 |  |  |  |  |  |  |  | 10yx1 |  |  |  |  |  |
| 1x101 | yx101 |  |  |  |  |  |  |  |  |  |  |  |  |
| 1011x |  |  |  |  |  |  |  |  |  | 10y1x |  |  |  |
| 1x110 |  |  |  |  |  |  |  |  |  |  |  |  |  |

Продолжение таблицы 3.2.3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1x111 |  |  |  |  |  |  |  |  |  |  |  |  | 1xy11 |
| 11x11 |  |  |  |  |  |  |  |  |  |  |  | 1yx11 |  |
| 1110x |  |  | y110x |  |  |  |  |  |  |  |  |  |  |
| 111x0 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 111x1 |  |  |  |  | y11x1 |  |  |  |  |  |  |  |  |
| 1111x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A2 | xx101 | xx101 | x110x | x110x | x11x1 | x11x1 | Ø | 10xx1 | 10xx1 | 10x1x | 10x1x | 1xx11 | 1xx11 |

Продолжение таблицы 3.2.3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C1\*C1 | 1010x | 101x0 | 1x100 | 101x1 | 1x101 | 1011x | 1x110 | 1x111 | 11x11 | 1110x | 111x0 | 111x1 | 1111x |
| 0x101 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0101 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0110x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x1100 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 011x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x1101 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x1111 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 100x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10x01 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1001x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10x10 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10x11 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1x011 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1010x | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 101x0 |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 1x100 |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 101x1 |  | 101xy |  | - |  |  |  |  |  |  |  |  |  |
| 1x101 |  |  | 1x10y |  | - |  |  |  |  |  |  |  |  |
| 1011x | 101yx |  |  |  |  | - |  |  |  |  |  |  |  |
| 1x110 |  |  | 1x1y0 |  |  |  | - |  |  |  |  |  |  |

Продолжение таблицы 3.2.3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1x111 |  |  |  |  | 1x1y1 |  | 1x11y | - |  |  |  |  |  |
| 11x11 |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 1110x | 1y10x |  |  |  |  |  |  |  |  | - |  |  |  |
| 111x0 |  | 1y1x0 |  |  |  |  |  |  |  |  | - |  |  |
| 111x1 |  |  |  | 1y1x1 |  |  |  |  |  |  | 111xy | - |  |
| 1111x |  |  |  |  |  | 1y11x |  |  |  | 111yx |  |  | - |
| А2 | 101xx  1x10x | 101xx  1x1x0 | 1x10x  1x1x0 | 1x1x1 | 1x1x1 | 1x11x | 1x11x | Ø | Ø | 111xx | 111xx | Ø | Ø |

В результате образовалось множество С*2* кубов второй размерности:

##### С2 = {xx101; x110x; x11x1; 10xx1; 10x1x; 1xx11; 101xx; 1x10x; 1x1x0; 1x1x1; 1x11x; 111xx}

Множество *Z*1 кубов, не участвовавших в образовании новых кубов, пустое.

В таблице 3.2.4 приведён следующий шаг поиска простых импликант – операция С2\*C2

Таблица 3.2.4 – Поиск простых импликант С2\*C2

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C2\*C2 | xx101 | x110x | x11x1 | 10xx1 | 10x1x | 1xx11 | 101xx | 1x10x | 1x1x0 | 1x1x1 | 1x11x | 111xx |
| xx101 | - |  |  |  |  |  |  |  |  |  |  |  |
| x110x |  | - |  |  |  |  |  |  |  |  |  |  |
| x11x1 |  |  | - |  |  |  |  |  |  |  |  |  |
| 10xx1 |  |  |  | - |  |  |  |  |  |  |  |  |
| 10x1x |  |  |  |  | - |  |  |  |  |  |  |  |
| 1xx11 |  |  |  |  |  | - |  |  |  |  |  |  |
| 101xx |  |  |  |  |  |  | - |  |  |  |  |  |
| 1x10x |  |  |  |  |  |  |  | - |  |  |  |  |
| 1x1x0 |  |  |  |  |  |  |  |  | - |  |  |  |
| 1x1x1 |  |  |  |  |  |  |  |  | 1x1xy | - |  |  |
| 1x11x |  |  |  |  |  |  |  | 1x1yx |  |  | - |  |
| 111xx |  |  |  |  |  |  | 1y1xx |  |  |  |  | - |
| A3 | Ø | Ø | Ø | Ø | Ø | Ø | 1x1xx | 1x1xx | 1x1xx | Ø | Ø | Ø |

В результате образовалось множество *С*3кубов третьей размерности:

##### *С*3 = {1x1xx}.

##### Получено множество Z2*=*{xx101; x110x; x11x1; 10xx1; 10x1x; 1xx11}*.*

В таблице 3.2.5 приведён следующий шаг поиска простых импликант – операция *С*3*\*С*3.

Таблица 3.2.5 – Поиск простых импликант *С*3*\*С*3

|  |  |
| --- | --- |
| C3 \* C3 | 1x1xx |
| 1x1xx | - |
| A4 | Ø |

Новых кубов (четвертой размерности) не образовалось.

Получено множество Z3*=* {1x1xx}

На этом заканчивается этап поиска простых импликант.

Множество простых импликант:

Z =Z0 ⋃ Z1⋃ Z2 ⋃ Z3 = { xx101; x110x; x11x1; 10xx1; 10x1x; 1xx11, 1x1xx }

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

Таблица 3.2.6 – Поиск L-экстремалей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | xx101 | x110x | x11x1 | 10xx1 | 10x1x | 1xx11 | 1x1xx |
| xx101 | - | x1100 | x1111 | 100x1 10x11 | 10x1x | 1xx11 | 1x11x 1x1x0 |
| x110x 1 | x0101 | - | x1111 | 100x1 10x11 | 10x1x | 1xx11 | 1x11x 101x0 1x110 |

Продолжение таблицы 3.2.6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x11x1 | x0101 | x1100 | - | 100x1 10x11 | 10x1x | 10x11 1x011 | 1011x 1x110 101x0 1x110 |
| 10xx1 | 00101 | x1100 | x1111 | - | 10x10 | 11011 | 10110 1x110 101x0 1x110 |
| 10x1x | 00101 | x1100 | x1111 | 10001 | - | 11011 | 11110 10100 11110 |
| 1xx11 | 00101 | x1100 | 01111 | 10001 | 10x10 | - | 11110 10100 11110 |
| 1x1xx | 00101 | 01100 | 01111 | 10001 | 10010 | 11011 | - |
| Остаток | 00101 | 01100 | 01111 | 10001 | 10010 | 11011 | 11110 10100 11110 |

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

Получили кубы, “подозрительные” на *L*-экстремальность. Проверяем их в таблице 3.2.7.

Таблица 3.2.7 – Проверка на *L-*экстремальность

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Z#(Z-z) ⋂L | 00101 | 01100 | 01101 | 01111 | 10001 | 10010 | 10011 | 10100 | 10101 | 10110 | 10111 | 11011 | 11100 | 11101 | 11110 | 11111 |
| 00101 | 00101 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 01100 | Ø | 01100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 01111 | Ø | Ø | Ø | 01111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 10001 | Ø | Ø | Ø | Ø | 10001 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 10010 | Ø | Ø | Ø | Ø | Ø | 10010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 11011 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11011 | Ø | Ø | Ø | Ø |
| 11110 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11110 | Ø |
| 10100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 11110 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11110 | Ø |

После проверки на L-экстремальность Ž = Z – E не образовалось (Ž = {Ø}), поэтому не покрытые наборы искать не надо.

Исходные кубы не надо анализировать так как все они покрываются найденной L-экстремалью. Поиск минимального покрытия завершён.

##### П = { xx101; x110x; x11x1; 10xx1; 10x1x; 1xx11; 1x1xx }=12 +212 +21 +12 +122 +12 +11

**Минимизация функции П**

Минимизацию функции **П** проведем с помощью карт Карно. Для функции **П** заполненная карта приведена на рисунке 3.2.6.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a2 | 000  b1b2p | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 10 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Рисунок 3.2.6 — Минимизация функции П картой Карно

Следовательно: П = 12 +212 +21 +12 +122 +12 +11

Запишем результат в базисе И-НЕ:

П =

Эффективность минимизации:

K = = 3,793

**Минимизация функции S**1

Минимизацию функции **S1** проведем с помощью карт Карно. Для функции **S1** заполненная карта приведена на рисунке 3.2.7.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a2 | 000  b1b2p | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 11 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Рисунок 3.2.7 — Минимизация функции S1 картой Карно

Следовательно: S1= 212 + 112 + 212 +

112+ 212 + 112 + 212 + 112

Запишем результат в базисе И-НЕ:

S1 =

Эффективность минимизации:

K = = 3,02

**Минимизация функции S2**

Минимизацию функцииS2 проведем с помощью карт Карно. Для функции S2 заполненная карта приведена на рисунке 3.2.8.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a2 | 000  b1b2p | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 01 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 11 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 10 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |

Рисунок 3.2.8 — Минимизация функции S2 картой Карно

Следовательно:

S2= a1b12 + a2b12p + a1b1b2p + 2b1b2 + 11b2 + a212 + 112p + 21b2p

Запишем результат в базисе И-НЕ:

S2 =

Эффективность минимизации:

K = = 3,02

**3.3. Логический синтез преобразователя множителя**

Преобразователь множителя (ПМ) – это устройство, которое преобразовывает диады множителя в соответствии с методом умножения.

При умножении в дополнительных кодах ПМ заменяет диаду 11 (34) на триаду .

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

Таблица 3.3.1 – Таблица истинности ПМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Входная диада** | | **Младший разряд** | **Знак** | **Выходная диада** | |
| ***Q*n** | ***Q*n-1** | ***Q*n-2** | ***P*** | ***S*1** | ***S*2** |
| **1** | **2** | **3** | **4** | **5** | **6** |
| 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 |

Минимизацию переключательных функций произведём с помощью карт Карно и реализуем их в полном базисе.

**Функция P**

Для функции P заполненная карта приведена на рисунке 3.3.1.

*Q*n-1 *Q*n-2

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

*Q*n

Рисунок 3.3.1 – Минимизация функции P при помощи карты Карно

Следовательно:

**Функция S1**

Для функции S1 Заполненная карта приведена на рисунке 3.3.2.

*Q*n-1 *Q*n-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 0  *Q*n | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |

Рисунок 3.3.2 – Минимизация функции S1 при помощи карты Карно

Следовательно:

**Функция S2**

Для функции S2 заполненная карта приведена на рисунке 3.3.3.

*Q*n-1 *Q*n-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 0  *Q*n | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |

Рисунок 3.3.3 – Минимизация функции S2 при помощи карты Карно

Следовательно:

Функциональная схема ПМ приведена в приложении Д.

# **4. Синтез комбинационных схем устройств**

# **на основе мультиплексоров**

*Мультиплексор* – это логическая схема, которая имеет *n* информационных входов, *m* управляющих входов и один выход. При этом должно выполнятся условие .

На выход мультиплексора может быть пропущен без изменений один любой 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, подаваемым на управляющие входы.

Переключательные функции (ПФ) от пяти переменных (как, например, ОЧС) можно реализовать на мультиплексоре «один из восьми». Управляющее поле такого мультиплексора будет определяться тремя переменными, следовательно, число групп с одинаковыми значениями этих переменных будет равно восьми. Также, реализация нескольких ПФ требует для каждой ПФ отдельного мультиплексора.

Для определения управляющего поля мультиплексора возьмём переменные *a*1, *a*2 и *b*1.

Таблица истинности для синтеза ПФ ОЧС приведена в таблице 4.1.

Таблица 4.1 – Таблица истинности для синтеза ПФ ОЧС

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a1** | **a2** | **b1** | **b2** | **p** | **П** | **Функция** | **S1** | **Функция** | **S2** | **Функция** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | b**2** p | 0 | b**2** p |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | p | 1 | b2 **+** | 0 | b2 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | b**2** p | 1 |  |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 2 **+** p | 0 | b**2** | 0 | p |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Продолжение таблицы 4.1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | b**2** p | 1 |  | 0 | b**2** p |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 2 **+** p | 1 | b2 **+** |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | b**2** p | 1 |  | 1 |  |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | p | 1 | 2 **+** p |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

При синтезе ОЧС на основе мультиплексоров получается более эффективная схема (если считать, что количество входов мультиплексора = n + 2n, не включая входы элементов, из которых он состоит).

Функциональная схема ОЧС на основе мультиплексоров представлена в приложении Г.

**5 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ**

Формула расчёта временных затрат на умножение:

Т=n\*(TПМ + TФДК + (n + 1) \* TОЧУС + TОЧС), где

TПМ – время преобразования множителя,

TФДК – время формирования дополнительного кода множимого,

TОЧУС – время работы ОЧУС,

TОЧС – время работы ОЧС,

n – количество разрядов множителя.

Минимизация функций позволила в несколько раз удешевить схему сумматора-умножителя и уменьшить затраты времени на выполнение за счет уменьшения количества элементов.

**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. Для минимизации функций многих переменных удобно использовать алгоритм Рота, который полностью формализует алгоритмы минимизации и делает минимизацию доступной для выполнения компьютерной программой.

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**

Луцик, Ю. А. Учебное пособие по курсу «Арифметические и логические основы вычислительной техники» / Ю. А. Луцик, И. В. Лукьянова. - Минск : БГУиР, 2014. – 76с.

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

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

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

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

**ПРИЛОЖЕНИЕ А**

**(**обязательное**)**

Сумматор-умножитель второго типа. Схема электрическая структурная

**ПРИЛОЖЕНИЕ Б**

**(**обязательное**)**

Одноразрядный четверичный сумматор. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ В**

**(**обязательное**)**

Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Г**

**(**обязательное**)**

Однозарядный четверичный сумматор. Схема электрическая функциональная на основе мультиплексоров

**ПРИЛОЖЕНИЕ Д**

**(**обязательное**)**

Преобразователь множителя. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Е**

**(**обязательное**)**

Ведомость документов