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

\_\_\_\_\_\_\_\_\_\_ Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 223 ПЗ

Студент

Руководитель

Д.А. Снитко

Ю. А. Луцик

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 вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_ Б. В. Никульшин

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

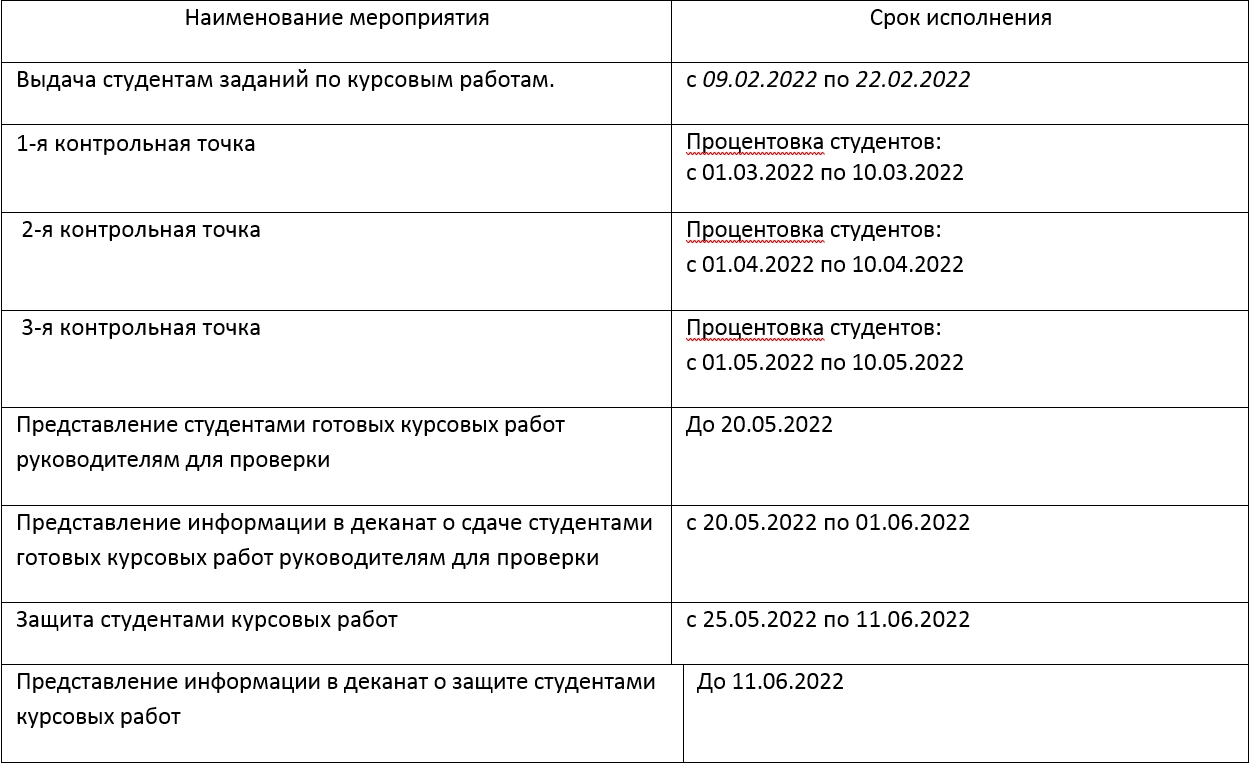
ЗАДАНИЕ

по курсовой работе студента  
 Снитко Даниила Александровича

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:
   1. исходные сомножители: = 82,48; = 43,35;
   2. алгоритм умножения: В;
   3. метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;
   4. коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования; 04 – 01, 14 – 10, 24 – 11, 34 – 00;
   5. тип синтезируемого умножителя: 1;
   6. тип синтезируемого умножителя: структурные схемы приведены для умножителя 1-ого типа (ОЧУ, ОЧС, аккумулятор);
   7. логический базис для реализации ОЧС: И, ИЛИ, НЕ; метод минимизации –алгоритм Рота;
   8. логический базис для реализации ОЧУ: НЕ; метод минимизации – карты Карно-Вейча.
4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

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 первого типа. Схема электрическая структурная.
   2. Одноразрядный четвертичный сумматор. Схема электрическая функциональная.
   3. Одноразрядный четвертичный умножитель. Схема электрическая функциональная.
   4. Регистр-аккумулятор. Схема электрическая функциональная.
   5. Одноразрядный четвертичных сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.



Дата выдачи задания: 10 февраля 2022 г.

Руководитель Ю. А. Луцик

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………..…21

5. Логический синтез одноразрядного четвертичного сумматора на

основе мультиплексора …………………..………………………………… 25

6. Логический синтез преобразователя множителя ………………………. 27

7. Оценка эффективности минимизации переключательных функций…….. 28

8. Временные затраты на умножение .……………………………………... 29

Заключение ………………………………………………………………….. 30

Список использованных источников ...………………………………….… 31

Приложение А ………………………………………………………………. 32

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

Приложение В ………………………………………………………………. 32

Приложение Г ………………………………………………………………. 32

Приложение Д ………………………………………………………………. 32

**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

82 | 4 0,48 Мн4 = 1102,132

80 20| 4 \* 4 в соответствии с заданной

2 20 5| 4 1,92 кодировкой множимого

0 4 1 \* 4 Мн2/4 = 10100111,100011

1 3,68

\* 4

2,72

Множитель

43 | 4 0,35 Мт4 = 223,112

40 10 | 4 \* 4

3 8 2 1,40 Мт2/4 =101011,010110

2 \* 4 *множитель представляется*

1,60 *обычным весомозначным*

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

2,40 для всех вариантов

Запишем сомножители в форме с плавающей запятой в прямом коде:

Мн = 0,10100111100011 РМн = 0.1001 + 0410 – закодировано по заданию

Мт = 0,101011010110 РМт = 0.0011 + 0310 – закодировано традиционно

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

Порядок произведения будет равен:

|  |  |
| --- | --- |
| = 0. 1001 | +104 |
| = 0. 0011 | +034 |
| P = 0.1000 | +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. Диада 11(34) заменяется на триаду 1. Преобразованный множитель имеет вид: Мтп4 = 1 или Мтп2=010.

Алгоритм “В”.

Таблица 1.1 - Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Четверичная с/c | | Двоично-четверичная с/с | | Комментарии |
| **1** | | **2** | | **3** |
| 0. | 0000000000000 | 01. | 01010101010101010101010101 | ∑0 ч |
| 0. | 0000001102132 | 01. | 01010101010110100111100011 | П1Ч=Мн∙b1 |
| 0. | 0000001102132 | 01. | 01010101010110100111100011 | ∑1 ч |
| 0. | 0000011021320 | 01. | 01010101011010011110001101 | ∑1 ч∙ 4 |
| 3. | 3333332231202 | 00. | 00000000000011110010110011 | П2Ч = Мн∙b2 |
| 0. | 0000003313122 | 01. | 01010101010100001000101111 | ∑2 ч |
| 0. | 0000033131220 | 01. | 01010101010000100010111101 | ∑2 ч∙ 4 |
| 3. | 3333332231202 | 00. | 00000000000011110010110111 | П3Ч= Мн∙b3 |
| 0. | 0000032023022 | 01. | 01010101010011011100011111 | ∑3 ч |
| 0. | 0000320230220 | 01. | 01010101001101110001111101 | ∑3 ч∙ 4 |
| 3. | 3333332231202 | 00. | 00000000000011110010110111 | П4Ч= Мн∙b4 |
| 0. | 0000313122022 | 01. | 01010101001000101111001111 | ∑4 ч |
| 0. | 0003131220220 | 01. | 01010100100010111101111101 | ∑4 ч∙ 4 |
| 0. | 0000001102132 | 01. | 01010101010110100111100011 | П5Ч= Мн∙b5 |
| 0. | 0003132323012 | 01. | 01010100100011001100011011 | ∑5 ч |
| 0. | 0031323230120 | 01. | 01010010001100110001101101 | ∑5 ч∙ 4 |
| 0. | 0000001102132 | 00. | 01010101010110100111100011 | П6Ч= Мн∙b6 |
| 0. | 0031330332312 | 01. | 01010010000001000011001011 | ∑6 ч |
| 0. | 0313303323120 | 01. | 01001000000100001100101101 | ∑6 ч∙ 4 |
| 0. | 0000011021320 | 01. | 01010101011010011110001101 | П7Ч= Мн∙b7 |
| 0. | 0313321011100 | 01. | 01001000001110011010100101 | ∑7 ч |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн∙Мт4 = 0, 112012000021, РМн∙Мт = 7) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн∙Мт4 = 313321,011100 РМн∙Мт = 0;

Мн∙Мт10 = 3577,082

Результат прямого перемножения операндов дает следующее значение:

Мн10 · Мт10 = 82,48 · 43,35 = 3575,508.

Абсолютная погрешность:

Δ = 3577,082- 3575,508= 1,574.

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 «В» приведена в приложении А.

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**

**3.1. 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. 3.1.1).

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

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

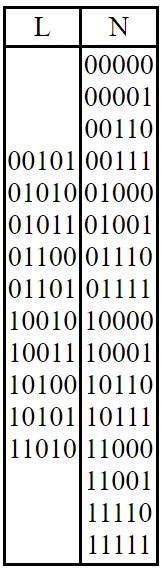
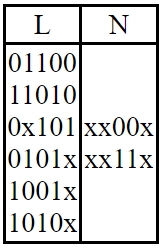
*Таблица 3.2.1 – Таблица истинности ОЧС*

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

*Продолжение таблицы 3.2.1*

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

Для минимизации функции S1 воспользуемся алгоритмом Рота. Определим множество единичных и безразличных кубов, а после произвели их склеивание:

#### 

#### Поиск простых импликант - воспользуемся операцией умножения (\*) над множествами С0, С1 и т. д., пока в результате операции будут образовываться новые кубы большей размерности.

*Поиск простых импликант C0\*C0:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C0\*C0 | 01100 | 11010 | 0x101 | 0101x | 1001x | 1010x | xx00x | xx11x |
| 01100 | - |  |  |  |  |  |  |  |
| 11010 |  | - |  |  |  |  |  |  |
| 0x101 | 0110y |  | - |  |  |  |  |  |
| 0101x |  | y1010 |  | - |  |  |  |  |
| 1001x |  | 1y010 |  |  | - |  |  |  |
| 1010x |  |  | y0101 |  |  | - |  |  |
| xx00x | 01y00 | 110y0 | 0xy01 | 010yx | 100yx | 10y0x | - |  |
| xx11x | 011y0 | 11y10 | 0x1y1 | 01y1x | 10y1x | 101yx |  | - |
| A1 | 0110x 01x00 011x0 | x1010 1x010 110x0 11x10 | x0101 0xx01 0x1x1 | 010xx 01x1x | 100xx 10x1x | 10x0x 101xx | Ø | Ø |

##### A1 = { 0110x; 01x00; 011x0; x1010; 1x010; 110x0; 11x10; x0101; 0xx01; 0x1x1; 010xx; 01x1x; 100xx; 10x1x; 10x0x; 101xx }

##### Z0 = { Ø } - Множество Z кубов, не участвовавших в образовании новых кубов, пустое.

##### B1= C0 — Z0 = { 01100; 11010; 0x101; 0101x; 1001x; 1010x; xx00x; xx11x }

##### C1 = A1 ∪ B1 = { 0110x; 01x00; 011x0; x1010; 1x010; 110x0; 11x10; x0101; 0xx01; 0x1x1; 010xx; 01x1x; 100xx; 10x1x; 10x0x; 101xx; xx00x; xx11x }

*Поиск простых импликант C1\*C1:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C1\*C1 | 0110x | 01x00 | 011x0 | x1010 | 1x010 | 110x0 | 11x10 | x0101 | 0xx01 | 0x1x1 | 010xx | 01x1x | 100xx | 10x1x | 10x0x | 101xx |
| 0110x | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01x00 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 011x0 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x1010 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 1x010 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 110x0 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 11x10 |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |
| x0101 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 0xx01 |  | 01x0y |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| 0x1x1 |  |  | 011xy |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 010xx | 01y0x |  | 01yx0 |  |  | y10x0 |  |  |  | 01yx1 | - |  |  |  |  |  |
| 01x1x | 011yx | 01xy0 |  |  |  |  | y1x10 |  | 01xy1 |  |  | - |  |  |  |  |
| 100xx |  |  |  |  |  | 1y0x0 |  |  |  |  |  |  | - |  |  |  |
| 10x1x |  |  |  |  |  |  | 1yx10 |  |  |  |  |  |  | - |  |  |
| 10x0x |  |  |  |  |  |  |  |  | y0x01 |  |  |  |  | 10xyx | - |  |
| 101xx |  |  |  |  |  |  |  |  |  | y01x1 |  |  | 10yxx |  |  | - |
| xx00x | 01y0x |  |  | x10y0 | 1x0y0 |  |  | x0y01 |  | 0xy01 |  | 010yx |  | 100yx |  | 10y0x |
| xx11x | 011yx |  |  | x1y10 | 1xy10 |  |  | x01y1 | 0x1y1 |  | 01y1x |  | 10y1x |  | 101yx |  |
| A2 | 01x0x 011xx 01x0x 011xx | 01x0x 01xx0 | 011xx 01xx0 | x10x0 x1x10 | 1x0x0 1xx10 | x10x0 1x0x0 | x1x10 1xx10 | x0x01 x01x1 | 01xx1 x0x01 0x1x1 | 01xx1 x01x1 0xx01 | 01x1x | 010xx | 10xxx 10x1x | 10xxx 100xx | 101xx | 10x0x |

##### A2 = { 01x0x; 011xx; 01xx0; x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 01xx1; 0x1x1; 0xx01; 01x1x; 010xx; 10xxx }

##### Z1 = { Ø }- Множество Z кубов, не участвовавших в образовании новых кубов, пустое.

##### B2 = C1 — Z1 = { 0110x; 01x00; 011x0; x1010; 1x010; 110x0; 11x10; x0101; 0xx01; 0x1x1; 010xx; 01x1x; 100xx; 10x1x; 10x0x; 101xx; xx00x; xx11x }

##### C2 = A2 ∪ B2 = { 01x0x; 011xx; 01xx0; x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 01xx1; 0x1x1; 0xx01; 01x1x; 010xx; 10xxx; xx00x; xx11x }

*Поиск простых импликант C2\*C2:*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C2\*C2 | 01x0x | 011xx | 01xx0 | x10x0 | x1x10 | 1x0x0 | 1xx10 | x0x01 | x01x1 | 01xx1 | 10xxx |
| 01x0x | - |  |  |  |  |  |  |  |  |  |  |
| 011xx |  | - |  |  |  |  |  |  |  |  |  |
| 01xx0 |  |  | - |  |  |  |  |  |  |  |  |
| x10x0 |  |  |  | - |  |  |  |  |  |  |  |
| x1x10 |  |  |  |  | - |  |  |  |  |  |  |
| 1x0x0 |  |  |  |  |  | - |  |  |  |  |  |
| 1xx10 |  |  |  |  |  |  | - |  |  |  |  |
| x0x01 |  |  |  |  |  |  |  | - |  |  |  |
| x01x1 |  |  |  |  |  |  |  |  | - |  |  |
| 01xx1 |  |  | 01xxy |  |  |  |  |  |  | - |  |
| 0x1x1 |  |  |  |  |  |  |  |  |  |  | - |
| 0xx01 |  |  |  |  |  |  |  |  |  |  |  |
| 01x1x | 01xyx |  |  |  |  |  |  |  |  |  |  |
| 010xx |  | 01yxx |  |  |  |  |  |  |  |  |  |
| 10xxx |  |  |  |  |  |  |  |  |  |  |  |
| xx00x |  |  |  |  |  |  |  |  |  |  |  |
| xx11x |  |  |  |  |  |  |  |  |  |  |  |
| A3 | 01xxx | 01xxx | 01xxx | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

##### A3 = { 01xxx }

##### Z2 = { x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 0x1x1; 0xx01; 10xxx; xx00x; xx11x }

##### B3 = C2 — Z2 = { 01x0x; 011xx; 01xx0; 01xx1; 01x1x; 010xx }

##### C3 = A3 ∪ B3 = { 01xxx }

*Поиск простых импликант C3\*C3:*

|  |  |
| --- | --- |
| C3\*C3 | 01xxx |
| 01xxx | - |
| A4 | Ø |

##### A4 = { Ø }

##### Конечное множество простых импликант Z = { x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 0x1x1; 0xx01; 10xxx; xx00x; xx11x; 01xxx }

Множество Z может быть избыточным. Необходимо выявить обязательные простые импликанты, называемые в алгоритме Рота L-экстремалями.

L-экстремаль – это единственный куб, который покрывает вершину из множества L, не покрываемую другими кубами из множества Z.

Для определения L-экстремалей воспользуемся операциями вычитания “#” и пересечения “∩” кубов.

#### Поиск L-экстремалей (из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты):

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | x10x0 | x1x10 | 1x0x0 | 1xx10 | x0x01 | x01x1 | 0x1x1 | 0xx01 | 10xxx | xx00x | xx11x | 01xxx |
| x10x0 | - | x1110 | 100x0 | 10x10 1x110 | x0x01 | x01x1 | 0x1x1 | 0xx01 | 10xxx | x000x xx001 | xx11x | 011xx 01xx1 |
| x1x10 | x1000 | - | 100x0 | 10x10 10110 | x0x01 | x01x1 | 0x1x1 | 0xx01 | 10xxx | x000x xx001 | x011x xx111 | 0110x 011x1 01xx1 |
| 1x0x0 | 01000 | x1110 | - | 10110 10110 | x0x01 | x01x1 | 0x1x1 | 0xx01 | 101xx 10xx1 | 0000x x0001 xx001 | x011x xx111 | 0110x 011x1 01xx1 |
| 1xx10 | 01000 | 01110 | 10000 | - | x0x01 | x01x1 | 0x1x1 | 0xx01 | 1010x 101x1 10xx1 | 0000x x0001 xx001 | 0011x x0111 xx111 | 0110x 011x1 01xx1 |
| x0x01 | 01000 | 01110 | 10000 | 10110 10110 | - | x0111 | 011x1 0x111 | 01x01 | 10100 10111 10x11 | 00000 x1001 | 0011x x0111 xx111 | 0110x 011x1 01xx1 |
| x01x1 | 01000 | 01110 | 10000 | 10110 10110 | x0001 | - | 011x1 01111 | 01x01 | 10100 10011 | 00000 x1001 | 00110 x1111 | 0110x 011x1 01xx1 |
| 0x1x1 | 01000 | 01110 | 10000 | 10110 10110 | x0001 | 10111 | - | 01001 | 10100 10011 | 00000 x1001 | 00110 11111 | 01100 010x1 |
| 0xx01 | 01000 | 01110 | 10000 | 10110 10110 | 10001 | 10111 | 01111 01111 | - | 10100 10011 | 00000 11001 | 00110 11111 | 01100 01011 |
| 10xxx | 01000 | 01110 | Ø | Ø | Ø | Ø | 01111 01111 | 01001 | - | 00000 11001 | 00110 11111 | 01100 01011 |
| xx00x | Ø | 01110 | Ø | Ø | Ø | Ø | 01111 01111 | Ø | 10100 10011 | - | 00110 11111 | 01100 01011 |
| xx11x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10100 10011 | 00000 11001 | - | 01100 01011 |
| 01xxx | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10100 10011 | 00000 11001 | 00110 11111 | - |
| Остаток | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10100 10011 | 00000 11001 | 00110 11111 | 01100 01011 |

Получили кубы, подозрительные на L-экстремальность.

##### Далее для расчетов используется исходное множество L без произведенного склеивания (проверка на L-экстремальность):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) ∩ L | 00101 | 01010 | 01011 | 01100 | 01101 | 10010 | 10011 | 10100 | 10101 | 11010 |
| 10100 | y010y Ø | yyyy0 Ø | yyyyy Ø | yy100 Ø | yy10y Ø | 10yy0 Ø | 10yyy Ø | 10100 | 1010y Ø | 1yyy0 Ø |
| 10011 | y0yy1 Ø | yy01y Ø | yy011 Ø | yyyyy Ø | yyyy1 Ø | 1001y Ø | 10011 | 10yyy Ø | 10yy1 Ø | 1y01y Ø |
| 00000 | 00y0y Ø | 0y0y0 Ø | 0y0yy Ø | 0yy00 Ø | 0yy0y Ø | y00y0 Ø | y00yy Ø | y0y00 Ø | y0y0y Ø | yy0y0 Ø |
| 11001 | yyy01 Ø | y10yy Ø | y10y1 Ø | y1y0y Ø | y1y01 Ø | 1y0yy Ø | 1y0y1 Ø | 1yy0y Ø | 1yy01 Ø | 110yy Ø |
| 00110 | 001yy Ø | 0yy10 Ø | 0yy1y Ø | 0y1y0 Ø | 0y1yy Ø | y0y10 Ø | y0y1y Ø | y01y0 Ø | y01yy Ø | yyy10 Ø |
| 11111 | yy1y1 Ø | y1y1y Ø | y1y11 Ø | y11yy Ø | y11y1 Ø | 1yy1y Ø | 1yy11 Ø | 1y1yy Ø | 1y1y1 Ø | 11y1y Ø |
| 01100 | 0y10y Ø | 01yy0 Ø | 01yyy Ø | 01100 | 0110y Ø | yyyy0 Ø | yyyyy Ø | yy100 Ø | yy10y Ø | y1yy0 Ø |
| 01011 | 0yyy1 Ø | 0101y Ø | 01011 | 01yyy Ø | 01yy1 Ø | yy01y Ø | yy011 Ø | yyyyy Ø | yyyy1 Ø | y101y Ø |

##### Множество L-экстремалей E = { 10xxx; 01xxx } – эти кубы обязательно должны войти в минимальное покрытие.

##### Z´ = Z - E = { x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 0x1x1; 0xx01; xx00x; xx11x }

Поиск непокрытых наборов:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00101 | 01010 | 01011 | 01100 | 01101 | 10010 | 10011 | 10100 | 10101 | 11010 |
| 10xxx | 00101 | 01010 | 01011 | 01100 | 01101 | Ø | Ø | Ø | Ø | 11010 |
| 01xxx | 00101 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11010 |
| Остаток | 00101 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11010 |

##### Множество кубов, непокрываемых L-экстремалями, L´ = L # E = { 00101; 11010 } – чтобы покрыть их, нужно воспользоваться множеством простых импликант, не являющихся L-экстремалями.

##### Покрытие оставшихся кубов:

Fmin S1 = { 10xxx; 01xxx; 1xx10; x01x1 } =  *+ \** *+*

|  |  |  |
| --- | --- | --- |
| Z´ ∩ L´ | 00101 | 11010 |
| x10x0 | 0yy0y Ø | 11010 |
| x1x10 | 0y1yy Ø | 11010 |
| 1x0x0 | y0y0y Ø | 11010 |
| 1xx10 | y01yy Ø | 11010 |
| x0x01 | 00101 | 1y0yy Ø |
| x01x1 | 00101 | 1yy1y Ø |
| 0x1x1 | 00101 | y1y1y Ø |
| 0xx01 | 00101 | y10yy Ø |
| xx00x | 00y01 Ø | 110y0 Ø |
| xx11x | 001y1 Ø | 11y10 Ø |

Проверку минимизации функции ***S1*** выполним при помощи карты Карно

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

Рисунок 3.1.2 – Минимизация функции***S1***при помощи карты Карно.

*+ \** *+*

Минимизацию функций ***П*** и ***S2*** проведем при помощи карт Карно.

Для функции ***П***:

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

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

*+*

Для функции ***S2***:

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

Рисунок 3.1.4 – Минимизация функции***S2***при помощи карты Карно.

*+*   *\**  \*  *+*  *\* \* + \* \**

Функциональная схема реализации ОЧС на мультиплексорах приведена в приложении В.

**3.2. 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.3.2.1).

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

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

Таблица 3.2.1 - 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 | 0 | 1 | 0 | 1 | 3∙0=00 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 3∙1=03 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | Выход 03 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 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 | 0 | 1 | 0 | 1 | 0∙0=00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Выход 00 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0∙0=00 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Выход 00 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0∙1=00 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход 00 |
| 0 | 1 | 1 | 1 | 0 | x | x | x | x | 0∙3=00 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | x | Выход 00 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1∙0=00 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Выход 01 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1∙1=01 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Выход 01 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1∙1=01 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Выход 01 |
| 1 | 0 | 1 | 1 | 0 | x | x | x | x | 1∙3=03 |

*Продолжение таблицы 3.2.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 1 | 0 | 1 | 1 | 1 | x | x | x | x | Выход 01 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2∙0=00 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Выход 02 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 2∙1=02 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Выход 02 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 2∙2=10 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Выход 02 |
| 1 | 1 | 1 | 1 | 0 | x | x | x | x | 2∙3=12 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | x | Выход 02 |

Минимизацию функций ***P1****,* ***P2***, ***P3*** и ***P4*** проведем при помощи карт Карно-Вейча.

Для функции ***P1***:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |  |  |
|  | 0 | 0 | \* | \* | \* | \* | 0 | 0 |  |  |
|  |  | 0 | 0 | \* | \* | \* | \* | 0 | 0 |  |
|  |  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |
|  |  |  | h | |  | | h | |  |  |  |

Рисунок 3.2.2 – Минимизация функции***P1***при помощи карты Вейча.

Для функции ***P2***:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |  |  |
|  | 1 | 0 | \* | \* | \* | \* | 0 | 1 |  |  |
|  |  | 1 | 0 | \* | \* | \* | \* | 0 | 1 |  |
|  |  | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |
|  |  |  | h | |  | | h | |  |  |  |

Рисунок 3.2.3 – Минимизация функции***P2***при помощи карты Вейча.

*= + \** \*  *+*  \* \*

Для функции ***P3***:

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

Рисунок 3.2.4 – Минимизация функции***P3***при помощи карты Карно.

*+ \** *+*  \* \* *+*  *\* \* + \* \**

Для функции ***P4***:

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

Рисунок 3.2.5 – Минимизация функции***P4***при помощи карты Карно.

*+ \** *+* \* \*

Функциональная схема реализации ОЧУ на мультиплексорах приведена в приложении Б.

**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.

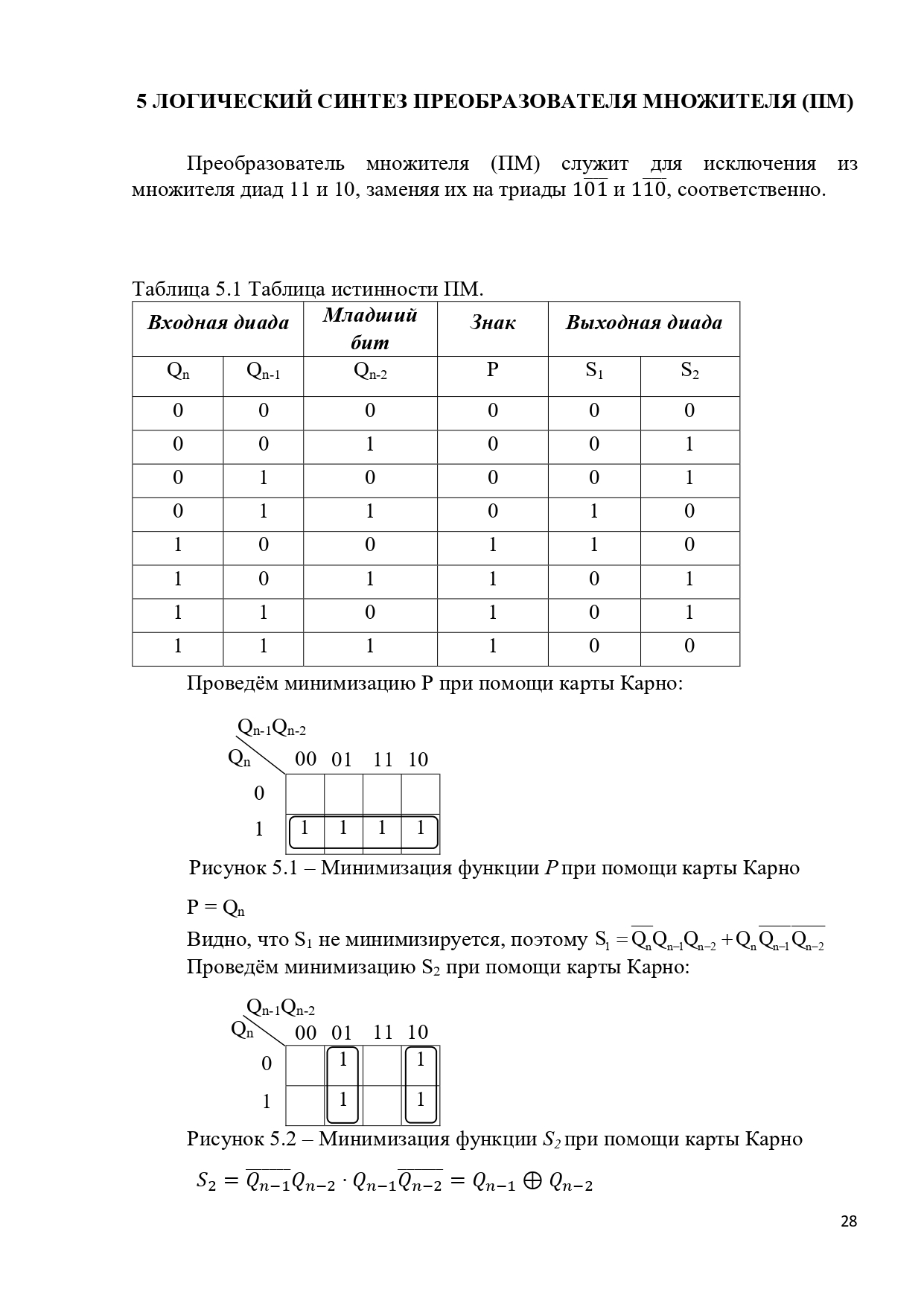
Функции ОЧС зависят от пяти переменных. Удобно взять мультиплексор с тремя адресными входами, это позволит упростить одну функцию от пяти аргументов до восьми функций от двух переменных. Функции от двух переменных достаточно просты для того, чтобы самостоятельно заметить их минимальную форму.

Таблица 4.1. – Таблица истинности для ОЧС на мультиплексорах

**!p**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***№*** | ***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 | **1**  **p**  **p** | X  **0** |  | X | **p** |
| 1 | 0 | 0 | 0 | 0 | 1 | X | X | X |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1  **!p** |
| 5 | 0 | 0 | 1 | 0 | 1 | 1 | 1  **p** | 0 |
| 6 | 0 | 0 | 1 | 1 | 0 | X | X | X |
| 7 | 0 | 0 | 1 | 1 | 1 | X | X | X |
| 8 | 0 | 1 | 0 | 0 | 0 | X | **0** | X  **1** |  | X |  |
| 9 | 0 | 1 | 0 | 0 | 1 | X | X | X |
| 10 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 11 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 12 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0  **p** |
| 13 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1  **1** |
| 14 | 0 | 1 | 1 | 1 | 0 | X | X | X |
| 15 | 0 | 1 | 1 | 1 | 1 | X | X | X |
| 16 | 1 | 0 | 0 | 0 | 0 | X | **p** | X | **1** | X  **1** | **p** |
| 17 | 1 | 0 | 0 | 0 | 1 | X | X  **0** | X |
| 18 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 19 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 20 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1  **!p** |
| 21 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 22 | 1 | 0 | 1 | 1 | 0 | X | X | X |
| 23 | 1 | 0 | 1 | 1 | 1 | X | X | X |
| 24 | 1 | 1 | 0 | 0 | 0 | X  **0** | **p** | X |  | X  **!p** | **!p** |
| 25 | 1 | 1 | 0 | 0 | 1 | X | X | X |
| 26 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 27 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 28 | 1 | 1 | 1 | 0 | 0 | 0 | 0  **0** | 0  **p** |
| 29 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 30 | 1 | 1 | 1 | 1 | 0 | X | X | X |
| 31 | 1 | 1 | 1 | 1 | 1 | X | X | X |

Функциональная схема реализации ОЧС на мультиплексорах приведена в приложении Г.

****

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=3\*5+3+5=23 | с=2\*4+4+2=14 | 1,64 |
| P2 | с=10\*5+10+5=65 | с=2\*1+3\*2+4+3=15 | 4,33 |
| P3 | с=11\*5+11+5=71 | с=2\*2+3\*3+3+5=21 | 3,38 |
| P4 | с=15\*5+15+5=95 | c=1+2\*3+4+3=14 | 6,79 |

Таблица 6.2 – Эффективность минимизации ОЧС

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вых.  схемы | Рассчитанная цена схемы | | | Эфф.  мин. k |
| До минимизации | | После минимизации |
|  | c=5\*5+5+5=35 |  | с=3\*3+2+3=14 | 2,5 |
|  | с=10\*5+5+10=65 |  | с=2\*2+2\*3+4+3=17 | 3,82 |
|  | с=8\*5+5+8=53 |  | с=4\*3+2+4=18 | 2,94 |

1. **ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ**

**Временные затраты на умножение**

Формула расчёта временных затрат на умножение:

Т=n\*(TПМ+TФДК+TОЧУ+(n+1)\*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”. – Минск.: БГУИР, 2016 г.

**ПРИЛОЖЕНИЕ А**

*(обязательное)*

Сумматор-умножитель первого типа. Схема электрическая структурная

**ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Одноразрядный четверичный умножитель.

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