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

\_\_\_\_\_\_\_\_\_\_ Лукьянова И.В.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 401 ПЗ

Студент

Руководитель

И.А. Лужков

И.В. Лукьянова

МИНСК 2021

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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2021 г.

ЗАДАНИЕ

по курсовой работе студента  
 Лужкова Ивана Андреевича

1. Тема работы: Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел.
2. Срок сдачи студентом законченной работы: 20 мая 2021 г.
3. Исходные данные к работе:
   1. исходные сомножители: = 17,41; = 79,53.
   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 – 10, 14 – 00, 24 – 01, 34 – 11.
   5. тип синтезируемого умножителя: структурные схемы приведены для умножителя 1-ого типа (ОЧУ, ОЧС, аккумулятор).
   6. логический базис для реализации ОЧС: А7; метод минимизации – алгоритм Рота.
   7. логический базис для реализации ОЧУ: А5; метод минимизации – карты Карно-Вейча.
4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

1. Разработка алгоритм умножения. 2. Разработка структурной схемы сумматора-умножителя. 3. Разработка функциональных схем основных узлов сумматора-умножителя. 4. Синтез комбинационных схем устройств на основе мультиплексоров. 5. Оценка результатов разработки. Заключение. Список литературы.

1. Перечень графического материала:
   1. Сумматор-умножитель первого типа. Схема электрическая структурная.
   2. Одноразрядный четвертичный сумматор. Схема электрическая функциональная.
   3. Регистр-аккумулятор. Схема электрическая функциональная.
   4. Одноразрядный четвертичных сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.

Дата выдачи задания: 26 февраля 2021 г.

Руководитель Лукьянова И.В.

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

**СОДЕРЖАНИЕ**

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

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

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

3. Логический синтез одноразрядного четверичного сумматора ………….. 10

4. Логический синтез одноразрядного четверичного умножителя………..…15

5. Логический синтез одноразрядного четвертичного сумматора на

основе мультиплексора …………………..………………………………… 23

6. Логический синтез преобразователя множителя ………………………. 25

7. Временные затраты на умножение .……………………………………... 26

Заключение ………………………………………………………………….. 29

Список использованных источников ...………………………………….… 30

Приложение А ………………………………………………………………. 31

Приложение Б ……………………………………………………………….. 32

Приложение В ………………………………………………………………. 33

Приложение Г ………………………………………………………………. 34

Приложение Д ………………………………………………………………. 35

**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 (ОЧУ), 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

17 | 4 0,41 Мн4 = 101,122

16 4| 4 \* 4 в соответствии с заданной

1 4 1 1,64 кодировкой множимого

0 \* 4 Мн2/4 = 001000,000101

2,56

\* 4

2,24

Множитель

79 | 4 0,53 Мт4 = 1033,20

76 19| 4 \* 4

3 16 4| 4 2,12 Мт2/4 = 01001111,1000

3 4 1 \* 4 *множитель представляется*

0 0,48 *обычным весомозначным*

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

для всех вариантов

Запишем сомножители в форме с плавающей запятой в прямом коде:

Мн = 0,001000000101 РМн = 0.1011 + 0310 – закодировано по заданию

Мт = 0,010011111000 РМт = 0.0100 + 0410 – закодировано традиционно

Умножение двух чисел с плавающей запятой на два разряда множителя одновременно в прямых кодах. Это сводится к сложению порядков, формированию знака произведения, преобразованию разрядов множителя согласно алгоритму, и перемножению мантисс сомножителей.

Порядок произведения будет равен:

|  |  |
| --- | --- |
| = 0. 1011 | +034 |
| = 0. 0100 | +104 |
| P = 0. 0011 | +134 |

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 = или Мтп2= 010100. 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. | 000000 |  | 10. | 101010101010 |  | ∑0 ч |
| 0. | 000000 |  | 10. | 101010101010 |  | П1Ч= 0 |
| 0. | 000000 |  | 10. | 101010101010 |  | ∑1 ч |
| 0. | 000000 | 0 | 10. | 101010101010 | 10 | ∑1 ч∙ |
| 0. | 202310 |  | 10. | 011001110010 |  | П2Ч = Мн ∙ 2 |
| 0. | 202310 | 0 | 10. | 011001110010 | 10 | ∑2 ч |
| 0. | 020231 | 00 | 10. | 100110011100 | 1010 | ∑2 ч∙ |
| 3. | 202012 |  | 11. | 011001100001 |  | П3Ч= Мн∙(-1) |
| 3. | 222303 | 00 | 11. | 010101111011 | 1010 | ∑3 ч |
| 3. | 322230 | 300 | 11. | 110101011110 | 11111010 | ∑3 ч∙ |
| 0. | 000000 |  | 00. | 101010101010 |  | П4Ч= 0 |
| 3. | 322230 | 300 | 11. | 110101011110 | 111010 | ∑4 ч |
| 3. | 332223 | 0300 | 11. | 111101010111 | 10111010 | ∑4 ч∙ |
| 0. | 101122 |  | 00. | 001000000101 |  | П5Ч= Мн |
| 0. | 100011 | 0300 | 00. | 001010100000 | 10111010 | ∑5 ч |
| 0. | 010001 | 10300 | 00. | 100010101000 | 0010111010 | ∑5 ч∙ |
| 0. | 101122 |  | 00. | 001000000101 |  | П6Ч= Мн |
| 0. | 111123 | 10300 | 00. | 000000000111 | 0010111010 | ∑6 ч |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн∙Мт4 = 0,12121330300, РМн∙Мт = 0) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн∙Мт4 = 111123,10300 РМн∙Мт = 0;

Мн∙Мт10 = 1371,2968

Результат прямого перемножения операндов дает следующее значение:

Мн10 · Мт10 = 17,41 · 79,53 = 1384,6173.

Абсолютная погрешность:

Δ = 1384,6173 – 1371,2968= 13,3205.

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 первого типа для алгоритма умножения «А» представлена на схеме 1.

Структура первого типа строится на базе заданных узлов ОЧУ, ОЧС, формирователя дополнительного кода и регистра результата. Управление режимами работы схемы осуществляется внешним сигналом *Mul/sum,* который определяет вид текущей арифметической операции (умножение или суммирование).

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 множимого |
| 1 | 1 | Дополнительный код слагаемого |

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

**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 – 11; 1 – 01; 2 – 10; 3 – 00.

В таблице 3.2.1 выделено 24 безразличных наборов, т.к. со старших выходов не могут прийти коды «2» и «3».

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 | x | x | x | 3+3+0=12 |
| 0 | 0 | 0 | 0 | 1 | x | x | x | 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 | x | x | x | 3+2+0=11 |
| 0 | 0 | 1 | 0 | 1 | x | x | x | 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 | x | x | x | 1+3+0=10 |
| 0 | 1 | 0 | 0 | 1 | x | x | x | 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 | x | x | x | 1+2+0=03 |
| 0 | 1 | 1 | 0 | 1 | x | x | x | 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 | x | x | x | 2+3+0=11 |
| 1 | 0 | 0 | 0 | 1 | x | x | x | 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 | x | x | x | 2+2+0=10 |
| 1 | 0 | 1 | 0 | 1 | x | x | x | 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 | x | x | x | 0+3+0=03 |

*Продолжение таблицы 3.2.1*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 1 | 1 | 0 | 0 | 1 | x | x | x | 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 | x | x | x | 0+2+0=02 |
| 1 | 1 | 1 | 0 | 1 | x | x | x | 0+2+1=03 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0+0+0=00 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0+0+1=01 |

Минимизацию функций *П, S1* и *S2*проведем при помощи карт Вейча.

Для функции *П*:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | \* | \* | \* | \* | \* | \* | \* | \* |  |  |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  |  |
|  |  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |  |
|  |  | \* | \* | \* | \* | \* | \* | \* | \* |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |
|  |  |  | p | |  | | p | |  |  |  |

Рисунок 3.2.2 – Минимизация функции *П*при помощи карты Вейча.

*=*

Для функции *S1*:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | \* | \* | \* | \* | \* | \* | \* | \* |  |  |
|  | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |  |  |
|  |  | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |  |
|  |  | \* | \* | \* | \* | \* | \* | \* | \* |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |
|  |  |  | p | |  | | p | |  |  |  |

Рисунок 3.2.3 – Минимизация функции *S1* при помощи карты Вейча.

*=*

*= .*

Для функции *S2*:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | \* | \* | \* | \*  Я | \* | \* | \* | \* |  |  |
|  | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |  |  |
|  |  | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |  |
|  |  | \* | \* | \* | \* | \* | \* | \* | \* |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |
|  |  |  | p | |  | | p | |  |  |  |

Рисунок 3.2.4 – Минимизация функции *S2* при помощи карты Карно.

*=*

**3.1. Логический синтез одноразрядного четверичного умножителя**

Одноразрядный четверичный умножитель – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда из регистра Мн, 2 разряда из регистра Мт и управляющий вход h) и 4 двоичных выхода. Принцип работы ОЧУ описывается с помощью таблицы истинности (табл.3.1.1).

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

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

Таблица 3.1.1 - Таблица истинности ОЧУ

| **Мн**  **Мт** | | **Мт**  **Перенос** | | **Упр.** | **Ст. разряды** | | **Мл. разряды**  **Мт** | | **Результат операции**  **в четверичной с/c** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***X1*** | ***X2*** | ***y1*** | ***y2*** | ***h*** | ***P1*** | ***P2*** | ***P3*** | ***P4*** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 3 ∙ 0 = 00 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 3 ∙ 1 = 03 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 3 ∙ 2 = 12 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 1 | 1 | 0 | x | x | x | x | 3 ∙ 3 = 21 |
| 0 | 0 | 1 | 1 | 1 | x | x | x | x | Выход 03 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 ∙ 0 = 00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Выход 01 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 ∙ 1 = 01 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Выход 01 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 ∙ 2 = 02 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход 01 |
| 0 | 1 | 1 | 1 | 0 | x | x | x | x | 1 ∙ 3 = 03 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | x | Выход 01 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 ∙ 0 = 00 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Выход 02 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 2 ∙ 1 = 02 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Выход 02 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 2 ∙ 2 = 10 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Выход 02 |
| 1 | 0 | 1 | 1 | 0 | x | x | x | x | 2 ∙ 3 = 12 |

*Продолжение таблицы 3.1.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 1 | 0 | 1 | 1 | 1 | x | x | x | x | Выход 02 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 ∙ 0 = 00 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Выход 00 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 ∙ 1 = 00 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Выход 00 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 ∙ 2 = 00 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Выход 00 |
| 1 | 1 | 1 | 1 | 0 | x | x | x | x | 0 ∙ 3 = 00 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | x | Выход 00 |

Минимизацию функций ***P1****,* ***P2*** и ***P3***проведем при помощи карт Карно.

Для функции ***P1***:

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

Рисунок 3.1.2 – Минимизация функции***P1***при помощи карты Карно.

*=*

Для функции ***P2***:

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

Рисунок 3.1.3 – Минимизация функции***P2***при помощи карты Карно.

*=*

Для функции ***P3***:

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

Рисунок 3.1.4 – Минимизация функции***P3***при помощи карты Карно.

*=*

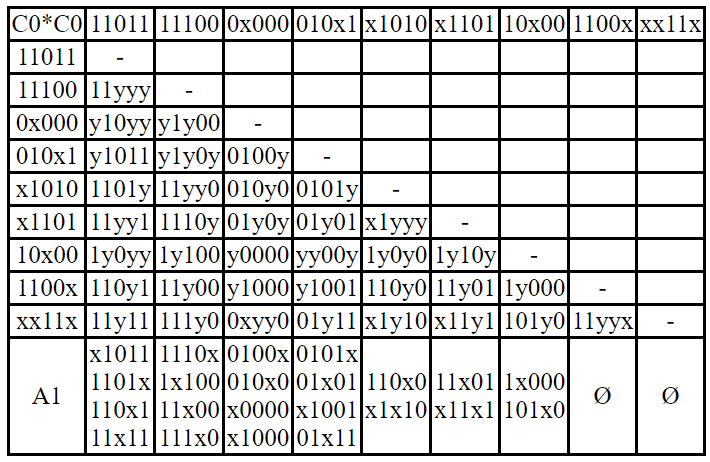
Минимизацию функции ***P4*** проведём алгоритмом Рота:

|  |  |
| --- | --- |
| *L* | *N* |
| 00000 01000 01001 01010 01011 01101 10000 10100 11000 11001 11010 11011 11100 11101 | 00110 00111 01110 01111 10110 10111 11110 11111 |

В исходных множествах произвели склеивание кубов и получили:

|  |  |
| --- | --- |
| L | N |
| 11011 11100 0x000 010x1 x1010 x1101 10x00 1100x | xx11x |

Поиск простых импликант:

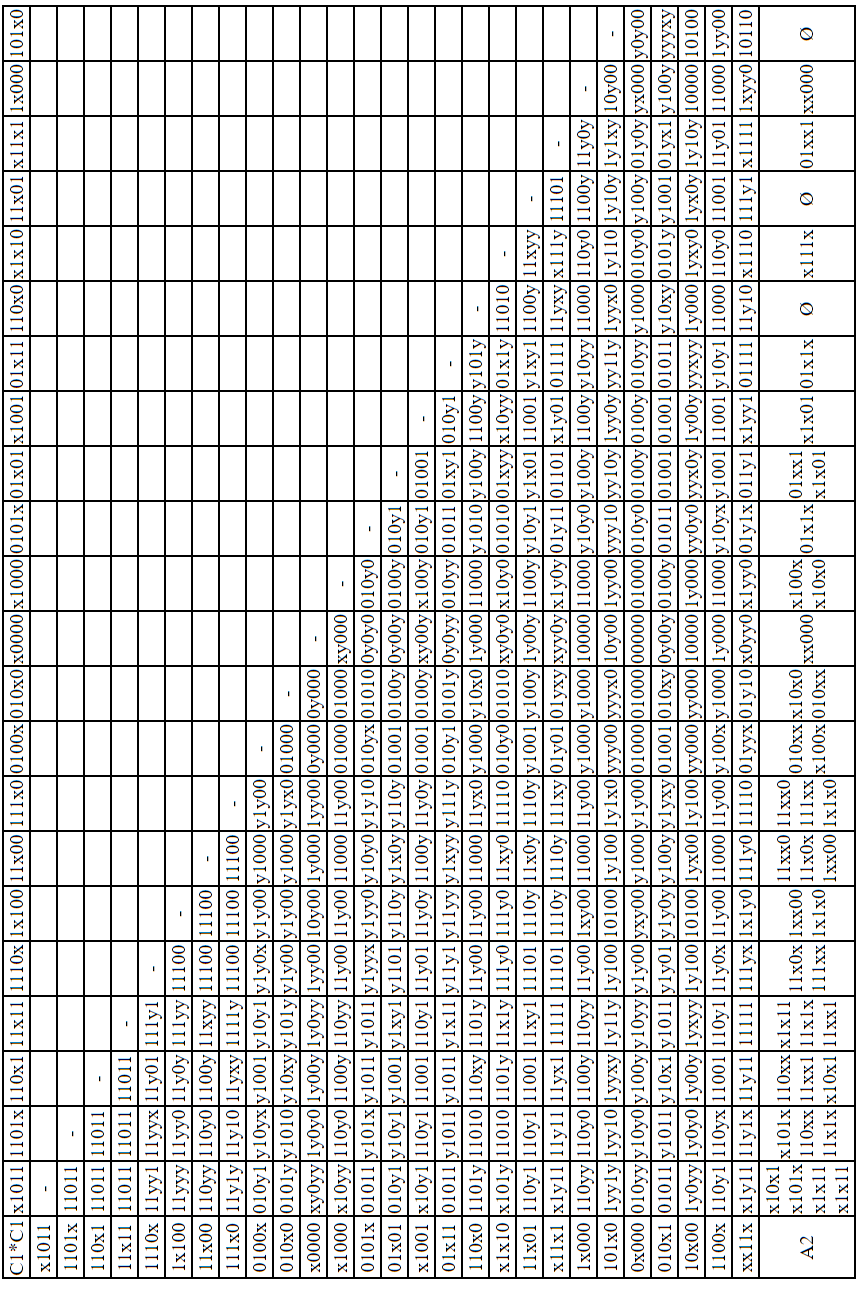


##### A1 = { x1011; 1101x; 110x1; 11x11; 1110x; 1x100; 11x00; 111x0; 0100x; 010x0; x0000; x1000; 0101x; 01x01; x1001; 01x11; 110x0; x1x10; 11x01; x11x1; 1x000; 101x0 }

Z0 = { Ø }

##### B1 = { 11011; 11100; 0x000; 010x1; x1010; x1101; 10x00; 1100x; xx11x }

##### C1 = { x1011; 1101x; 110x1; 11x11; 1110x; 1x100; 11x00; 111x0; 0100x; 010x0; x0000; x1000; 0101x; 01x01; x1001; 01x11; 110x0; x1x10; 11x01; x11x1; 1x000; 101x0; 0x000; 010x1; 10x00; 1100x; xx11x }

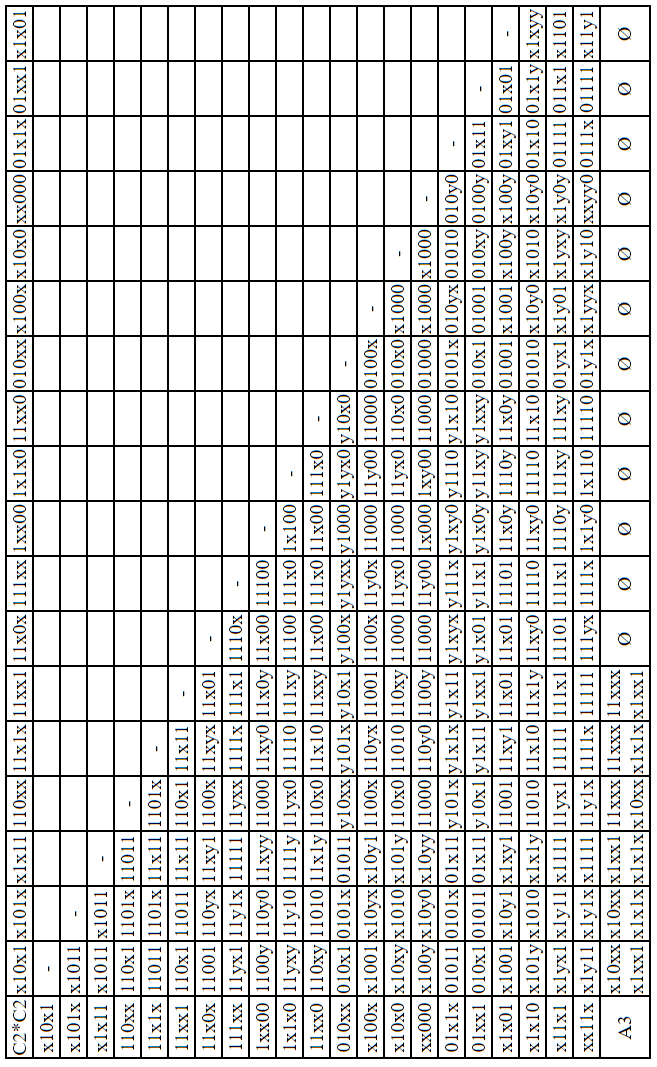


##### A2 = { x10x1; x101x; x1x11; 110xx; 11x1x; 11xx1; 11x0x; 111xx; 1xx00; 1x1x0; 11xx0; 010xx; x100x; x10x0; xx000; 01x1x; 01xx1; x1x01; x111x }

Z1 = { Ø }

##### B2 = { x1011; 1101x; 110x1; 11x11; 1110x; 1x100; 11x00; 111x0; 0100x; 010x0; x0000; x1000; 0101x; 01x01; x1001; 01x11; 110x0; x1x10; 11x01; x11x1; 1x000; 101x0; 0x000; 010x1; 10x00; 1100x; xx11x }

##### C2 = { x10x1; x101x; x1x11; 110xx; 11x1x; 11xx1; 11x0x; 111xx; 1xx00; 1x1x0; 11xx0; 010xx; x100x; x10x0; xx000; 01x1x; 01xx1; x1x01; x1x10; x11x1; xx11x }

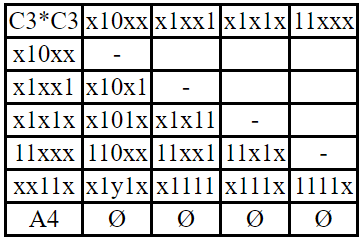


##### A3 = { x10xx; x1xx1; x1x1x; 11xxx }

##### Z2 = { 1xx00; 1x1x0; xx000 }

##### B3 = { x10x1; x101x; x1x11; 110xx; 11x1x; 11xx1; 11x0x; 111xx; 11xx0; 010xx; x100x; x10x0; 01x1x; 01xx1; x1x01; x1x10; x11x1; xx11x }

##### C3 = { x10xx; x1xx1; x1x1x; 11xxx; xx11x }



##### A4 = { Ø }

##### Конечное множество простых импликант Z = { 1xx00; 1x1x0; xx000; x10xx; x1xx1; x1x1x; 11xxx; xx11x }

**Поиск L-экстремалей.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 1xx00 | 1x1x0 | xx000 | x10xx | x1xx1 | x1x1x | 11xxx | xx11x |
| 1xx00 | - | 1x110 | 0x000 | 010xx x101x x10x1 | x1xx1 | x1x1x | 11x1x 11xx1 | xx11x |
| 1x1x0 | 1x000 | - | 0x000 | 010xx x101x x10x1 | x1xx1 | 01x1x x101x x1x11 | 1101x 11x11 11xx1 | 0x11x xx111 |
| xx000 | Ø | 1x110 | - | 0101x 010x1 x101x x10x1 | x1xx1 | 01x1x x101x x1x11 | 1101x 11x11 11xx1 | 0x11x xx111 |
| x10xx | Ø | 1x110 | 00000 | - | x11x1 | 0111x x1111 | 11111 111x1 | 0x11x xx111 |
| x1xx1 | Ø | 1x110 | 00000 | 01010 x1010 | - | 01110 | Ø | 0011x 0x110 x0111 |
| x1x1x | Ø | 10110 | 00000 | Ø | x1101 | - | Ø | 0011x 00110 x0111 |
| 11xxx | Ø | 10110 | 00000 | Ø | 01101 | 01110 | - | 0011x 00110 x0111 |
| xx11x | Ø | Ø | 00000 | Ø | 01101 | Ø | Ø | - |
| Остаток | Ø | Ø | 00000 | Ø | 01101 | Ø | Ø | 0011x 00110 x0111 |

**Далее для расчетов используется исходное множество L (без произведенного склеивания).**

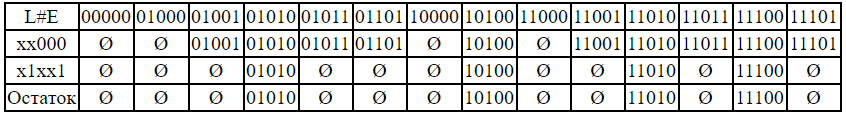
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) n L | 00000 | 01000 | 01001 | 01010 | 01011 | 01101 | 10000 | 10100 | 11000 | 11001 | 11010 | 11011 | 11100 | 11101 |
| 00000 | 00000 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 01101 | Ø | Ø | Ø | Ø | Ø | 01101 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 0011x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 00110 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| x0111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

##### **Множество L-экстремалей E = { xx000; x1xx1 }**

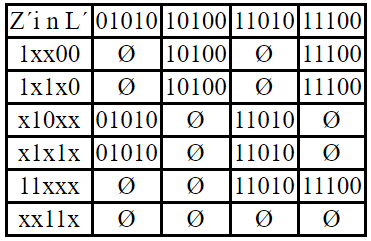
##### Z´ = Z - E = { 1xx00; 1x1x0; x10xx; x1x1x; 11xxx; xx11x }

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00000 | 01000 | 01001 | 01010 | 01011 | 01101 | 10000 | 10100 | 11000 | 11001 | 11010 | 11011 | 11100 | 11101 |
| xx000 | Ø | Ø | 01001 | 01010 | 01011 | 01101 | Ø | 10100 | Ø | 11001 | 11010 | 11011 | 11100 | 11101 |
| x1xx1 | Ø | Ø | Ø | 01010 | Ø | Ø | Ø | 10100 | Ø | Ø | 11010 | Ø | 11100 | Ø |
| Остаток | Ø | Ø | Ø | 01010 | Ø | Ø | Ø | 10100 | Ø | Ø | 11010 | Ø | 11100 | Ø |

##### Z´ = Z - E = { 1xx00; 1x1x0; x10xx; x1x1x; 11xxx; xx11x }



##### Множество кубов, непокрываемых L-экстремалями, L´ = L # E = { 01010; 10100; 11010; 11100 }



*=*

**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

*Таблица 4.1 – Таблица синтеза дополнительных логических схем для ПФ ОЧС*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***№*** | ***a1*** | ***a2*** | ***b1*** | ***b2*** | ***p*** | **П** | **Вы-ход** |  | **Вы-ход** |  | **Вы-ход** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 0 | 0 | 0 | 0 | 0 | 0 | X | *«0»* | X | *!b2+p* | X |  |
| 1 | 0 | 0 | 0 | 0 | 1 | X | X | X |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | X | X | X |
| 5 | 0 | 0 | 1 | 0 | 1 | X | X | X |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | X | *«0»* | X | !b2\*p | X |  |
| 9 | 0 | 1 | 0 | 0 | 1 | X | X | X |
| 10 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 11 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 12 | 0 | 1 | 1 | 0 | 0 | X | X | X |
| 13 | 0 | 1 | 1 | 0 | 1 | X | X | X |
| 14 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 15 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 16 | 1 | 0 | 0 | 0 | 0 | X | *!b2+p* | X | *b2\*!p* | X |  |
| 17 | 1 | 0 | 0 | 0 | 1 | X | X | X |
| 18 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 19 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 20 | 1 | 0 | 1 | 0 | 0 | X | X | X |
| 21 | 1 | 0 | 1 | 0 | 1 | X | X | X |
| 22 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 23 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 24 | 1 | 1 | 0 | 0 | 0 | X | *«0»* | X | b2+!p | X |  |
| 25 | 1 | 1 | 0 | 0 | 1 | X | X | X |
| 26 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 27 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 28 | 1 | 1 | 1 | 0 | 0 | X | X | X |
| 29 | 1 | 1 | 1 | 0 | 1 | X | X | X |
| 30 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 31 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Функциональная схема реализации ОЧС на мультиплексорах приведена в приложении Г.

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

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 |

1. **ОЦЕНКА ЭФФЕКТИВНОСТИ МИНИМИЗАЦИИ ПЕРЕКЛЮЧАТЕЛЬНЫХ ФУНКЦИЙ**

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

Таблица 6.1 – Эффективность минимизации ОЧУ

|  |  |  |  |
| --- | --- | --- | --- |
| Вых.  схемы | Рассчитанная цена схемы | | Эфф.  мин. k |
| До минимизации | После минимизации |
| P1 | с=127 | с=1 | 127 |
| P2 | с=22\*5+22+5+1=138 | с=6+4+2+1=13 | 10,6 |
| Q1 | с=10\*6+5+1=66 | с=7+1+3+1=12 | 5,5 |
| Q2 | с=16\*5+16+5+1=102 | c=2+3+1+1=7 | 14,57 |

Таблица 6.2 – Эффективность минимизации ОЧС

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вых.  схемы | Рассчитанная цена схемы | | | Эфф.  мин. k |
| До минимизации | | После минимизации |
|  | c=4\*5+5+4+1=30 |  | с=3\*3=15 | 2 |
|  | с=8\*5=54 |  | с=19 | 1,8 |
|  | с=8\*5=54 |  | с=12+4+3+1=20 | 2,7 |

1. **ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ**

**Временные затраты на умножение**

Формула расчёта временных затрат на умножение:

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

TПМ – время преобразования множителя;

TФДК – время формирования дополнительного кода множимого;

TОЧУ – время умножения на ОЧУ;

TОЧС – время формирования единицы переноса в ОЧС;

Tсдвига – время сдвига частичной суммы;

n – количество разрядов множителя.

**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.

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 первого типа. Схема электрическая структурная

**ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Одноразрядный четверичный умножитель-сумматор.

Схема электрическая функциональная

**ПРИЛОЖЕНИЕ В**

*(обязательное)*

Одноразрядный четверичный сумматор. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Г**

*(обязательное)*

Одноразрядный четверичный сумматор.

Реализация на мультиплексорах.

Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Д**

*(обязательное)*

Преобразователь множителя. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Е**

*(обязательное)*

Ведомость документов