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 01 323 ПЗ

Студент И.И. Федорович

Руководитель И.В. Лукьянова

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

«\_\_\_\_» \_\_\_\_\_\_\_\_\_2021г.

ЗАДАНИЕ

по курсовой работе студента

Федоровича Ильи Игоревича

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел»
2. Срок сдачи студентом законченной работы: 20 мая;
3. Исходные данные к работе:

**3.1** Исходные сомножители: Мн=23,58; Мт=97,47;

* 1. 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: 04 – 11, 14 – 00, 24 – 10, 34 – 01;

**3.5** Тип синтезируемого умножителя: 2;

**3.6** Логический базис для синтеза ОЧС: ИЛИ, СУММА ПО МОДУЛЮ 2, КОНСТАНТА 1; метод минимизации – алгоритм Рота;

**3.7** Логический базис для синтеза ОЧУС: НЕ, И, ИЛИ; метод минимизации – карты Карно-Вейча;

1. 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. Реализация на

мультиплексорах. Схема электрическая функциональная.

**5.5** Одноразрядный четверичный сумматор. Реализация на

мультиплексорах. Схема электрическая функциональная.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объем этапа, % | Срок выполнения этапа | Примечания |
| Разработка алгоритма умножения | 10 | 9.02-20.02 |  |
| Разработка структурной схемы сумматора-умножителя | 10 | 21.02-09.03 | С выполнением чертежа |
| Разработка функциональных схем основных узлов сумматора-умножителя | 50 | 10.03-30.04 | С выполнением чертежа |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 | 1.05-15.05 | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 | 15.05-20.05 |  |

КАЛЕНДАРНЫЙ ПЛАН

Дата выдачи задания: 11 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 ………………….10

3. Разработка функциональных схем основных узлов сумматора-умножителя…………………………………………………………………....11

3.1. Логический синтез одноразрядного четвертичного сумматора….. 11

3.2. Логический синтез одноразрядного четверичного умножителя-сумматора………………………………………………………………….……. 21

4. Логический синтез одноразрядного четвертичного сумматора на

основе мультиплексора …………………..………………………………… 26

5. Логический синтез преобразователя множителя ………………………. 28

6. Временные затраты на умножение………………………………………..... 30

Заключение ………………………………………………………………….. 31

Список использованных источников ...………………………………….… 32

Приложение А ………………………………………………………………. 33

Приложение Б ………………………………………………………………..34

Приложение В ………………………………………………………………..35

Приложение Г ………………………………………………………………. 36

Приложение Д ………………………………………………………………. 37

Приложение Е ………………………………………………………………. 38

**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 цифровых устройств”, 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**

Исходные данные:

-исходные сомножители: Мн = 23,58; Мт =97,47;

-алгоритм умножения: В;

-метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;

-кодирование четверичных цифр множимого для перехода к двоично- четверичной системе кодирования: 04 – 11, 14 – 00, 24 – 10, 34 – 01;

-тип синтезируемого умножителя: 2-й;

1.1. Перевод сомножителей из десятичной системы счисления в четверичную.

## Множимое

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \_ | 23 | 4 |  |  | 0,58 |
|  | 20 | \_5 | 4 | \* | 4 |
|  | 3 | 4 | 1 |  | 2,32 |
|  |  | 1 |  | \* | 4 |
|  |  |  |  |  | 1,28 |
|  |  |  |  | \* | 4 |
|  |  |  |  |  | 1,12 |

Мн4 = 113,211.

В соответствии с заданной кодировкой множимого: Мн2/4 = 000001,100000.

## Множитель

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \_ | 97 | 4 |  |  |  | 0,47 |
|  | 96 | \_ 24 | 4 |  | \* | 4 |
|  | 1 | 24 | \_6 | 4 |  | 1,88 |
|  |  | 0 | 4 | 1 | \* | 4 |
|  |  |  | 2 |  |  | 3,52 |

Мт4 = 1201,13.

В соответствии с обычной весомозначной кодировкой множителя:

Мт2/4 = 01100001,0111.

Запишем сомножители в форме с плавающей запятой в прямом коде:

|  |  |  |
| --- | --- | --- |
| Мн = 0,113211 | РМн = 0.0011 +0310 | – закодировано по заданию, |
| Мт = 0,120113 | РМт = 0.0100 +0410 | – закодировано традиционно. |

Порядок произведения будет следующим:

|  |
| --- |
| Pмн=+0310 |
| Pмт=+0410 |
| P = +0710 |
|  |

Результат закодирован в соответствии с заданием на кодировку множимого. Знак произведения определяется суммой по модулю двух знаков сомножителей, т.е.:

зн Мн ⊕ зн Мт = 0 ⊕ 0 = 0.

Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в дополнительных кодах диада 10(24) заменяется на диаду 1. Преобразованный множитель имеет вид МтП4 =0,2012. Перемножение мантисс по алгоритму “B” приведено в таблице 1.1

Таблица 1.1 – Перемножение мантисс

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Четверичная с/с | | | Двоично четверичная с/с | | | Комментарии |
| 0,  0, | 000000000000  000000233022 | 0,  0, | | 11 11 11 11 11 11 11 11 11 11 11 11  11 11 11 11 11 11 10 01 01 11 10 10 | ∑0  ∏1=[2Мн]д | |
| 0,  0,  3, | 000000233022  000002330220  333333100312 | 0,  0,  3, | | 11 11 11 11 11 11 10 01 01 11 10 10  11 11 11 11 11 10 01 01 11 10 10 11  01 01 01 01 01 01 00 11 11 01 00 10 | ∑1  ∑1·41  ∏2=[-2Мн]д | |
| 0,  0,  0, | 000002031132  000020311320  000000000000 | 0,  0,  0, | | 11 11 11 11 11 10 11 01 00 00 01 10  11 11 11 11 10 11 01 00 00 01 10 11  11 11 11 11 11 11 11 11 11 11 11 11 | ∑2  ∑2·41  ∏3=0 | |
| 0,  0,  0, | 000020311320  000203113200  000000113211 | 0,  0,  0, | | 11 11 11 11 10 11 01 00 00 01 10 11  11 11 11 10 11 01 00 00 01 10 11 11  11 11 11 11 11 11 00 00 01 10 00 00 | ∑3  ∑3·41  ∏4=[Мн]д | |
| 0,  0,  0, | 000203233011  002032330110  000000233022 | 0,  0,  0, | | 11 11 11 10 11 01 10 01 01 11 00 00  11 11 10 11 01 10 01 01 11 00 00 11  11 11 11 11 11 11 10 01 01 11 10 10 | ∑4  ∑4­·41  ∏5=[2Мн]д | |
| 0,  0,  3, | 002033223132  020332231320  333333220130 | 0,  0,  3, | | 11 11 10 11 01 01 10 10 01 00 01 10  11 10 11 01 01 10 10 01 00 01 10 11  01 01 01 01 01 01 10 10 11 00 01 11 | ∑5  ∑5·41  ∏6=[-Мн]д | |
| 0, | 020332032110 | 0, | | 11 10 11 01 01 10 11 01 10 00 00 11 | ∑6 | |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (Мн·Мт4= 0,020332032110, РМн**·**Мт = 7) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн · Мт4= 203320,31110 РМн **·** Мт = 0;

Мн · Мт10= 2296,8945.

Результат прямого перемножения операндов даёт следующее значение:

Мн10 · Мт10 = 23,58 · 97,47 = 2298,3426.

Абсолютная погрешность:

Δ = =1,4481.

Относительная погрешность:

δ= ==0,00063

δ=0,063%

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**

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

Структурная схема второго типа строится на базе заданных узлов ОЧУС, 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).

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».

Если устройство работает как умножитель, то множимое и множитель

помещаются в соответствующие регистры, а на управляющий вход ФДК F2 поступает «0».

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 |

**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).

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

Поскольку ОЧС синтезируется для схемы 2-го типа, то безразличные наборы в таблице истинности отсутствуют.

Таблица 3.1 – Таблица истинности ОЧС

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **p** | **П** |  |  | **Пример 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 |

Продолжение таблицы 3.1

**Минимизация функции П**

Проведем минимизацию с помощью алгоритма Рота. Определим множество единичных кубов:

L=

N={}

C0=

Первым этапом алгоритма Рота является нахождение множества простых импликант.

Для реализации этого этапа будем использовать операцию умножения (\*) над множествами С0, С1 и т. д., пока в результате операции будут образовываться новые кубы большей размерности.

Первый шаг умножения (С0\*С0) приведён в таблице 3.2

В результате этой операции и последующего поглощения сформируется новое множество кубов:

C1=

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С0\*С0 | 00010 | 00010 | 00101 | 01000 | 01001 | 01010 | 01011 | 01100 | 01101 | 01111 | 10001 | 10010 | 10011 | 10100 | 10101 | 11011 |
| 00010 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00011 | 0001y | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 00101 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01000 |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 01001 |  |  |  | 0100y | - |  |  |  |  |  |  |  |  |  |  |  |
| 01010 | 0y010 |  |  | 010y0 |  | - |  |  |  |  |  |  |  |  |  |  |
| 01011 |  | 0y011 |  |  | 010y1 | 0101y | - |  |  |  |  |  |  |  |  |  |
| 01100 |  |  |  | 01y00 |  |  |  | - |  |  |  |  |  |  |  |  |
| 01101 |  |  | 0y101 |  | 01y01 |  |  | 0110y | - |  |  |  |  |  |  |  |
| 01111 |  |  |  |  |  |  | 01y11 |  | 011y1 | - |  |  |  |  |  |  |
| 10001 |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 10010 | y0010 |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 10011 |  | y0011 |  |  |  |  |  |  |  |  | 100y1 | 1001y | - |  |  |  |
| 10100 |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |
| 10101 |  |  | y0101 |  |  |  |  |  |  |  | 10y01 |  |  | 1010y | - |  |
| 11011 |  |  |  |  |  |  | y1011 |  |  |  |  |  | 1y011 |  |  | - |
| А1 | 0001x  0x010  x0010 | 0x011  x0011 | 0x101  X0101 | 0100x  010x0  01x00 | 010x1  01x01 | 0101x | 01x11  X1011 | 0110x | 011x1 | Ø | 100x1  10x01 | 1001x | 1x011 | 1010x | Ø | Ø |

Таблица 3.2 - Поиск простых импликант С0\*С0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С1\*С1  Таблица 3.3-Поиск простых импликант С1\*С1 | 0001x | 0x010 | x0010 | 0x011 | X0011 | 0x101 | X0101 | 0100x | 010x0 | 01x00 | 010x1 | 01x01 | 0101x | 01x11 | X1011 | 0110x | 011x1 | 100x1 | 10x01 | 1001x | 1x011 | 1010x |
| 0001x | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x010 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| X0010 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x011 |  | 0x01y |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| X0011 |  |  | x001y |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x101 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| X0101 |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0100x |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 010x0 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01x00 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 010x1 |  |  |  |  |  |  |  |  | 010xy |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 01x01 |  |  |  |  |  |  |  |  |  | 01x0y |  | - |  |  |  |  |  |  |  |  |  |  |
| 0101x | 0y01x |  |  |  |  |  |  | 010yx |  |  |  |  | - |  |  |  |  |  |  |  |  |  |
| 01x11 |  |  |  |  |  |  |  |  |  |  |  | 01xy1 |  | - |  |  |  |  |  |  |  |  |
| X1011 |  |  |  |  | xy011 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |  |
| 0110x |  |  |  |  |  |  |  | 01y0x |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 011x1 |  |  |  |  |  |  |  |  |  |  | 01yx1 |  |  |  |  |  | - |  |  |  |  |  |
| 100x1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 10x01 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |  |
| 1001x | y001x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |  |
| 1x011 |  |  |  | yx011 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |  |
| 1010x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - |
| А2 | 0x01x  x001x | 0x01x | x001x | xx011 | xx011 | Ø | Ø | 010xx  01x0x | 010xx | 01x0x | 01xx1 | 01xx1 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

В результате образовалось множество С2 кубов второй размерности:

C2={0x01x,x001x,xx011,010xx,01x0x,01xx}

Множество Z1 кубов, не участвовавших в образовании новых кубов имеет вид:

Z1={0x101,x0101,100x1,10x01,1010x}

В таблице 3.4 приведён следующий шаг поиска простых импликант с помощью операции С2\*С2.

Таблица 3.4 – Поиск простых импликант С2\*С2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| С2\*С2 | 0x01x | x001x | xx011 | 010xx | 01x0x | 01xx1 |
| 0x01x | - |  |  |  |  |  |
| x001x |  | - |  |  |  |  |
| xx011 |  |  | - |  |  |  |
| 010xx |  |  |  | - |  |  |
| 01x0x |  |  |  |  | - |  |
| 01xx1 |  |  |  |  |  | - |
| А3 | Ø | Ø | Ø | Ø | Ø | Ø |

Новых кубов (третей размерности) не образовалось.

Получено множество Z2 = {0x01x,x001x,xx011,010xx,01x0x,01xx1}.

На этом заканчивается этап поиска простых импликант, т. к. |С3| ≤ 1.

Множество простых импликант:

Z=={

Следующий этап – поиск L-экстремалей на множестве простых импликант (таблица 3.5). Для этого используется операция # (решётчатое вычитание).

Таблица 3.5 – Поиск L-экстремалей

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-z) | 0x101 | x0101 | 100x1 | 10x01 | 1010x | 0x01x | x001x | xx011 | 010xx | 01x0x | 01xx1 |
| 0x101 | - | 1zzzz  10101 | yzy1z  100x1 | yz0zz  10x01 | yzzz0  1010x | zzyy0  0x01x | 1zyy0  x001x | 1zyyz  xx011 | zzy10  010xx | zz0z0  0100x  01x00 | zz01z  010x1  01x11 |
| x0101 | z1zzz  01101 | - | zzy1z  100x1 | zz0zz  10001 | zzzz0  10100 | z1yy0  0x01x | zzyy0  x001x | z1yyz  xx011 | zzy10  010xx | zyyz0 zy0z0  0100x 01x00 | zyz1z zy01z  01x11 010x1 |
| 100x1 | yyy1z  01101 | z0zyzz  10101 | - | zzzzz  Ø | zzyzy  10100 | y1zzz  0x01x | 0zzz0  0001x  x0010 | 01zzz  0x011  x1011 | yyzzz  010xx | yy1z0 yy1z0  0100x 01x00 | yy1zz yyzzz  01x11 010x1 |
| 10x01 | yyzzz  01101 | zzzzz  Ø | zzz1z  10011 | - | zzzzy  10100 | y1zyz  0x01x | yzzy0 0zzyy  0001x x0010 | x1zyz 0yzyz  0x011 x1011 | yyz1z  010xx | yyzz0 yyzz0  0100x 01x00 | yyzyz yy11z  01x11 010x1 |
| 1010x | yyzzz  01101 | Ø | zzyyz  10011 | Ø | - | y1zyz  0x01x | yzyyz 0zyyz  0001x x0010 | y1yyz 0yyyz  0x011 x1011 | yyy1z  010xx | yy0zz yy0zz  0100x 01x00 | yy0yz yyy1z  01x11 010x1 |
| 0x01x | zzyyz  01101 | Ø | yzzzz  10011 | Ø | yzyyz  10100 | - | zzzzz 1zzzz  Ø 10010 | zzzzz 1zzzz  Ø 11011 | zzz0z  0100x | zzzyz zzyyz  0100x 01x00 | zz1zz zzz0z  01111 01001 |
| x001x | zyyyz  01101 | Ø | zzzzz  Ø | Ø | yzyyz  10100 | y1zzz  0101x | - | zyzzz  11011 | zyzyz  0100x | zyzyz zy1yz  0100x 01x00 | zyyzz zyzyz  01111 01001 |
| xx011 | zzyyz  01101 | Ø | Ø | Ø | zzyyy  10100 | zzzz1  01011 | zzzzy  10010 | - | zzzyz  0100x | zzzyz zz1yy  0100x 01x00 | zzyzz zzzyz  01111 01001 |
| 010xx | zzyzz  01101 | Ø | Ø | Ø | yyyzz  10100 | zzzzz  Ø | yyzzz  10010 | yzzzz  11011 | - | zzzzz zz1zz  Ø 01100 | zzyzz zzzzz  01111 Ø |
| 01x0x | zzzzz  Ø | Ø | Ø | Ø | yyzzz  10100 | Ø | yyzyz  10010 | yzzyz  11011 | zzzzz  Ø | - | Zzzyz  01111 |
| 01xx1 | Ø | Ø | Ø | Ø | yyzzy  10100 | Ø | yyzzy  10010 | yzzzz  11011 | Ø | zzzzy  01100 | - |
| Остаток | Ø | Ø | Ø | Ø | 10100 | Ø | 10010 | 11011 | Ø | 01100 | 01111 |

В таблице 3.5 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 = {1010𝑥, 𝑥001𝑥, 𝑥𝑥011, 01𝑥0𝑥, 01𝑥𝑥1}.

После выявления L-экстремалей следует выяснить, не являются ли некоторые из них простыми импликантами, остатки которых покрывают только некоторое подмножество кубов комплекса N, которое нет необходимости покрывать, вводя в минимальное покрытие соответствующие наборы. Для этого необходимо выполнить операцию пересечения остатков, полученных при выполнении операции z#(Z− z) с кубами из комплекса L. Во множестве E необходимо оставить только те кубы, остатки от которых пересекаются с кубами из комплекса L.

Таблица 3.6 – Проверка L-экстремалей

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z-  z) ⋂ L | 00010 | 00011 | 00101 | 01000 | 01001 | 01010 | 01011 | 01100 | 01101 | 01111 | 10001 | 10010 | 10011 | 10100 | 10101 | 11011 |
| 10100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10100 | Ø | Ø |
| 10010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10010 | Ø | Ø | Ø | Ø |
| 11011 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11011 |
| 01100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 01100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 01111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 01111 | Ø | Ø | Ø | Ø | Ø | Ø |

По результатам таблицы 3.6 все L-экстремали оказались не связанными с безразличными наборами, а значит все они обязательно должны войти в минимальное покрытие.

Далее выясним, какие из вершин комплекса L не покрываются L-экстремалями. Для этого из каждого куба комплекса L вычтем (#) элементы множества Е (таблица 3.7). В результате вычитания получим L1 = L#Е.

Таблица 3.7 – Поиск непокрытых исходных наборов

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00010 | 00011 | 00101 | 01000 | 01001 | 01010 | 01011 | 01100 | 01101 | 01111 | 10001 | 10010 | 10011 | 10100 | 10101 | 11011 |
| 1010x | 00010 | 00011 | 00101 | 01000 | 01001 | 01010 | 01011 | 01100 | 01101 | 01111 | 10001 | 10010 | 10011 | Ø | Ø | 11011 |
| x001x | Ø | Ø | 00101 | 01000 | 01001 | 01010 | 01011 | 01100 | 01101 | 01111 | 10001 | Ø | Ø | Ø | Ø | 11011 |
| 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 |

Из таблицы 3.7 видно, что L-экстремалью не покрыты три единичных куба:

L1 = {00101, 01010, 10001}.

Чтобы их покрыть, воспользуемся множеством покрытых импликант, не являющиеся L-экстремалями (таблица 3.8).

Таблица 3.8–Покрытие оставшихся кубов

|  |  |  |  |
| --- | --- | --- | --- |
| Z’L1 | 00101 | 01010 | 10001 |
| 0x101 | 00101 | 01yyy  Ø | y0y01  Ø |
| x0101 | 00101 | 0yyyy  Ø | 10y01  Ø |
| 100x1 | y0y01  Ø | yy01y  Ø | 10001 |
| 10x01 | y0101  Ø | yy0yy  Ø | 10001 |
| 0x01x | 00yy1  Ø | 01010 | y00y1  Ø |
| 010xx | 0yy01  Ø | 01010 | yy001  Ø |

Из таблицы 3.8 видно, что каждый из непокрытых единичных кубов может быть покрыт двумя равнозначными способами.

Следовательно, существуют восемь тупиковых (минимальных) форм:

Fmin1 = {0x101, 100x1, 0x01x, 1010x, x001x, xx011, 01x0x, 01xx1}, Fmin2 = {x0101, 100x1, 0x01x, 1010x, x001x, xx011, 01x0x, 01xx1}, Fmin3 = {0x101, 10x01, 0x01x, 1010x, x001x, xx011, 01x0x, 01xx1}, Fmin4 = {x0101, 10x01, 0x01x, 1010x, x001x, xx011, 01x0x, 01xx1}, Fmin5 = {0x101, 100x1, 010xx, 1010x, x001x, xx011, 01x0x, 01xx1}, Fmin6 = {x0101, 100x1, 010xx, 1010x, x001x, xx011, 01x0x, 01xx1}, Fmin7 = {0x101, 10x01, 010xx, 1010x, x001x, xx011, 01x0x, 01xx1}, Fmin8 = {x0101, 10x01, 010xx, 1010x, x001x, xx011, 01x0x, 01xx1}.

##### FminДнф ={x0101; 100x1; 0x01x; 1010x; x001x; xx011; 01x0x; 01xx1}

П=2b12p+a121p+11b2+a12b12+21b2+1b2p+1а22+1а2p

Запишем результат в данном базисе:

П=(a2+(b1⊕1)+b2+(p⊕1)) ⊕1+((a1⊕1)+a2+b1+(p⊕1)) ⊕1+

+(a1+b1+(b2⊕1)) ⊕1+((a1⊕1)+a2+(b1⊕1)+b2) ⊕1+

+(a2+b1+(b2⊕1)) ⊕1+(b1+(b2⊕1)+(p⊕1)) ⊕1+

+(a1+(a2⊕1)+b2) ⊕1+(a1+(a2⊕1)+(p⊕1)) ⊕1

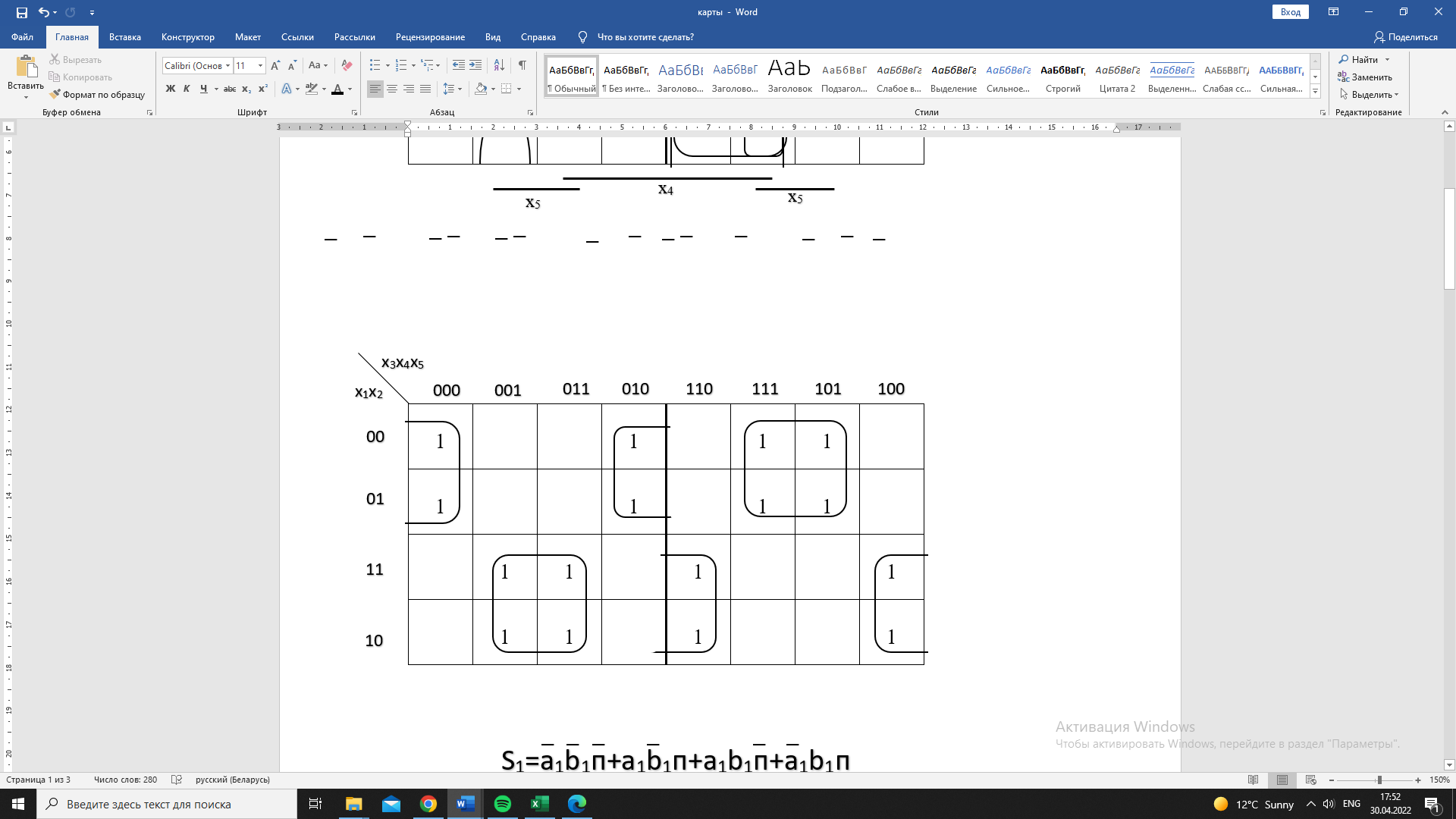
Эффективность минимизации можно оценить отношением числа входов схем, реализующих переключательную функцию до и после минимизации:

КП==2,525

**Минимизация функции S1**

Проведем минимизацию с помощью карты Карно. Для функции S1 заполненная карта приведена на рисунке 3.1.

b1b2p



a1a2

Рисунок3.1 Минимизация S1 при помощи карты Карно

S1=11+a11p+a1b1+1b1p

Запишем результат в данном базисе:

S1=(a1+b1+p) ⊕1+((a1⊕1)+b1+(p⊕1)) ⊕1+((a1⊕1)+(b1⊕1)+p)+ ⊕1+(a1+(b1⊕1)+(p⊕1)) ⊕1

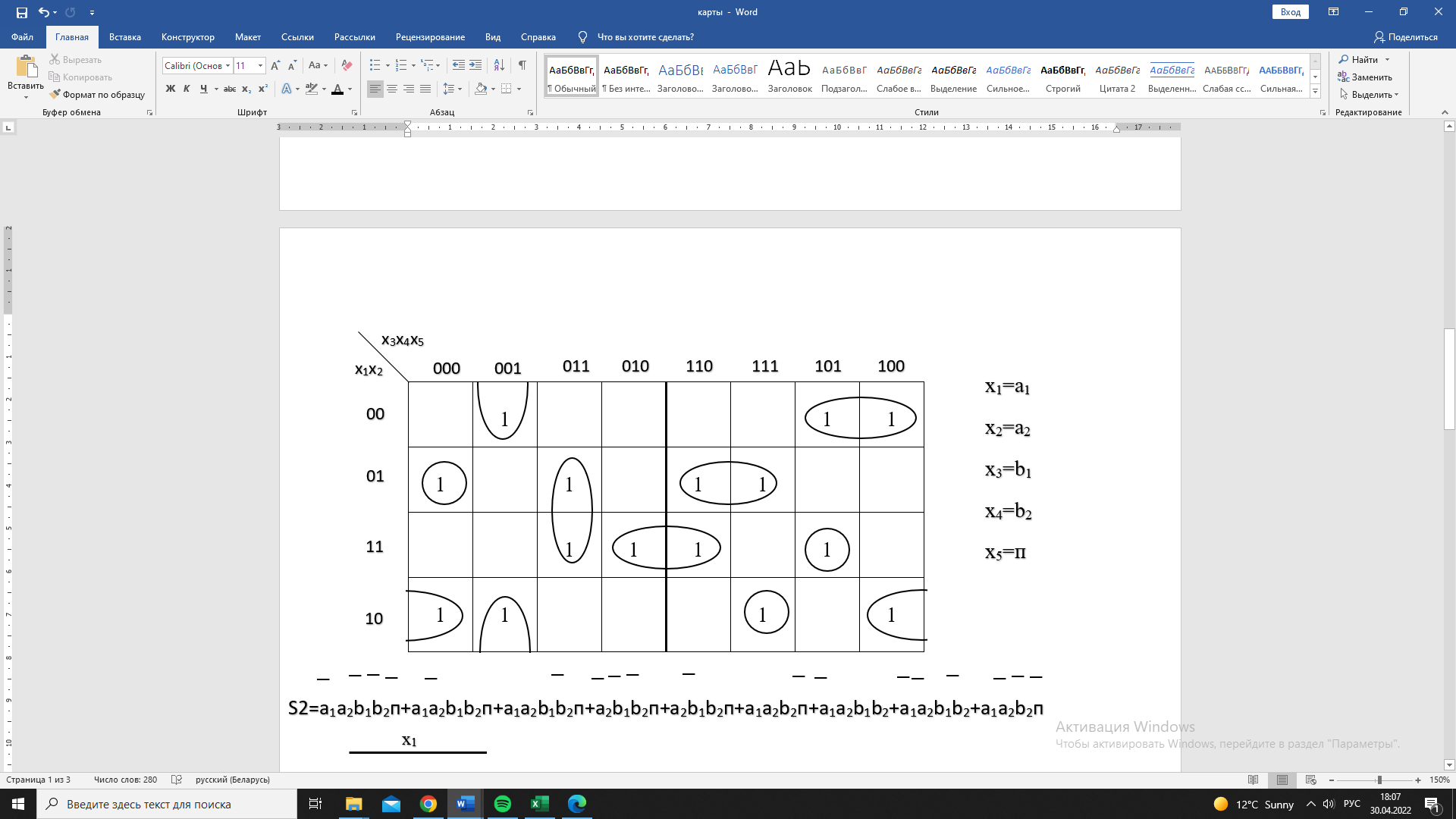
Найдем коэффициент минимизации:

КS1==5,32

**Минимизация функции S2**

Проведем минимизацию с помощью карты Карно. Для функции S1 заполненная карта приведена на рисунке 3.2.

b1b2p



a1a2

Рисунок 3.2 - Минимизация S1 при помощи карты Карно

S2=1a212+a12b1b2p+a1a2b12p+212p+a21b2p+a1a2b2+1a2b2p+12b12+a122

Запишем результат в данном базисе:

S2=(a1+(a2⊕1)+b1+b2+p)⊕1+((a1⊕1)+a2+(b1⊕1)+(b2⊕1)+(p⊕1))⊕1+((a1⊕1)+(a2⊕1)+(b1⊕1)+b2+(p⊕1))⊕1+(a2+b1+b2+(p⊕1))⊕1+ ((a2⊕1)+b1+(b2⊕1)+(p⊕1))⊕1+((a1⊕1)+(a2⊕1)+(b2⊕1)+p)⊕1+ (a1+(a2⊕1)+(b2⊕1)+(p⊕1))⊕1+(a1+a2+(b1⊕1)+b2)⊕1+((a1⊕1)+a2+b2+p)⊕1

Найдем коэффициент минимизации:

КS2==1,91

Функциональная схема ОЧС приведена в приложении Б.

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

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.

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, и добавление переноса; 1 – вывод на выходы без изменения значений разрядов, поступивших из регистра множимого.

Принцип работы ОЧУС представлен с помощью таблицы истинности (таблица 3.9).

Таблица 3.9 – Таблица истинности ОЧУС

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **p** | **x1** | **x2** | **y1** | **y2** | **h** | **П** | **Q1** | **Q2** | **результат в 4 С/С** |
| **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 |
| 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=01 |
| 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=01 |
| 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=01 |
| 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 3.9 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.

**Минимизация функции П**

Проведем минимизацию с помощью карты Вейча. Для функции П заполненная карта приведена на рисунке 3.3.

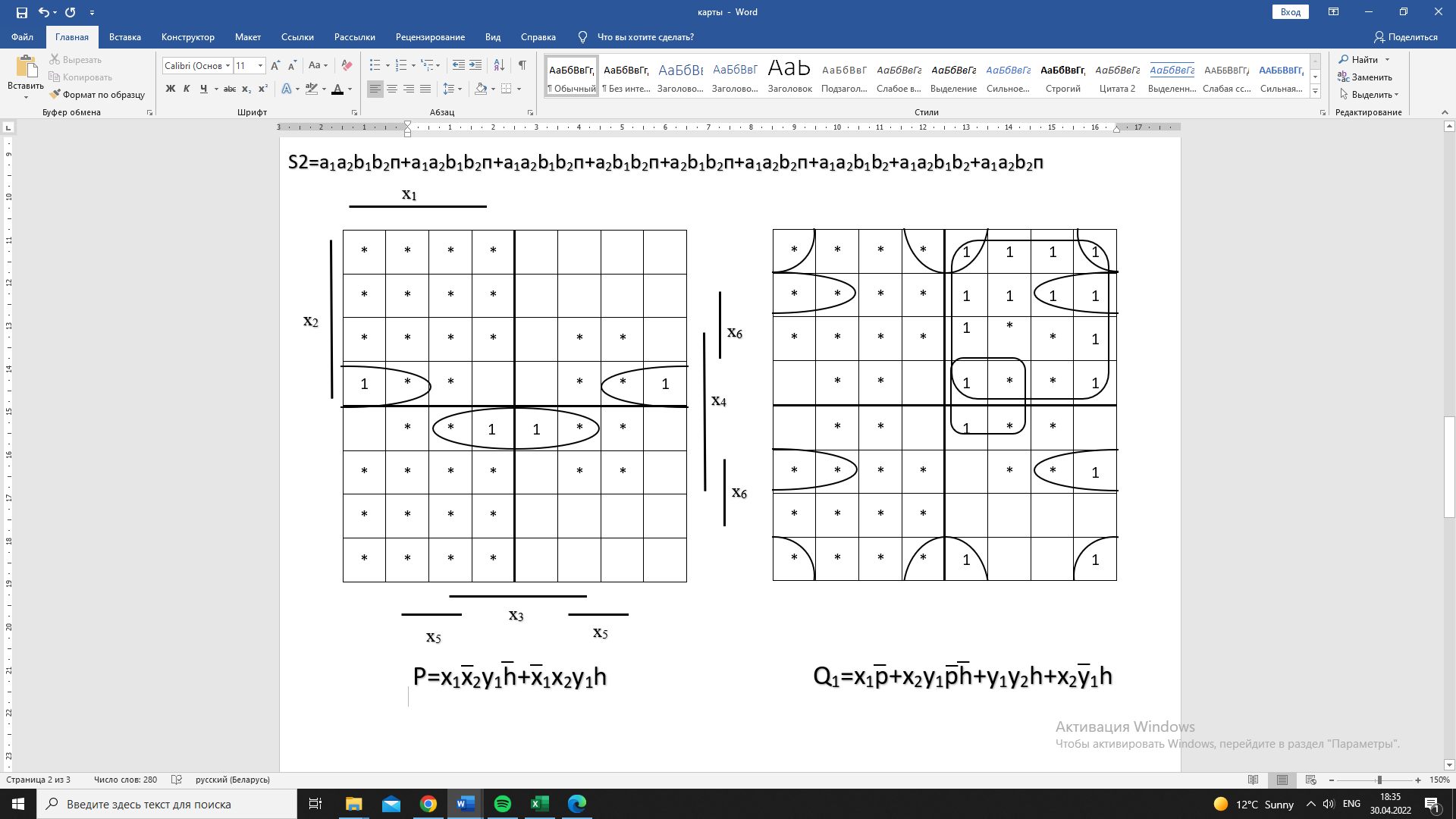


Рисунок 3.3 - Минимизация П при помощи карты Карно

Из карты (рис 3.4) видно, что минимизированная карта П имеет вид:

П=x12y1+1x2y1h

Запишем в нужном базисе:

П=x12y1+1x2y1h

Найдем коэффициент минимизации:

КП==2,57

**Минимизация функции Q1**

Проведем минимизацию с помощью карты Вейча. Для функции Q1 заполненная карта приведена на рисунке 3.4.

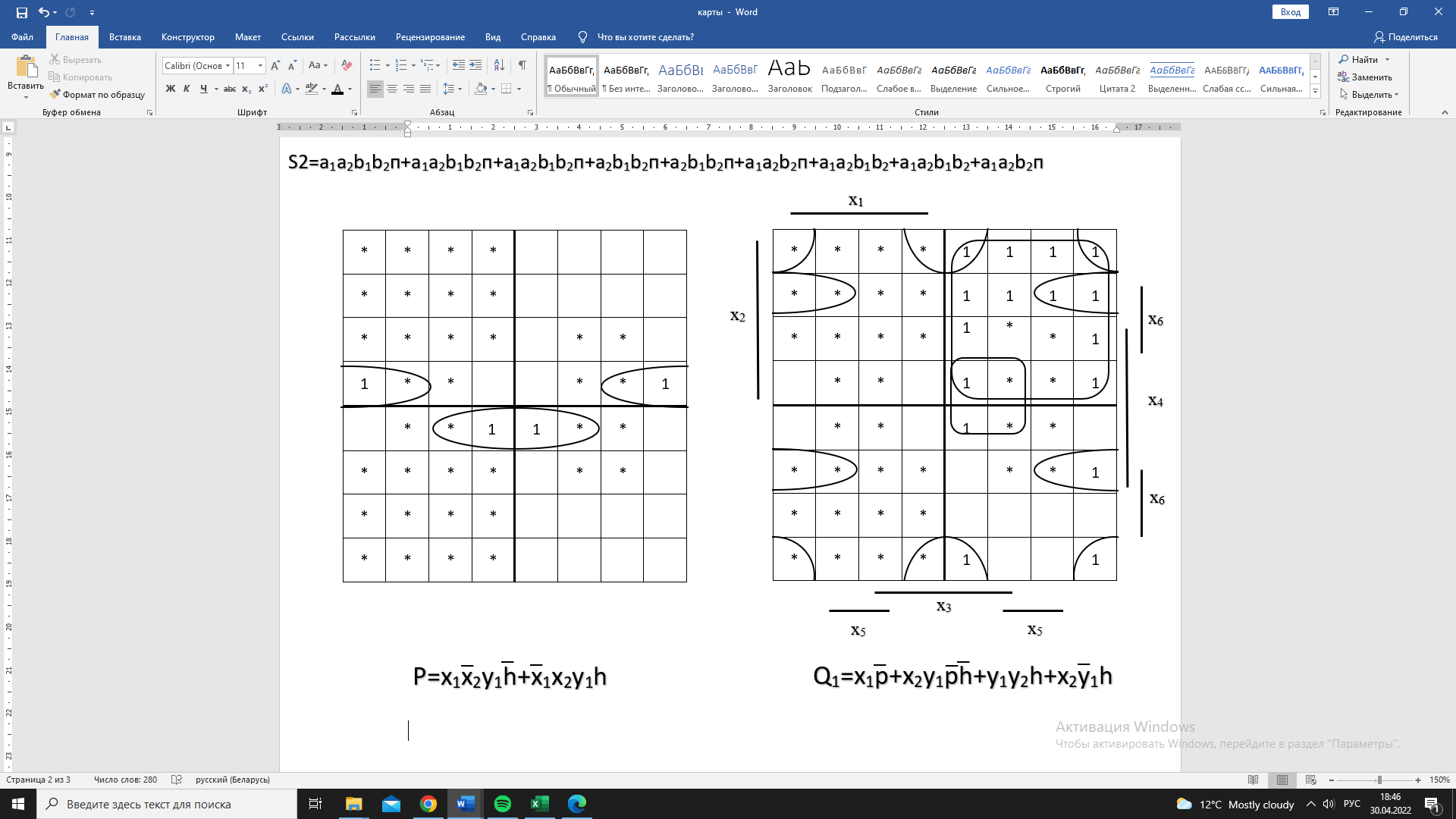


Рисунок 3.4 - Минимизация Q1 при помощи карты Карно

Из карты (рис 3.5) видно, что минимизированная карта Q1 имеет вид:

Q1=1+x121h+12y1h+x2y1h

Запишем в нужном базисе:

Q1=1+x121h+12y1h+x2y1h

Найдем коэффициент минимизации:

КП==5,36

**Минимизация функции Q2**

Проведем минимизацию с помощью карты Вейча. Для функции Q2 заполненная карта приведена на рисунке 3.5.

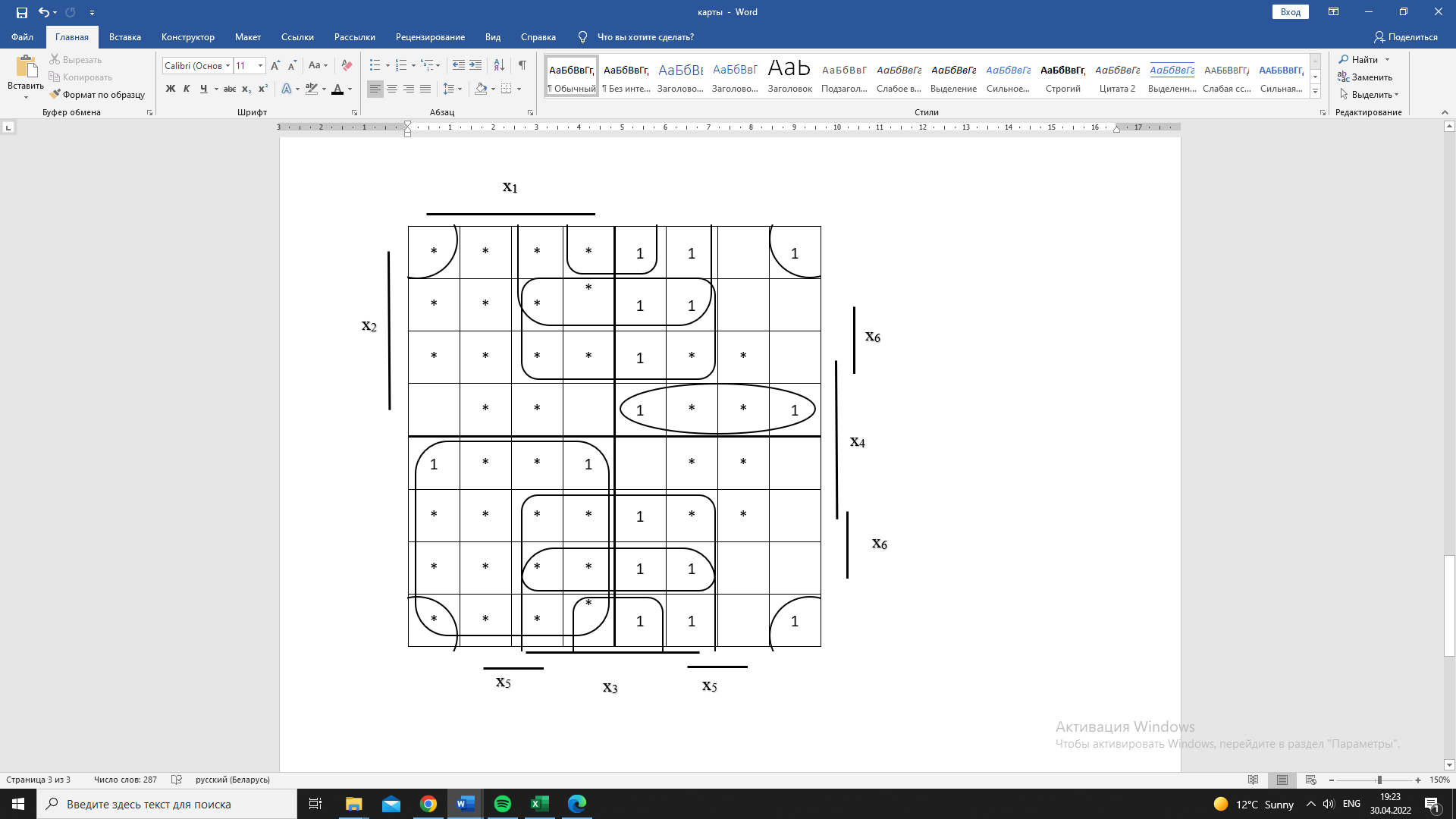


Рисунок 3.5 - Минимизация Q2 при помощи карты Карно

Из карты (рис 3.5) видно, что минимизированная карта Q2 имеет вид:

Q2=1p+12h+x21+x2h+x1y1ph

Запишем в нужном базисе:

Q2=1p+12h+x21+x2h+x1y1ph

Найдем коэффициент минимизации:

КП==5,62

Функциональная схема ОЧУС приведена в приложении В.

**4. ЛОГИЧЕСКИЙ СИНТЕЗ ОДНОРАЗРЯДНОГО ЧЕТВЕРИЧНОГО СУММАТОРА НА ОСНОВЕ МУЛЬТИПЛЕКСОРА**

Мультиплексор – это логическая схема, имеющая *n* информационных входов, *m* управляющих входов и один выход.

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.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 0 | 1 | «1» | 1 |  | 1 | 2+b2p |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 2+p | 0 | p | 0 | b2 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

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 самостоятельно заметить их минимальную форму.

Синтез дополнительных логических схем для ПФ ОЧC приведен в таблице 4.1.

Таблица 4.1. – Таблица истинности для ОЧС на мультиплексорах

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **p** | **П** | **ВЫХОД П** |  | **ВЫХОД S1** |  | **ВЫХОД S2** |
| 0 | 0 | 0 | 0 | 0 | 0 | b2 | 1 |  | 0 | b2⊕p |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 2p | 0 | p | 1 | 2 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | b2+p | 0 | p | 1 | 2 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 2 | 1 |  | 1 | 2+b2p |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 2+p | 0 | *p* | 0 | b2 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | «0» | 1 |  | 0 | b2⊕p |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Продолжение таблицы 4.1

Главным преимуществом построения ОЧС на мультиплексорах является значительно меньшее количество элементов, необходимых для функционирования схемы.

Функциональная схема ОЧС на базе мультиплексоров приведена в приложении Г.

**5 ЛОГИЧЕСКИЙ СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ МНОЖИТЕЛЯ (ПМ)**

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11, заменяя их на триады 10.

Таблица 5.1 - Таблица истинности ПМ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Мл. бит | Вх. диада | | Зн. | Вых. диада | |
| P | Q1 | Q2 | P | S1 | S2 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Минимизируем P, S1, S2 с помощью карт Карно.

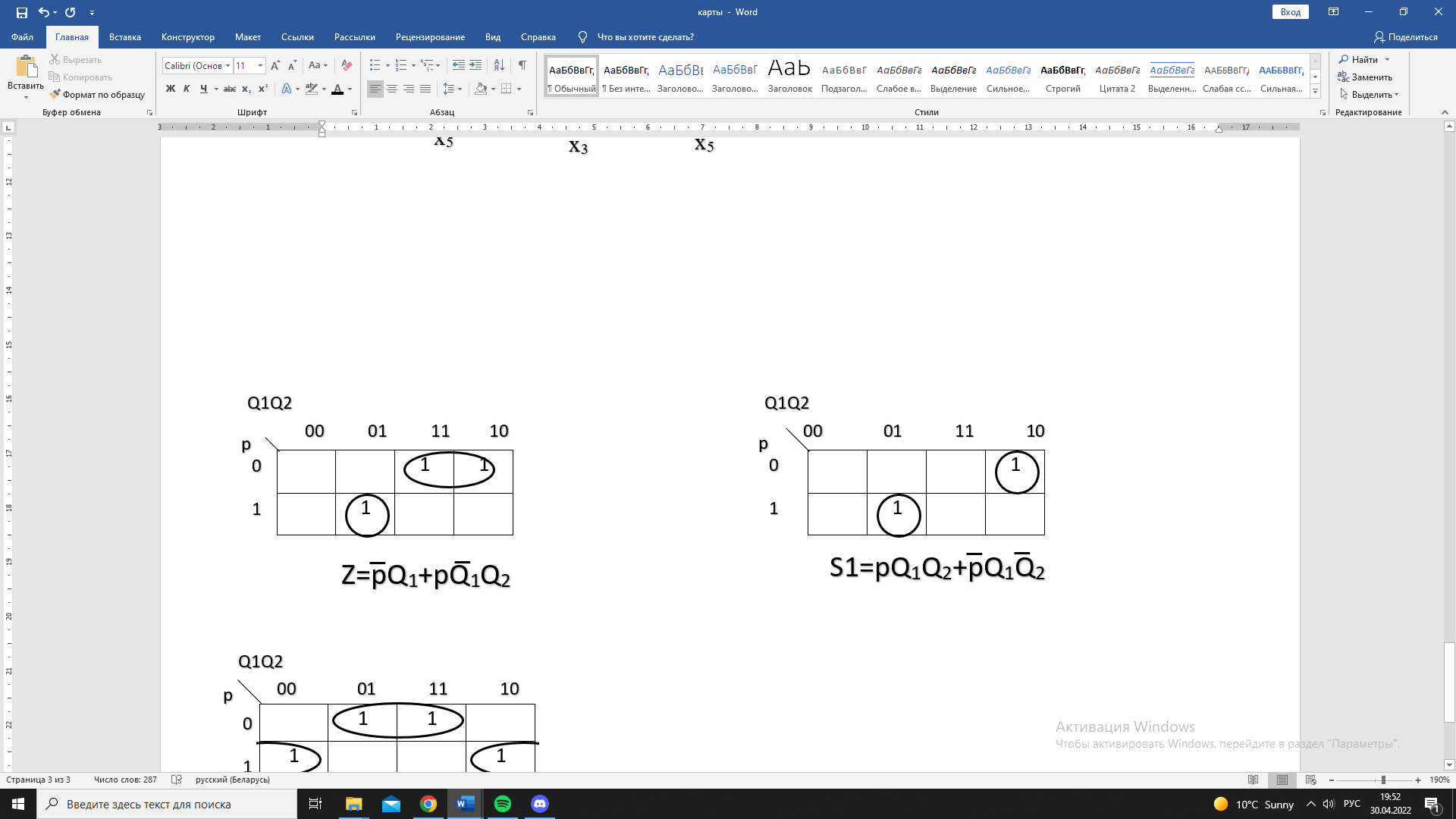


Рисунок 3.6 – Минимизация P при помощи карты Карно

Из карты (рис 3.6) видно, что минимизированная карта Q2 имеет вид:

P=Q1+p1Q2

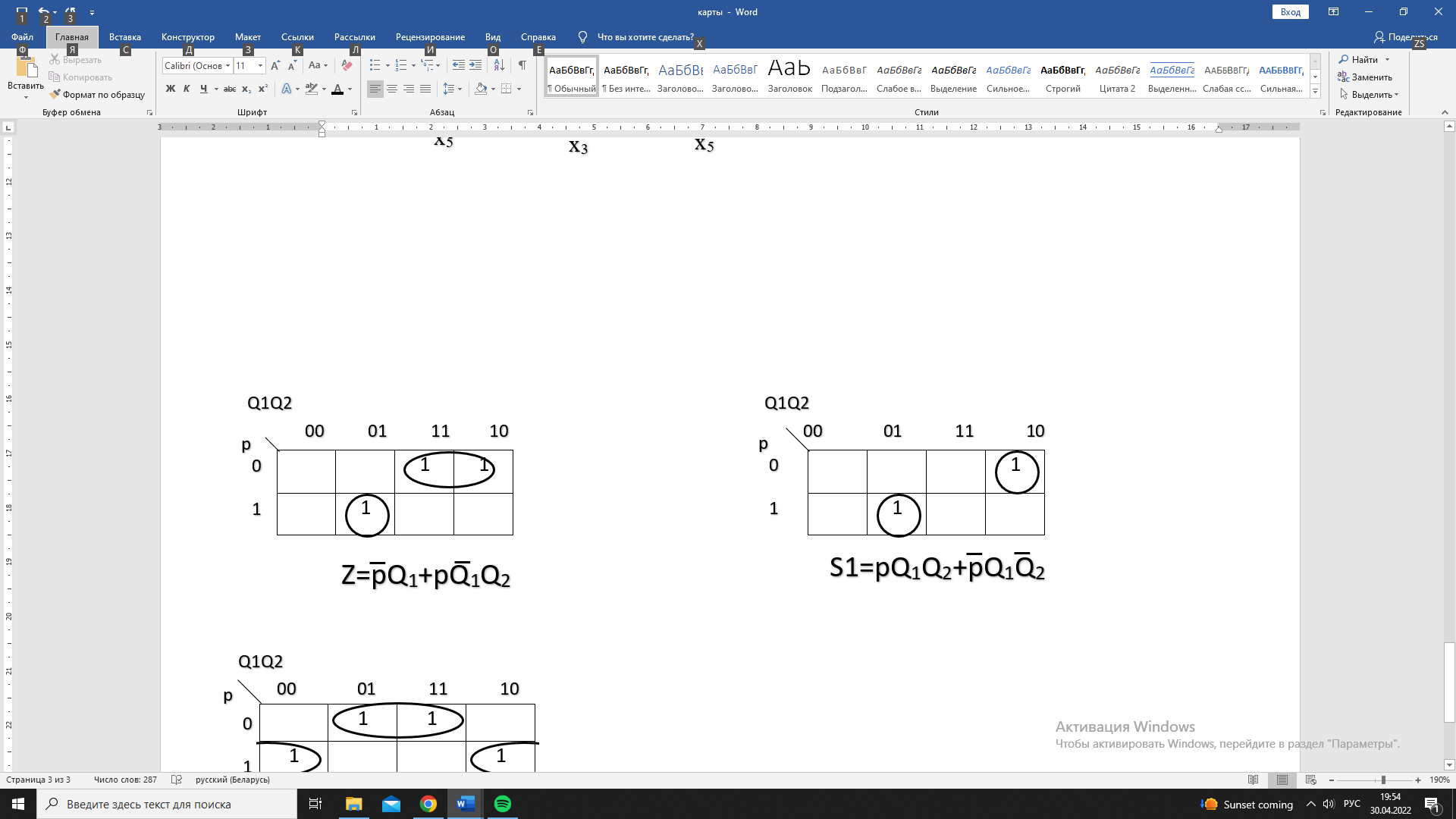


Рисунок 3.7 – Минимизация S1 при помощи карты Карно

Из карты (рис 3.7) видно, что минимизированная карта Q2 имеет вид:

S1=p1Q2+Q12

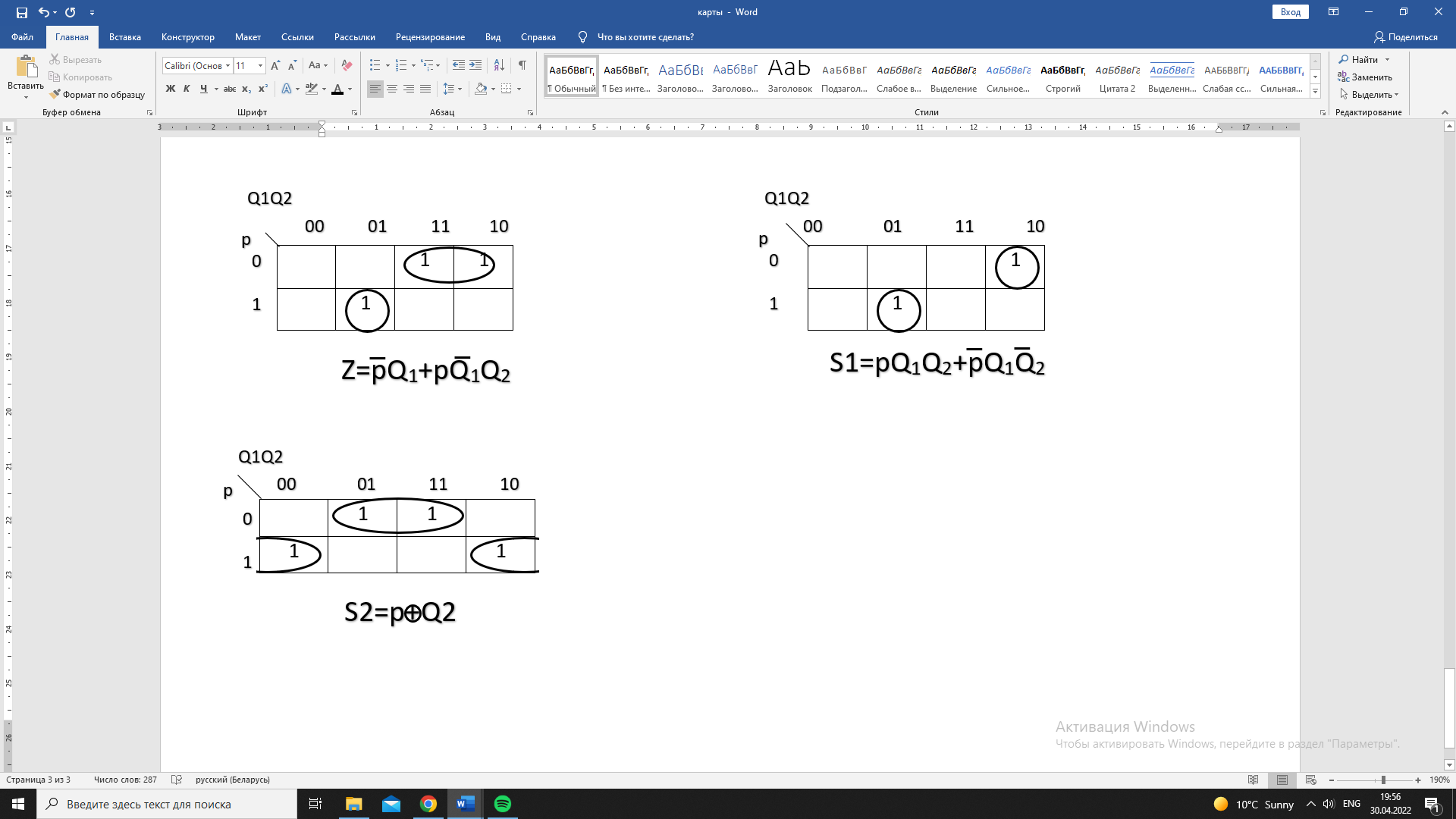


Рисунок 3.8 – Минимизация P при помощи карты Карно

Из карты (рис 3.8) видно, что минимизированная карта Q2 имеет вид:

S2=p⊕Q2

Функциональная схема ПМ приведена в приложении Д.

**6 ВРЕМЕННЫЕ ЗАТРАТЫ НА УМНОЖЕНИЕ**

Формула расчёта временных затрат на умножение:

Т=𝑛∗(𝑇ПМ+𝑇ФДК+m∗𝑇ОЧУC+(m+2)\*𝑇ОЧС+𝑇сдвига), где

𝑇ПМ – время преобразования множителя;

𝑇ФДК – время формирования дополнительного кода множимого;

𝑇ОЧУC – время умножения на ОЧУC;

𝑇ОЧС – время формирования единицы переноса в ОЧС;

𝑇сдвига – время сдвига частичной суммы;

n – количество разрядов множителя.

m – количество разрядов множимого

**ЗАКЛЮЧЕНИЕ**

По итогам выполнения курсового проекта по дисциплине «Арифметические и логические основы цифровых устройств» цель была достигнута, знания, полученные в результате выполнения, были успешно применены в теории и на практике.

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 функций различными алгоритмами, это позволило выявить их достоинства.

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.

Функциональные схемы основных узлов устройства были построены в различных базисах, что позволило закрепить знания основных законов булевой алгебры.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Луцик Ю.А., Лукьянова И.В. – Учебное пособие по курсу "Арифметические и логические основы вычислительной техники". – Минск: БГУИР, 2014 г.

Луцик Ю.А., Лукьянова И.В. – Методические указания к курсовому проекту по курсу “Арифметические и логические основы вычислительной техники”. – Мн.: БГУИР, 2004 г.

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

**ПРИЛОЖЕНИЕ А**

*(обязательное)*

Сумматор-умножитель второго типа. Схема электрическая структурная

**ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Одноразрядный четверичный сумматор.

Схема электрическая функциональная

**ПРИЛОЖЕНИЕ В**

*(обязательное)*

Одноразрядный четверичный умножитель-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