Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра вычислительных машин, систем и сетей

Дисциплина: Арифметические и логические основы

цифровых устройств

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_\_\_\_\_\_ И. В. Лукьянова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-

УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 420 ПЗ

Студент П. А. Щербо

(гр. 250504)

Руководитель И. В. Лукьянова

МИНСК 2023

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

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

ЗАДАНИЕ

по курсовой работе студента

Щербо Павла Андреевича

1. Тема работы: «Проектирование и логический синтез сумматора- умножителя двоично-десятичных чисел»
2. Срок сдачи студентом законченной работы: до 20 мая 2023г.
3. Исходные данные к работе:
   1. Исходные сомножители: Мн = 36,39; Мт = 53,25;
   2. Алгоритм умножения: В;
   3. Метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в дополнительных кодах;
   4. Коды четверичных цифр множимого для перехода к двоично- четверичной системе кодирования: 04 – 11, 14 – 10, 24 – 00, 34 – 01;
   5. Тип синтезируемого умножителя: 1;
   6. Логический базис для реализации ОЧУ: ИЛИ, Константная единица, Сумма по модулю; метод минимизации – карты Карно-Вейча.
   7. Логический базис для реализации ОЧС: ИЛИ-НЕ; метод минимизации – Алгоритм Рота.
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. Умножитель-сумматор 1 типа. Схема электрическая структурная.
   2. Однозарядный четверичный умножитель. Схема электричская

функциональная.

* 1. Одноразрядный четверичный сумматор. Схема электрическая функциональная.
  2. Преобразователь множителя. Схема электрическая функциональная.
  3. Однозарядный 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 | 15.05-20.05 |  |

Дата выдачи задания: 10 февраля 2023г.

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

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_\_П. А. Щербо

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ .............................................................................................................5

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

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

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

3.1. Логический синтез одноразрядного четверичного умножителя.............................................................................................................

3.2. Логический синтез одноразрядного четверичного сумматора................................................................................................................

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 первого типа. Схема электрическая структурная .................................................................................

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

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

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

ПРИЛОЖЕНИЕ Д Одноразрядный четверичный сумматор. Реализация на мультиплексорах. 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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \_ 36 | 4 |  |  |  |  | 0,39 |
| 36 | \_ 9 | 4 |  |  | \* | 4 |
| 0 | 8 | 2 |  |  |  | 1,56 |
|  | 1 |  |  |  | \* | 4 |
|  |  |  |  |  | \* | 2,24  4 |
|  |  |  |  |  |  | 0,96 |

Мн4 = 210,120.

В соответствии с заданной кодировкой множимого:

Мн2/4 = 001011,100011

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \_ 53 | 4 |  |  |  |  | 0,25 |
| 52 | \_ 13 | 4 |  |  | \* | 4 |
| 1 | 12 | 3 |  |  |  | 1,00 |
|  | 1 |  |  |  |

Мт4 = 311,100.

В соответствии с заданной кодировкой множителя:

Мт2/4 = 110101,010000

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

Мн = 0,210120 РМн = 0.1101 +0310

Мт = 0,311100 РМт = 0.0011 +0310

Мн = 0,011110010110 РМн = 0.1001 +0310

Мт = 0,010001101001 РМт = 0.0011 +0310

Порядок произведения:

|  |  |  |  |
| --- | --- | --- | --- |
| РМн | = | 0.1101 | 034 |
| РМт | = | 0.0011 | 034 |
| РМн∙Мт | = | 1.0000 | 124 |
|  |  |  |  |

Знак произведения определяется суммой по модулю два знаков

сомножителей:

зн Мн ⊕ зн Мт = 0 ⊕ 0 = 0.

При умножении чисел в дополнительных кодах диада 11(34) заменяется на триаду , диада 10(24) заменяется на триаду Преобразованный множитель имеет вид М = 11100.

Перемножение мантисс по алгоритму «В» представлено в таблице 1.1

Таблица 1.1 — Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Четверичная С/С** | | **Двоично-четверичная С/C** | | **Комментарии** |
| **1** | | **2** | | **3** |
| 0. | 000000 000000 | 11. | 111111111111 111111111111 | ∑0ч |
| 0. | 000000 210120 | 11. | 111111111111 001011100011 | П1ч = Мн |
| 0.  0. | 000000 210120  000002 101200 | 11.  11. | 111111111111 001011100011  111111111100 101110001111 | ∑1ч  ∑1ч \* 41 |
| 3. | 333333 123220 | 01. | 010101010101 100001000011 | П2ч = -[Мн] |
| 0.  0. | 000001 231020  000012 310200 | 11.  11. | 111111111110 000110110011  111111111000 011011001111 | ∑2ч  ∑2ч \* 41 |
| 0. | 000000 210120 | 11. | 111111111111 001011100011 | П3ч = Мн |
| 0.  0. | 000013 120320  000131 203200 | 11.  11. | 111111111001 100011010011  111111100110 001101001111 | ∑3ч  ∑3ч \* 41 |
| 0. | 000000 210120 | 11. | 111111111111 001011100011 | П4ч = Мн |
| 0.  0. | 000132 013320  001320 133200 | 11.  11. | 111111100100 111001010011  111110010011 100101001111 | ∑4ч  ∑4ч \* 41 |
| 0. | 000000 210120 | 11. | 111111111111 001011100011 | П5ч = Мн |
| 0.  0. | 001321 003320  013210 033200 | 11.  11. | 111110010010 111101010011  111001001011 110101101111 | ∑5ч  ∑5ч \* 41 |
| 0. | 000000 000000 | 11. | 111111111111 111111111111 | П6ч = 0 |
| 0. | 013210 033200 | 11. | 111001001011 110101001111 | ∑6ч |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн4 ∙ Мт4 = 0,032211 312130 , РМн ∙ Мт = 7) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн4 ∙ Мт4 = 132100,33200 РМн ∙ Мт = 0;

Мн10 ∙ Мт10 = 1 936,96875.

Результат прямого перемножения операндов дает следующее:

Мн10 ∙ Мт10 = 52,95\*17,65 = 1 937,7675.

Абсолютная погрешность:

Δ = 1 937,7675 – 1 936,96875 = .

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* (на входе Mul/sum – «1»), то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода (ФДК) *F2* поступает «1» (рисунок 2.1).

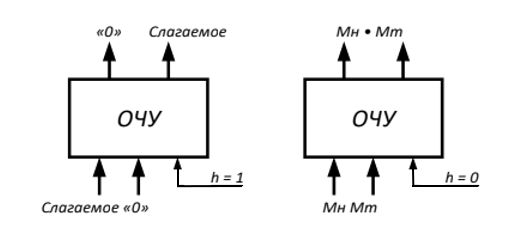


Рисунок 2.1 – Режимы работы ОЧУ

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

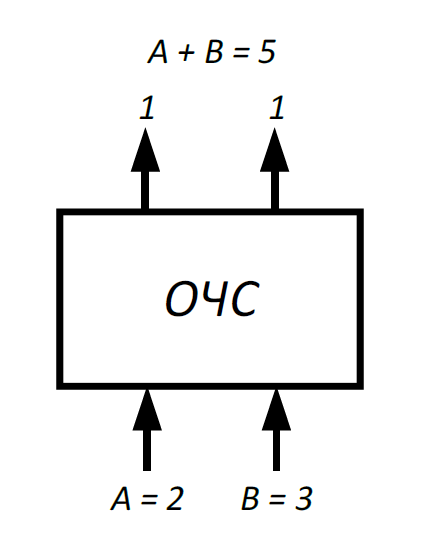


Рисунок 2.2 – Одноразрядный четверичный сумматор

Структурная схема сумматора-умножителя приведена в приложении А.

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

**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 представлен с помощью таблицы истинности (таблица 3.1.1).

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

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

Управляющий вход *h* определяет тип операции:

«0» – умножение закодированных цифр, поступивших на информационные входы;

«1» – вывод на выходы без изменения значения разрядов, поступивших из регистра множимого.

В таблице 3.1.1 выделено восемь безразличных наборов, т.к. на входы ОЧУ из разрядов множителя не может поступить код «11».

Таблица 3.1.1 — Таблица истинности ОЧУ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Мн** | | **Мт** | | **Упр.** | **Старшие**  **разряды** | | **Младшие**  **разряды** | | **Пример операции в четверичной с/с** |
| ***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 | 2\*0=00 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Выход – код «02» |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 2\*1=02 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | Выход – код «02» |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 2\*2=10 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | Выход – код «02» |
| 0 | 0 | 1 | 1 | 0 | x | x | x | x | 2\*3=12 |
| 0 | 0 | 1 | 1 | 1 | x | x | x | x | Выход – код «02» |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 3\*0=00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Выход – код «03» |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 3\*1=03 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Выход – код «03» |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 3\*2=12 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход – код «03» |
| 0 | 1 | 1 | 1 | 0 | x | x | x | x | 3\*3=21 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | x | Выход – код «03» |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1\*0=00 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Выход – код «01» |

Продолжение таблицы 3.1.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1\*1=01 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Выход – код «01» |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1\*2=02 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Выход – код «01» |
| 1 | 0 | 1 | 1 | 0 | x | x | x | x | 1\*3=03 |
| 1 | 0 | 1 | 1 | 1 | x | x | x | x | Выход – код «01» |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0\*0=00 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Выход – код «00» |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0\*1=00 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Выход – код «00» |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0\*2=00 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Выход – код «00» |
| 1 | 1 | 1 | 1 | 0 | x | x | x | x | 0\*3=00 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | x | Выход – код «00» |

Минимизацию функций *Р* проведем с помощью карт Карно. В рисунках 3.1.1 – 3.1.4 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

**Минимизация функции *Р*1:**

y1y2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1x2h | 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 3.1.1 — Минимизация функции *Р*1 картой Карно

Следовательно:

P1min =

Запишем результат в базисе ИЛИ:

P1min =

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

*K = =* 18,6

**Минимизация функции *Р*2:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1x2h | 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 | | 0 | |  |  | |
| 11 | | 1 | | 1 | | 1 | | 1 | | x | | x | | 1 | | 1 | |  |
| 10 | | 1 | | 1 | | 1 | | 1 | | x | | x | | 1 | | 0 | |  |  | |

Рисунок 3.1.2 — Минимизация функции *Р*2 картой Карно

Следовательно:

P2min =

Запишем результат в базисе ИЛИ:

P2min =

Эффективность минимизации:

*K = =* 13,5

**Минимизация функции *Р*3:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1x2h | 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 | | 0 | |  |  | |
| 11 | | 1 | | 1 | | 1 | | 1 | | x | | x | | 1 | | 1 | |  |
| 10 | | 1 | | 1 | | 1 | | 1 | | x | | x | | 1 | | 0 | |  |  | |

Рисунок 3.1.3 — Минимизация функции *Р*3 картой Карно

Следовательно:

P3min =

Запишем результат в базисе ИЛИ:

P3min =

Эффективность минимизации:

*K = =* 6,5

**Минимизация функции *Р*4:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1x2h | 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 | | 0 | |  |  | |
| 11 | | 1 | | 1 | | 1 | | 1 | | x | | x | | 1 | | 1 | |  |
| 10 | | 1 | | 0 | | 0 | | 0 | | x | | x | | 0 | | 0 | |  |  | |

Рисунок 3.1.4 — Минимизация функции *Р*4 картой Карно

Следовательно:

P4min =

Запишем результат в базисе ИЛИ:

P4min =

Эффективность минимизации:

*K = =* 7,5

Функциональная схема ОЧУ приведена в приложении Б.

**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 – 11, 1 – 10, 2 – 00, 3 – 01;

Так как ОЧС синтезируется для схемы первого типа, то в таблице истинности необходимо выделить 16 безразличных наборов, т. к. со старших выходов ОЧУ не могут прийти коды «2» и «3».

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2+1+0=03 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 2+1+1=10 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2+0+0=02 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2+0+1=03 |
| 0 | 1 | 0 | 0 | 0 | x | x | x | 3+2+0=11 |
| 0 | 1 | 0 | 0 | 1 | x | x | x | 3+2+1=12 |
| 0 | 1 | 0 | 1 | 0 | x | x | x | 3+3+0=12 |
| 0 | 1 | 0 | 1 | 1 | x | x | x | 3+3+1=13 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 3+1+0=10 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 3+1+1=11 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 3+0+0=03 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3+0+1=10 |
| 1 | 0 | 0 | 0 | 0 | x | x | x | 1+2+0=03 |
| 1 | 0 | 0 | 0 | 1 | x | x | x | 1+2+1=10 |
| 1 | 0 | 0 | 1 | 0 | x | x | x | 1+3+0=10 |
| 1 | 0 | 0 | 1 | 1 | x | x | x | 1+3+1=11 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1+1+0=02 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1+1+1=03 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1+0+0=01 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1+0+1=02 |
| 1 | 1 | 0 | 0 | 0 | x | x | x | 0+2+0=02 |
| 1 | 1 | 0 | 0 | 1 | x | x | x | 0+2+1=03 |
| 1 | 1 | 0 | 1 | 0 | x | x | x | 0+3+0 =03 |
| 1 | 1 | 0 | 1 | 1 | x | x | x | 0+3+1=10 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0+1+0=01 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0+1+1=02 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0+0+0=00 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0+0+1=01 |

Продолжение таблицы 3.2.1

**Минимизация функции *П*:**

Минимизацию функции*П* проведем с помощью карт Вейча. Для функции *П* заполненная карта приведена на рисунке 3.2.2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | a1 | | | | | | | |  | |  | |  | |  | |  | |  | | |
|  |  |  | |  | |  | |  | |  | |  | |  | |  | |  | | |  | | |
| b1 |  | | 0 | | 0 | | 0 | | 0 | | 1 | | 1 | | 1 | | 0 | |  | | |  | | | |
|  | | 0 | | 0 | | 0 | | 0 | | 0 | | 1 | | 0 | | 0 | |  | | | b2 | | |
|  |  | | x | | x | | x | | x | | x | | x | | x | | x | |  | | |
|  |  | | x | | x | | x | | x | | x | | x | | x | | x | |  | | |  | | |
|  |  |  | |  | |  | |  | |  | |  | |  | |  | |  | | |  | | |
|  |  |  | |  | | a2 | | | | | | | |  | |  | |  | | |  | | |
|  |  |  | | p | | | |  | |  | | p | | | |  | |  | | |  | | |

Рисунок 3.2.2 — Минимизация функции *П* картой Вейча

Следовательно:

Пmin =

Запишем результат в базисе ИЛИ-НЕ:

Пmin =

Эффективность минимизации:

K = = 2,0

**Минимизация функции *S1*:**

Определим множество единичных кубов:

и множество безразличных кубов:

На рисунке 3.2.3 приведена карта Карно для минимизации множества N.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a2 | b1b2П |  | | | | | | | |  | |  | |  | |  | |  | |  | | |
|  |  | 000 | | 001 | | 011 | | 010 | | 110 | | 111 | | 101 | | 100 | |  | | |  | | |
|  | 00 | | x | | x | | x | | x | |  | |  | |  | |  | |  | | |  | | | |
| 01 | | x | | x | | x | | x | |  | |  | |  | |  | |  | | |  | | |
|  | 11 | | x | | x | | x | | x | |  | |  | |  | |  | |  | | |
|  | 10 | | x | | x | | x | | x | |  | |  | |  | |  | |  | | |  | | |
|  |  |  | |  | |  | |  | |  | |  | |  | |  | |  | | |  | | |
|  |  |  | |  | |  | | | | | | | |  | |  | |  | | |  | | |

Рисунок 3.2.3 – Карта Вейча для минимизации множества N

Следовательно:

Сформируем множество С0 = L ⋃ N:

C0 = {00101, 01100, 01101, 01111, 10110, 11100, 11110, 11111, xx0xx }

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 3.2.4

Таблица 3.2.4 – Поиск простых импликант (С0\*С0)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C0\*C0 | 00101 | 01100 | 01101 | 01111 | 10110 | 11100 | 11110 | 11111 | xx0xx |
| 00101 | - |  |  |  |  |  |  |  |  |
| 01100 |  | - |  |  |  |  |  |  |  |
| 01101 | 0y101 | 0110y | - |  |  |  |  |  |  |
| 01111 |  |  | 011y1 | - |  |  |  |  |  |
| 10110 |  |  |  |  | - |  |  |  |  |
| 11100 |  | y1100 |  |  |  | - |  |  |  |
| 11110 |  |  |  |  | 1y110 | 111y0 | - |  |  |
| 11111 |  |  |  | y1111 |  |  | 1111y | - |  |
| xx0xx | 00y01 | 01y00 | 01y01 | 01y11 | 10y10 | 11y00 | 11y10 | 11y11 | - |
| A1 | 0x101 00x01 | 0110x x1100 01x00 | 011x1 01x01 | x1111 01x11 | 1x110 10x10 | 111x0 11x00 | 1111x 11x10 | 11x11 | Ø |

В результате этой операции сформируется новое множество кубов:

С1 = {0x101, 00x01, 0110x, x1100, 01x00, 011x1, 01x01, x1111, 01x11, 1x110, 10x10, 111x0, 11x00, 1111x, 11x10, 11x11, xx0xx}

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

В таблице 3.2.5 приведён следующий шаг поиска простых импликант с помощью операции С1\*С

Таблица 3.2.5 – Поиск простых импликант (С1\*С1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C1\*C1 | 0x101 | 00x01 | 0110x | x1100 | 01x00 | 011x1 | 01x01 | x1111 | 01x11 | 1x110 | 10x10 | 111x0 | 11x00 | 1111x | 11x10 | 11x11 |
| 0x101 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x01 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0110x |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x1100 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 01x00 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 011x1 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 01x01 |  | 0yx01 |  |  | 01x0y |  | - |  |  |  |  |  |  |  |  |  |
| x1111 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 01x11 |  |  |  |  |  |  | 01xy1 |  | - |  |  |  |  |  |  |  |
| 1x110 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 10x10 |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 111x0 |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 11x00 |  |  |  |  | y1x00 |  |  |  |  |  |  |  | - |  |  |  |
| 1111x |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |
| 11x10 |  |  |  |  |  |  |  |  |  |  | 1yx10 |  | 11xy0 |  | - |  |
| 11x11 |  |  |  |  |  |  |  |  | y1x11 |  |  |  |  |  | 11x1y | - |
| xx0xx | 0xy01 |  | 01y0x | x1y00 |  | 01yx1 |  | x1y11 |  | 1xy10 |  | 11yx0 |  | 11y1x |  |  |
| A2 | 0xx01 | 0xx01 | 01x0x | x1x00 | 01x0x x1x00 | 01xx1 | 01xx1 | x1x11 | x1x11 | 1xx10 | 1xx10 | 11xx0 | 11xx0 | 11x1x | 11x1x | Ø |

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

С2 = {0xx01, 01x0x, x1x00, 01xx1, x1x11, 1xx10, 11xx0, 11x1x, xx0xx}

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

В таблице 3.2.6 приведён следующий шаг поиска простых импликант с помощью операции С2\*С2.

Таблица 3.2.6 – Поиск простых импликант (С2\*С2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C2\*C2 | 0xx01 | 01x0x | x1x00 | 01xx1 | x1x11 | 1xx10 | 11xx0 | 11x1x |
| 0xx01 | - |  |  |  |  |  |  |  |
| 01x0x |  | - |  |  |  |  |  |  |
| x1x00 |  |  | - |  |  |  |  |  |
| 01xx1 |  |  |  | - |  |  |  |  |
| x1x11 |  |  |  |  | - |  |  |  |
| 1xx10 |  |  |  |  |  | - |  |  |
| 11xx0 |  |  |  |  |  |  | - |  |
| 11x1x |  |  |  |  |  |  |  | - |
| xx0xx |  |  |  |  |  |  |  |  |
| A3 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Новых кубов (третьей размерности) не образовалось.

На этом заканчивается этап поиска простых импликант. Множество простых импликант:

Z = Z0 ⋃ Z1 ⋃ Z2 = {0xx01, 01x0x, x1x00, 01xx1, x1x11, 1xx10, 11xx0, 11x1x, xx0xx}

Поиск L-экстремалей на множестве простых импликант (таблица 3.2.7).

Таблица 3.2.7 – Поиск L-экстремалей

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 0xx01 | 01x0x | x1x00 | 01xx1 | x1x11 | 1xx10 | 11xx0 | 11x1x |
| 0xx01 | - | 01x00 | x1x00 | 01x11 | x1x11 | 1xx10 | 11xx0 | 11x1x |
| 01x0x | 00x01 | - | 11x00 | 01x11 | x1x11 | 1xx10 | 11xx0 | 11x1x |
| x1x00 | 00x01 | Ø | - | 01x11 | x1x11 | 1xx10 | 11x10 | 11x1x |
| 01xx1 | 00x01 | Ø | 11x00 | - | 11x11 | 1xx10 | 11x10 | 11x1x |
| x1x11 | 00x01 | Ø | 11x00 | Ø | - | 1xx10 | 11x10 | 11x10 |
| 1xx10 | 00x01 | Ø | 11x00 | Ø | 11x11 | - | Ø | Ø |
| 11xx0 | 00x01 | Ø | Ø | Ø | 11x11 | 10x10 | - | Ø |
| 11x1x | 00x01 | Ø | Ø | Ø | Ø | 10x10 | Ø | - |
| xx0xx | 00101 | Ø | Ø | Ø | Ø | 10110 | Ø | Ø |
| Остаток | 00101 | Ø | Ø | Ø | Ø | 10110 | Ø | Ø |

В таблице 3.2.7 из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты *z#(Z-z)*, за исключением импликанты xx0xx, т.к. она находиться во множестве безразличных наборов и не содержит(не покрывает) ни один куб из множества L.

Множество L-экстремалей E = {0xx01, 1xx10}.

Необходимо проверить, нет ли среди полученных L-экстремалей таких, которые стали L-экстремалями за счёт безразличных кубов (таблица 3.2.8).

Таблица 3.2.8 – Проверка L-экстремалей

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *z#(Z-z) ∩ L* | 00101 | 01100 | 01101 | 01111 | 10110 | 11100 | 11110 | 11111 |
| 00101 | 00101 |  |  |  |  |  |  |  |
| 10110 |  |  |  |  | 10110 |  |  |  |
| 1000x |  |  |  |  |  |  |  |  |
| 100x1 |  |  |  |  |  |  |  |  |
| 10011 |  |  |  |  |  |  |  |  |
| 100x1 |  |  |  |  |  |  |  |  |

По результатам таблицы 3.2.8:

E = {0xx01, 1xx10}.

Далее необходимо проанализировать, какие из исходных единичных кубов (множество L) не покрыты найденными L-экстремалями (таблица 3.2.9).

Таблица 3.2.9 – Поиск непокрытых исходных наборов

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00101 | 01100 | 01101 | 01111 | 10110 | 11100 | 11110 | 11111 |
| 0xx01 | Ø | 01100 | Ø | 01111 | 10110 | 11100 | 11110 | 11111 |
| 1xx10 | Ø | 01100 | Ø | 01111 | Ø | 11100 | Ø | 11111 |
| Остаток | Ø | 01100 | Ø | 01111 | Ø | 11100 | Ø | 11111 |

##### Теперь из полученного множества надо выбрать куб с минимальной ценой (максимальной размерностью). Из таблицы видно, что не покрываются L-экстремалями кубы L = { 01100; 01111; 11100; 11111}. После чего выполним пересечение набора из множества L с кубами из . Результат пересечения приведен в таблице 3.1.10:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L ∩ Ž | 01100 | 01111 | 11100 | 11111 |
| 01x0x | 01100 | Ø | Ø | Ø |
| x1x00 | 01100 | Ø | 11100 | Ø |
| 01xx1 | Ø | 01111 | Ø | Ø |
| x1x11 | Ø | 01111 | Ø | 11111 |
| 11xx0 | Ø | Ø | 11100 | Ø |
| 11x1x | Ø | Ø | Ø | 11111 |
| xx0xx | Ø | Ø | Ø | Ø |

Следовательно:

S1min =

Запишем результат в базисе ИЛИ-НЕ:

S1min =

Эффективность минимизации:

K = = 2,8

**Минимизация функции *S2*:**

Минимизацию функции*S2* проведем с помощью карт Карно. Для функции *S2* заполненная карта приведена на рисунке 3.2.10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | a1 | | | | | | | |  | |  | |  | |  | |  | |  | | |
|  |  |  | |  | |  | |  | |  | |  | |  | |  | |  | | |  | | |
| b1 |  | | 0 | | 1 | | 0 | | 0 | | 1 | | 0 | | 1 | | 1 | |  | | |  | | | |
|  | | 0 | | 1 | | 0 | | 1 | | 1 | | 1 | | 1 | | 0 | |  | | | b2 | | |
|  |  | | x | | x | | x | | x | | x | | x | | x | | x | |  | | |
|  |  | | x | | x | | x | | x | | x | | x | | x | | x | |  | | |  | | |
|  |  |  | |  | |  | |  | |  | |  | |  | |  | |  | | |  | | |
|  |  |  | |  | | a2 | | | | | | | |  | |  | |  | | |  | | |
|  |  |  | | p | | | |  | |  | | p | | | |  | |  | | |  | | |

Рисунок 3.2.10 — Минимизация функции S2 картой Вейча

Следовательно:

S2min=

Запишем результат в базисе ИЛИ-НЕ:

S2min=

Эффективность минимизации:

K = = 2,8

Функциональная схема ОЧС приведена в приложении В.

**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 диады 11 (34) и 10 (24) на триады и соответственно.

Таблица истинности для синтеза ПМ приведена в таблице 3.3.1.

Таблица 3.3.1 – Таблица истинности ПМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Входная диада** | | **Младший разряд** | **Знак** | **Выходная диада** | |
| ***a*1** | ***a*2** | ***p*** | ***Q*** | ***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 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Минимизацию переключательных функций произведём с помощью карт Вейча.

**Минимизация функции Q:**

Для функции Q заполненная карта приведена на рисунке 3.3.1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | a2 | |  |  |
|  |  |  |  |  |  |
| a1 |  | 1 | 0 | 1 | 1 |
|  |  | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |
|  |  |  | p | |  |

Рисунок 3.3.1 – Минимизация функции Q при помощи карты Вейча

Следовательно:

Эффективность минимизации:

K = = 2,5

**Минимизация функции S1:**

Для функции S1 Заполненная карта приведена на рисунке 3.3.2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | *a*2 | |  |  |
|  |  |  |  |  |  |
| *a*1 |  | 0 | 0 | 0 | 1 |
|  |  | 0 | 1 | 0 | 0 |
|  |  |  |  |  |  |
|  |  |  | *p* | |  |

Рисунок 3.3.2 – Минимизация функции S1 при помощи карты Вейча

Следовательно:

Эффективность минимизации:

K = = 1

**Минимизация функции S2:**

Для функции S2 заполненная карта приведена на рисунке 3.3.3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | *a*2 | |  |  |
|  |  |  |  |  |  |
| *a*1 |  | 1 | 0 | 1 | 0 |
|  |  | 1 | 0 | 1 | 0 |
|  |  |  |  |  |  |
|  |  |  | *p* | |  |

Рисунок 3.3.3 – Минимизация функции S2 при помощи карты Вейча

Следовательно:

Эффективность минимизации:

K = = 9,5

Функциональная схема ПМ приведена в приложении Г.

# **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 | x | **-** | x | **-** | x | **-** |
| 0 | 0 | 0 | 0 | 1 | x | x | x |
| 0 | 0 | 0 | 1 | 0 | x | x | x |
| 0 | 0 | 0 | 1 | 1 | x | x | x |
| 0 | 0 | 1 | 0 | 0 | 0 | p | 0 | p | 1 |  |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | x | **-** | x | **-** | x | **-** |
| 0 | 1 | 0 | 0 | 1 | x | x | x |
| 0 | 1 | 0 | 1 | 0 | x | x | x |
| 0 | 1 | 0 | 1 | 1 | x | x | x |
| 0 | 1 | 1 | 0 | 0 | 1 |  | 1 |  | 1 |  |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Продолжение таблицы 4.1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | x | **-** | x | **-** | x | **-** |
| 1 | 0 | 0 | 0 | 1 | x | x | x |
| 1 | 0 | 0 | 1 | 0 | x | x | x |
| 1 | 0 | 0 | 1 | 1 | x | x | x |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |  | 0 | p |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | x | **-** | x | **-** | x | **-** |
| 1 | 1 | 0 | 0 | 1 | x | x | x |
| 1 | 1 | 0 | 1 | 0 | x | x | x |
| 1 | 1 | 0 | 1 | 1 | x | x | x |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |  | 0 |  |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

Функциональная схема ПФ ОЧС приведена в приложении Д.

**5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ**

Формула расчёта временных затрат на умножение:

Т=𝑛∗(𝑇ПМ+𝑇ФДК+𝑇ОЧУ+6\*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 в регистрах;

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.

**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**)**

Преобразователь множителя. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Д**

**(**обязательное**)**

Однозарядный четверичный сумматор. Схема электрическая функциональная на основе мультиплексоров

**ПРИЛОЖЕНИЕ Е**

**(**обязательное**)**

Ведомость документов