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 СУММАТОРАУМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 130 ПЗ

Студент А. П. Якуш (гр. 458301)

Руководитель Ю. А. Луцик

МИНСК 2025

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы

цифровых устройств

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_ Б. В. Никульшин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

ЗАДАНИЕ

по курсовой работе студента

Якуша Алексея Павловича

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-десятичных чисел»
2. Срок сдачи студентом законченной работы: 10 мая 2025 г.
3. Исходные данные к работе:
4. исходные сомножители: Мн = 92,27; Мт = 33,34;
5. алгоритм умножения: В;
6. метод умножения – умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в дополнительных кодах;
7. тип реализуемой структурной схемы – 1;
8. коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования:
9. – 00, – 11, – 01, – 10;
10. логический базис для реализации ОЧС: И-НЕ; метод минимизации – алгоритм Рота;
11. логический базис для реализации ОЧУ: И, НЕ; метод минимизации – карты Карно-Вейча;
12. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

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

1. Перечень графического материала:

* Сумматор-умножитель первого типа. Схема электрическая структурная.
* Одноразрядный четверичный сумматор. Схема электрическая функциональная.
* Одноразрядный четверичный умножитель. Схема электрическая функциональная.
* Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.
* Преобразователь множителя. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объём этапа,  % | Срок выполнения  этапа | Примечания |
| Разработка алгоритма умножения | 10 |  |  |
| Разработка структурной схемы  сумматора-умножителя | 10 |  | С выполнением  чертежа |
| Разработка функциональных схем  основных узлов сумматора- умножителя | 50 |  | С выполнением чертежей |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 |  | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 |  |  |

**Содержание**

Введение

1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ
2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ

**Введение**

1. **РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ**
2. Перевод сомножителей из десятичной системы счисления в четверичную

**Множимое**

|  |  |
| --- | --- |
| \* | 0,27 |
| 4 |
| \* | 1,08 |
| 4 |
|  | 0,32 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| - | 92 | 4 |  |  |  |  |
| 8 | - | 23 | 4 |  |  |
| - | 12 | 20 | - | 5 | 4 |
| 12 |  | 3 | 4 | 1 |
|  | 0 |  |  |  | 1 |  |

.

В соответствии с кодировкой множимого:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| - | 33 | 4 |  |  |
| 32 | - | 8 | 4 |
|  | 1 | 8 | 2 |
|  |  | 0 |  |
|  |  |  |  |  |

**Множитель**

|  |  |
| --- | --- |
| \* | 0,34 |
| 4 |
| \* | 1,36 |
| 4 |
| \* | 1,44 |
| 4 |
|  | 1,76 |

.

В соответствии с обычной весомозначной кодировкой множителя:

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

|  |  |  |
| --- | --- | --- |
| Мн = 0,111110001100 |  | – закодировано по заданию |
| Мт = 0,100001010101 | 11 | – закодировано традиционно |

1. Умножение двух чисел с плавающей запятой на два разряда множителя одновременно в прямых кодах

Порядок произведения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| + | = |  |  |  |
| = | 0.0011 |  |  |
| = | | 0.0111 |  |  |

Результат закодирован в соответствии с заданием на кодировку множимого.

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

зн Мн ⊕ зн Мт = 0 ⊕ 0 = 0.

1. Для умножения мантисс необходимо предварительно преобразовать множитель. При умножении чисел в прямых кодах диада 11() заменяется на триаду , диада 10() заменяется на триаду . Преобразованный множитель имеет вид: или . Перемножение мантисс по алгоритму «В» представлено в таблице 1.1

Таблица 1.1 – Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Четверичная с/с | | Двоичная с/с | | Комментарии |
| 1 | | 2 | | 3 |
| 0 | 000000000000 | 0 | 000000000000000000000000 |  |
| 0 | 000000113010 | 0 | 000000000000111110001100 | Мн |
| 0 | 000000113010 | 0 | 000000000000111110001100 |  |
| 0 | 000001130100 | 0 | 000000000011111000110000 |  |
| 3 | 333333101320 | 1 | 101010101010110011100100 | -2Мн |
| 0 | 000000232020 | 0 | 000000000000011001000100 |  |
| 0 | 000002320200 | 0 | 000000000001100100010000 |  |
| 0 | 000000000000 | 0 | 000000000000000000000000 | 0 |
| 0 | 000002320200 | 0 | 000000000001100100010000 |  |
| 0 | 000023202000 | 0 | 000000000110010001000000 |  |
| 0 | 000000113010 | 0 | 000000000000111110001100 | Мн |
| 0 | 000023321010 | 0 | 000000000110100111001100 |  |
| 0 | 000233210100 | 0 | 000000011010011100110000 |  |
| 0 | 000000113010 | 0 | 000000000000111110001100 | Мн |
| 0 | 000233323110 | 0 | 000000011010100110111100 |  |
| 0 | 002333231100 | 0 | 000001101010011011110000 |  |
| 0 | 000000113010 | 0 | 000000000000111110001100 | Мн |
| 1 | | 2 | | 3 |
| 0 | 003000010110 | 0 | 000010000000001100111100 |  |
| 0 | 030000101100 | 0 | 001000000000110011110000 |  |
| 0 | 000000113010 | 0 | 000000000000111110001100 | Мн |
| 0 | 030000220110 | 0 | 001000000000010100111100 |  |

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение (= 0,030000220110, = 7) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

= 0;

= 3074.5195.

Результат прямого перемножения даёт следующее значение:

3076,2818.

Абсолютная погрешность:

.

Относительная погрешность:

1. **РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ**
2. Структура сумматора-умножителя первого типа строится на базе заданных узлов ОЧУ, ОЧС и аккумулятора (накапливающего сумматора). Управление режимами работы схемы осуществляется внешним сигналом Mul/sum, который определяет вид текущей арифметической операции (умножение или сложение).
3. Если устройство работает как сумматор (на входе Mul/sum – «1»), то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющие входы h ОЧУ поступает «1».
4. Если устройство работает как умножитель (на входе Mul/sum – «0»), то множимое и множитель помещаются в соответствующие регистры, а на управляющие входы h ОЧУ поступает «0».
5. Частичные суммы хранятся в аккумуляторе, где и происходят их очередные сдвиги.

СТРУКТУРНАЯ СХЕМА СУММАТОРА-УМНОЖИТЕЛЯ ПРЕДСТАВЛЕНА В **ПРИЛОЖЕНИИ A.**

1. **РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ**
2. **Логический синтез одноразрядного четверичного умножителя**

ОЧУ – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда из регистра Мн, 2 разряда из регистра Мт и управляющий вход *h*) и 4 выхода. Принцип работы ОЧУ представлен с помощью таблицы истинности (таблица 3.1).

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

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

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.

В таблице 3.1 выделено 8 безразличных наборов, так как на входы ОЧУ из разрядов множителя не может поступить код «11».

Таблица 3.1 – таблица истинности ОЧУ

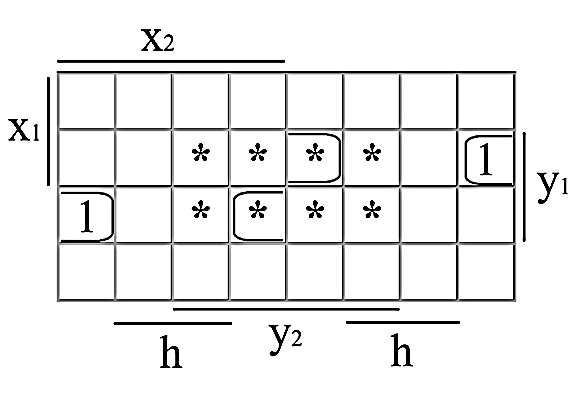
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Мн** | | **Мт** | | **Упр.** | **Старшие разряды** | | **Младшие разряды** | | **Пример операции в четверичной с/с** |
| ***x1*** | ***x2*** | ***y1*** | ***y2*** | ***h*** | ***P1*** | ***P2*** | ***P3*** | ***P4*** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 \* 0 = 00 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Выход - код «00» |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 \* 1 = 00 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | Выход - код «00» |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 \* 2 = 00 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | Выход - код «00» |
| 0 | 0 | 1 | 1 | 0 | x | x | x | x | 0 \* 3 = 00 |
| 0 | 0 | 1 | 1 | 1 | x | x | x | x | Выход - код «00» |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 \* 0 = 00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Выход - код «01» |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 \* 1 = 02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Выход - код «01» |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 2 \* 2 = 10 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход - код «01» |
| 0 | 1 | 1 | 1 | 0 | x | x | x | x | 2 \* 3 = 12 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | x | Выход - код «01» |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 \* 0 = 00 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | Выход - код «10» |

*Продолжение таблицы 3.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 3 \* 1 = 03 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | Выход – код «10» |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 3 \* 2 = 12 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Выход – код «10» |
| 1 | 0 | 1 | 1 | 0 | x | x | x | x | 3 \* 3 = 21 |
| 1 | 0 | 1 | 1 | 1 | x | x | x | x | Выход – код «11» |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 \* 0 = 00 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | Выход – код «11» |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 \* 1 = 01 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Выход – код «11» |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 \* 2 = 02 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | Выход – код «11» |
| 1 | 1 | 1 | 1 | 0 | x | x | x | x | 1 \* 3 = 03 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | x | Выход – код «11» |

Минимизация функций ОЧУ проведена с помощью карт Карно-Вейча.

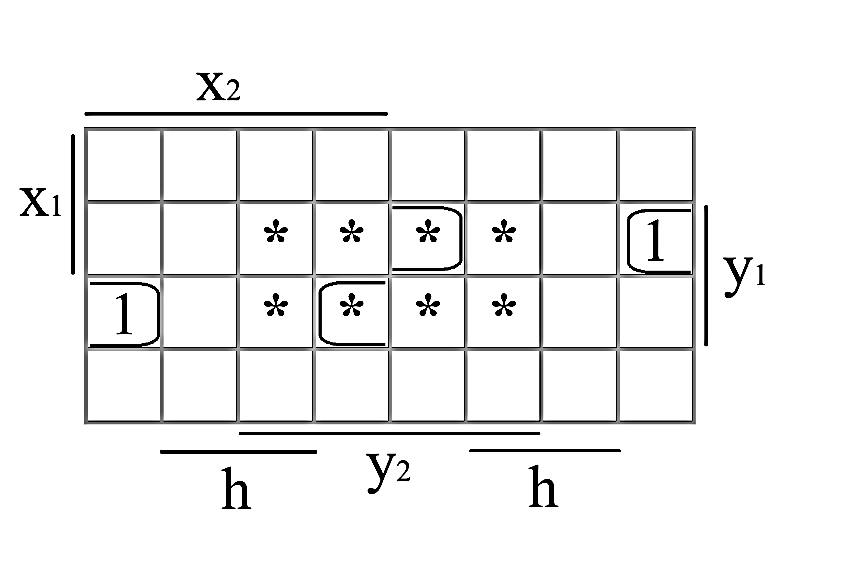
**Карты Вейча минимизации функций ОЧУ**

****Минимизация функции *P1***

*P1* = x2y1+x1y1

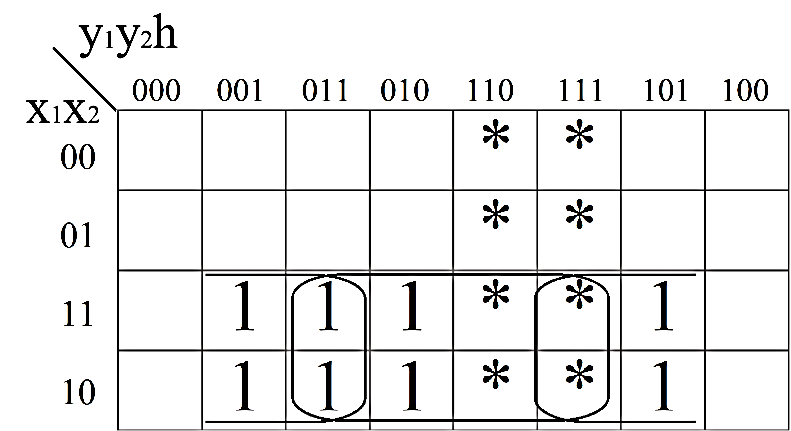
Эффективность минимизации по количеству входов логических элементов:

В базисе И, НЕ: *P1* =

****Минимизация функции *P2***

*P2* = x2y1+x1y1 = *P1*

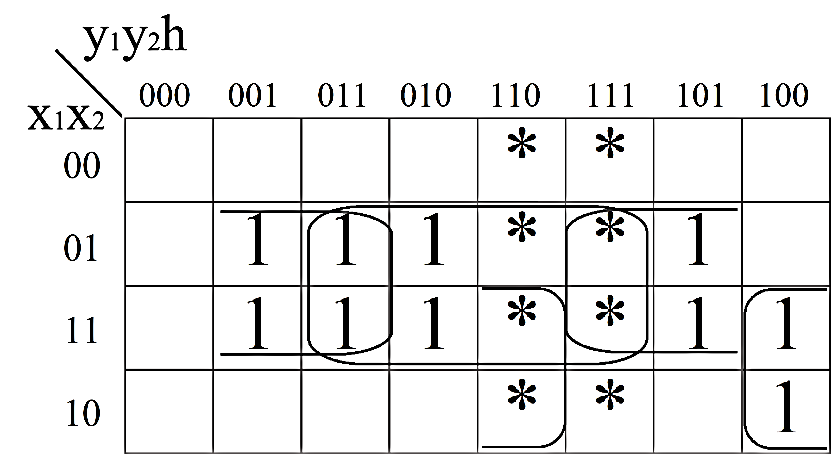
**Карты Карно минимизации функций ОЧУ**

**Минимизация функции *P3***

*P3 =*

Эффективность минимизации по количеству входов логических элементов:

В базисе И, НЕ: *P3 =*

**Минимизация функции *P4***

*P4=*

Эффективность минимизации по количеству входов логических элементов:

В базисе И, НЕ: *P4* =

1. **Логический синтез одноразрядного четверичного сумматора**

Одноразрядный четверичный сумматор (ОЧС) – 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 (таблица 3.2).

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

В таблице имеется 16 безразличных наборов т.к. со старших разрядов ОЧУ не могут прийти коды «2» и «3».

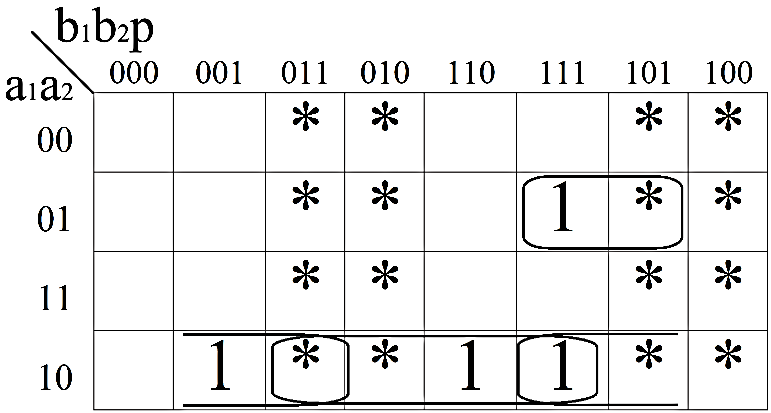
Таблица 3.2 – таблица истинности ОЧС

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a1** | **a2** | **b1** | **b2** | **p** | **П** | **S1** | **S2** | **Пример операции в четверичной с/с** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 + 0 + 0 = 00 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 + 0 + 1 = 01 |
| 0 | 0 | 0 | 1 | 0 | x | x | x | 0 + 2 + 0 = 02 |
| 0 | 0 | 0 | 1 | 1 | x | x | x | 0 + 2 + 1 = 03 |
| 0 | 0 | 1 | 0 | 0 | x | x | x | 0 + 3 + 0 = 03 |
| 0 | 0 | 1 | 0 | 1 | x | x | x | 0 + 3 + 1 = 10 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 + 1 + 0 = 01 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 + 1 + 1 = 02 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 2 + 0 + 0 = 02 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 2 + 0 + 1 = 03 |
| 0 | 1 | 0 | 1 | 0 | x | x | x | 2 + 2 + 0 = 10 |
| 0 | 1 | 0 | 1 | 1 | x | x | x | 2 + 2 + 1 = 11 |
| 0 | 1 | 1 | 0 | 0 | x | x | x | 2 + 3 + 0 = 11 |
| 0 | 1 | 1 | 0 | 1 | x | x | x | 2 + 3 + 1 = 12 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 2 + 1 + 0 = 03 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 2 + 1 + 1 = 10 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 + 0 + 0 = 03 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 3 + 0 + 1 = 10 |
| 1 | 0 | 0 | 1 | 0 | x | x | x | 3 + 2 + 0 = 11 |
| 1 | 0 | 0 | 1 | 1 | x | x | x | 3 + 2 + 1 = 12 |
| 1 | 0 | 1 | 0 | 0 | x | x | x | 3 + 3 + 0 = 12 |
| 1 | 0 | 1 | 0 | 1 | x | x | x | 3 + 3 + 1 = 13 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 3 + 1 + 0 = 10 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3 + 1 + 1 = 11 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 + 0 + 0 = 01 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 + 0 + 1 = 02 |

*Продолжение таблицы 3.2*

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

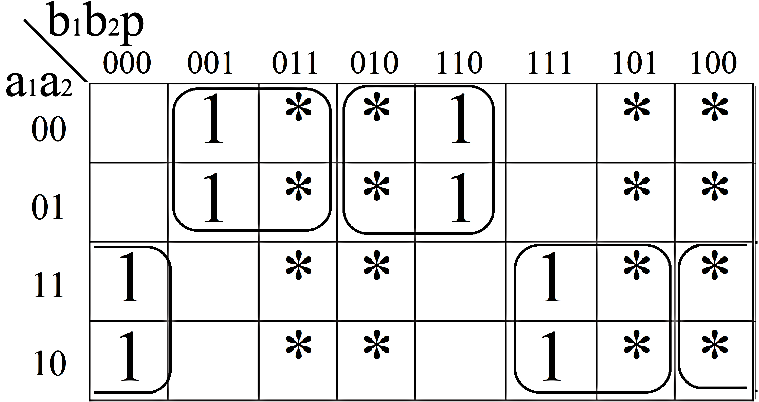
**Карты Карно минимизации функций ОЧС**

**Минимизация функции П**

П =

Эффективность минимизации по количеству входов логических элементов:

В базисе И-НЕ: П =

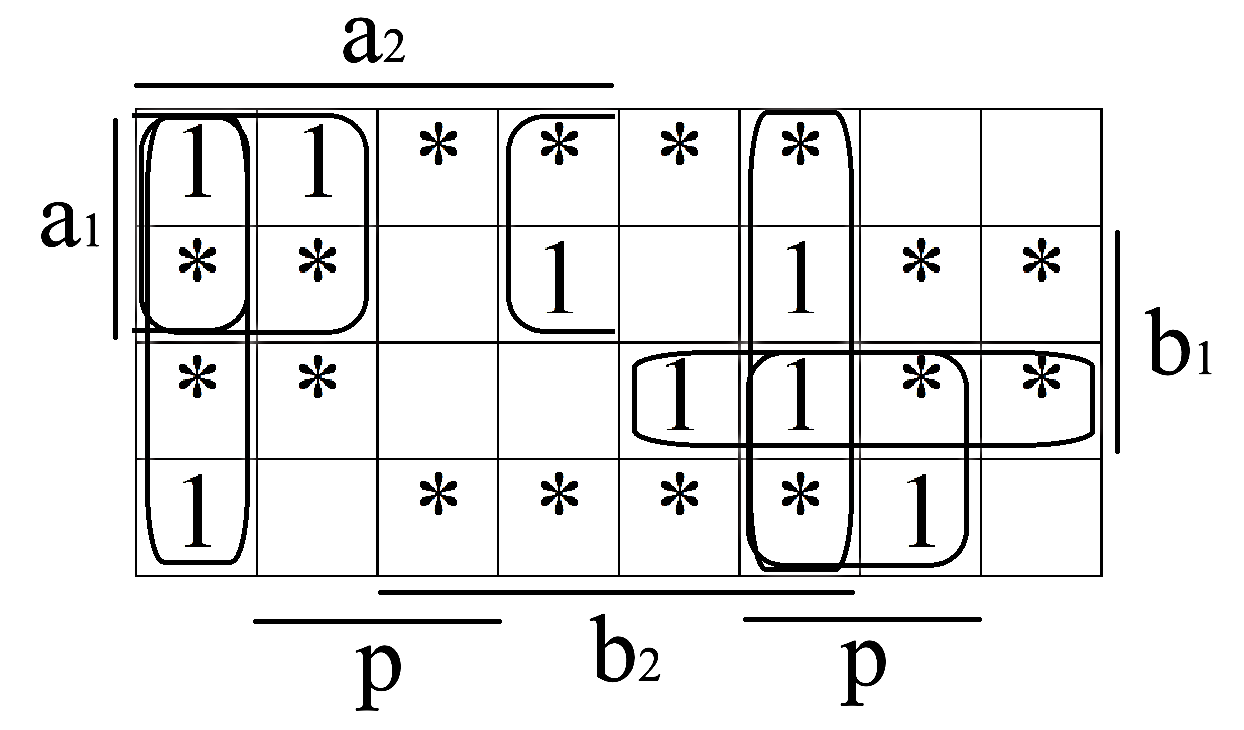
**Минимизация функции S1**

S1=

Эффективность минимизации по количеству входов логических элементов:

В базисе И-НЕ: S1=

**Карты Вейча минимизации функций ОЧС**

**Минимизация функции S2**

S2 =

Эффективность минимизации по количеству входов логических элементов:

В базисе И-НЕ: S2 = [