

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические
основы цифровых устройств

К ЗАЩИТЕ ДОПУСТИТЬ

_____ Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-
УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 223 ПЗ

Студент

Д.А. Снитко

Руководитель

Ю. А. Луцик

МИНСК 2022

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические
основы вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Б. В. Никульшин
«___» _____ 2022 г.

ЗАДАНИЕ

по курсовой работе студента
Снитко Даниила Александровича

- 1 Тема работы: Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел.
- 2 Срок сдачи студентом законченной работы: 20 мая 2022 г.
- 3 Исходные данные к работе:
 - 3.1 исходные сомножители: $M_{N_{10}} = 82,48$; $M_{T_{10}} = 43,35$;
 - 3.2 алгоритм умножения: В;
 - 3.3 метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;
 - 3.4 коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования; $0_4 - 01$, $1_4 - 10$, $2_4 - 11$, $3_4 - 00$;
 - 3.5 тип синтезируемого умножителя: 1;
 - 3.6 тип синтезируемого умножителя: структурные схемы приведены для умножителя 1-ого типа (ОЧУ, ОЧС, аккумулятор);
 - 3.7 логический базис для реализации ОЧС: И, ИЛИ, НЕ; метод минимизации – алгоритм Рота;
 - 3.8 логический базис для реализации ОЧУ: НЕ; метод минимизации – карты Карно-Вейча.

- 4** Содержание пояснительной записки (перечень подлежащих разработке вопросов):
1. Разработка алгоритма умножения.
 2. Разработка структурной схемы сумматора-умножителя.
 3. Разработка функциональных схем основных узлов сумматора-умножителя.
 4. Синтез комбинационных схем устройств на основе мультиплексоров.
 5. Оценка результатов разработки.
- Заключение. Список литературы.
- 5** Перечень графического материала:
- 5.1** Сумматор-умножитель первого типа. Схема электрическая структурная.
 - 5.2** Одноразрядный четвертичный сумматор. Схема электрическая функциональная.
 - 5.3** Одноразрядный четвертичный умножитель. Схема электрическая функциональная.
 - 5.4** Регистр-аккумулятор. Схема электрическая функциональная.
 - 5.5** Одноразрядный четвертичных сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.

Наименование мероприятия	Срок исполнения
Выдача студентам заданий по курсовым работам.	с 09.02.2022 по 22.02.2022
1-я контрольная точка	<u>Процентровка</u> студентов: с 01.03.2022 по 10.03.2022
2-я контрольная точка	<u>Процентровка</u> студентов: с 01.04.2022 по 10.04.2022
3-я контрольная точка	<u>Процентровка</u> студентов: с 01.05.2022 по 10.05.2022
Представление студентами готовых курсовых работ руководителям для проверки	До 20.05.2022
Представление информации в деканат о сдаче студентами готовых курсовых работ руководителям для проверки	с 20.05.2022 по 01.06.2022
Защита студентами курсовых работ	с 25.05.2022 по 11.06.2022
Представление информации в деканат о защите студентами курсовых работ	До 11.06.2022

Дата выдачи задания: 10 февраля 2022 г.

Руководитель

Ю. А. Луцик

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ

СОДЕРЖАНИЕ

Введение	5
1. Разработка алгоритма умножения	6
2. Разработка структурной схемы сумматора-умножителя	9
3. Логический синтез одноразрядного четверичного сумматора	10
4. Логический синтез одноразрядного четверичного умножителя.....	21
5. Логический синтез одноразрядного четверичного сумматора на основе мультиплексора	25
6. Логический синтез преобразователя множителя	27
7. Оценка эффективности минимизации переключательных функций.....	28
8. Временные затраты на умножение	29
Заключение	30
Список использованных источников	31
Приложение А	32
Приложение Б	32
Приложение В	32
Приложение Г	32
Приложение Д	32

ВВЕДЕНИЕ

Данная курсовая работа посвящена разработке алгоритмов выполнения операций умножения и сложения. На основе полученных алгоритмов требуется разработать и синтезировать следующие устройства: одноразрядный четвертичный сумматор (ОЧС), одноразрядный четвертичный умножитель (ОЧУ), а также переключательные функции ОЧС на мультиплексорах. Минимизация перечисленных устройств осуществляется с помощью карт Карно-Вейча и алгоритма извлечения Рота. На основе полученных данных требуется построить схемы этих устройств и проанализировать результаты (эффективность минимизации и время выполнения операций).

1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

1.1. Перевод сомножителей из десятичной системы счисления в четверичную.

Множимое

$$\begin{array}{r} 82 \mid 4 \\ \underline{80} \quad 20 \mid 4 \\ 2 \quad 20 \quad 5 \mid 4 \\ \quad \quad 0 \quad 4 \quad 1 \\ \quad \quad \quad 1 \end{array}$$

$$\begin{array}{r} 0,48 \\ * \quad 4 \\ \hline 1,92 \\ * \quad 4 \\ \hline 3,68 \\ * \quad 4 \\ \hline 2,72 \end{array}$$

$M_{H4} = 1102,132$
в соответствии с заданной
кодировкой множимого
 $M_{H2/4} = 10100111,100011$

Множитель

$$\begin{array}{r} 43 \mid 4 \\ \underline{40} \quad 10 \mid 4 \\ 3 \quad 8 \quad 2 \\ \quad \quad 2 \end{array}$$

$$\begin{array}{r} 0,35 \\ * \quad 4 \\ \hline 1,40 \\ * \quad 4 \\ \hline 1,60 \\ * \quad 4 \\ \hline 2,40 \end{array}$$

$M_{T4} = 223,112$
 $M_{T2/4} = 101011,010110$
множитель представляется
обычным весомозначным
кодом: 0₄ - 00, 1₄ - 01, 2₄ - 10, 3₄ - 11
для всех вариантов

Запишем сомножители в форме с плавающей запятой в прямом коде:
 $M_H = 0,10100111100011$ $P_{MH} = 0.1001 + 04_{10}$ – закодировано по заданию
 $M_T = 0,101011010110$ $P_{MT} = 0.0011 + 03_{10}$ – закодировано традиционно

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

Порядок произведения будет равен:

$$\begin{array}{r} P_{MH} = 0.1001 + 10_4 \\ P_{MT} = 0.0011 + 03_4 \\ \hline P = 0.1000 + 13_4 \end{array}$$

Результат закодирован в соответствии с заданием на кодировку множимого.

Знак произведения определяется суммой по модулю “два” знаков сомножителей:

$$\text{зн } M_H \oplus \text{зн } M_T = 0 \oplus 0 = 0$$

Для умножения мантисс необходимо предварительно преобразовать множитель. Диада $11(3_4)$ заменяется на триаду $1\overline{0}\overline{1}$. Преобразованный множитель имеет вид: $M_T^{п_4} = 1\overline{1}\overline{1}\overline{1}112$ или $M_T^{п_2} = 010\overline{1}0\overline{1}0\overline{1}010110$.

Алгоритм “В”.

Таблица 1.1 - Перемножение мантисс

Четверичная с/с		Двоично-четверичная с/с		Комментарии
1		2		3
0.	00000000000000	01.	01010101010101010101010101	\sum_0^q
<u>0.</u>	<u>0000001102132</u>	<u>01.</u>	<u>010101010101110100111100011</u>	$\Pi_1^q = M_H \cdot b_1$
0.	0000001102132	01.	010101010101110100111100011	\sum_1^q
0.	0000011021320	01.	01010101011010011110001101	$\sum_1^q \cdot 4$
<u>3.</u>	<u>3333332231202</u>	<u>00.</u>	<u>00000000000011110010110011</u>	$\Pi_2^q = M_H \cdot b_2$
0.	0000003313122	01.	01010101010100001000101111	\sum_2^q
0.	0000033131220	01.	01010101010000100010111101	$\sum_2^q \cdot 4$
<u>3.</u>	<u>3333332231202</u>	<u>00.</u>	<u>00000000000011110010110111</u>	$\Pi_3^q = M_H \cdot b_3$
0.	0000032023022	01.	01010101010011011100011111	\sum_3^q
0.	0000320230220	01.	01010101001101110001111101	$\sum_3^q \cdot 4$
<u>3.</u>	<u>3333332231202</u>	<u>00.</u>	<u>00000000000011110010110111</u>	$\Pi_4^q = M_H \cdot b_4$
0.	0000313122022	01.	01010101001000101111001111	\sum_4^q
0.	0003131220220	01.	01010100100010111101111101	$\sum_4^q \cdot 4$
<u>0.</u>	<u>0000001102132</u>	<u>01.</u>	<u>010101010101110100111100011</u>	$\Pi_5^q = M_H \cdot b_5$
0.	0003132323012	01.	01010100100011001100011011	\sum_5^q
0.	0031323230120	01.	01010010001100110001101101	$\sum_5^q \cdot 4$
<u>0.</u>	<u>0000001102132</u>	<u>00.</u>	<u>010101010101110100111100011</u>	$\Pi_6^q = M_H \cdot b_6$
0.	0031330332312	01.	01010010000001000011001011	\sum_6^q
0.	0313303323120	01.	01001000000100001100101101	$\sum_6^q \cdot 4$
<u>0.</u>	<u>0000011021320</u>	01.	<u>010101010111010011110001101</u>	$\Pi_7^q = M_H \cdot b_7$
0.	0313321011100	01.	01001000001110011010100101	\sum_7^q

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение ($M_H \cdot M_{T_4} = 0,112012000021$, $P_{M_H \cdot M_T} = 7$) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

$$\begin{aligned} M_H \cdot M_{T_4} &= 313321,011100 & P_{M_H \cdot M_T} &= 0; \\ M_H \cdot M_{T_{10}} &= 3577,082 \end{aligned}$$

Результат прямого перемножения операндов дает следующее значение:

$$M_{H_{10}} \cdot M_{T_{10}} = 82,48 \cdot 43,35 = 3575,508.$$

Абсолютная погрешность:

$$\Delta = 3577,082 - 3575,508 = 1,574.$$

Относительная погрешность:

$$\delta = \frac{\Delta}{M_H \cdot M_T} = \frac{1,574}{3575,508} = 0,00044022 \quad (\delta = 0,044022\%)$$

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

2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ

Структурная схема сумматора-умножителя первого типа для алгоритма умножения «В» приведена в приложении А.

Структура первого типа строится на базе заданных узлов ОЧУ, ОЧС, формирователя дополнительного кода и регистра результата. Управление режимами работы схемы осуществляется внешним сигналом *Mul/sum*, который определяет вид текущей арифметической операции (умножение или суммирование).

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

Таблица 2.1 – Режимы работы формирователя дополнительного кода

Сигналы на входах ФДК		Результат на выходах ФДК
F ₁	F ₂	
0	0	Дополнительный код множимого
0	1	Дополнительный код слагаемого
1	0	Меняется знак множимого
1	1	Дополнительный код слагаемого

3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ

3.1. Логический синтез одноразрядного четверичного сумматора

Одноразрядный четверичный сумматор – это комбинационное устройство, имеющее 5 входов (2 разряда одного слагаемого, 2 разряда второго слагаемого и вход переноса) и 3 выхода. Принцип работы ОЧС представлен с помощью таблицы истинности (табл. 3.1.1).

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

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

Таблица 3.2.1 – Таблица истинности ОЧС

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

Продолжение таблицы 3.2.1

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

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

L	N
	00000
	00001
	00110
00101	00111
01010	01000
01011	01001
01100	01110
01101	01111
10010	10000
10011	10001
10100	10110
10101	10111
11010	11000
	11001
	11110
	11111

L	N
01100	
11010	
0x101	xx00x
0101x	xx11x
1001x	
1010x	

Поиск простых импликант - воспользуемся операцией умножения (*) над множествами C_0 , C_1 и т. д., пока в результате операции будут образовываться новые кубы большей размерности.

*Поиск простых импликант $c_0 * c_0$:*

$C_0 * C_0$	01100	11010	0x101	0101x	1001x	1010x	xx00x	xx11x
01100	-							
11010		-						
0x101	0110y		-					
0101x		y1010		-				
1001x		1y010			-			
1010x			y0101			-		
xx00x	01y00	110y0	0xy01	010yx	100yx	10y0x	-	
xx11x	011y0	11y10	0x1y1	01y1x	10y1x	101yx		-
A1	0110x 01x00 011x0	x1010 1x010 110x0 11x10	x0101 0xx01 0x1x1	010xx 01x1x	100xx 10x1x	10x0x 101xx	\emptyset	\emptyset

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

$Z0 = \{ \emptyset \}$ - Множество Z кубов, не участвовавших в образовании новых кубов, пустое.

$B1 = C_0 - Z_0 = \{ 01100; 11010; 0x101; 0101x; 1001x; 1010x; xx00x; xx11x \}$

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

*Поиск простых импликант $C1 * C1$:*

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

$A_2 = \{ 01x0x; 011xx; 01xx0; x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 01xx1; 0x1x1; 0xx01; 01x1x; 010xx; 10xxx \}$

$Z_1 = \{ \emptyset \}$ - Множество Z кубов, не участвовавших в образовании новых кубов, пустое.

$B_2 = C_1 - Z_1 = \{ 0110x; 01x00; 011x0; x1010; 1x010; 110x0; 11x10; x0101; 0xx01; 0x1x1; 010xx; 01x1x; 100xx; 10x1x; 10x0x; 101xx; xx00x; xx11x \}$

$C_2 = A_2 \cup B_2 = \{ 01x0x; 011xx; 01xx0; x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 01xx1; 0x1x1; 0xx01; 01x1x; 010xx; 10xxx; xx00x; xx11x \}$

*Поиск простых импликант $C_2 * C_2$:*

$C_2 * C_2$	01x0x	011xx	01xx0	x10x0	x1x10	1x0x0	1xx10	x0x01	x01x1	01xx1	10xxx
01x0x	-										
011xx		-									
01xx0			-								
x10x0				-							
x1x10					-						
1x0x0						-					
1xx10							-				
x0x01								-			
x01x1									-		
01xx1			01xxy							-	
0x1x1											-
0xx01											
01x1x	01xyx										
010xx		01yxx									
10xxx											
xx00x											
xx11x											
A3	01xxx	01xxx	01xxx	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

$$A_3 = \{ 01xxx \}$$

$$Z_2 = \{ x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 0x1x1; 0xx01; 10xxx; xx00x; xx11x \}$$

$$B_3 = C_2 - Z_2 = \{ 01x0x; 011xx; 01xx0; 01xx1; 01x1x; 010xx \}$$

$$C_3 = A_3 \cup B_3 = \{ 01xxx \}$$

*Поиск простых импликант $C_3 * C_3$:*

$C_3 * C_3$	01xxx
01xxx	-
A_4	\emptyset

$$A_4 = \{ \emptyset \}$$

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

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

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

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

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

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

Получили кубы, подозрительные на L-экстремальность.

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

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

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

$$Z' = Z - E = \{ x10x0; x1x10; 1x0x0; 1xx10; x0x01; x01x1; 0x1x1; 0xx01; xx00x; xx11x \}$$

Поиск непокрытых наборов:

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

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

Покрытие оставшихся кубов:

$Z' \cap L'$	00101	11010
x10x0	0yy0y ∅	11010
x1x10	0y1yy ∅	11010
1x0x0	y0y0y ∅	11010
1xx10	y01yy ∅	11010
x0x01	00101	1y0yy ∅
x01x1	00101	1yy1y ∅
0x1x1	00101	y1y1y ∅
0xx01	00101	y10yy ∅
xx00x	00y01 ∅	110y0 ∅
xx11x	001y1 ∅	11y10 ∅

$$F_{\min} S1 = \{ 10xxx; 01xxx; 1xx10; x01x1 \} =$$

$$= a_1 * \overline{a_2} + \overline{a_1} * a_2 + a_1 * b_2 * \overline{p} + \overline{a_2} * b_1 * p$$

Проверку минимизации функции S_I выполним при помощи карты Карно

$\begin{matrix} b_1 b_2 h \\ a_1 b_2 \end{matrix}$	000	001	011	010	110	111	101	100
00	x	x	0	0	x	x	1	0
01	x	x	1	1	x	x	1	1
11	x	x	0	1	x	x	0	0
10	x	x	1	1	x	x	1	1

Рисунок 3.1.2 – Минимизация функции S_I при помощи карты Карно.

$$S_{I \text{ СДНФ}} = a_1 * \overline{a_2} + \overline{a_1} * a_2 + a_1 * b_2 * \overline{p} + \overline{a_2} * b_1 * p$$

Минимизацию функций Π и S_2 проведем при помощи карт Карно.

Для функции Π :

$\begin{matrix} b_1 b_2 h \\ a_1 b_2 \end{matrix}$	000	001	011	010	110	111	101	100
00	*	*	1	0	*	*	1	1
01	*	*	0	0	*	*	0	0
11	*	*	0	0	*	*	1	0
10	*	*	0	0	*	*	1	0

Рисунок 3.1.3 – Минимизация функции Π при помощи карты Карно.

$$\Pi_{\text{СДНФ}} = a_1 * b_1 * p + \overline{a_1} * \overline{a_2} * b_1 + \overline{a_1} * \overline{a_2} * p$$

Для функции **S2**:

$b_1 b_2 h$ $a_1 b_2$	000	001	011	010	110	111	101	100
00	*	*	1	0	*	*	0	1
01	*	*	0	1	*	*	1	0
11	*	*	0	1	*	*	1	0
10	*	*	1	0	*	*	0	1

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

$$S_2 \text{ СДНФ} = a_2 * b_1 * p + \overline{a_2} * b_1 * \overline{p} + a_2 * b_2 * \overline{p} + \overline{a_2} * b_2 * p$$

Функциональная схема реализации ОЧС на мультиплексорах приведена в приложении В.

3.2. Логический синтез одноразрядного четверичного умножителя

Одноразрядный четверичный умножитель – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда из регистра Мн, 2 разряда из регистра Мт и управляющий вход h) и 4 двоичных выхода. Принцип работы ОЧУ описывается с помощью таблицы истинности (табл.3.2.1).

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

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

Таблица 3.2.1 - Таблица истинности ОЧУ

Мн		Мт		Упр.	Ст. разряды		Мл. разряды		Результат операции в четверичной с/с
X_1	X_2	y_1	y_2	h	P_1	P_2	P_3	P_4	
1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	1	0	1	3·0=00
0	0	0	0	1	0	0	0	0	Выход 03
0	0	0	1	0	0	1	0	0	3·1=03
0	0	0	1	1	0	0	0	0	Выход 03
0	0	1	0	0	1	0	1	1	3·2=12
0	0	1	0	1	0	0	0	0	Выход 03
0	0	1	1	0	х	х	х	х	3·3=21
0	0	1	1	1	х	х	х	х	Выход 03
0	1	0	0	0	0	1	0	1	0·0=00
0	1	0	0	1	0	0	0	1	Выход 00
0	1	0	1	0	0	1	0	1	0·0=00
0	1	0	1	1	0	0	0	1	Выход 00
0	1	1	0	0	0	1	0	1	0·1=00
0	1	1	0	1	0	0	0	1	Выход 00
0	1	1	1	0	х	х	х	х	0·3=00
0	1	1	1	1	х	х	х	х	Выход 00
1	0	0	0	0	0	1	0	1	1·0=00
1	0	0	0	1	0	0	1	0	Выход 01
1	0	0	1	0	0	1	1	0	1·1=01
1	0	0	1	1	0	0	1	0	Выход 01
1	0	1	0	0	0	1	1	0	1·1=01
1	0	1	0	1	0	0	1	0	Выход 01
1	0	1	1	0	х	х	х	х	1·3=03

Продолжение таблицы 3.2.1

1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	х	х	х	х	Выход 01
1	1	0	0	0	1	1	1	1	2·0=00
1	1	0	0	1	0	0	1	1	Выход 02
1	1	0	1	0	0	1	1	1	2·1=02
1	1	0	1	1	0	0	1	1	Выход 02
1	1	1	0	0	1	0	0	1	2·2=10
1	1	1	0	1	0	0	1	1	Выход 02
1	1	1	1	0	х	х	х	х	2·3=12
1	1	1	1	1	х	х	х	х	Выход 02

Минимизацию функций P_1 , P_2 , P_3 и P_4 проведем при помощи карт Карно-Вейча.

Для функции P_1 :

		x_2								
x_1		1	0	0	1	0	0	0	0	
		0	0	*	*	*	*	0	0	
		0	0	*	*	*	*	0	0	
		0	0	0	0	1	0	0	0	
		y_1								
		h				h				
		y_2								

Рисунок 3.2.2 – Минимизация функции P_1 при помощи карты Вейча.

$$P_{1 \text{ СДНФ}} = \overline{x_1} * \overline{x_2} * y_1 * \overline{h} + x_1 * x_2 * \overline{y_2} * \overline{h} = \overline{x_1 + x_2 + y_1 + h} + \overline{x_1 + x_2 + y_2 + h}$$

Для функции P_2 :

		x_2							
x_1		1	0	0	0	1	0	0	1
		1	0	*	*	*	*	0	1
		1	0	*	*	*	*	0	1
		1	0	0	1	0	0	0	1
		y_1							
		h				h			

Рисунок 3.2.3 – Минимизация функции P_2 при помощи карты Вейча.

$$P_{2 \text{ СДНФ}} = \overline{y_1} * \overline{h} + x_1 * \overline{x_2} * \overline{h} + \overline{x_1} * x_2 * \overline{h} = \overline{y_1 + h} + \overline{x_1 + x_2 + h} + x_1 + x_2 + h$$

Для функции P_3 :

$y_1 y_2 h$									
$x_1 x_2$		000	001	011	010	110	111	101	100
	00	0	0	0	0	x	x	0	1
	01	0	0	0	0	x	x	0	0
	11	1	1	1	1	x	x	1	0
	10	0	1	1	1	x	x	1	1

Рисунок 3.2.4 – Минимизация функции P_3 при помощи карты Карно.

$$P_{3 \text{ СДНФ}} = x_1 * y_2 + x_1 * h + x_1 * \overline{x_2} * y_1 + \overline{x_2} * y_1 * \overline{h} + x_1 * x_2 * \overline{y_1} = \overline{x_1 + y_2} + \overline{x_1 + h} + \overline{x_1 + x_2 + y_1} + \overline{x_2 + y_1 + h} + \overline{x_1 + x_2 + y_1}$$

Для функции P_4 :

$y_1 y_2 h$ $x_1 x_2$	000	001	011	010	110	111	101	100
00	1	0	0	0	x	x	0	1
01	1	1	1	1	x	x	1	1
11	1	1	1	1	x	x	1	1
10	1	0	0	0	x	x	0	0

Рисунок 3.2.5 – Минимизация функции P_4 при помощи карты Карно.

$$P_{4 \text{ СДНФ}} = x_2 + \overline{x_1} * y_1 * \overline{h} + \overline{y_1} * \overline{y_2} * \overline{h} = x_2 + \overline{x_1 + y_1 + h} + \overline{y_1 + y_2 + h}$$

Функциональная схема реализации ОЧУ на мультиплексорах приведена в приложении Б.

4. ЛОГИЧЕСКИЙ СИНТЕЗ ОДНОРАЗРЯДНОГО ЧЕТВЕРИЧНОГО СУММАТОРА НА ОСНОВЕ МУЛЬТИПЛЕКСОРА

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

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

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

Таблица 4.1. – Таблица истинности для ОЧС на мультиплексорах

N_2	a_1	a_2	b_1	b_2	p	П	Вы- ход	s_1	Вы- ход	s_2	Вы- ход
1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	X	p	X	0	X	p
1	0	0	0	0	1	X		X		X	
2	0	0	0	1	0	0		0		0	
3	0	0	0	1	1	1		0		1	
4	0	0	1	0	0	1	1	0	p	1	!p
5	0	0	1	0	1	1		1		0	
6	0	0	1	1	0	X		X		X	
7	0	0	1	1	1	X		X		X	
8	0	1	0	0	0	X	p	X	1	X	!p

9	0	1	0	0	1	X		X		X	
10	0	1	0	1	0	0		1		1	
11	0	1	0	1	1	0		1		0	
12	0	1	1	0	0	0	0	1	1	0	p
13	0	1	1	0	1	0		1		1	
14	0	1	1	1	0	X		X		X	
15	0	1	1	1	1	X		X		X	
16	1	0	0	0	0	X	0	X	1	X	p
17	1	0	0	0	1	X		X		X	
18	1	0	0	1	0	0		1		0	
19	1	0	0	1	1	0		1		1	
20	1	0	1	0	0	0	p	1	1	1	!p
21	1	0	1	0	1	1		1		0	
22	1	0	1	1	0	X		X		X	
23	1	0	1	1	1	X		X		X	
24	1	1	0	0	0	X	0	X	!p	X	!p
25	1	1	0	0	1	X		X		X	
26	1	1	0	1	0	0		1		1	
27	1	1	0	1	1	0		0		0	
28	1	1	1	0	0	0	p	0	0	0	p
29	1	1	1	0	1	1		0		1	
30	1	1	1	1	0	X		X		X	
31	1	1	1	1	1	X		X		X	

Функциональная схема реализации ОЧС на мультиплексорах приведена в приложении Г.

5 ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ (ПМ)

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

Таблица 5.1 Таблица истинности ПМ.

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

Проведём минимизацию P при помощи карты Карно:

		$Q_{n-1}Q_{n-2}$			
Q_n		00	01	11	10
	0				
	1	1	1	1	1

Рисунок 5.1 – Минимизация функции P при помощи карты Карно

$$P = Q_n$$

Видно, что S_1 не минимизируется, поэтому $S_1 = \bar{Q}_n \bar{Q}_{n-1} Q_{n-2} + Q_n \bar{Q}_{n-1} \bar{Q}_{n-2}$

Проведём минимизацию S_2 при помощи карты Карно:

		$Q_{n-1}Q_{n-2}$			
Q_n		00	01	11	10
	0		1		1
	1		1		1

Рисунок 5.2 – Минимизация функции S_2 при помощи карты Карно

$$S_2 = \bar{Q}_{n-1} Q_{n-2} \cdot Q_{n-1} \bar{Q}_{n-2} = Q_{n-1} \oplus Q_{n-2}$$

6 ОЦЕНКА ЭФФЕКТИВНОСТИ МИНИМИЗАЦИИ ПЕРЕКЛЮЧАТЕЛЬНЫХ ФУНКЦИЙ

Для проведения оценки эффективности минимизации переключательных функций необходимо посчитать цену схемы до минимизации и цену схемы после минимизации. Эффективность минимизации k определяется как:

$$k = \frac{C_{\text{до мин.}}}{C_{\text{после мин.}}}$$

Таблица 6.1 – Эффективность минимизации ОЧУ

Вых. схемы	Рассчитанная цена схемы		Эфф. мин. k
	До минимизации	После минимизации	
P ₁	c=3*5+3+5=23	c=2*4+4+2=14	1,64
P ₂	c=10*5+10+5=65	c=2*1+3*2+4+3=15	4,33
P ₃	c=11*5+11+5=71	c=2*2+3*3+3+5=21	3,38
P ₄	c=15*5+15+5=95	c=1+2*3+4+3=14	6,79

Таблица 6.2 – Эффективность минимизации ОЧС

Вых. схемы	Рассчитанная цена схемы		Эфф. мин. k
	До минимизации	После минимизации	
П	c=5*5+5+5=35	c=3*3+2+3=14	2,5
S ₁	c=10*5+5+10=65	c=2*2+2*3+4+3=17	3,82
S ₂	c=8*5+5+8=53	c=4*3+2+4=18	2,94

7 ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ

Формула расчёта временных затрат на умножение:

$$T = n * (T_{\text{пм}} + T_{\text{фдк}} + T_{\text{очу}} + (n + 1) * T_{\text{очс}} + T_{\text{сдвига}}), \text{ где}$$

$T_{\text{пм}}$ – время преобразования множителя;

$T_{\text{фдк}}$ – время формирования дополнительного кода множимого;

$T_{\text{очу}}$ – время умножения на ОЧУ;

$T_{\text{очс}}$ – время формирования единицы переноса в ОЧС;

$T_{\text{сдвига}}$ – время сдвига частичной суммы;

n – количество разрядов множителя.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы была разработана структурная схема сумматора-умножителя второго типа, а также функциональные схемы основных узлов данного устройства. Для уменьшения стоимости логических схем были выполнены минимизации переключательных функций различными способами. Такой подход позволил выявить достоинства и недостатки этих алгоритмов.

В качестве главного достоинства минимизации картами Карно-Вейча можно выделить простоту и минимальные затраты времени. Однако применение данного способа для функций многих переменных будет затруднительно. Функциональные схемы были построены в различных логических базисах. Это позволило закрепить теоретические знания основных законов булевой алгебры, например, правило де Моргана. Также можно отметить, что необходимо сократить количество уровней в логической схеме для уменьшения времени работы данного устройства.

Реализация переключательных функций на основе мультиплексоров позволила облегчить процесс минимизации этих функций.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Луцик Ю.А., Лукьянова И.В. – Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". – Минск: БГУИР, 2014 г.
2. Луцик Ю.А., Лукьянова И.В. – Методические указания к курсовому проекту по курсу "Арифметические и логические основы вычислительной техники". – Минск.: БГУИР, 2016 г.

ПРИЛОЖЕНИЕ А

(обязательное)

Сумматор-умножитель первого типа. Схема электрическая структурная

ПРИЛОЖЕНИЕ Б

(обязательное)

Одноразрядный четверичный умножитель.
Схема электрическая функциональная

ПРИЛОЖЕНИЕ В

(обязательное)

Одноразрядный четверичный сумматор. Схема электрическая
функциональная

ПРИЛОЖЕНИЕ Г

(обязательное)

Одноразрядный четверичный сумматор.
Реализация на мультиплексорах.
Схема электрическая функциональная

ПРИЛОЖЕНИЕ Д

(обязательное)

Преобразователь множителя. Схема электрическая функциональная

ПРИЛОЖЕНИЕ Е

(обязательное)

Ведомость документов