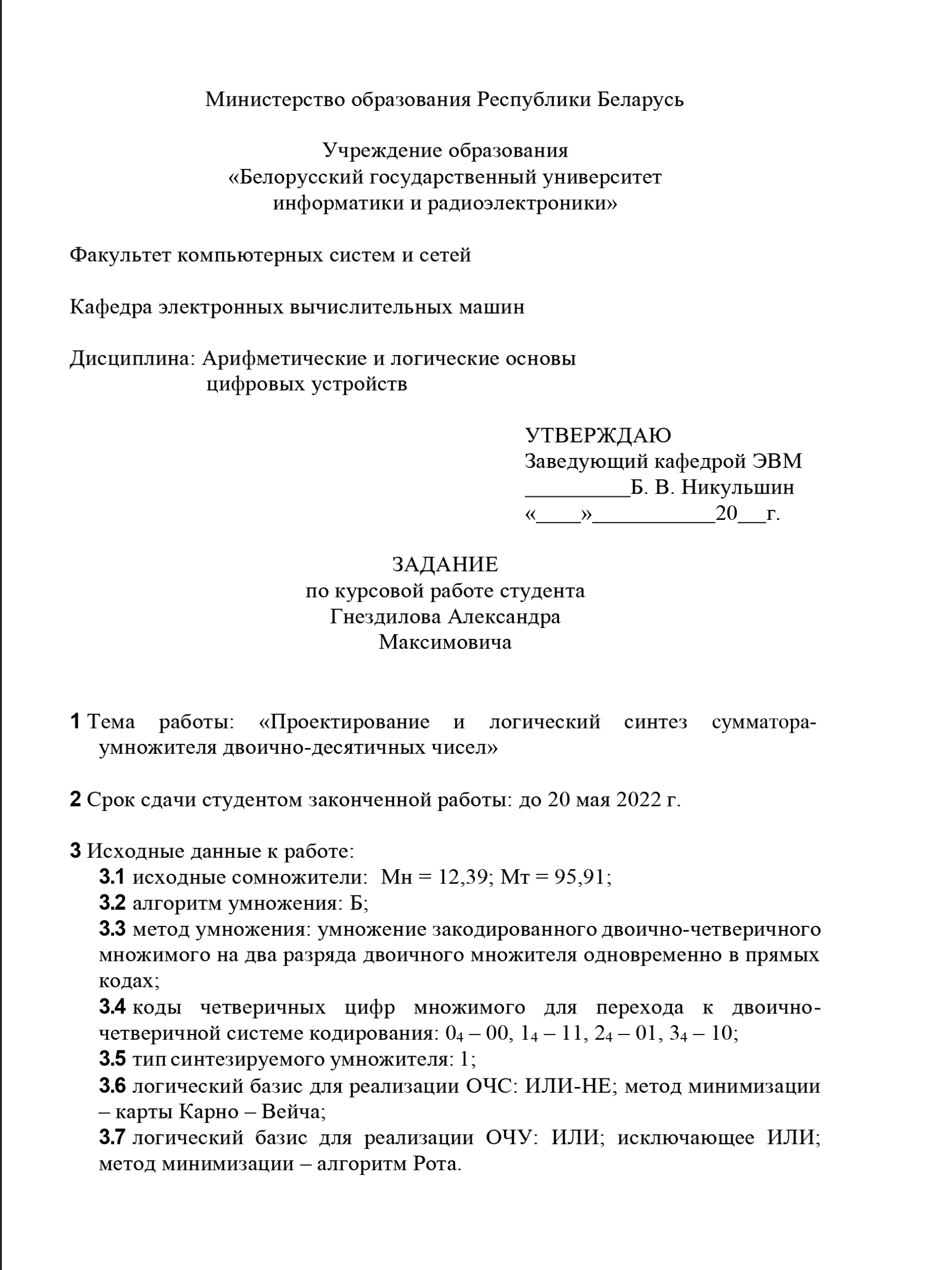
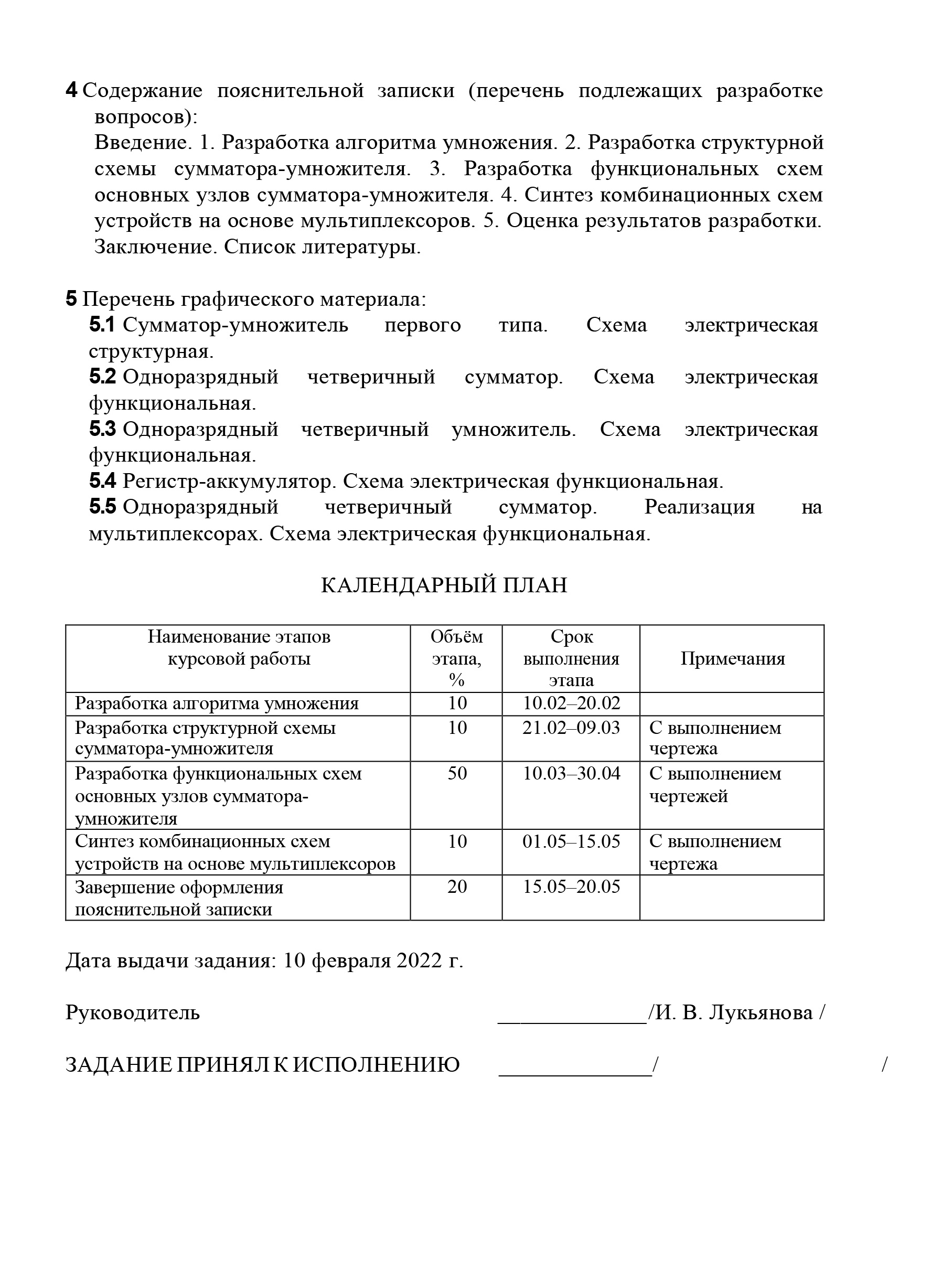


****

****

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc14589)

[1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ 6](#_Toc13259)

[2. СИНТЕЗ СТРУКТУРЫ СУММАТОРА-УМНОЖЕНИТЕЛЯ ПЕРВОГО ТИПА 9](#_Toc31350)

[3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ](#_Toc25127) 12

[3.1. Логический синтез ОЧС](#_Toc25127) 12

[3.2. Логический синтез ОЧУ](#_Toc25127) 20

[4. ЭФФЕКТИВНОСТЬ МИНИМИЗАЦИИ 2](#_Toc11814)4

[5. СИНТЕЗ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРА 25](#_Toc8283)

[6. ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ 27](#_Toc419)

[7. ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ](#_Toc30179) 28

[ЗАКЛЮЧЕНИЕ 2](#_Toc11148)9

[ЛИТЕРАТУРА 3](#_Toc832)0

[ПРИЛОЖЕНИЕ А](#_Toc16970) 31

[ПРИЛОЖЕНИЕ Б](#_Toc25132) 32

[ПРИЛОЖЕНИЕ В](#_Toc7475) 33

[ПРИЛОЖЕНИЕ Г](#_Toc18181) 34

[ПРИЛОЖЕНИЕ Д](#_Toc30732) 35

[ПРИЛОЖЕНИЕ Е 3](#_Toc30732)6

**ВВЕДЕНИЕ**

Целью данной работы является разработка сумматора-умножителя первого типа для алгоритма умножения «Б» в дополнительном коде на два разряда одновременно. Чтобы это осуществить нам нужно решить ряд задач:

Разработать алгоритм умножения и оценить погрешности вычислений.

Разработать структурную схему сумматора-умножителя первого типа.

Разработать функциональные схемы основных узлов сумматора-умножителя в заданных логических базисах.

Разработать комбинационную схему на основе мультиплексора.

Рассчитать временные затраты на умножение.

1. **РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ**

Перевод данных сомножителей из десятичной системы счисления в четверичную систему счисления:

Множимое:

|  |  |  |
| --- | --- | --- |
| \* | 0,91  4 | |
| \* | 1,56 | |
| 4 | |
| \* | 2,24  4 | |
| \* | 0,96  4 |
|  | 3,84 | |

12 | 4

12 3 | 4

0 0 0

3

=30,1203

=1000,11010010

Множитель:

|  |  |
| --- | --- |
| \* | 0,91  4 |
| \* | 3,64 |
| 4 |
|  | 2,56 |
|  |  |

95 | 4

92 23 | 4

3 20 5 | 4

3 4 1

1

=1133,32

=01011111,1110

Cомножители в форме с плавающей запятой в прямом коде:

Мн = 0, 100011010010 = 0.0001 (закодирован согласно заданию)

Мт = 0, 010111111110 = 0.0100 (закодирован традиционно)

Cложении порядков:

|  |  |
| --- | --- |
| = 0.0001 | +02 |
| = 0.0100 | +10 |
| P = 0.1101 | +12 |

Результат операции закодирован согласно задания на кодировку множимого.

Знак произведения определяем суммой по модулю двух знаков сомножителей:

зн Мн зн Мт = 0 + 0 = 0

Для умножения мантисс предварительно преобразуем множитель, чтобы исключить диаду 11 (34), заменив ее на триаду 101.

Преобразованный множитель имеет вид: =

Мн=301203

=113332

=12002

=

= 3.032131

= 0,1203012

Умножение мантисс по алгоритму «Б» приведено в таблице 1.1.

Таблица 1.1 - Умножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Четверичная с/с** | | **Двоично-четверичная с/с** | | **Комментарии** |
| **1** | | **2** | | **3** |
| 0.  0.  0.  3.  3.  0.  3.  0.  3.  0.  0.  0.  0. | 000000000000  000001203012  000001203012  333330321310  333332130322  000000000000  333332130322  000000000000  333332130322  012030120000  012022310322  030120300000  102203210322 | 0.  0.  0.  1.  1.  0.  1.  0.  1.  0.  0.  0.  0. | 000000000000000000000000  000000000011010010001101  000000000011010010001101  101010101000100111101100  101010101001111000100101  000000000000000000000000  101010101001111000100101  000000000000000000000000  101010101001111000100101  001101001000110100000000  001101000101101100100101  001000110100100000000000  110001010010011100100101 | 0  = Мн 2 20    = Мн (-1)21    = Мн 022    = Мн 023    = Мн 224    = Мн 1 25 |

Окончив умножение, необходимо оценить погрешность вычислений. Для этого полученное произведение = 0,102203210322

(\* ) приводим к нулевому порядку, а затем переводим в десятичную систему счисления:

= 102203,210322 (\* 0)

= 1187.5766

= 1188,3249

Абсолютная погрешность:

Δ = 1188,3249- 1187,5766= 0,7483

*δ* =

*δ* =

**2.РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ**

Структурная схема сумматора-умножителя 1-го типа алгоритма умножения “Б” приведена в приложении А.

*Если устройство работает как сумматор* (на входе mul/sum – “1”), то оба слагаемых последовательно записываются в регистр множимого, а на управляющий вход формирователя дополнительного кода (ФДК) F2 поступает «1». Нужно помнить, что числа представлены в форме с плавающей запятой. Поэтому, перед тем как складывать мантиссы, необходимо выровнять порядки. В блоке порядков нужно обеспечить сравнение порядков, используя сумматор порядков, и в зависимости от знака результата сдвигать первое или второе слагаемое. Реализация сдвига мантиссы числа с меньшим порядком будет зависеть от алгоритма умножения. Этим будет определяться порядок подачи слагаемых на операцию и то, где будет сдвигаться мантисса (в регистре множимого или в регистре результата). На выходах ФДК формируется дополнительный код одного слагаемого с учетом знака. Это слагаемое может быть записано в регистр множимого, при этом управляющие сигналы, поступающие на входы «h» всех ОЧУ, дают возможность переписать на выходы ОЧУ разряды слагаемого без изменений. Если есть необходимость выравнивания порядков, то в регистре-аккумуляторе может выполняться сдвиг мантиссы первого слагаемого. Если на вход «h» поступает «0», то ОЧУ перемножает разряды Мн и Мт.

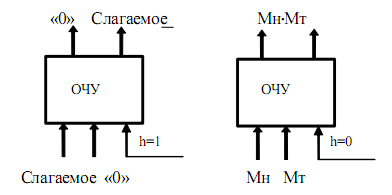


Рисунок 2.1 – Режимы работы ОЧУ

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

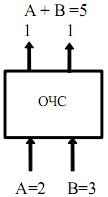


Рисунок 2.2 – Однозарядный четверичный сумматор

В ОЧС первое слагаемое складывается с нулем, так как на старших выходах ОЧУ будут формироваться только коды нуля. После первое

слагаемое попадает в регистр-аккумулятор, который изначально обнулен.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУ и ОЧС попадает в аккумулятор, где складывается с первым слагаемым. Таким образом, аккумулятор (накапливающий сумматор) складывает операнды и хранит результат. Разрядность аккумулятора должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения при суммировании переноса.

*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 этой старшей диады (выход 1 ПМ). В регистре множителя в конце каждого такта умножения содержимое сдвигается на 2 двоичных разряда, и в последнем такте умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения при умножении по алгоритму “Б “ ( 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 приведен 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 3,4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУ вместе с диадами множимого.

ОЧУ предназначен только 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 предназначены ОЧС. Значит, чтобы полностью сформировать частичное произведение четверичных сомножителей, нужна комбинация цепочек ОЧУ и ОЧС. Частичные суммы формируются в аккумуляторе. На первом этапе он обнулен, и первая частичная сумма получается за счет сложения первого частичного произведения (сформированного на выходах ОЧС) и нулевой частичной суммы (хранящейся в аккумуляторе). Потом в аккумуляторе происходит сложение i-й частичной суммы с (i+1)-м частичным произведением, результат сложения сохраняется. Содержимое аккумулятора сдвигается на один четверичный разряд влево в конце каждого такта умножения по алгоритму «Б».

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

3 · 2 = 1 2.

max maх

Мн Мт

От сюда следует, 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, так как алгоритм умножения «Б» предполагает возможность синхронного сдвига этих устройств. Количество тактов умножения определяется разрядностью Мт.

**3.РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ**

**3.1Логический синтез одноразрядного четверичного сумматора**

ОЧС - это комбинационное устройство (5 входов и 3 выхода):

˗ 2 разряда одного слагаемого (множимого);

- 2 разряда второго слагаемого (регистр результата);

- вход переноса из младшего ОЧС.

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

Принцип работы ОЧС представлен с помощью таблицы 3.1.

Таблица 3.1 - Таблица истинности ОЧС

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

*Продолжение таблицы 3.1*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 1 | x | x | x | 1+3+1=11 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1+1+0=02 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1+1+1=03 |

Минимизацию выходов ОЧС проведем с помощью карты Карно и Алгоритма Рота, минимизацию и *П* проведем с помощью карт Карно.

**Минимизация функции Пкартами Карно:**

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

Минимизировав функцию, получим:

**Минимизация функции картами Карно:**

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

Минимизировав функцию, получим:

**Минимизация функции картами Карно:**

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

Минимизировав функцию, получим:

**Минимизация функции алгоритмом Рота:**

Определим количество единичных кубов:

L= { 00001; 00110; 00111; 01000; 10111; 11000; 11001; 11110}

и множество безразличных кубов:

N **=** { 00010; 00011; 00100; 00101; 01010; 01011; 01100; 01101; 10010; 10011; 10100; 10101; 11010; 11011; 11100; 11101}

Таблица 3.2 – Минимизация N

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

N**=**{xx01x; xx10x}

={ 00001; 00110; 00111; 01000; 10111; 11000; 11001; 11110, xx01x; xx10x}

Первым этапом алгоритма Рота является нахождение множества простых импликант. Для реализации этого этапа будем использовать операцию умножения (\*) над множествами Со, С1 и т.д., пока в результате операции будут образовываться новые кубы большей размерности. Первый шаг умножения (C0\*Co) приведен в табл. 3.3. В результате этой операции сформируем новое множество кубов:

Таблица 3.3 – Поиск простых импликант (\* )

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C0\*C0 | 00001 | 00110 | 00111 | 01000 | 10111 | 11000 | 11001 | 11110 | xx01x | xx10x |
| 00001 | - |  |  |  |  |  |  |  |  |  |
| 00110 |  | - |  |  |  |  |  |  |  |  |
| 00111 |  | 0011y | - |  |  |  |  |  |  |  |
| 01000 |  |  |  | - |  |  |  |  |  |  |
| 10111 |  |  | y0111 |  | - |  |  |  |  |  |
| 11000 |  |  |  | y1000 |  | - |  |  |  |  |
| 11001 |  |  |  |  |  | 1100y | - |  |  |  |
| 11110 |  |  |  |  |  |  |  | - |  |  |
| xx01x | 000y1 | 00y10 | 00y11 | 010y0 | 10y11 | 110y0 | 110y1 | 11y10 | - |  |
| xx10x | 00y01 | 001y0 | 001y1 | 01y00 | 101y1 | 11y00 | 11y01 | 111y0 |  | - |
| A1 | 000x1 00x01 | 0011x 00x10 001x0 | x0111 00x11 001x1 | x1000 010x0 01x00 | 10x11 101x1 | 1100x 110x0 11x00 | 110x1 11x01 | 11x10 111x0 | Ø | Ø |

A1 = { 000x1; 00x01; 0011x; 00x10; 001x0; x0111; 00x11; 001x1; x1000; 010x0; 01x00; 10x11; 101x1; 1100x; 110x0; 11x00; 110x1; 11x01; 11x10; 111x0 }

Z0 = { Ø }

B1 = { 00001; 00110; 00111; 01000; 10111; 11000; 11001; 11110; xx01x; xx10x }

C1 = { 000x1; 00x01; 0011x; 00x10; 001x0; x0111; 00x11; 001x1; x1000; 010x0; 01x00; 10x11; 101x1; 1100x; 110x0; 11x00; 110x1; 11x01; 11x10; 111x0; xx01x; xx10x }

Таблица 3.4 – Поиск простых импликант (\* )

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C1\*C1 | 000x1 | 00x01 | 0011x | 00x10 | 001x0 | x0111 | 00x11 | 001x1 | x1000 | 010x0 | 01x00 | 10x11 | 101x1 | 1100x | 110x0 | 11x00 | 110x1 | 11x01 | 11x10 | 111x0 | xx01x | xx10x |
| 000x1 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x01 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0011x |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x10 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 001x0 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x0111 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00x11 |  | 00xy1 |  | 00x1y |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 001x1 | 00yx1 |  |  |  | 001xy |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| x1000 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 010x0 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 01x00 |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 10x11 |  |  |  |  |  |  | y0x11 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| 101x1 |  |  |  |  |  |  |  | y01x1 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |
| 1100x |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |
| 110x0 |  |  |  |  |  |  |  |  |  | y10x0 |  |  |  |  | - |  |  |  |  |  |  |  |
| 11x00 |  |  |  |  |  |  |  |  |  |  | y1x00 |  |  |  |  | - |  |  |  |  |  |  |
| 110x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 110xy |  | - |  |  |  |  |  |
| 11x01 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 11x0y |  | - |  |  |  |  |
| 11x10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 11xy0 |  |  | - |  |  |  |
| 111x0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 11yx0 |  |  |  |  | - |  |  |
| xx01x |  |  | 00y1x |  |  | x0y11 |  |  | x10y0 |  |  |  |  | 110yx |  |  |  |  |  |  | - |  |
| xx10x |  |  | 001yx |  |  | x01y1 |  |  | x1y00 |  |  |  |  | 11y0x |  |  |  |  |  |  |  | - |
| A2 | 00xx1 | 00xx1 | 00x1x 001xx | 00x1x | 001xx | x0x11 x01x1 | x0x11 | x01x1 | x10x0 x1x00 | x10x0 | x1x00 | Ø | Ø | 110xx 11x0x | 110xx 11xx0 | 11x0x 11xx0 | Ø | Ø | Ø | Ø | Ø | Ø |

##### A2 = { 00xx1; 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; 11xx0 }

##### Z1 = { Ø }

##### B2 = { 000x1; 00x01; 0011x; 00x10; 001x0; x0111; 00x11; 001x1; x1000; 010x0; 01x00; 10x11; 101x1; 1100x; 110x0; 11x00; 110x1; 11x01; 11x10; 111x0; xx01x; xx10x }

##### C2 = { 00xx1; 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; 11xx0; xx01x; xx10x }

Таблица 3.5 – Поиск простых импликант (\* )

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C2\*C2 | 00xx1 | 00x1x | 001xx | x0x11 | x01x1 | x10x0 | x1x00 | 110xx | 11x0x | 11xx0 | xx01x | xx10x |
| 00xx1 | - |  |  |  |  |  |  |  |  |  |  |  |
| 00x1x |  | - |  |  |  |  |  |  |  |  |  |  |
| 001xx |  |  | - |  |  |  |  |  |  |  |  |  |
| x0x11 |  |  |  | - |  |  |  |  |  |  |  |  |
| x01x1 |  |  |  |  | - |  |  |  |  |  |  |  |
| x10x0 |  |  |  |  |  | - |  |  |  |  |  |  |
| x1x00 |  |  |  |  |  |  | - |  |  |  |  |  |
| 110xx |  |  |  |  |  |  |  | - |  |  |  |  |
| 11x0x |  |  |  |  |  |  |  |  | - |  |  |  |
| 11xx0 |  |  |  |  |  |  |  |  |  | - |  |  |
| xx01x |  |  |  |  |  |  |  |  |  |  | - |  |
| xx10x |  |  |  |  |  |  |  |  |  |  |  | - |
| A3 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

A3 = { Ø }

Новых кубов (третьей размерности) не образовалось. На этом заканчивается этап поиска простых импликант, так как |C3|<=1. Конечное множество простых импликант Z = { 00xx1; 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; 11xx0; xx01x; xx10x }

Следующий этап - поиск L-экстремалей на множестве простых импликант. Для этого используется операция # (решетчатое вычитание). В табл. 3.6 из каждой простой импликанты поочередно вычитаются все остальные простые импликанты Z#(Z\z), результат операции (последняя строка таблицы) указывает на то, что L-экстремалями стали следующие простые импликанты:

##### E = { 00xx1; 11xx0 }.

Таблица 3.6 – Поиск L-экстремалей

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 00xx1 | 00x1x | 001xx | x0x11 | x01x1 | x10x0 | x1x00 | 110xx | 11x0x | 11xx0 |
| 00xx1 | - | zzzz0  00x10 | zzzz0  001x0 | 1zzzz  10x11 | 1zzzz  101x1 | 1yzzy  x10x0 | 1yzzy  x1x00 | yyzz0  110xx | yyzz0  11x0x | yyzzy  11xx0 |
| 00x1x | zzz0z  00x01 | - | zzz0z  00100 | yzzzz  10x11 | yzz0z  101x1 | 1yz0z  x10x0 | 1yzyz  x1x00 | yyz0z  110xx | yyzyz  11x0x | yyz0z  11xx0 |
| 001xx | zz0zz  00001 | zz0zz  00010 | - | yz0zz  10x11 | yzzzz  101x1 | 1yyzz  x10x0 | 1y0zz  x1x00 | yyyzz  110xx | yy0zz  11x0x | yy0zz  11xx0 |
| x0x11 | zzzyz  00001 | zzzzy  00010 | zzzyy  00100 | - | zzz0z  10101 | zyz0y  x10x0 | zyzyy  x1x00 | zyz00  110xx | zyzy0  11x0x | zyz0y  11xx0 |
| x01x1 | zzyzz  00001 | zzyzy  00010 | zzzzy  00100 | zz0zz  10011 | - | zyyzy  x10x0 | zy0zy  x1x00 | zyyz0  110xx | zy0z0  11x0x | zy0zy  11xx0 |
| x10x0 | zyzzy  00001 | zyzzz  00010 | zyyzz  00100 | zyzzy  10011 | zyyzy  10101 | - | zz1zz  x1100 | zzzz1  110x1 | zz1z1  1110x 11x01 | zz1zz  111x0 |
| x1x00 | zyzzy  00001 | zyzyz  00010 | zyzzz  00100 | zyzyy  10011 | zyzzy  10101 | zzz1z  x1010 | - | zzz1y  110x1 | zzzz1  zzzzy  11101 11x01 | zzz1z  11110 |
| 110xx | yyzzz  00001 | yyzzz  00010 | yyyzz  00100 | zyzzz  10011 | zyyzz  10101 | 0zzzz  01010 | 0zyzz  x1100 | - | zzyzz  zz1zz  11101 11101 | zzyzz  11110 |
| 11x0x | yyzzz  00001 | yyzyz  00010 | yyzzz  00100 | zyzyz  10011 | zyzzz  10101 | yzzyz  01010 | 0zzzz  01100 | zzz1z  11011 | - | zzzyz  11110 |
| 11xx0 | yyzzy  00001 | yyzzz  00010 | yyzzz  00100 | zyzzy  10011 | zyzzy  10101 | yzzzz  01010 | yzzzz  01100 | zzzzy  11011 | zzzzy  11101 11101 | - |
| xx01x | zzzyz  00001 | zzzzz  Ø | zzyyz  00100 | zzzzz  Ø | zzyyz  10101 | zzzzz  Ø | zzyyz  01100 | zzzzz  Ø | zzyyz  11101 11101 | zzyzz  11110 |
| xx10x | zzyzz  00001 | Ø | zzzzz  Ø | Ø | zzzzz  Ø | Ø | zzzzz  Ø | Ø | zzzzz  Ø | zzzyz  11110 |
| Остаток | 00001 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11110 |

Проверка L-экстремалий показана в таблице 3.7

Таблица 3.7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) n L | 00001 | 00110 | 00111 | 01000 | 10111 | 11000 | 11001 | 11110 |
| 00001 | 00001 | 00yyy Ø | 00yy1 Ø | 0y00y Ø | y0yy1 Ø | yy00y Ø | yy001 Ø | yyyyy Ø |
| 11110 | yyyyy Ø | yy110 Ø | yy11y Ø | y1yy0 Ø | 1y11y Ø | 11yy0 Ø | 11yyy Ø | 11110 |
| 10010 | y00yy Ø | y0y10 Ø | y0y1y Ø | yy0y0 Ø | 10y1y Ø | 1y0y0 Ø | 1y0yy Ø | 1yy10 Ø |
| 01011 | 0y0y1 Ø | 0yy1y Ø | 0yy11 Ø | 010yy Ø | yyy11 Ø | y10yy Ø | y10y1 Ø | y1y1y Ø |
| 10010 | y00yy Ø | y0y10 Ø | y0y1y Ø | yy0y0 Ø | 10y1y Ø | 1y0y0 Ø | 1y0yy Ø | 1yy10 Ø |
| 10100 | y0y0y Ø | y01y0 Ø | y01yy Ø | yyy00 Ø | 101yy Ø | 1yy00 Ø | 1yy0y Ø | 1y1y0 Ø |
| 01101 | 0yy01 Ø | 0y1yy Ø | 0y1y1 Ø | 01y0y Ø | yy1y1 Ø | y1y0y Ø | y1y01 Ø | y11yy Ø |
| 10100 | y0y0y Ø | y01y0 Ø | y01yy Ø | yyy00 Ø | 101yy Ø | 1yy00 Ø | 1yy0y Ø | 1y1y0 Ø |

##### E = { 00xx1; 11xx0 }

##### Z´ = Z - E = { 00x1x; 001xx; x0x11; x01x1; x10x0; x1x00; 110xx; 11x0x; xx01x; xx10x }

Таблица 3.8 – Множество L-экстремалей

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00001 | 00110 | 00111 | 01000 | 10111 | 11000 | 11001 | 11110 |
| 00xx1 | zzzzz  Ø | zzzzy  00110 | zzzzz  Ø | zyzzy  01000 | yzzzz  10111 | yyzzy  11000 | yyzzz  11001 | yyzzy  11110 |
| 11xx0 | Ø | yyzzz  00110 | Ø | yzzzz  01000 | zyzzy  10111 | zzzzz  Ø | zzzzy  11001 | zzzzz  Ø |
| Остаток | Ø | 00110 | Ø | 01000 | 10111 | Ø | 11001 | Ø |

##### Множество кубов, непокрываемых L-экстремалями, L´ = L # E = { 00110; 01000; 10111; 11001 }

Таблица 3.9– Множество кубов, непокрываемых L-экстремалями

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Z´i n L´ | 00110 | 01000 | 10111 | 11001 |
| 00x1x | 00110 | Ø | Ø | Ø |
| 001xx | 00110 | Ø | Ø | Ø |
| x0x11 | Ø | Ø | 10111 | Ø |
| x01x1 | Ø | Ø | 10111 | Ø |
| x10x0 | Ø | 01000 | Ø | Ø |
| x1x00 | Ø | 01000 | Ø | Ø |
| 110xx | Ø | Ø | Ø | 11001 |
| 11x0x | Ø | Ø | Ø | 11001 |
| xx01x | Ø | Ø | Ø | Ø |
| xx10x | Ø | Ø | Ø | Ø |

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

##### Fmin1={00x1x; x01x1; x1x00; 11x0x; 00xx1; 11xx0}=

##### =

Преобразуем функции *П, S1, S2* к заданному логическому базису:

*П =*

*S1 =*

*S2 =*

Функциональная схема в заданном логическом базисе ОЧС представлена в приложении Б.

**3.2. Логический синтез одноразрядного четверичного умножителя**

ОЧУ - это комбинационное устройство, имеющее 5 входов и 4 выхода.

Принцип работы ОЧУ представлен с помощью таблицы истинности (таблица 3.2).

Таблица 3.2 – Таблица истинности ОЧУ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | h |  |  |  |  | Пример |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0\*0=00 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Выход «00» |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0\*0=00 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | Выход «00» |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0\*2=00 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | Выход «00» |
| 0 | 0 | 1 | 1 | 0 | x | x | x | x | 0\*3=00 |
| 0 | 0 | 1 | 1 | 1 | x | x | x | x | Выход «00» |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2\*0=00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Выход «02» |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2\*1=02 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | Выход «02» |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 2\*2=10 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход «02» |
| 0 | 1 | 1 | 1 | 0 | x | x | x | x | 2\*3=12 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | x | Выход «02» |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3\*0=00 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Выход «03» |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 3\*1=03 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Выход «03» |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 3\*2=12 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Выход «03» |
| 1 | 0 | 1 | 1 | 0 | x | x | x | x | 3\*3=21 |
| 1 | 0 | 1 | 1 | 1 | x | x | x | x | Выход «03» |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1\*0=00 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Выход «01» |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1\*1=01 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Выход «01» |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1\*2=02 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Выход «01» |
| 1 | 1 | 1 | 1 | 0 | x | x | x | x | 1\*3=03 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | x | Выход «01» |

Минимизацию выходов ОЧУ , проведем с помощью карт Вейча.

**Минимизация функции  картами Вейча:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | *x1* | | | | |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  | 1 | |  |  |  | 1 | |  | |  | |  | |  | |  | |
|  | \* | | \* | \* | \* | \* | | \* | | \* | | \* | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  | | |  | |  | |  | |  | |  | |
|  |  |  | | *h* | |  | | | *h* | | | |  | |  | |  | |

Минимизировав функцию, получим:

**Минимизация функции  картами Вейча:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | *x1* | | | | |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  | 1 | |  |  |  | 1 | |  | |  | |  | |  | |  | |
|  | \* | | \* | \* | \* | \* | | \* | | \* | | \* | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  | | |  | |  | |  | |  | |  | |
|  |  |  | | *h* | |  | | | *h* | | | |  | |  | |  | |

Минимизировав функцию, получим:

**Минимизация функции  картами Вейча:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | *x1* | | | | |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  |  | | 1 | 1 |  |  | |  | |  | |  | |  | |  | |
|  | \* | | \* | \* | \* | \* | | \* | | \* | | \* | |  | |  | |
|  |  |  | | 1 | 1 | 1 | 1 | |  | |  | |  | |  | |
|  |  | 1 | | 1 | 1 |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  | | |  | |  | |  | |  | |  | |
|  |  |  | | *h* | |  | | | *h* | | | |  | |  | |  | |

Минимизировав функцию, получим:

**Минимизация функции картами Вейча:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | *x1* | | | | |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  | 1 | |  | 1 | 1 |  | | 1 | |  | |  | |  | |  | |
|  | \* | | \* | \* | \* | \* | | \* | | \* | | \* | |  | |  | |
|  |  |  | |  | 1 | 1 | 1 | | 1 | |  | |  | |  | |
|  |  |  | |  | 1 |  |  | | 1 | |  | |  | |  | |  | |
|  |  |  | |  |  |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  |  | | |  | |  | |  | |  | |  | |
|  |  |  | | *h* | |  | | | *h* | | | |  | |  | |  | |

Минимизировав функцию, получим:

h

Преобразуем функции *, S1, S2* к заданному логическому базису:

***/****=*

*=*

*=*

Функциональная схема в заданном логическом базисе ОЧУ представлена в приложении В.

**4.ЭФФЕКТИВНОСТЬ МИНИМИЗАЦИИ**

Для проведения оценки эффективности минимизации переключательных функций необходимо посчитать цену схемы до минимизации и цену схемы после минимизации.

Все рассчитанные данные сведены в таблицу 4.1 и 4.2

Таблица 4.1 - Эффективность минимизации ОЧС

|  |  |  |  |
| --- | --- | --- | --- |
| Выход  схемы | Количество входов до минимизации | Количество входов после минимизации | Эффективность минимизации |
| *П* | c=5+4\*5+4=29 | с=10+2+3=15 | 1,93 |
| S1 | с=5+8\*5+8=53 | с=12+4+4=20 | 2,65 |
| S2 | с=5+8\*5+8=53 | с=18+4+6=28 | 1,89 |

Таблица 4.2 - Эффективность минимизации ОЧУ

|  |  |  |  |
| --- | --- | --- | --- |
| Выход  схемы | Количество входов до минимизации | Количество входов после минимизации | Эффективность минимизации |
| P1 /P2 | с=5+2\*5+2=17 | с=8+3+2=13 | 1,3 |
| P3 | с=5+9\*5+9=59 | с=9+3+3=15 | 3,93 |
| P4 | с=5+10\*5+10=65 | c=11+2+4=17 | 3,82 |

**5.СИНТЕЗ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРА**

Мультиплексор – это логическая схема, имеющая n входов,m управляющих входов и один выход. При этом должно выполняться равенство . На выход мультиплексора может быть пропущен без изменений любой (один) логический сигнал, поступающий на информационные входы. Порядковый номер информационного входа, значение с которого в данный момент должно быть передано на выход, должно быть передано на выход, определяется двоичным кодам на управляющих входах. Для синтеза ОЧС будем использовать мультиплексор «один из восьми» (1 из 8-ми). Входы – это информационные входы мультиплексора. Входы – управляющие входы.

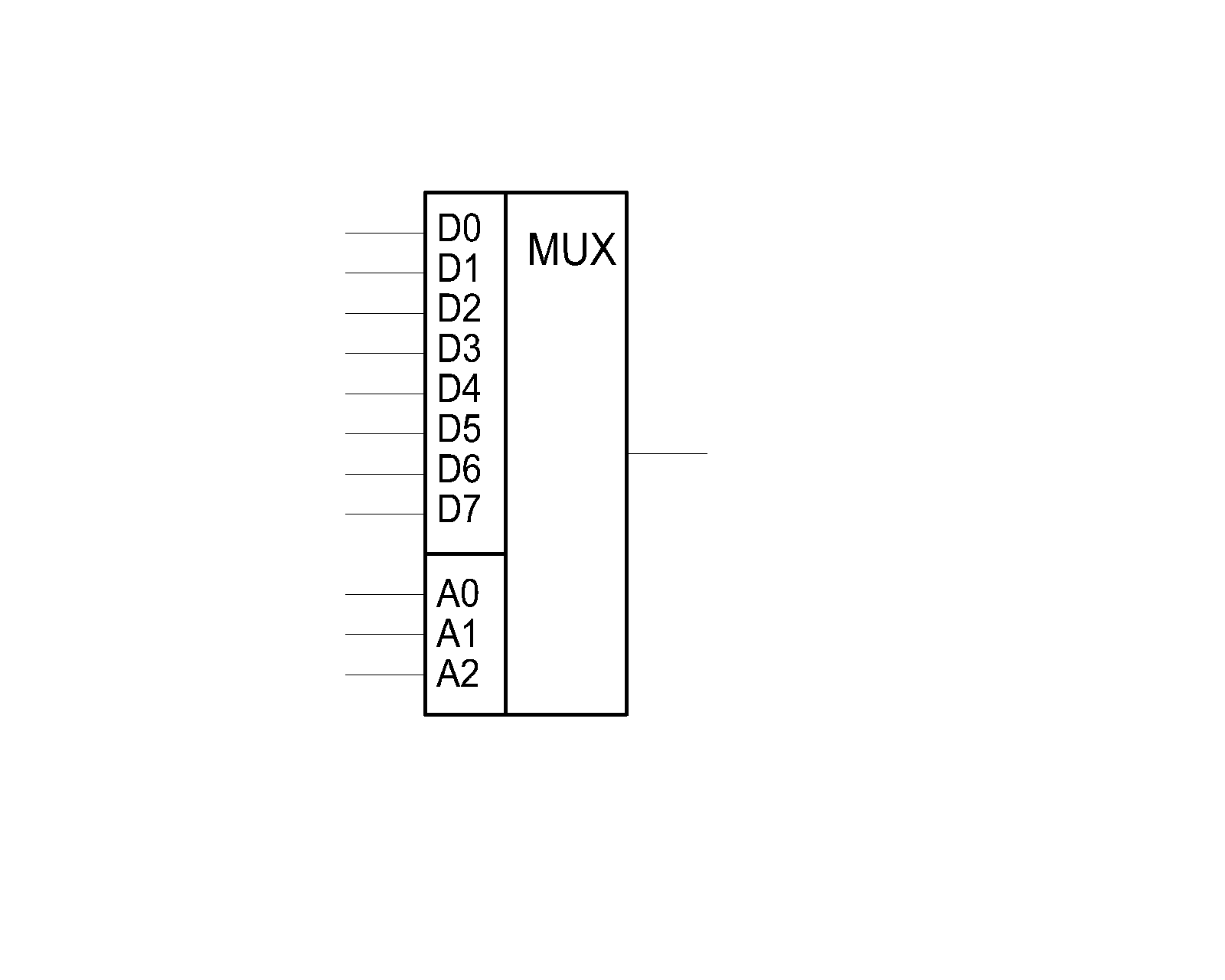


Рисунок 5.1 - Мультиплексор «один из восьми»

Используя таблицу истинности ОЧС, составим таблицу истинности для построения ОЧС на мультиплексорах (таблица 5.1).

Управление мультиплексором осуществляется тремя переменными: , а вход соответствующих значений функций на информационные входы обеспечивается реализацией этих функций на дополнительных логических элементах.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | p | П |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | “0” | 0 | \*P | 0 | \*P |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | x | x | x |
| 0 | 0 | 0 | 1 | 1 | x | x | x |
| 0 | 0 | 1 | 0 | 0 | x | “0” | x |  | x | “1” |
| 0 | 0 | 1 | 0 | 1 | x | x | x |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | “0” | 0 | \*P | 1 |  |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | x | x | x |
| 0 | 1 | 0 | 1 | 1 | x | x | x |
| 0 | 1 | 1 | 0 | 0 | x |  | x |  | x | “0” |
| 0 | 1 | 1 | 0 | 1 | x | x | x |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | \*P | 1 |  | 0 | “0” |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | x | x | x |
| 1 | 0 | 0 | 1 | 1 | x | x | x |
| 1 | 0 | 1 | 0 | 0 | x | “1” | x |  | x |  |
| 1 | 0 | 1 | 0 | 1 | x | x | x |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | “0” | 1 |  | 1 | “1” |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | x | x | x |
| 1 | 1 | 0 | 1 | 1 | x | x | x |
| 1 | 1 | 1 | 0 | 0 | x | “0” | x |  | x |  |
| 1 | 1 | 1 | 0 | 1 | x | x | x |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

Функциональная схема ОЧС на основе мультиплексора представлена в приложении Г.

**6.ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ**

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11, заменяя их на триады .

Таблица 6 - таблица истинности ПМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вх. диада | | Tригер | Зн. | Вых. диада | |
| Q1 | Q2 | T | П | S1 | S2 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

Проведём минимизацию Ппри помощи карты Карно:

Q2T

Q1

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

P = Q1T+Q1Q2

Очевидно, что S1 не минимизируется, поэтому

Проведём минимизацию S2при помощи карты Карно:

Q2T

Q1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S2 | 00 | 01 | 11 | 10 |
| 0 |  | 1 |  | 1 |
| 1 |  | 1 |  | 1 |

Реализация ПМ представлена в приложении Д.

**7.ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ**

Временные затраты на умножение сомножителей определяются в основном затратами на образование частичных произведений, получаемых на выходах ОЧС.

, где

– время преобразования множителя;

– время формирования дополнительного кода множимого;

– время умножения на ОЧУ;

– время формирования единицы переноса в ОЧС;

– время сдвига частичной суммы;

n – количество разрядов на множителе.

**ЗАКЛЮЧЕНИЕ**

При выполнения курсового проекта были разработаны структурная схема сумматора-умножителя первого типа и функциональные схемы ОЧС и ОЧУ.

Одноразрядный четверичный умножитель – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда и регистра Мн, 2 разряда из регистра Мт и управляющий вход h) и 4 двоичных выхода.

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.

Мультиплексор-это логическая схема, имеющая n информационных входов, m управляющих входов и один выход. При этом должно выполняться условие n=2m.

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

Также удалось усвоить навыки работы с различными типами минимизации переключательных функций, с проектированием функциональных схем в различных базисах и с расчетом времени работы схем.

**ЛИТЕРАТУРА**

[1] Савельев А.Я. Прикладная теория цифровых автоматов. М.: Высшая школа, 1985.

[2] Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. Мн.: Вышейшая школа, 1980.

[3] Лысиков Б.Г. Цифровая вычислительная техника. Мн.: , 2003 г.

[4] Луцик Ю.А., Лукьянова И.В., Ожигина М.П. – Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". -Мн.: ротапринт МРТИ ,2001 г.

[5] Луцик Ю.А., Лукьянова И.В.– Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". -Мн.:ротапринт МРТИ ,2004 г.