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

\_\_\_\_\_\_\_\_\_\_ Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-

УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 116 ПЗ

Студент Е.О. Лукьянов

Руководитель Ю.А. Луцик

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

Лукьянова Евгения Олеговича

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 = 52,33; Мт = 37,62;
   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 – 01, 14 – 10, 24 – 00, 34 – 11;
   5. Тип синтезируемого умножителя: 2;
   6. Логический базис для реализации ОЧС: И, НЕ; метод минимизации – алгоритм Рота.
   7. Логический базис для реализации ОЧУС: И, Исключающее ИЛИ, Генератор единицы; метод минимизации – карты Карно-Вейча
4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

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

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

функциональная.

* 1. Однозарядный четверичный умножитель. Схема электрическая функциональная. Однозарядный четверичный умножитель. Схема электрическая функциональная.
  2. Однозарядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.
  3. Преобразователь множителя. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объём этапа, % | Срок выполнения этапа | Примечания |
| Разработка алгоритма умножения | 10 | 14.02-20.02 |  |
| 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: 14 февраля 2023г.

Руководитель \_\_\_\_\_\_\_\_\_Ю.А. Луцик

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_\_Е.О. Лукьянов

**СОДЕРЖАНИЕ**

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

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

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

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

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

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

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

4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ

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

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

ПРИЛОЖЕНИЕ А Сумматор-умножитель первого типа. Схема электрическая структурная .................................................................................

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

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

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

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

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

**ВВЕДЕНИЕ**

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

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

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

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

**1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ**

1. Перевод сомножителей из десятичной системы счисления в четверичную.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| \_ 52 | 4 |  |  |  |  |  | 0,33 |
| 52 | \_ 13 | 4 |  |  |  | \* | 4 |
| 0 | 12 | 3 |  |  |  |  | 1,32 |
|  | 1 |  |  |  |  | \* | 4 |
|  |  |  |  |  |  |  | 1,28 |
|  |  |  |  |  |  | \* | 4 |
|  |  |  |  |  |  | \* | 1,12  4 |
|  |  |  |  |  |  |  | 0,48 |

Мн4 = 310,1110.

В соответствии с заданной кодировкой множимого:

Мн2/4 = 111001,10101001

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| \_ 37 | 4 |  |  |  |  |  | 0,62 |
| 36 | \_ 9 | 4 |  |  |  | \* | 4 |
| 1 | 8 | 2 |  |  |  |  | 2,48 |
|  | 1 |  |  |  |  | \* | 4 |
|  |  |  |  |  |  |  | 1,92 |
|  |  |  |  |  |  | \* | 4 |
|  |  |  |  |  |  | \* | 3,68  4 |
|  |  |  |  |  |  |  | 2,72 |

Мт4 = 211,2132.

В соответствии с заданной кодировкой множителя:

Мт2/4 = 100101,10011110

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

Мн = 0,11100110101001 РМн = 0.0111 + 0310

Мт = 0,10010110011110 РМт = 0.0011 + 0310

Порядок произведения:

|  |  |  |  |
| --- | --- | --- | --- |
| РМн | = | 0.0111 | 034 |
| РМт | = | 0.0011 | 034 |
| РМн∙Мт | = | 0.1000 | 124 |
|  |  |  |  |

Знак произведения определяется суммой по модулю два знаков

сомножителей:

зн. Мн ⊕ зн. Мт = 0 ⊕ 0 = 0.

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

Перемножение мантисс по алгоритму «А» представлено в таблице 1.1

Таблица 1.1 — Перемножение мантисс

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Четверичная с/с** | | | **Двоично-четверичная с/с** | | | **Комментарии** |
| 1 | | | 2 | | | 3 |
| 0.  0.  0.  0.  3.  3.  3.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. | 00000000  12202220  12202220  01220222  30232230  32113112  33211311  12202220  12020131  01202013  12202220  20010233  02001023  03101110  11102133  01110213  03101110  10211323  01021132  12202220  13230012  01323001 | 0  0  20  20  120  120  3120  3120  33120  33120  333120  333120  2333120 | 0.  0.  0.  0.  1.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. | 0101010101010101  1000000100000001  1000000100000001  0110000001000000  1101001100001101  1100101011101000  1111001010111010  1000000100000001  1000010001101110  0110000100011011  1000000100000001  0001011001001111  0100010110010011  0111100110101001  1010100100101111  0110101001001011  0111100110101001  1001001010110011  0110010010101100  1000000100000001  1011001101011000  0110110011010110 | 01  01  0001  0001  100001  100001  11100001  11100001  1111100001  1111100001  111111100001  111111100001  00111111100001 | ∑0ч  П1ч =Мн2  ∑1ч  ∑1ч 4-1  П2ч =-1Мн  ∑2ч  ∑2ч4-1  П3ч =Мн2  ∑3ч  ∑3ч4-1  П4ч =Мн2  ∑4ч  ∑4ч4-1  П5ч =Мн1  ∑5ч  ∑5ч4-1  П6ч =Мн1  ∑6ч  ∑6ч4-1  П7ч =Мн2  ∑7ч  ∑7ч4-1 |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн4 Мт4 = 0.013230012 333120, = 7) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

= 132300,12333120 = 0;

= 1968,4368.

Результат прямого перемножения операндов дает следующее:

Мн10 Мт10 = 52,33 37,62= 1968,6546.

Абсолютная погрешность:

Δ = 1968,6546 – 1968,4368 = 0,2178.

Относительная погрешность:

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

**2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ**

*Если устройство работает как сумматор*, то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода F2 поступает «1». Необходимо обеспечить выполнение алгоритма сложения чисел, представленных в форме с плавающей запятой, базируясь на схеме умножителя, реализующего заданный алгоритм умножения.

Первое слагаемое переписывается в регистр результата под действием управляющих сигналов, поступающих на входы h всех ОЧУС (рисунок 2.1).

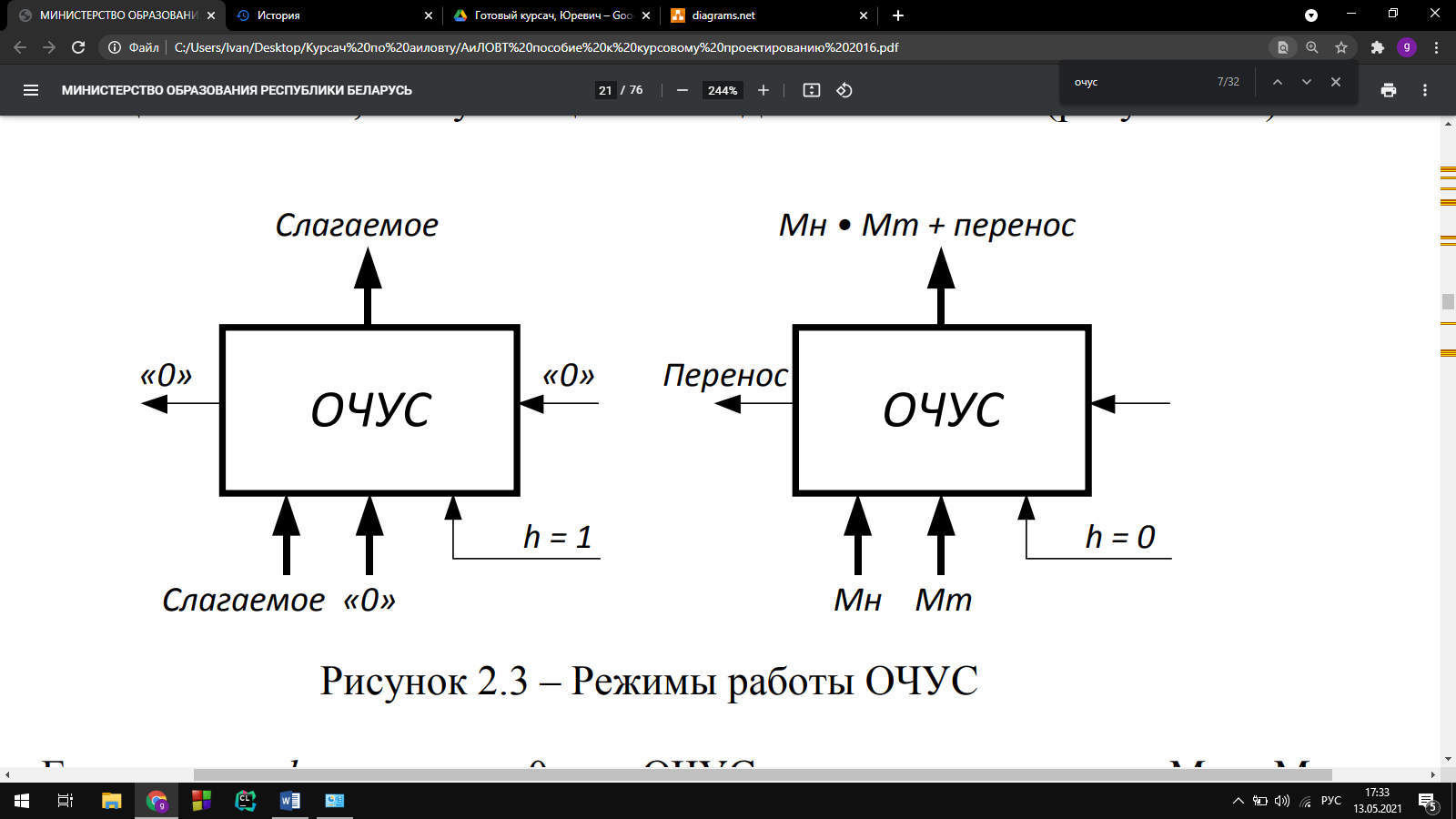


Рисунок 2.1 – Режимы работы ОЧУС

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

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

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

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

*Если устройство работает как умножитель*, то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК F2 поступает «0».

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

В регистре множителя после каждого такта умножения содержимое сдвигается на два двоичных разряда, и в конце умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения при умножении по алгоритму «А».

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

Принцип работы ФДК в зависимости от управляющих сигналов отражён в таблице 2.1.

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

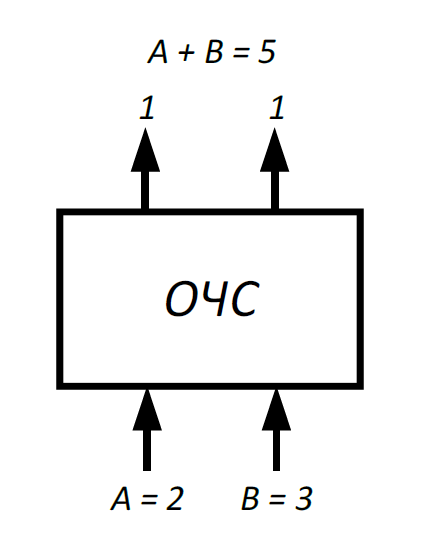
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | ∙ | 2 | = | 12 | (+1 в случае переноса из предыдущего ОЧУС) | |
| max |  | max |  | max | |  | |
| Мн |  | Мт |  | Перенос | |  | |

Так как на входы ОЧУС из регистра Мт не могут поступить коды «3», в таблице истинности работы ОЧУС будут содержаться 16 безразличных входных наборов.

Частичные произведения, получаемые на выходах ОЧУС, складываются с накапливаемой частичной суммой из регистра результата с помощью цепочки ОЧС (на первом такте выполняется сложение с нулём).

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

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

Рисунок 2.2 – Одноразрядный четверичный сумматор

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

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

Разрядность аккумулятор должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения переноса при суммировании.

*Если устройство работает как умножитель* (на входе *Mul/sum* - «0»), то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК *F2* поступает «0».

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

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

Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание (01). В этом случае инициализируется управляющий вход F1 формирователя дополнительного кода (ФДК) и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на «-1»). Принцип работы ФДК, в зависимости от управляющих сигналов, приведён в таблице 2.1.

Таблица 2.1 – Режимы работы формирователя дополнительного кода

|  |  |  |
| --- | --- | --- |
| **Сигналы на входах ФДК** | | **Результат на выходах ФДК** |
| *F1* | *F2* |
| 0 | 0 | Дополнительный код множимого |
| 0 | 1 | Дополнительный код слагаемого |
| 1 | 0 | Меняется знак Мн |
| 1 | 1 | Меняется знак слагаемого |

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

ОЧУ предназначен лишь для умножения двух четверичных цифр. Если в процессе умножения возникает перенос в следующий разряд, необходимо предусмотреть возможность его прибавления.

Для суммирования результата умножения текущей диады Мн·Мт с переносом из предыдущей диады предназначен ОЧС. Следовательно, чтобы полностью сформировать частичное произведение четверичных сомножителей, необходима комбинация цепочек ОЧУ и ОЧС.

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

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

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

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

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

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

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

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

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

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

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

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

Управляющий вход *h* определяет тип операции:

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

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

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

Таблица 3.1.1 — Таблица истинности ОЧУС

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Пер.** | **Мн.** | | **Мт.** | | **Упр.** | **Перенос** | **Результат** | | **Пример операции в четверичной с/с** |
| ***p*** | ***x1*** | ***x2*** | ***y1*** | ***y2*** | ***h*** | ***P*** | ***Q1*** | ***Q2*** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 0 + 0 = 00 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | Выход – код «02» |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 1 + 0 = 02 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | Выход – код «02» |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 2 2 + 0 = 10 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | Выход – код «02» |
| 0 | 0 | 0 | 1 | 1 | 0 | x | x | x | 2 3 + 0 = 12 |
| 0 | 0 | 0 | 1 | 1 | 1 | x | x | x | Выход – код «02» |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 0 + 0= 00 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Выход – код «00» |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 1 + 0 = 00 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | Выход – код «00» |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 2 + 0 = 00 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Выход – код «00» |
| 0 | 0 | 1 | 1 | 1 | 0 | x | x | x | 0 3 + 0 = 00 |
| 0 | 0 | 1 | 1 | 1 | 1 | x | x | x | Выход – код «00» |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 0 + 0 = 00 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | Выход – код «01» |

Продолжение таблицы 3.1.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 1 + 0 = 01 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Выход – код «01» |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 2 + 0 = 02 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Выход – код «01» |
| 0 | 1 | 0 | 1 | 1 | 0 | x | x | x | 1 3 + 0 = 03 |
| 0 | 1 | 0 | 1 | 1 | 1 | x | x | x | Выход – код «01» |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 3 0 + 0 = 00 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Выход – код «03» |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 3 1 + 0 = 03 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | Выход – код «03» |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 3 2 + 0 = 12 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | Выход – код «03» |
| 0 | 1 | 1 | 1 | 1 | 0 | x | x | x | 3 3 + 0 = 21 |
| 0 | 1 | 1 | 1 | 1 | 1 | x | x | x | Выход – код «03» |
| 1 | 0 | 0 | 0 | 0 | 0 | х | х | х | 2 0 + 1 = 02 |
| 1 | 0 | 0 | 0 | 0 | 1 | х | х | х | Выход – код «02» |
| 1 | 0 | 0 | 0 | 1 | 0 | х | х | х | 2 1 + 1 = 03 |
| 1 | 0 | 0 | 0 | 1 | 1 | х | х | х | Выход – код «02» |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 2 2 + 1 = 11 |
| 1 | 0 | 0 | 1 | 0 | 1 | х | х | х | Выход – код «02» |
| 1 | 0 | 0 | 1 | 1 | 0 | x | x | x | 2 3 + 1 = 13 |
| 1 | 0 | 0 | 1 | 1 | 1 | x | x | x | Выход – код «02» |
| 1 | 0 | 1 | 0 | 0 | 0 | х | х | х | 0 0 + 1 = 01 |
| 1 | 0 | 1 | 0 | 0 | 1 | х | х | х | Выход – код «00» |
| 1 | 0 | 1 | 0 | 1 | 0 | х | х | х | 0 1 + 1 = 01 |
| 1 | 0 | 1 | 0 | 1 | 1 | х | х | х | Выход – код «00» |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 2 + 1 = 01 |
| 1 | 0 | 1 | 1 | 0 | 1 | х | х | х | Выход – код «00» |
| 1 | 0 | 1 | 1 | 1 | 0 | x | x | x | 0 3 + 1 = 01 |
| 1 | 0 | 1 | 1 | 1 | 1 | x | x | x | Выход – код «00» |
| 1 | 1 | 0 | 0 | 0 | 0 | х | х | х | 1 0 + 1 = 01 |
| 1 | 1 | 0 | 0 | 0 | 1 | х | х | х | Выход – код «01» |
| 1 | 1 | 0 | 0 | 1 | 0 | х | х | х | 1 1 + 1 = 02 |
| 1 | 1 | 0 | 0 | 1 | 1 | х | х | х | Выход – код «01» |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 2 + 1 = 03 |
| 1 | 1 | 0 | 1 | 0 | 1 | х | х | х | Выход – код «01» |
| 1 | 1 | 0 | 1 | 1 | 0 | x | x | x | 1 3 + 1 = 10 |
| 1 | 1 | 0 | 1 | 1 | 1 | x | x | x | Выход – код «01» |
| 1 | 1 | 1 | 0 | 0 | 0 | х | х | х | 3 0 + 1 = 01 |
| 1 | 1 | 1 | 0 | 0 | 1 | х | х | х | Выход – код «03» |
| 1 | 1 | 1 | 0 | 1 | 0 | х | х | х | 3 1 + 1 = 10 |

Продолжение таблицы 3.1.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 1 | 1 | х | х | х | Выход – код «03» |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 3 2 + 1 = 13 |
| 1 | 1 | 1 | 1 | 0 | 1 | х | х | х | Выход – код «03» |
| 1 | 1 | 1 | 1 | 1 | 0 | x | x | x | 3 3 + 1 = 22 |
| 1 | 1 | 1 | 1 | 1 | 1 | x | x | x | Выход – код «03» |

**Минимизация функции P:**

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | X1 | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| p |  | x | x | x | x | x | x | x | x |  |  |  |
|  | x | x | x | x | x | x | x | x |  |  | h |
|  | x | x | x | x | x | x | x | x |  |  |  |
|  | 0 | x | x | 1 | 0 | x | x | 1 |  | Y1 |  |
|  |  | 0 | x | x | 1 | 0 | x | x | 1 |  |  |  |
|  |  | 0 | x | x | 0 | 0 | x | x | 0 |  |  | h |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | X2 |  |  |  |  |  |  |  |
|  |  |  | Y2 | |  |  | Y2 | |  |  |  |  |

Рисунок 3.1.1 — Минимизация функции Р картой Вейча

Следовательно:

P = x1y1 + x2y1

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

P = y1 ∙ (h1) ∙ (x1x2)

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

*K = =* 2,62

**Минимизация функции Q1:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | X1 | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | x | x | x | x | x | x | x | x |  |  |  |
| p |  | x | x | x | x | x | x | x | x |  |  | h |
|  |  | x | x | x | x | x | x | x | x |  | Y1 |
|  |  | 1 | x | x | 1 | 1 | x | x | 1 |  |  |
|  |  | 0 | x | x | 0 | 0 | x | x | 0 |  |  |
|  |  | 1 | x | x | 1 | 0 | x | x | 0 |  | h |
|  |  | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |  |  |
|  |  | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | X2 |  |  |  |  |  |  |  |
|  |  |  | Y2 | |  |  | Y2 | |  |  |  |  |

Рисунок 3.1.2 — Минимизация функции Q1 картой Вейча

Следовательно:

Q1= + x1y2 + x1h

Запишем результат в базисе И-Константная единица-Сумма по модулю: Q1 = (((p1) ∙ (x1y2 1) ∙ (x1h 1)) 1)

Эффективность минимизации:

K = = 11,25

**Минимизация функции Q2:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | X1 | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | x | x | x | x | x | x | x | x |  |  |  |
| p |  | x | x | x | x | x | x | x | x |  |  | h |
|  |  | x | x | x | x | x | x | x | x |  | Y1 |
|  |  | 1 | x | x | 1 | 0 | x | x | 0 |  |  |
|  |  | 0 | x | x | 0 | 1 | x | x | 1 |  |  |
|  |  | 0 | x | x | 1 | 1 | x | x | 0 |  | h |
|  |  | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |  |  |
|  |  | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | X2 |  |  |  |  |  |  |  |
|  |  |  | Y2 | |  |  | Y2 | |  |  |  |  |

Рисунок 3.1.3 — Минимизация функции Q2 картой Вейча

Следовательно:

Q2 = + x2h + x2y1 + + y1

Запишем результат в базисе И-Константная единица-Сумма по модулю: Q2 = (((((y1  1) ∙ (y2 1) ∙ (h 1)) 1) ∙ (x2h 1) ∙ (x2y1 1) ∙ ∙ ( 1) ∙ ((( 1) ∙ ( 1) ∙ y1 ∙ (h 1)) 1)) 1)

Эффективность минимизации:

K = = 5,13

Функциональная схема ОЧУС приведена в приложении В.

**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 – 01, 1 – 10, 2 – 00, 3 – 11;

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 | 1 | 0 | 1 | 2 + 2 + 0 = 10 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2 + 2 + 1 = 11 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 + 0 + 0 = 02 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 2 + 0 + 1 = 03 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 + 1 + 0 = 03 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 2 + 1 + 1 = 10 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 2 + 3 + 0 = 11 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2 + 3 + 1 = 12 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 + 2 + 0 = 02 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 + 2 + 1 = 03 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 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 | 0 | 0 | 0 + 1 + 1 = 02 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 + 3 + 0 = 03 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 + 3 + 1 = 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 + 2 + 0 = 03 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 + 2 + 1 = 10 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 + 0 + 0 = 01 |

Продолжение таблицы 3.2.1

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

**Минимизация функции П:**

Минимизацию функцииП проведем с помощью карт Карно. Для функции П заполненная карта приведена на рисунке 3.2.2.

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

Рисунок 3.2.2 — Минимизация функции П картой Карно

Следовательно:

П = 1212 + a1a22 + a1a21p + a112p+ 2b1b2 + a1b1b2 + b1b2p + 12b1p

Запишем результат в базисе И-НЕ:

П=

Эффективность минимизации:

K = = 2,46

**Минимизация функции S**1

Минимизацию функции **S1** проведем с помощью карт Карно. Для функции **S1** заполненная карта приведена на рисунке 3.2.3.

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

Рисунок 3.2.3 — Минимизация функции S1 картой Карно

Следовательно: S1= 11p+ a1b1p + a11 + 1b1

Запишем результат в базисе И-НЕ:

S1 =

Эффективность минимизации:

K = = 5,32

**Минимизация функции S2**

Определим множество единичных кубов:

Множество безразличных кубов пустое.

Сформируем множество С0 = L ⋃ N:

C0 = {00000, 00011, 00100, 00101, 00110, 01001, 01010, 01110, 01111, 10000, 10001, 10101, 10110, 11010, 11011, 11100, 11111}

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)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С0\*С0 | 00000 | 00011 | 00100 | 00101 | 00110 | 01001 | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | 10110 | 11010 | 11011 |
| 00000 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00011 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00100 | 00x00 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 00101 |  |  | 0010x | - |  |  |  |  |  |  |  |  |  |  |  |
| 00110 |  |  | 001x0 |  | - |  |  |  |  |  |  |  |  |  |  |
| 01001 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |
| 01010 |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 01110 |  |  |  |  | 0x110 |  | 01x10 | - |  |  |  |  |  |  |  |
| 01111 |  |  |  |  |  |  |  | 0111x | - |  |  |  |  |  |  |
| 10000 | x0000 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 10001 |  |  |  |  |  |  |  |  |  | 1000x | - |  |  |  |  |
| 10101 |  |  |  | x0101 |  |  |  |  |  |  | 10x01 | - |  |  |  |
| 10110 |  |  |  |  | x0110 |  |  |  |  |  |  |  | - |  |  |
| 11010 |  |  |  |  |  |  | x1010 |  |  |  |  |  |  | - |  |
| 11011 |  |  |  |  |  |  |  |  |  |  |  |  |  | 1101x | - |
| 11100 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11111 |  |  |  |  |  |  |  |  | x1111 |  |  |  |  |  | 11x11 |

Продолжение таблицы 3.2.4

|  |  |  |
| --- | --- | --- |
| С0\*С0 | 11100 | 11111 |
| 00000 |  |  |
| 00011 |  |  |
| 00100 |  |  |
| 00101 |  |  |
| 00110 |  |  |
| 01001 |  |  |
| 01010 |  |  |
| 01110 |  |  |
| 01111 |  |  |
| 10000 |  |  |
| 10001 |  |  |
| 10101 |  |  |
| 10110 |  |  |
| 11010 |  |  |
| 11011 |  |  |
| 11100 | - |  |
| 11111 |  | - |

В результате этой операции сформируется новое множество кубов:

С1 = {00x00, x0000, 0010x, 001x0, x0101, 0x110, x0110, 01x10, x1010, 0111x, x1111, 1000x, 10x01, 1101x, 11x11}

Z0 = {00011, 01001, 11100}

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

Таблица 3.2.5 – Поиск простых импликант С1\* С1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C1\*C1 | 0001x | x0000 | 0010x | 001x0 | x0101 | 0x110 | x0110 | 01x10 | x1010 | 0111x | x1111 | 1000x | 10x01 | 1101x | 11x11 |
| 0001x | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0000 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0010x |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 001x0 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| x0101 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 0x110 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |
| x0110 |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 01x10 |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| x1010 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 0111x |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| x1111 |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 1000x |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |
| 10x01 |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |
| 1101x |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |
| 11x11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |

Новых кубов (второй размерности) не образовалось.

##### Конечное множество простых импликант Z = {00011; 01001; 11100; x0110; 1101x; 11x11; 0010x; 001x0; x0000; 10x01; 01x10; 0111x; 00x00; x0101; 0x110; x1010; x1111; 1000x}

На этом заканчивается этап поиска простых импликант.

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

Таблица 3.2.6 – Поиск L-экстремалей

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 00011 | 01001 | 11100 | x0110 | 1101x | 11x11 | 0010x | 001x0 | x0000 | 10x01 | 01x10 | 0111x | 00x00 | x0101 |
| 00011 | - | zyzyz  01001 | zyzyz  11100 | 1zyzy  x0110 | yyzz0  1101x | yy1zz  11x11 | zzyy0  0010x | zzy0y  001x0 | 1zzyy  x0000 | yz1yz  10x01 | zy1zy  01x10 | zyyz0  0111x | zz1yy  00x00 | 1zyyz  x0101 |
| 01001 | zyzyz  00011 | - | yzyzy  11100 | 1yyyy  x0110 | yzzy0  1101x | yz1yz  11x11 | zyyz0  0010x | zyy1y  001x0 | 1yzzy  x0000 | yy1zz  10x01 | zz1yy  01x10 | zzyy0  0111x | zy1zy  00x00 | 1yyzz  x0101 |
| 11100 | yyyyy  00011 | yzyzy  01001 | - | 0yzyz  x0110 | zzyy1  1101x | zz0yy  11x11 | yyzz1  0010x | yyz1z  001x0 | 0yyzz  x0000 | zy0zy  10x01 | yz0yz  01x10 | yzzy1  0111x | yy0zz  00x00 | 0yzzy  x0101 |
| x0110 | zzyzy  00011 | zyyyy  01001 | zyzyz  11100 | - | zyyz1  1101x | zy0zy  11x11 | zzzy1  0010x | zzz0z  00100 | zzyyz  x0000 | zz0yy  10x01 | zy0zz  01x10 | zyzz1  0111x | zz0yz  00x00 | zzzyy  x0101 |
| 1101x | yyzzz  00011 | yzzyz  01001 | zzyyz  11100 | 0yyzz  x0110 | - | zz1zz  11111 | yyyyz  0010x | yyyyz  00100 | 0yzyz  x0000 | zy1yz  10x01 | yz1zz  01x10 | yzyzz  0111x | yy1yz  00x00 | 0yyyz  x0101 |
| 11x11 | yyzzz  00011 | yzzyz  01001 | zzzyy  11100 | 0yzzy  x0110 | zzzz0  11010 | - | yyzy0  0010x | yyzyy  00100 | 0yzyy  x0000 | zyzyz  10x01 | yzzzy  01x10 | yzzz0  0111x | yyzyy  00x00 | 0yzyz  x0101 |
| 0010x | zzyyz  00011 | zyyzz  01001 | yyzzz  11100 | 1zzyz  x0110 | yyyyz  11010 | yyzyz  11111 | - | zzzzz  Ø | 1zyzz  x0000 | yz0zz  10x01 | zy0yz  01x10 | zyzyz  0111x | zz0zz  00000 | 1zzzz  10101 |
| 001x0 | zzyzy  00011 | zyyzy  01001 | yyzzz  11100 | 1zzzz  10110 | yyyzz  11010 | yyzzy  11111 | zzzz1  00101 | - | 1zyzz  x0000 | yz0zy  10x01 | zy0zz  01x10 | zyzz1  0111x | zzyzz  00000 | yzzzy  10101 |
| x0000 | zzzyy  00011 | zyzzy  01001 | zyyzz  11100 | zzyyz  10110 | zyzyz  11010 | zyyyy  11111 | zzyzy  00101 |  | - | zz1zy  10x01 | zy1yz  01x10 | zyyy1  0111x | zzzzz  Ø | zzyzy  10101 |
| 10x01 | yzzyz  00011 | yyzzz  01001 | zyzzy  11100 | zzzyy  10110 | zyzyy  11010 | zyzyz  11111 | yzzzz  00101 |  | 0zzzy  x0000 | - | yyzyy  01x10 | yyzy0  0111x |  | zzzzz  Ø |
| 01x10 | zyzzy  00011 | zzzyy  01001 | yzzyz  11100 | yyzzz  10110 | yzzzz  11010 | yzzzy  11111 | zyzyy  00101 |  | 1yzyz  x0000 | yyzyy  10x01 | - | zzzz1  01111 |  |  |
| 0111x | zyyzz  00011 | zzyyz  01001 | yzzyz  11100 | yyzzz  10110 | yzyzz  11010 | yzzzz  11111 | zyzyz  00101 |  | 1yyyz  x0000 | yy0yz  10x01 | zz0zz  01010 | - |  |  |

Продолжение таблицы 3.2.6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00x00 | zzzyy  00011 | zyzzy  01001 | yyzzz  11100 | yzzyz  10110 | yyzyz  11010 | yyzyy  11111 | zzzzy  00101 |  | 1zzzz  10000 | yzzzy  10x01 | zyzyz  01010 | zyzyy  01111 | - |  |
| x0101 | zzyyz  00011 | zyyzz  01001 | zyzzy  11100 | zzzyy  10110 | zyyyy  11010 | zyzyz  11111 | zzzzz  Ø |  | zzyzy  10000 | zz0zz  10001 | zyyyy  01010 | zyzyz  01111 |  | - |
| 0x110 | zzyzy  00011 | zzyyy  01001 | yzzyz  11100 | yzzzz  10110 | yzyzz  11010 | yzzzy  11111 |  |  | yzyyz  10000 | yzyyy  10001 | zzyzz  01010 | zzzzy  01111 |  |  |
| x1010 | zyzzy  00011 | zzzyy  01001 | zzyyz  11100 | zyyzz  10110 | zzzzz  Ø | zzyzy  11111 |  |  | zyzyz  10000 | zyzyy  10001 | zzzzz  Ø | zzyzy  01111 |  |  |
| x1111 | zyyzz  00011 | zzyyz  01001 | zzzyy  11100 | zyzzy  10110 |  | zzzzz  Ø |  |  | zyyyy  10000 | zyyyz  10001 |  | zzzzz  Ø |  |  |
| 1000x | yzzyz  00011 | yyzzz  01001 | zyyzz  11100 | zzyyz  10110 |  |  |  |  | zzzzz  Ø | zzzzz  Ø |  |  |  |  |
| Остаток | 00011 | 01001 | 11100 | 10110 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Продолжение таблицы 3.2.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| z#(Z-z) | 0x110 | x1010 | x1111 | 1000x |
| 00011 | z1yzy  0x110 | 1yzzy  x1010 | 1yyzz  x1111 | yzzy0  1000x |
| 01001 | z0yyy  0x110 | 1zzyy  x1010 | 1zyyz  x1111 | yyzz0  1000x |
| 11100 | y0zyz  0x110 | 0zyyz  x1010 | 0zzyy  x1111 | zyyz1  1000x |
| x0110 | z1zzz  01110 | zyyzz  x1010 | zyzzy  x1111 | zzyy1  1000x |
| 1101x | yzyzz  01110 | 0zzzz  01010 | 0zyzz  x1111 | zyzyz  1000x |
| 11x11 | yzzzy  01110 | yzzzy  01010 | 0zzzz  01111 | zyzy0  1000x |
| 0010x | zyzyz  01110 | zyyyz  01010 | zyzyz  01111 | yzyzz  1000x |
| 001x0 | zyzzz  01110 | zyyzz  01010 | zyzzy  01111 | yzyz1  1000x |
| x0000 | zyyyz  01110 | zyzyz  01010 | zyyyy  01111 | zzzz1  10001 |
| 10x01 | yyzyy  01110 | yyzyy  01010 | yyzyz  01111 | zzzzz  Ø |
| 01x10 | zzzzz  Ø | zzzzz  Ø | zzzzy  01111 |  |
| 0111x |  |  | zzzzz  Ø |  |

Продолжение таблицы 3.2.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00x00 |  |  |  |  |
| x0101 |  |  |  |  |
| 0x110 | - |  |  |  |
| x1010 |  | - |  |  |
| x1111 |  |  | - |  |
| 1000x |  |  |  | - |
| Остаток | Ø | Ø | Ø | Ø |

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

Множество L-экстремалей = E = {00011, 01001, 11100, x0110}

Следующий этап – поиск непокрытых кубов из множества L (таблица 3.2.7). Для этого используется операция # (вычитание).

Таблица 3.2.7 – Поиск непокрытых кубов из множества L

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L *#* E | 00000 | 00011 | 00100 | 00101 | 00110 | 01001 | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | 10110 | 11010 |
| 00011 | 00000 | Ø | 00100 | 00101 | 00110 | 01001 | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | 10110 | 11010 |
| 01001 | 00000 | Ø | 00100 | 00101 | 00110 | Ø | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | 10110 | 11010 |
| 11100 | 00000 | Ø | 00100 | 00101 | 00110 | Ø | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | 10110 | 11010 |
| x0110 | 00000 | Ø | 00100 | 00101 | Ø | Ø | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | Ø | 11010 |
| Остаток | 00000 | Ø | 00100 | 00101 | Ø | Ø | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | Ø | 11010 |

Продолжение таблицы 3.2.7

|  |  |  |  |
| --- | --- | --- | --- |
| L *#* E | 11011 | 11100 | 11111 |
| 00011 | 11011 | 11100 | 11111 |
| 01001 | 11011 | 11100 | 11111 |
| 11100 | 11011 | Ø | 11111 |
| x0110 | 11011 | Ø | 11111 |
| Остаток | 11011 | Ø | 11111 |

##### 

##### Из таблицы 3.2.7 следует, что множество кубов, непокрытых L-экстремалями, L´ = L # E = {00000; 00100; 00101; 01010; 01110; 01111; 10000; 10001; 10101; 11010; 11011; 11111}

Следующий этап – поиск минимального числа кубов из множества Z’ = Z \ E, которыми будут покрыты оставшиеся кубы из множества L’ (таблица 3.2.8). Для этого используется операция ∩ (пересечение).

##### Z´ = Z - E = {1101x; 11x11; 0010x; 001x0; x0000; 10x01; 01x10; 0111x; 00x00; x0101; 0x110; x1010; x1111; 1000x}

Из оставшихся кубов множества Z’ необходимо найти кубы с минимальной ценой и одновременно с этим кубы, которые максимально покрывают кубы множества L’.

Таблица 3.2.8 – Поиск не перекрытых кубов из множества L

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L’ ∩ Z’ | 00000 | 00100 | 00101 | 01010 | 01110 | 01111 | 10000 | 10001 | 10101 | 11010 | 11011 | 11111 |
| 1101x | yy0y0 Ø | yyyy0 Ø | yyyy1 Ø | y1010 Ø | y1y10 Ø | y1y11 Ø | 1y0y0 Ø | 1y0y1 Ø | 1yyy1 Ø | 11010 | 11011 | 11y11 Ø |
| 11x11 | yy0yy Ø | yy1yy Ø | yy1y1 Ø | y101y Ø | y111y Ø | y1111 Ø | 1y0yy Ø | 1y0y1 Ø | 1y1y1 Ø | 1101y Ø | 11011 | 11111 |
| 0010x | 00y00 Ø | 00100 | 00101 | 0yyy0 Ø | 0y1y0 Ø | 0y1y1 Ø | y0y00 Ø | y0y01 Ø | y0101 Ø | yyyy0 Ø | yyyy1 Ø | yy1y1 Ø |
| 001x0 | 00y00 Ø | 00100 | 0010y Ø | 0yy10 Ø | 0y110 Ø | 0y11y Ø | y0y00 Ø | y0y0y Ø | y010y Ø | yyy10 Ø | yyy1y Ø | yy11y Ø |
| x0000 | 00000 | 00y00 Ø | 00y0y Ø | 0y0y0 Ø | 0yyy0 Ø | 0yyyy Ø | 10000 | 1000y Ø | 10y0y Ø | 1y0y0 Ø | 1y0yy Ø | 1yyyy Ø |
| 10x01 | y000y Ø | y010y Ø | y0101 Ø | yy0yy Ø | yy1yy Ø | yy1y1 Ø | 1000y Ø | 10001 | 10101 | 1y0yy Ø | 1y0y1 Ø | 1y1y1 Ø |
| 01x10 | 0y0y0 Ø | 0y1y0 Ø | 0y1yy Ø | 01010 | 01110 | 0111y Ø | yy0y0 Ø | yy0yy Ø | yy1yy Ø | y1010 Ø | y101y Ø | y111y Ø |
| 0111x | 0yyy0 Ø | 0y1y0 Ø | 0y1y1 Ø | 01y10 Ø | 01110 | 01111 | yyyy0 Ø | yyyy1 Ø | yy1y1 Ø | y1y10 Ø | y1y11 Ø | y1111 Ø |
| 00x00 | 00000 | 00100 | 0010y Ø | 0y0y0 Ø | 0y1y0 Ø | 0y1yy Ø | y0000 Ø | y000y Ø | y010y Ø | yy0y0 Ø | yy0yy Ø | yy1yy Ø |
| x0101 | 00y0y Ø | 0010y Ø | 00101 | 0yyyy Ø | 0y1yy Ø | 0y1y1 Ø | 10y0y Ø | 10y01 Ø | 10101 | 1yyyy Ø | 1yyy1 Ø | 1y1y1 Ø |
| 0x110 | 00yy0 Ø | 001y0 Ø | 001yy Ø | 01y10 Ø | 01110 | 0111y Ø | y0yy0 Ø | y0yyy Ø | y01yy Ø | y1y10 Ø | y1y1y Ø | y111y Ø |

Продолжение таблицы 3.2.8

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1010 | 0y0y0 Ø | 0yyy0 Ø | 0yyyy Ø | 01010 | 01y10 Ø | 01y1y Ø | 1y0y0 Ø | 1y0yy Ø | 1yyyy Ø | 11010 | 1101y Ø | 11y1y Ø |
| x1111 | 0yyyy Ø | 0y1yy Ø | 0y1y1 Ø | 01y1y Ø | 0111y Ø | 01111 | 1yyyy Ø | 1yyy1 Ø | 1y1y1 Ø | 11y1y Ø | 11y11 Ø | 11111 |
| 1000x | y0000 Ø | y0y00 Ø | y0y01 Ø | yy0y0 Ø | yyyy0 Ø | yyyy1 Ø | 10000 | 10001 | 10y01 Ø | 1y0y0 Ø | 1y0y1 Ø | 1yyy1 Ø |

##### Могут быть получены следующие тупиковые формы:

##### Сmin1 = {11x11; 0010x; x0000; 10x01; 0111x; x1010; 00011; 01001; 11100; x0110}

##### Сmin2 = {1101x; 0010x; x0000; 10x01; 01x10; x1111; 00011; 01001; 11100; x0110}

##### Сmin3 = {11x11; 0111x; 00x00; x0101; x1010; 1000x; 00011; 01001; 11100; x0110}

##### Сmin4 = {1101x; 01x10; 00x00; x0101; x1111; 1000x; 00011; 01001; 11100; x0110}

Поиск минимального покрытия завершён.

**S2 =** a1a2b2p + 12b12 + 212 + a122p + 1a2b1b2 + a21b2 + 121b2p + 1a212p + a1a2b12 + 2b1b2

Запишем результат в базисе И-НЕ:

**S2** =

Эффективность минимизации:

K = = 1,84

Функциональная схема ОЧС приведена в приложении Б.

**3.3. Логический синтез преобразователя множителя**

Преобразователь множителя (ПМ) – это устройство, которое служит для исключения из множителя диад 11, заменяя их на триады 10. Если из Вых.2 поступает 1 в ФДК, то значит число – отрицательное.

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

Таблица 3.3.1 – Таблица истинности ПМ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Входная диада** | | **Пер. из триггера** | **Знак** | **Сигнал в ФДК** | **Выходная диада** | |
| ***Q1*** | ***Q2*** | ***T*** | ***D*** | ***F*** | ***Y1*** | ***Y2*** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Минимизацию переключательных функций произведём с помощью карт Вейча и реализуем их в базисе И, ИЛИ, НЕ, Исключающее ИЛИ.

**Функция D**

Для функции D заполненная карта приведена на рисунке 3.3.2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Q2 | |  |  |
|  |  |  |  |  |  |
| Q1 |  | 1 | 1 | 1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | T | |  |

Рисунок 3.3.2 – Минимизация функции Q при помощи карты Вейча

Следовательно:

**Функция F**

Для функции F заполненная карта приведена на рисунке 3.3.3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Q2 | |  |  |
|  |  |  |  |  |  |
| Q1 |  | 1 |  | 1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | T | |  |

Рисунок 3.3.3 – Минимизация функции F при помощи карты Вейча

Следовательно:

**Функция Y1**

Для функции Y1 заполненная карта приведена на рисунке 3.3.4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Q2 | |  |  |
|  |  |  |  |  |  |
| Q1 |  |  |  |  | 1 |
|  |  |  | 1 |  |  |
|  |  |  |  |  |  |
|  |  |  | T | |  |

Рисунок 3.3.4 – Минимизация функции Y1 при помощи карты Вейча

Следовательно:

**Функция Y2**

Для функции Y2 заполненная карта приведена на рисунке 3.3.5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Q2 | |  |  |
|  |  |  |  |  |  |
| Q*1* |  | 1 |  | 1 |  |
|  |  | 1 |  | 1 |  |
|  |  |  |  |  |  |
|  |  |  | T | |  |

Рисунок 3.3.5 – Минимизация функции Y2 при помощи карты Вейча

Следовательно:

Функциональная схема ПМ приведена в приложении Д.

# **4. Синтез комбинационных схем устройств**

# **на основе мультиплексоров**

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

На выход мультиплексора может быть пропущен без изменений один любой логический сигнал, поступающий на один из информационных входов. Порядковый номер информационного входа, значение которого в данный момент должно быть передано на выход, определяется двоичным кодом, подаваемым на управляющие входы.

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

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

Таблица истинности для синтеза ПФ ОЧС приведена в таблице 4.1.

Таблица 4.1 – Таблица истинности для синтеза ПФ ОЧС

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

Продолжение таблицы 4.1

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

Функциональная схема ОЧС на основе мультиплексоров представлена в приложении Г.

**5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ**

Формула расчёта временных затрат на умножение:

Т=𝑛\*(𝑇ПМ + 𝑇ФДК + m\*𝑇ОЧУC + 𝑇ОЧС+𝑇сдвига), где

𝑇ПМ – время преобразования множителя;

𝑇ФДК – время формирования дополнительного кода множимого;

𝑇ОЧУC – время умножения на ОЧУC;

𝑇ОЧС – время формирования единицы переноса в ОЧС;

𝑇сдвига – время сдвига в регистрах;

n – количество разрядов множителя;

m – количество разрядов множимого.

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.

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

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.

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

**ПРИЛОЖЕНИЕ А**

**(**обязательное**)**

Сумматор-умножитель второго типа. Схема электрическая структурная

**ПРИЛОЖЕНИЕ Б**

**(**обязательное**)**

Одноразрядный четверичный сумматор. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ В**

**(**обязательное**)**

Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Г**

**(**обязательное**)**

Однозарядный четверичный сумматор. Схема электрическая функциональная на основе мультиплексоров

**ПРИЛОЖЕНИЕ Д**

**(**обязательное**)**

Преобразователь множителя. Схема электрическая функциональная

**ПРИЛОЖЕНИЕ Е**

**(**обязательное**)**

Ведомость документов