

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические
основы вычислительной техники

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 01 523 ПЗ

Студент

В. С. Чеботарёв

Руководитель

И. В. Лукьянова

МИНСК 2021

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические
основы вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Б. В. Никульшин
«___» _____ 20__ г.

ЗАДАНИЕ
по курсовой работе студента
Чеботарёва Вячеслава Сергеевича

- 1 Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел»
- 2 Срок сдачи студентом законченной работы: 19 мая 2021г.
- 3 Исходные данные к работе:
 - 3.1 исходные сомножители: $M_n = 51,50$; $M_t = 85,17$.
 - 3.2 алгоритм умножения: В.
 - 3.3 метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в дополнительных кодах.
 - 3.4 коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования; $0_4 - 11$, $1_4 - 00$, $2_4 - 10$, $3_4 - 01$.
 - 3.5 тип синтезируемого умножителя: 2.
 - 3.6 логический базис для реализации ОЧС: НЕ, И; метод минимизации – алгоритм Рота для одного выхода, метод карты Карно – для остальных.
 - 3.7 логический базис для реализации ОЧУС: ИЛИ-НЕ; метод минимизации – карты 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.

5 Перечень графического материала:

5.1 Сумматор-умножитель второго типа. Схема электрическая структурная.

5.2 Одноразрядный четверичный сумматор. Схема электрическая функциональная.

5.3 Одноразрядный четверичный умножитель-сумматор. Схема электрическая функциональная.

5.4 Преобразователь множителя. Схема электрическая функциональная.

5.5 Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов курсовой работы	Объём этапа, %	Срок выполнения этапа	Примечания
Разработка алгоритма умножения	10	26.02-13.03	
Разработка структурной схемы сумматора-умножителя	10	14.03-27.03	С выполнением чертежа
Разработка функциональных схем основных узлов сумматора-умножителя	50	28.03-08.05	С выполнением чертежей
Синтез комбинационных схем устройств на основе мультиплексоров	10	09.05-22.05	С выполнением чертежа
Завершение оформления пояснительной записки	20	23.05-05.06	

Дата выдачи задания: 26 февраля 2021 г.

Руководитель

И. В. Лукьянова

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ	6
2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ ВТОРОГО ТИПА	9
3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ	
3.1 Логический синтез одноразрядного четверичного умножителя- сумматора	11
3.2 Логический синтез одноразрядного четверичного сумматора	14
4.3 Логический синтез преобразователя множителя	23
4 СИНТЕЗ СХЕМЫ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРА	25
5 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ	
5.1 Расчёт времени умножения на один разряд множителя	27
5.2 Расчёт времени умножения на n разрядов множителя	28
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30
ПРИЛОЖЕНИЕ А Сумматор-умножитель второго типа. Схема электрическая структурная	31
ПРИЛОЖЕНИЕ Б Одноразрядный четверичный умножитель-сумматор. Схема электрическая функциональная	32
ПРИЛОЖЕНИЕ В Одноразрядный четверичный сумматор. Схема электрическая функциональная	33
ПРИЛОЖЕНИЕ Г Преобразователь множителя. Схема электрическая функциональная	34
ПРИЛОЖЕНИЕ Д Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная	35
ПРИЛОЖЕНИЕ Е Ведомость документов	36

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.

$$M_H = 51,50; M_T = 85,17.$$

Множимое

$$\begin{array}{r|l} 51 & 4 \\ 40 & 12 \quad 4 \\ \hline 11 & 12 \quad 3 \\ 8 & 0 \\ \hline 3 & \end{array}$$

$$\begin{array}{r} 0.50 \\ * \quad 4 \\ \hline 2.00 \\ * \quad 4 \\ \hline 0.00 \\ * \quad 4 \\ \hline 0.00 \end{array}$$

$$M_{H4} = 303,200.$$

В соответствии с кодировкой множимого:

$$M_{H2/4} = 011101,101111.$$

Множитель

$$\begin{array}{r|l} 85 & 4 \\ 84 & 21 \quad 4 \\ 1 & 20 \quad 5 \quad 4 \\ & 1 \quad 4 \quad 1 \\ & 1 \end{array}$$

$$\begin{array}{r} 0.17 \\ * \quad 4 \\ \hline 0.68 \\ * \quad 4 \\ \hline 2.72 \\ * \quad 4 \\ \hline 2.88 \end{array}$$

$$M_{T4} = 1111.02$$

В соответствии с обычной весомозначной кодировкой множителя:

$$M_{T2/4} = 01010101.0010.$$

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

$$M_H = 0.011101101111 \quad P_{MH} = 0.1110 + 03_4 - \text{закодировано по заданию}$$

$$M_T = 0.010101010010 \quad P_{MT} = 0.0100 + 10_4 - \text{закодировано традиционно}$$

3. Умножение двух чисел с плавающей запятой на два разряда множителя одновременно в дополнительных кодах. Это сводится к сложению

порядков, формированию знака произведения, преобразованию разрядов множителя согласно алгоритму и перемножению мантисс сомножителей.

Порядок произведения будет следующим:

$$P_{\text{мн}} = 0.1110 + 03_4$$

$$P_{\text{мт}} = 0.0100 + 10_4$$

$$P_{\text{мн} \cdot \text{мт}} = 0.0001 + 13_4$$

Результат закодирован в соответствии с заданием на кодировку множимого.

Знак произведения определяется суммой по модулю два знаков сомножителей, т.е.:

$$\text{зн Мн} \oplus \text{зн Мт} = 0 \oplus 0 = 0$$

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в дополнительных кодах диада $11(3_4)$ заменяется на триаду $10\bar{1}(\bar{1}_4)$, диада $10(2_4)$ заменяется на триаду $1\bar{1}0(\bar{2}_4)$.

Преобразованный множитель имеет вид:

$$[M_T]_д = 11111\bar{2}$$

Для выполнения операции умножения нам понадобятся заготовки в виде:

$$[M_n]_д = 0.303200$$

$$[-2M_n]_д = 3.212100$$

Перемножение мантисс по алгоритму “В” приведено в таблице 1.1.

Таблица 1.1 - Перемножение мантисс

Четверичная с/с	Двоично-четверичная с/с	Комментарий
000000000000	11111111111111111111	$\Sigma_0^ч$
000000000000	11111111111111111111	$\Sigma_0^ч * 4$
000000303200	1111111111101110110111	$\Pi_1 = [M_n]_д$
000000303200	1111111111101110110111	$\Sigma_1^ч$
000003032000	1111111111011101101111	$\Sigma_1^ч * 4$
000000303200	1111111111101110110111	$\Pi_2 = [M_n]_д$
000010001200	1111111001111110010111	$\Sigma_2^ч$
000100012000	1111110011111100101111	$\Sigma_2^ч * 4$
000000303200	1111111111101110110111	$\Pi_3 = [M_n]_д$
000100321200	11111100111101100010111	$\Sigma_3^ч$
001003212000	1111001111011000101111	$\Sigma_3^ч * 4$
000000303200	1111111111101110110111	$\Pi_3 = [M_n]_д$
001010121200	11110011001100101110111	$\Sigma_4^ч$
010101212000	1100110011001011101111	$\Sigma_4^ч * 4$
000000303200	1111111111101110110111	$\Pi_4 = [M_n]_д$
010102121200	11001100111000100010111	$\Sigma_5^ч$
101021212000	0011001110001000101111	$\Sigma_5^ч * 4$
333332121000	01010101011000100011111	$\Pi_6 = [-2M_n]_д$
101013333000	00110011000101010111111	$\Sigma_6^ч$

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

$$\begin{aligned} [M_H \cdot M_T]_4 &= 1010133.33000, & P_{M_H \cdot M_T} &= 0; \\ [M_H \cdot M_T]_{10} &= 4383.9375. \end{aligned}$$

Результат прямого перемножения операндов дает результат:

$$M_{H10} \cdot M_{T10} = 51.50 * 85.17 = 4386.2550.$$

Абсолютная погрешность:

$$\Delta = 4386.2550 - 4383.9375 = 2.3175.$$

Относительная погрешность:

$$\delta = \frac{\Delta}{M_H \cdot M_T} = \frac{2.3175}{4386.2550} = 0.0005284 \quad (\delta = 0.053 \%).$$

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 ВТОРОГО ТИПА

Структурная схема сумматора-умножителя второго типа для алгоритма умножения «В» приведена на рисунке приложения А.

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).

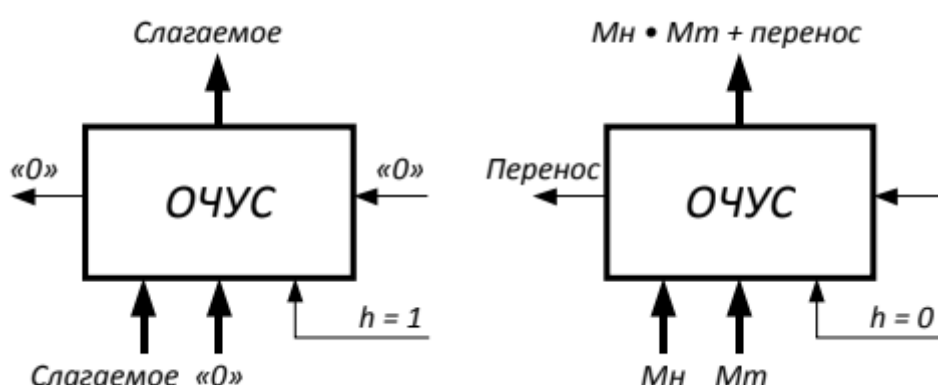


Рисунок 2.1 – Режимы работы ОЧУС

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

В ОЧУС 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 ОЧУС 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.

Выход 2 ПМ переходит в единичное состояние, если текущая диада со-

держит отрицание ($\overline{01}$). 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 2.2.

Таблица 2.2 – Режимы работы формирователя дополнительного кода

Сигналы на входах ФДК		Результат на выходах ФДК
F_1	F_2	
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 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»):

$$\begin{array}{ccccccc}
 3 & \cdot & 2 & = & 1 & 2 & (+1 \text{ в случае переноса из} \\
 \text{max} & & \text{max} & & \text{max} & & \text{предыдущего ОЧУС)} \\
 \text{Мн} & & \text{Мт} & & \text{перенос} & &
 \end{array}$$

Так как на входы ОЧУС из регистра Мт не могут поступить коды «3», в таблице истинности работы ОЧУС будут содержаться 36 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).

3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ

3.1 Логический синтез одноразрядного четверичного умножителя-сумматора

ОЧУС – это комбинационное устройство, имеющее шесть входов (два разряда из регистра множимого, два разряда из регистра множителя, вход переноса и управляющий вход h) и три выхода.

Принцип работы ОЧУС представлен с помощью таблицы истинности (таблица 3.1).

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

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

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.

Таблица 3.1

Пер.	Мн		Мт		Упр.	Перенос	Результат		Результат операции в четверичной системе счисления
P1	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	1	1	1*0+0=00
0	0	0	0	0	1	0	0	0	Выход-код "01"
0	0	0	0	1	0	0	0	0	1*1+0=01
0	0	0	0	1	1	0	0	0	Выход-код "01"
0	0	0	1	0	0	0	1	0	1*2+0=02
0	0	0	1	0	1	0	0	0	Выход-код "01"
0	0	0	1	1	0	x	x	x	1*3+0=03
0	0	0	1	1	1	x	x	x	Выход-код "01"
0	0	1	0	0	0	0	1	1	3*0+0=00
0	0	1	0	0	1	0	0	1	Выход-код "03"
0	0	1	0	1	0	0	0	1	3*1+0=03
0	0	1	0	1	1	0	0	1	Выход-код "03"
0	0	1	1	0	0	1	1	0	3*2+0=12
0	0	1	1	0	1	0	0	1	Выход-код "03"
0	0	1	1	1	0	x	x	x	3*3+0=21
0	0	1	1	1	1	x	x	x	Выход-код "03"
0	1	0	0	0	0	0	1	1	2*0+0=00
0	1	0	0	0	1	0	1	0	Выход-код "02"
0	1	0	0	1	0	0	1	0	2*1+0=02
0	1	0	0	1	1	0	1	0	Выход-код "02"
0	1	0	1	0	0	1	1	1	2*2+0=10
0	1	0	1	0	1	0	1	0	Выход-код "02"
0	1	0	1	1	0	x	x	x	2*3+0=12
0	1	0	1	1	1	x	x	x	Выход-код "02"
0	1	1	0	0	0	0	1	1	0*0+0=00

Продолжение таблицы 3.1

0	1	1	0	0	1	0	1	1	Выход-код "00"
0	1	1	0	1	0	0	1	1	$0*1+0=00$
0	1	1	0	1	1	0	1	1	Выход-код "00"
0	1	1	1	0	0	0	1	1	$0*2+0=00$
0	1	1	1	0	1	0	1	1	Выход-код "00"
0	1	1	1	1	0	x	x	x	$0*3+0=00$
0	1	1	1	1	1	x	x	x	Выход-код "00"
1	0	0	0	0	0	x	x	x	$1*0+1=00$
1	0	0	0	0	1	x	x	x	Выход-код "01"
1	0	0	0	1	0	x	x	x	$1*1+1=02$
1	0	0	0	1	1	x	x	x	Выход-код "01"
1	0	0	1	0	0	0	0	1	$1*2+1=03$
1	0	0	1	0	1	x	x	x	Выход-код "01"
1	0	0	1	1	0	x	x	x	$1*3+1=10$
1	0	0	1	1	1	x	x	x	Выход-код "01"
1	0	1	0	0	0	x	x	x	$3*0+1=00$
1	0	1	0	0	1	x	x	x	Выход-код "03"
1	0	1	0	1	0	x	x	x	$3*1+1=10$
1	0	1	0	1	1	x	x	x	Выход-код "03"
1	0	1	1	0	0	1	0	1	$3*2+1=13$
1	0	1	1	0	1	x	x	x	Выход-код "03"
1	0	1	1	1	0	x	x	x	$3*3+1=22$
1	0	1	1	1	1	x	x	x	Выход-код "03"
1	1	0	0	0	0	x	x	x	$2*0+1=00$
1	1	0	0	0	1	x	x	x	Выход-код "02"
1	1	0	0	1	0	x	x	x	$2*1+1=03$
1	1	0	0	1	1	x	x	x	Выход-код "02"
1	1	0	1	0	0	1	0	0	$2*2+1=11$
1	1	0	1	0	1	x	x	x	Выход-код "02"
1	1	0	1	1	0	x	x	x	$2*3+1=13$
1	1	0	1	1	1	x	x	x	Выход-код "02"
1	1	1	0	0	0	x	x	x	$0*0+1=01$
1	1	1	0	0	1	x	x	x	Выход-код "00"
1	1	1	0	1	0	x	x	X	$0*1+1=01$
1	1	1	0	1	1	x	x	x	Выход-код "00"
1	1	1	1	0	0	0	0	0	$0*2+1=01$
1	1	1	1	0	1	x	x	x	Выход-код "00"
1	1	1	1	1	0	x	x	x	$0*3+1=01$
1	1	1	1	1	1	x	x	x	Выход-код "00"

В таблице 3.1 выделено 36 безразличных наборов, т. к. на входы ОЧУС из разрядов множителя не может поступить код «11», при работе ОЧУС как сумматора на вход переноса не может поступить единица, а при умножении на ноль или единицу на вход переноса также не может поступить единица.

Минимизацию функций P , Q_1 и Q_2 проведем при помощи карт Вейча.

Для функции Р:

		y1		y2					
p1	x1	0	x	x	x	x	x	x	x
		1	x	x	x	x	x	x	x
		0	x	x	x	x	x	x	x
		1	x	x	x	x	x	x	x
		1	0	x	x	0	0	0	0
		0	0	x	x	0	0	0	0
		1	0	x	x	0	0	0	0
		0	0	x	x	0	0	0	0

Для функции Q_2 :

		y_1				y_2			
p_1	x_1	1	x	x	x	x	x	x	x
		1	x	x	x	x	x	x	x
		0	x	x	x	x	x	x	x
		0	x	x	x	x	x	x	x
	x_2	1	0	x	x	0	0	0	1
		1	1	x	x	1	1	1	1
		0	1	x	x	1	1	1	1
		0	0	x	x	0	0	0	1

Следовательно:

$$P = \overline{x_1} + x_2 + \overline{y_1} + h + x_1 + \overline{x_2} + \overline{y_1} + h$$

$$Q_1 = \overline{p_1} + \overline{x_1} + \overline{p_1} + y_2 + h$$

$$Q_1 = \overline{x_2} + y_1 + \overline{x_2} + h + \overline{p_1} + x_1 + y_1 + y_2 + h + \overline{p_1} + \overline{x_1} + y_2 + h$$

Функциональная схема ОЧУС в заданном базисе представлена в приложении

Б.

Оценка эффективности минимизации функций ОЧУС.

f	C_f	$C_{f_{min}}$	k
P	$4 \cdot 6 + 4 + 6 = 34$	$3 + 2 + 2 \cdot 4 = 13$	2.7
Q_1	$16 \cdot 6 + 16 + 6 = 118$	$3 + 2 + 2 + 3 = 10$	11.8
Q_2	$15 \cdot 6 + 15 + 6 = 111$	$5 + 5 + 3 \cdot 2 + 3 + 4 = 23$	4.8

3.2 Логический синтез одноразрядного четвертичного сумматора

Одноразрядный четвертичный 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).

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

Так как ОЧС синтезируется для схемы второго типа, то безразличные наборы в таблице истинности отсутствуют.

Таблица 3.2

a ₁	a ₂	b ₁	b ₂	p	П	S ₁	S ₂	Примеры операции в четверичной с/с
1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	1	0	1+1+0=02
0	0	0	0	1	0	0	1	1+1+1=03
0	0	0	1	0	1	1	1	1+3+0=10
0	0	0	1	1	1	0	0	1+3+1=11
0	0	1	0	0	0	0	1	1+2+0=03
0	0	1	0	1	1	1	1	1+2+1=10
0	0	1	1	0	0	0	0	1+0+0=01
0	0	1	1	1	0	1	0	1+0+1=02
0	1	0	0	0	1	1	1	3+1+0=10
0	1	0	0	1	1	0	0	3+1+1=11
0	1	0	1	0	1	1	0	3+3+0=12
0	1	0	1	1	1	0	1	3+3+1=13
0	1	1	0	0	1	0	0	3+2+0=11
0	1	1	0	1	1	1	0	3+2+1=12
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	0	0	1	2+1+0=03
1	0	0	0	1	1	1	1	2+1+1=10
1	0	0	1	0	1	0	0	2+3+0=11
1	0	0	1	1	1	1	0	2+3+1=12
1	0	1	0	0	1	1	1	2+2+0=10
1	0	1	0	1	1	0	0	2+2+1=11
1	0	1	1	0	0	1	0	2+0+0=02
1	0	1	1	1	0	0	1	2+0+1=03
1	1	0	0	0	0	0	0	0+1+0=01
1	1	0	0	1	0	1	0	0+1+1=02
1	1	0	1	0	0	0	1	0+3+0=03
1	1	0	1	1	1	1	1	0+3+1=10
1	1	1	0	0	0	1	0	0+2+0=02
1	1	1	0	1	0	0	1	0+2+1=03
1	1	1	1	0	0	1	1	0+0+0=00
1	1	1	1	1	0	0	0	0+0+1=01

Минимизацию переключательной функций II проведём с помощью алгоритма Рота, а функции $S1$ и $S2$ с помощью карты Карно.

Для функции $П$:

Множество единичных кубов:

$$L = \left\{ \begin{array}{l} 00010, 00011, 00101, 01000 \\ 01001, 01010, 01011, 01100 \\ 01101, 01111, 10001, 10010 \\ 10011, 10100, 10101, 11011 \end{array} \right\}$$

Множество безразличных кубов:

$$N = \{\emptyset\}.$$

Сформируем множество: $C_0 = L \cup N$

$$C_0 = \{00010, 00011, 00101, 01000, 01001, 01010, 01011, 01100, 01101, 01111, 10001, 10010, 10011, 10100, 10101, 11011\}$$

Первый шаг умножения ($C_0 * C_0$) приведен в таблице 3.3

В результате этой операции сформируется новое множество кубов:

$$C_1 = \{0001x; 0x010; x0010; 0x011; x0011; 0x101; x0101; 0100x; 010x0; 01x00; 010x1; 01x01; 0101x; 01x11; x1011; 0110x; 011x1; 100x1; 10x01; 1x011; 1001x; 1010x\}$$

Множество Z_0 кубов, не участвовавших в образовании новых кубов, пустое.

Следующий шаг поиска простых импликант с помощью операции $C_1 * C_1$ приведен в таблице 3.4

$$C_2 = \{0x01x; x001x; xx011; 010xx; 01x0x; 01xx1\}$$

Множество кубов не участвующих в образовании новых кубов имеет вид:

$$Z_1 = \{0x101; x0101; 100x1; 10x01; 1010x\}$$

Следующий шаг поиска простых импликант с помощью операции $C_2 * C_2$ приведен в таблице 3.5

Новых кубов (третьей размерности) не образовалось

Множество кубов не участвующих в образовании новых кубов имеет вид:

$$Z_2 = \{0x01x; x001x; xx011; 010xx; 01x0x; 01xx1\}$$

На этом заканчивается поиск простых импликант, т.к. $|C_3| \leq 1$.

Таблица 3.3 - Поиск простых импликант (C0*C0)

C0*C0	00010	00011	00101	01000	01001	01010	01011	01100	01101	01111	10001	10011	10100	10101	11011	10010
00010	-															
00011	0001y	-														
00101	00ywy	00yy1	-													
01000	0y0y0	0y0yy	0yy0y	-												
01001	0y0yy	0y0y1	0yy01	0100y	-											
01010	0y010	0y01y	0yyy	010y0	010yy	-										
01011	0y01y	0y011	0yyy1	010yy	010y1	0101y	-									
01100	0yyy0	0yyy	0y10y	01y00	01y0y	01yy0	01yy	-								
01101	0yyy	0yyy1	0y101	01y0y	01y01	01yy	01yy1	0110y	-							
01111	0yy1y	0yy11	0y1y1	01yy	01yy1	01y1y	01y11	011yy	011y1	-						
10001	y00yy	y00y1	y0y01	yy00y	yy001	yy0yy	yy0y1	yy0y	yy01	yyy1	-					
10011	y001y	y0011	y0yy1	yy0yy	yy0y1	yy01y	yy011	yyy	yyy1	yyy11	100y1	-				
10100	y0yy0	y0yy	y010y	yy00	yy0y	yyy0	yyy	yy100	yy10y	yy1yy	10y0y	10yy	-			
10101	y0yy	y0yy1	y0101	yy0y	yy01	yyy	yyy1	yy10y	yy101	yy1y1	10y01	10yy1	1010y	-		
11011	yy01y	yy011	yyy1	y10yy	y10y1	y101y	y1011	y1yy	y1yy1	y1y11	1y0y1	1y011	1yy	1yy1	-	
10010	y0010	y001y	y0yy	yy0y0	yy0yy	yy010	yy01y	yyy0	yyy	yyy1y	100yy	1001y	10yy0	10yy	1y01y	-

Таблица 3.4 - Поиск простых импликант C1*C1

C1*C1	0001x	0x010	x0010	0x011	x0011	0x101	x0101	0100x	010x0	01x00	010x1	01x01	0101x	01x11	x1011	0110x	011x1	100x1	10x01	1x011	1001x	1010x
0001x	-																					
0x010	00010	-																				
x0010	00010	00010	-																			
0x011	00011	0x01y	0001y	-																		
x0011	00011	0001y	x001y	00011	-																	
0x101	00y1	0x0yy	000yy	0x0y1	00y1	-																
x0101	00yy1	000yy	x00yy	00yy1	x0yy1	00101	-															
0100x	0y0yx	010y0	0y0y0	010y1	0y0y1	01y01	0yy01	-														
010x0	0y010	01010	0y010	0101y	0y01y	01y0y	0yy0y	01000	-													
01x00	0y0y0	010y0	0y0y0	010yy	0y0yy	0110y	0y10y	01000	-													
010x1	0y011	0101y	0y01y	01011	0y011	01y01	0yy01	0100y	0100y	0100y	-											
01x01	0y0y1	010yy	0y0yy	010y1	0y0y1	01101	0y101	01001	0100y	01x0y	01001	-										
0101x	0y01x	01010	0y010	01011	0y011	01yy1	0yy1	010yx	01010	010y0	01011	010y1	-									
01x11	0y011	0101y	0y01y	01011	0y011	011y1	0y1y1	010y1	0101y	01x0y	01011	01xy1	01011	-								
x1011	0y011	0101y	x010y	01011	x0101	01yy1	xyy1	010y1	0101y	010yy	01011	010y1	01011	01011	-							
0110x	0yyyx	01yy0	0yyy0	01yy1	0yy1	01101	0y101	01y0x	01y00	01100	01y01	01101	01yyx	011y1	01yy1	-						
011x1	0yy11	01y1y	0yy1y	01y11	0yy11	01101	0y101	01y01	01y0y	0110y	01yx1	01101	01y11	01111	01y11	01101	-					
100x1	y0011	y001y	1001y	y0011	10011	y0y01	10y01	y0001	y00xy	y000y	y00x1	y0001	y0011	y0011	1y011	y001	yyx1	-				
10x01	y00y1	y00yy	100yy	y00y1	100y1	y0101	10101	y0001	y000y	y0x0y	y0001	y0x01	y00y1	yyx1	1y0y1	y0101	yy101	10001	-			
1x011	y0011	y001y	1001y	y0011	10011	yyy1	10yy1	y10y1	y101y	y10yy	y1011	y10y1	y1011	y1011	11011	y1yy1	y1y11	10011	100y1	-		
1001x	y001x	y0010	10010	y0011	10011	y0y1	10yy1	y00yx	y010	y0y0	y011	y0y01	yy01x	yy011	1y011	yyyx	yy11	10011	100y1	10011	-	
1010x	y0yx	y0y0	10yy0	y0yy1	10yy1	y0101	10101	yy00x	yy00	yy100	yy01	yy101	yyyx	yy1y1	1yyy1	yy10x	yy101	10y01	10101	10yy1	10yyx	-

Таблица 3.5 – Поиск простых импликант $C_2 * C_2$

$C_2 * C_2$	0x01x	x001x	xx011	010xx	01x0x	01xx1
0x01x	-					
x001x	0001x	-				
xx011	0x011	x0011	-			
010xx	0101x	0y01x	01011	-		
01x0x	010yx	0y0yx	010y1	0100x	-	
01xx1	01011	0y011	01011	010x1	01x01	-

Новых кубов (третьей размерности) не образовалось

Множество кубов не участвующих в образовании новых кубов имеет вид:

$$Z_2 = \{0x01x; x001x; xx011; 010xx; 01x0x; 01xx1\}$$

На этом заканчивается поиск простых импликант, т.к. $|C_3| \leq 1$.

Множество простых импликант:

$$Z = Z_0 \cup Z_1 \cup Z_2 = \{0x101; x0101; 100x1; 10x01; 1010x; 0x01x; x001x; xx011; 010xx; 01x0x; 01xx1\}$$

Следующий этап – поиск L -экстремалей на множестве простых импликант (таблица 3.6). Для этого используется операция $\#$ (решетчатое вычитание).

В таблице 3.6 из каждой простой импликанты поочередно вычитаются все остальные простые импликанты $Z \# (Z \setminus z)$, результат операции (последняя строка таблицы) указывает на то, что L -экстремальями стали следующие простые импликанты:

$$E = \{1010x; x001x; xx011; 01x0x; 01xx1\}$$

Т.к. безразличных наборов нет, то все L -экстремали являются минимальным значением.

Далее необходимо проанализировать, какие из исходных единичных кубов (множество L) не покрыты найденной L -экстремалью. Этот анализ осуществляется с помощью таблицы 3.7

$$L' = \{00101; 01010; 10001\}$$

Чтобы их покрыть, воспользуемся множеством простых импликант, не являющихся L -экстремальями (таблица 3.8).

Из таблицы видно, что каждый из непокрытых единичных кубов может быть покрыт двумя равнозначными способами.

Таблица 3.6 - Поиск L-экстремалей

z#(Z-z)	0x101	x0101	100x1	10x01	1010x	0x01x	x001x	xx011	010xx	01x0x	01xx1
0x101	-	10101	100x1	10x01	1010x	0x01x	x001x	xx011	010xx	0100x 01x00	010x1 01x11
x0101	01101	-	100x1	10001	10100	0x01x	x001x	xx011	010xx	0100x 01x00	010x1 01x11
100x1	01101	10101	-	Ø	10100	0x01x	0001x x0010	0x011 x1011	010xx	0100x 01x00	010x1 01x11
10x01	01101	Ø	10011	-	10100	0x01x	0001x x0010	0x011 x1011	010xx	0100x 01x00	010x1 01x11
1010x	01101	Ø	10011	Ø	-	0x01x	0001x x0010	0x011 x1011	010xx	0100x 01x00	010x1 01x11
0x01x	01101	Ø	10011	Ø	10100	-	10010	11011	0100x	0100x 01x00	01001 01111
x001x	01101	Ø	Ø	Ø	10100	0101x	-	11011	0100x	0100x 01x00	01001 01111
xx011	01101	Ø	Ø	Ø	10100	01010	10010	-	0100x	0100x 01x00	01001 01111
010xx	01101	Ø	Ø	Ø	10100	Ø	10010	11011	-	01100	01111
01x0x	Ø	Ø	Ø	Ø	10100	Ø	10010	11011	Ø	-	01111
01xx1	Ø	Ø	Ø	Ø	10100	Ø	10010	11011	Ø	01100	-

Таблица 3.7 Поиск не покрытых исходных наборов

L#E	00010	00011	00101	01000	01001	01010	01011	01100	01101	01111	10001	10011	10100	10101	11011	10010
1010x	00010	00011	00101	01000	01001	01010	01011	01100	01101	01111	10001	10011	Ø	Ø	11011	10010
x001x	Ø	Ø	00101	01000	01001	01010	01011	01100	01101	01111	10001	Ø	Ø	Ø	11011	Ø
xx011	Ø	Ø	00101	01000	01001	01010	Ø	01100	01101	01111	10001	Ø	Ø	Ø	Ø	Ø
01x0x	Ø	Ø	00101	Ø	Ø	01010	Ø	Ø	Ø	01111	10001	Ø	Ø	Ø	Ø	Ø
01xx1	Ø	Ø	00101	Ø	Ø	01010	Ø	Ø	Ø	Ø	10001	Ø	Ø	Ø	Ø	Ø

Таблица 3.8 - Покрытие оставшихся кубов

$Z' \text{ n } L'$	00101	01010	10001
0x101	00101	∅	∅
x0101	00101	∅	∅
100x1	∅	∅	10001
10x01	∅	∅	10001
0x01x	∅	01010	∅
010xx	∅	01010	∅

Следовательно, существует 8 тупиковых (минимальных) форм:

$F_{min1} = \{0x101; 100x1; 0x01x; 1010x; x001x; xx011; 01x0x; 01xx1\}$

$F_{min2} = \{x0101; 100x1; 0x01x; 1010x; x001x; xx011; 01x0x; 01xx1\}$

$F_{min3} = \{0x101; 10x01; 0x01x; 1010x; x001x; xx011; 01x0x; 01xx1\}$

$F_{min4} = \{x0101; 10x01; 0x01x; 1010x; x001x; xx011; 01x0x; 01xx1\}$

$F_{min5} = \{0x101; 100x1; 010xx; 1010x; x001x; xx011; 01x0x; 01xx1\}$

$F_{min6} = \{x0101; 100x1; 010xx; 1010x; x001x; xx011; 01x0x; 01xx1\}$

$F_{min7} = \{0x101; 10x01; 010xx; 1010x; x001x; xx011; 01x0x; 01xx1\}$

$F_{min8} = \{x0101; 10x01; 010xx; 1010x; x001x; xx011; 01x0x; 01xx1\}$

Для функции S_1 :

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

Для функции S_2 :

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

Следовательно:

$$\begin{aligned}
 \Pi &= \overline{\overline{b_1 b_2 p}} * \overline{\overline{a_2 \overline{b_1} b_2}} * \overline{\overline{a_1 a_2 p}} * \overline{\overline{a_1 a_2 \overline{b_2}}} * \overline{\overline{a_1 a_2 \overline{b_1}}} * \overline{\overline{a_1 \overline{a_2} b_1 \overline{b_2}}} * \overline{\overline{a_2 b_1 \overline{b_2} p}} \\
 &\quad * \overline{\overline{a_1 \overline{a_2} b_2 p}} \\
 S_1 &= \overline{\overline{a_1 b_1 p}} * \overline{\overline{a_1 \overline{b_1} p}} * \overline{\overline{a_1 \overline{b_1} \overline{p}}} * \overline{\overline{a_1 b_1 \overline{p}}} \\
 S_2 &= \overline{\overline{a_1 \overline{a_2} b_1 \overline{b_2}}} * \overline{\overline{a_1 a_2 b_1 b_2}} * \overline{\overline{a_1 a_2 b_1 \overline{p}}} * \overline{\overline{a_2 \overline{b_1} b_2 p}} * \overline{\overline{a_1 a_2 \overline{b_1} \overline{b_2} \overline{p}}} * \overline{\overline{a_1 \overline{a_2} \overline{b_1} b_2 \overline{p}}} \\
 &\quad * \overline{\overline{a_1 \overline{a_2} b_1 b_2 p}} * \overline{\overline{a_1 a_2 b_1 \overline{b_2} p}} * \overline{\overline{a_2 \overline{b_1} \overline{b_2} p}} * \overline{\overline{a_1 \overline{a_2} \overline{b_2} \overline{p}}}
 \end{aligned}$$

Функциональная схема ОЧС в заданном базисе представлена в приложении В.

Оценка эффективности минимизации функций ОЧС.

f	C_f	$C_{f_{min}}$	k
Π	$16 \cdot 5 + 16 + 5 = 101$	$4 + 8 + 5 \cdot 3 + 3 \cdot 4 = 39$	2.6
S_1	$16 \cdot 5 + 16 + 5 = 101$	$3 + 4 + 3 \cdot 4 = 19$	5.3
S_2	$16 \cdot 5 + 16 + 5 = 101$	$4 \cdot 6 + 5 \cdot 3 + 5 + 10 = 54$	1.9

3.3 Логический синтез преобразователя множителя

Преобразователь множителя служит для исключения из множителя диад 11 и 10, заменяя их на триады $1\overline{0}1$ и $1\overline{1}0$ соответственно.

Таблица истинности для ПМ. Таблица 3.9

Входная диада		Младший бит	Знак	Выходная диада	
Q_n	Q_{n-1}	Q_{n-2}	P	S_1	S_2
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	1	0	0

Минимизируем выходные функции P , S_1 и S_2 .

Минимизацию проведем картами Карно.

Минимизация функции P :

	00	01	11	10
0	0	0	0	0
1	1	1	1	1

Минимизация функции S_1 :

	00	01	11	10
0	0	0	1	0
1	1	0	0	0

Минимизация функции S_2 :

	00	01	11	10
0	0	1	0	1
1	0	1	0	1

Следовательно:

$$P = Q_n;$$

$$S_1 = Q_n \overline{Q_{n-1}} \overline{Q_{n-2}} + \overline{Q_n} Q_{n-1} Q_{n-2};$$

$$S_2 = \overline{Q_{n-1}} Q_{n-2} + Q_{n-1} \overline{Q_{n-2}} = Q_{n-1} \oplus Q_{n-2}.$$

Функциональная схема преобразователя множителя представлена в приложении Г.

Оценка эффективности минимизации функций ПМ.

f	C_f	$C_{f_{min}}$	k
P	$3 \cdot 4 + 4 + 3 = 19$	0	∞
S_1	$3 \cdot 2 + 2 + 3 = 11$	$3 \cdot 2 + 2 + 3 = 11$	1
S_2	$3 \cdot 4 + 4 + 3 = 19$	2	9.5

4 СИНТЕЗ СХЕМЫ ОЧС НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ

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. Функции ОЧС зависят от пяти переменных. Удобно взять мультиплексор с тремя адресными входами, это позволит упростить одну нашу большую функцию от пяти аргументов до восьми функций от двух переменных. Функции от двух переменных достаточно просты для того, чтобы в самостоятельно заметить их минимальную форму. Таблица истинности ОЧС на базе мультиплексора приведена в таблице 4.1.

Таблица 4.1 – таблица истинности ОЧС на базе мультиплексора

№	a_1	a_2	b_1	b_2	p	П	Выход	S_1	Выход	S_2	Выход
1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	b_2	1	\bar{p}	0	$b_2\bar{p} + \bar{b}_2p$
2	0	0	0	0	1	0		0		1	
3	0	0	0	1	0	1		1		1	
4	0	0	0	1	1	1		0		0	
5	0	0	1	0	0	0	\bar{b}_2p	0	p	1	\bar{b}_2
6	0	0	1	0	1	1		1		1	
7	0	0	1	1	0	0		0		0	
8	0	0	1	1	1	0		1		0	
9	0	1	0	0	0	1	“1”	1	\bar{p}	1	$\bar{b}_2\bar{p} + b_2p$
10	0	1	0	0	1	1		0		0	
11	0	1	0	1	0	1		1		0	
12	0	1	0	1	1	1		0		1	
13	0	1	1	0	0	1	$p + \bar{b}_2$	0	p	0	b_2
14	0	1	1	0	1	1		1		0	
15	0	1	1	1	0	0		0		1	
16	0	1	1	1	1	1		1		1	
17	1	0	0	0	0	0	$p + b_2$	0	p	1	\bar{b}_2
18	1	0	0	0	1	1		1		1	
19	1	0	0	1	0	1		0		0	
20	1	0	0	1	1	1		1		0	
21	1	0	1	0	0	1	\bar{b}_2	1	\bar{p}	1	$\bar{b}_2\bar{p} + b_2p$
22	1	0	1	0	1	1		0		0	
23	1	0	1	1	0	0		1		0	
24	1	0	1	1	1	0		0		1	

Продолжение таблицы 4.1

25	1	1	0	0	0	0	b_2p	0	p	0	b_2
26	1	1	0	0	1	0		1		0	
27	1	1	0	1	0	0		0		1	
28	1	1	0	1	1	1		1		1	
29	1	1	1	0	0	0	“0”	1	\bar{p}	0	$b_2\bar{p} + \overline{b_2}p$
30	1	1	1	0	1	0		0		1	
31	1	1	1	1	0	0		1		1	
32	1	1	1	1	1	0		0		0	

Функциональная схема ОЧС на базе мультиплексоров приведена в приложении Д.

5 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ

5.1 Расчёт времени умножения на один разряд множителя

Время умножения на один разряд множителя является суммой временных затрат на нескольких различных этапах умножения. Для понимания процесса расчёта следует использовать структурную схему разрабатываемого устройства (см. приложение А).

Процесс умножения можно разбить на несколько этапов:

1. Преобразование разряда множителя
2. Формирование дополнительного кода множимого в ФДК, если это необходимо
3. Умножение дополнительного кода множимого в ОЧУС
4. Добавление полученного произведения в регистр результата посредством ОЧС

Заметим, процессы умножения в ОЧУС и сложения в ОЧС могут происходить параллельно. После того, как первый ОЧУС получит свой результат, он может быть сразу передан в блок ОЧС для обработки. В этот момент независимо друг от друга могут начать работу первый ОЧС и второй ОЧУС. После того, как отработает второй ОЧУС, он может незамедлительно передать свой результат во второй ОЧС, который сможет приступить к сложению только после того, как получит перенос из первого ОЧС. Таким образом, если ОЧС работает медленнее, чем ОЧУС, то именно ОЧС будет обуславливать задержку. Если же ОЧУС работает медленнее, то ОЧУС будет вызывать задержку. Выполнение блока ОЧС, который работает с $(n-1)$ старшими разрядами регистра результата, займёт всегда $(n-1)*t_{очс}$. Формула для расчёта времени имеет следующий вид:

$$T_{умн} = t_{пр} + t_{флк} + t_{очус} + \max(m*t_{очус}, (m+1)*t_{очс}) + (n-1)*t_{очс}$$

где $T_{умн}$ – общее время умножения на один разряд множителя, $t_{пр}$ – время преобразования разряда множителя, $t_{флк}$ – время формирования дополнительного кода, $t_{очус}$ и $t_{очс}$ – время работы ОЧУС и ОЧС.

Приведём краткое пояснение. После того, как получен результат с первого ОЧУС, он передается на первый ОЧС и второй ОЧУС. Для того, чтобы блок из $(n-1)$ старших ОЧС мог начать работать, должны отработать ещё $m+1$ ОЧС и m ОЧУС. Но так как они могут работать параллельно, мы берём максимальное время их работы.

5.2 Расчёт времени умножения на n разрядов множителя

В этом случае рассуждения аналогичны с теми, что приведены в предыдущем разделе. Итоговая формула имеет следующий вид:

$$T_{\text{умн}} = n \cdot (t_{\text{пр}} + t_{\text{флк}} + t_{\text{очус}} + \max(m \cdot t_{\text{очус}}, (m+1) \cdot t_{\text{очс}}) + (n-1) \cdot t_{\text{очс}})$$

где $T_{\text{умн}}$ – общее время умножения, $t_{\text{пр}}$ – время преобразования разряда множителя, $t_{\text{флк}}$ – время формирования дополнительного кода, $t_{\text{очус}}$ и $t_{\text{очс}}$ – время работы ОЧУС и ОЧС.

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

1. Искра Н. А. Арифметические и логические основы вычислительной техники: пособие / Н. А. Искра, И. В. Лукьянова, Ю. А. Луцик. – Минск: БГУИР, 2016. – 75 с.
2. Луцик Ю.А. Арифметические и логические основы вычислительной техники: Учеб. пособие / Ю. А. Луцик, И. В. Лукьянова. – Минск: БГУИР, 2014. – 165с.

ПРИЛОЖЕНИЕ А

(обязательное)

Сумматор-умножитель второго типа. Схема электрическая структурная

ПРИЛОЖЕНИЕ Б
(обязательное)

Одноразрядный четверичный умножитель-сумматор. Схема электрическая функциональная

ПРИЛОЖЕНИЕ В
(обязательное)

Одноразрядный четверичный сумматор. Схема электрическая
функциональная

ПРИЛОЖЕНИЕ Г
(обязательное)

Преобразователь множителя. Схема электрическая функциональная

ПРИЛОЖЕНИЕ Д
(обязательное)

Одноразрядный четверичный сумматор. Реализация на мультиплексорах.
Схема электрическая функциональная

ПРИЛОЖЕНИЕ Е
(обязательное)

Ведомость документов