ВВЕДЕНИЕ

Темой курсовой работы является «Проектирование и логический синтез сумматора-умножителя двоично-четвертичных чисел». Это устройство акту-ально, поскольку мы его используем ежедневно. Оно устанавливается в бытовые приборы, мобильные телефоны и компьютеры.

В ходе курсовой работы разрабатывается алгоритм умножения, основ-ные узлы устройства, устройство на базу мультиплексоров.

*Цель курсовой работы* по дисциплине «Арифметические и логические основы вычислительной техники»:

1) освоение, углубление и систематизация теоретических и практических знаний, полученных в процессе изучения данной дисциплины;

2) развитие навыков самостоятельной работы;

3) подготовка студента к выполнению дипломного проектирования, а также к самостоятельной инженерной работе.

*Основные задачи курсовой работы:*

1) приобретение практических навыков и развитие творческих подходов к решению конкретной инженерной задачи, а именно разработке и синтезу цифровых схем арифметических устройств, выполняющих определенные операции над числами, представленными в заданной форме и кодировке;

2) формирование умения использовать справочную литературу, работать со всеми видами научно-технической литературы в заданной предметной области, осуществлять патентный поиск;

3) формирование навыков по разработке структуры аппаратного обеспе- чения в соответствии с заданием на базе системного подхода;

4) формирование навыка практической самостоятельной реализации устройств посредством логического синтеза под руководством преподавателя;

5) приобретение навыков по оформлению текстовой и графической документации согласно требованиям государственных стандартов и Стандарта предприятия СТП 01–2013.

**1** РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

**1.1** Перевод сомножителей из десятичной системы счисления в \_\_\_\_четверичную

|  |  |
| --- | --- |
| \* | 0,35 |
| 4 |
| \* | 1,40 |
| 4 |
| \* | 1,60 |
| 4 |
| \* | 2,40 |
| 4 |
|  | 1,60 |

*Множимое*

|  |  |  |
| --- | --- | --- |
| \_ 22 | 4 |  |
| 20 | \_ 5 | 4 |
| 2 | 4 | 1 |
|  | 1 |  |

Мн4 = 112,112.

|  |  |
| --- | --- |
| \* | 0,54 |
| 4 |
| \* | 2,16 |
| 4 |
| \* | 0,64 |
| 4 |
| \* | 2,56 |
| 4 |
|  | 2,24 |

В соответствии с заданной кодировкой множимого:

Мн2/4 = 010110,010110.

*Множитель*

|  |  |  |
| --- | --- | --- |
| \_ 49 | 4 |  |
| 4 | \_ 12 | 4 |
| \_ 9 | 12 | 3 |
| 8 | 0 |  |
| 1 |  |  |

Мт4 = 301,202.

В соответствии с обычной весомозначной кодировкой множителя:

Мт2/4 =110001,100010.

**1.2** Представление сомножителей в форме с плавающей запятой в \_\_\_\_\_\_\_\_прямом коде

Мн=0,010110010110 РМн =0.0011 +0310  − закодировано по заданию,

Мт=0,110001100010 РМт =0.0011 +0310  − закодировано традиционно.

**1.3** Умножение двух чисел с плавающей запятой на два разряда \_\_\_\_\_\_\_\_\_множителя одновременно в прямых кодах

РМн  = 0.0011 034

РМт = 0.0011 034

РМн·Мт = 0.0110 124

Знак произведения:

зн Мн ⊕ зн Мт = 0 ⊕ 0 = 0.

Преобразованный множитель:

= 101202;

= 010001100010.

Перемножение мантисс по алгоритму «А» приведено в таблице 1.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Четвертичная с/с** | | | **Двоично-четвертичная с/с** | | | **Комментарии** |
| **1** | | | **2** | | | **3** |
| 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  3.  3.  3.  0.  0. | 000000  230230  230230  023023  002302  230230  233132  023313  112112  202031  020203  002020  221222  223302  322330  112112  101102 | 0  30  30  230  230  1230  31230  31230  231230  231230 | 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  1.  0.  0. | 00 00 00 00 00 00  10 11 00 10 11 00  10 11 00 10 11 00  00 10 11 00 10 11 00 00 10 11 00 10  10 11 00 10 11 00  10 11 11 01 11 10  00 10 11 11 01 11  01 01 10 01 01 10  10 00 10 00 11 01  00 10 00 10 00 11  00 00 10 00 10 00  10 10 01 10 10 10  10 10 11 11 00 10  11 10 10 11 11 00  01 01 10 01 01 10  01 00 01 01 00 10 | 00  11   00  11 00  10 11 00  10 11 00  01 10 11 00  11 01 10 11 00  11 01 10 11 00  10 11 01 10 11 00  10 11 01 10 11 00 | ∑0 = 0  П1 = Мн ∙ 2  ∑1  ∑1 ∙ 4-1(П2, ∑2)  ∑2 ∙ 4-1  П3 = Мн ∙ 2  ∑3  ∑3 ∙ 4-1  П4 = Мн ∙ 1  ∑4  ∑4 ∙ 4-1(П5, ∑5)  ∑5 ∙ 4-1  П6 = Мн ∙ (-1)  ∑6  ∑6 ∙ 4-1  П7 = Мн ∙1  ∑7 |

Таблица 1.1 – Перемножение мантисс

Полученное произведение (Мн ∙ Мт4 = 0,101102231230, РМн ∙ Мт =124) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

Мн ∙ Мт4 = 101102,231230 РМн ∙ Мт = 0;

Мн ∙ Мт10  = 1106,7138.

Результат прямого перемножения:

Мн10 ∙ Мт10 = 1107,2190.

**1.4** Расчет погрешностей

Абсолютная погрешность рассчитывается по формуле

Δ = Мн10 ∙ Мт10 - Мн ∙ Мт10, (1.1)

где Мн10 ∙ Мт10 – результат прямого умножения;

Мн ∙ Мт10 – результат умножение в четвертичной системе счисления.

В результате подстановки полученных результатов в формулу (1.1) аб-солютная погрешность составит:

Δ=1107,2190 - 1106,7138 = 0,5052.

Относительная погрешность рассчитывается по формуле

*δ* = , (1.2)

где Мн, Мт – операнды.

В результате подстановки полученных результатов в формулу (1.1) от-носительная погрешность составит:

*δ* = = 0,000456 (*δ* = 0,0456 %).

Данная погрешность объясняется приближённым переводом из деся-тичной системы счисления в четвертичную обоих сомножителей, а также округлением полученного результата произведения.

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

**2.1** Синтез структуры одноразрядного четвертичного умножителя

Устройство работает как сумматор (на входе *Mul/sum* – «1»): оба слага-емых последовательно (за два такта) заносятся в регистр множимого, на управляющий вход формирователя дополнительного кода (ФДК) *F2* поступает «1».  
 Т.к. числа представлены в форме с плавающей запятой, прежде чем складывать мантиссы, необходимо выровнять порядки. Для этого необходимо в блоке порядков обеспечить сравнение порядков. После использования сумматора порядков в зависимости от знака результата сдвигается (уменьша-ется) мантисса числа, порядок которого меньше, чем порядок другого числа.  
 Сдвиг мантиссы числа с меньшим порядком будет реализован согласно используемому алгоритму «А». Порядок подачи слагаемых на операцию начинается с младших разрядов, мантисса сдвигается в регистре результата.

На выходах ФДК формируется дополнительный код одного из слагаемых с учётом знака. Это слагаемое может быть записано в регистр результата, при этом управляющие сигналы, поступающие на входы *h* всех ОЧУ, дают возможность переписать на выходы ОЧУ разряды слагаемого без изменений (рисунок 2.1).

*ОЧУ*

*Слагаемое*

*«0»*

*Слагаемое*

*«0»*

*h=1*

*ОЧУ*

*Мн* ***∙*** *Мт*

*Мн*

*Мт*

*h=0*

Рисунок 2.1 – Режимы работы ОЧУ

Если на вход *h* поступает «0», то ОЧУ перемножает разряды Мн и Мт.  
 Одноразрядный четверичный сумматор предназначен для сложения двух двоично-четверичных цифр, подаваемых на его входы (рисунок 2.2).

*А + В =4*

*А = 3*

*ОЧУ*

*В = 1*

*1*

*0*

Рисунок 2.2 – Одноразрядный четвертичный сумматор

В ОЧС первое слагаемое складывается с нулем, т.к. на старших выходах ОЧУ будут формироваться только коды нуля. Затем первое слагаемое попадает в регистр-аккумулятор, который изначально обнулён.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУ и ОЧС попадает в аккумулятор, где складывается с первым слагаемым. Таким образом, аккумулятор (накапливающий сумматор) складывает операнды и хранит результат.

Разрядность аккумулятора должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения переноса при суммировании.

Устройство работает как умножитель (на входе *Mul/sum* – «0»): множимое и множитель помещаются в соответствующие регистры, на управляющий вход ФДК *F2* поступает «0».

**2.2** Синтез структуры преобразователя множителя

Диада множителя поступает на входы преобразователя множителя (ПМ). Задачей ПМ является преобразование диады множителя в соответствии с алгоритмом «А». При этом в случае образования единицы переноса в старшую диаду множителя она должна быть учтена при преобразовании следующей старшей диады (выход 1 ПМ), т. е. сохраняться до следующего такта на триггер.

В регистре множителя в конце каждого такта умножения содержимое сдвигается на два двоичных разряда, и в последнем такте умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения.

Выход 2 ПМ переходит в единичное состояние, если текущая диада со- держит отрицание (̅0̅1̅). В этом случае инициализируется управляющий вход *F1* формирователя дополнительного кода (ФДК), и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на – 1).

Принцип работы ФДК в зависимости от управляющих сигналов приведён в таблице 2.2.

Таблица 2.2 – Режимы работы формирователя дополнительного кода

|  |  |  |
| --- | --- | --- |
| **Сигналы на входах ФДК** | | **Результат на выходах ФДК** |
| ***F1*** | ***F2*** |
| 0 | 0 | Дополнительный код множимого |
| 0 | 1 | Дополнительный код слагаемого |
| 1 | 0 | Меняется знак Мн |
| 1 | 1 | Меняется знак слагаемого |

На выходах 3 и 4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУ вместе с диадами множимого.

**2.3** Синтез структуры одноразрядного четвертичного сумматора

ОЧУ предназначен для умножения двух четверичных цифр. Поэтому необходимо предусмотреть возможность прибавления переноса из предыдущей диады. Для суммирования результата умножения текущей диады Мн∙Мт с переносом из предыдущей диады предназначены ОЧС. Следовательно, чтобы полностью сформировать частичное произведение четверичных сомножителей, необходима комбинация цепочек ОЧУ и ОЧС.

Частичные суммы формируются в аккумуляторе. На первом этапе он обнулён, и первая частичная сумма получается за счёт сложения первого частичного произведения (сформированного на выходах ОЧС) и нулевой частичной суммы (хранящейся в аккумуляторе). В аккумуляторе происходит сложение i-й частичной суммы с (i+1)-м частичным произведением, результат сложения сохраняется. Содержимое аккумулятора сдвигается на один четверичный разряд вправо в конце каждого такта умножения по заданному алгоритму «А».

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

3 ∙ 2 = 1 2

max max

Мн Мт

Это означает, что на младшие входы ОЧС никогда не поступят диады цифр, соответствующие кодам «2» и «3», следовательно, в таблице истинности работы ОЧС будут содержаться 16 безразличных входных наборов.

Частичные суммы хранятся в аккумуляторе и регистре множителя, т. к. алгоритм умножения «А» предполагает возможность синхронного сдвига этих устройств. Количество тактов умножения определяется разрядностью Мт.

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

**3** РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ \_\_\_\_\_\_.СУММАТОРА- УМНОЖИТЕЛЯ

**3.1** 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 представлен с помощью таблицы истинности (таблица 3.1).

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

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

Управляющий вход h определяет тип операции:

«0» – умножение закодированных цифр, поступивших на информа-ционные входы;

«1» – вывод на выходы без изменения значения разрядов, поступивших из регистра множимого.

В таблице 3.1 выделено восемь безразличных наборов, т. к. 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 четвертичной с/c** |
| *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 | 1∙0=00 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Выход – код «01» |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1∙1=01 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | Выход – код «01» |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1∙2=02 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход – код «01» |
| 0 | 1 | 1 | 1 | 0 | x | x | x | x | 1∙3=03 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | x | Выход – код «01» |

*Продолжение таблицы 3.1*

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

Минимизация функций *P1*, *P2*, *P3*, *P4* производится с помощью алгоритма Рота.

В связи с тем, что для каждой функции набор безразличных значений одинаков, проведем его минимизацию с помощью карты Карно(рисунок 3.1).

Множество безразличных значений *N*:

00110, 00111

01110, 01111

*N = .*

10110, 10111

11110, 11111

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x1x2\y1y2h* | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 |  |  |  |  | x | x |  |  |
| 01 |  |  |  |  | x | x |  |  |
| 11 |  |  |  |  | x | x |  |  |
| 10 |  |  |  |  | x | x |  |  |

Рисунок 3.1 – Минимизация функции *N* при помощи карты Карно

Следовательно:

*N = xx*11*x .*

Следует отметить, что функция *P1* для каждого набора принимает значение «0».

Следовательно:

*P1* min = .

**3.1.1** Минимизация функции *P2*

Множество единичных кубов *L* и множество безразличных кубов *N*:

10100

*L =* , *N* = *xx*11*x* .

11100

Откуда образуется множество *C0 = L* ⋃ *N* :

10100

*C0 =* 11100 .

*xx*11*x*

Первый этапом минимизации является нахождения множества простых импликант, для реализации которого будет использовать операция умножения (\*) над множествами *C0*, *C1, C2* и т.д., до тех пор пока будут образовываться кубы большей размерности.

Умножение *С0\*С0*приведено в таблице 3.2.  
  
Таблица 3.2 – Поиск простых импликант (*С0\*С0*)

|  |  |  |  |
| --- | --- | --- | --- |
| *С0\*С0* | 10100 | 11100 | *xx*11*x* |
| 10100 | ----- |  |  |
| 11100 | 1*y*100 | ----- |  |
| *xx*11*x* | 101*y*0 | 111*y*0 | ----- |

В результате умножения образовалось множество кубов первой размер-ности *A1* и пустое множество простых импликант *Z0*.

1*y*100

*A1* = 101*y*0 , *Z0* = Ø.

111*y*0

После чего формируется множество *B1* = *C0* – *Z0* = *C0* и множество *C1* = *A1* ⋃ *B1*, которое включает в себя следующий набор импликант (здесь и на следующих шагах множество представлено после выполнения операции поглощения кубами множества *A* кубов множества *B*):

1*x*100

101*x*0

*C1= .*

111*x*0

*xx*11*x*

Умножение *С1\*С1* приведено в таблице 3.3.

Таблица 3.3 – Поиск простых импликант (*С1\*С1*)

|  |  |  |  |
| --- | --- | --- | --- |
| *С1\*С1* | 1*x*100 | 101*x*0 | 111*x*0 |
| 1*x*100 | ----- |  |  |
| 101*x*0 |  | ----- |  |
| 111*x*0 |  | 1*y*1*x*0 | ----- |
| *xx*11*x* | 1*x*1*y*0 |  |  |

В следствие умножения *С1\*С1* образовались множество кубов второй размерности *A2*, пустое множество *Z1* и множество *B2*:

*A2* = 1*x*1*x*0 , *Z1* = Ø, *B2* = *С1.*

В результате образовалось множество кубов *С2*:

1*x*1*x*0

*C2= .*

*xx*11*x*

В таблице 3.4 приведен следующий шаг поиска простых импликант –6операция *С2\*С2*.

Таблица 3.4 – Поиск простых импликант (*С2\*С2*)

|  |  |
| --- | --- |
| *С2\*С2* | 1*x*1*x*0 |
| 1*x*1*x*0 | ----- |
| *xx*11*x* |  |

В результате выполнения умножения *С2\*С2* не было получено кубов новой размерности, т.е. множество *A3* – пустое множество, а во множество *Z2* пошли все кубы множества *C2*:

*A3* = Ø, *Z2* = 1*x*1*x*0, *xx*11*x .*

На этом этап нахождения простых импликант можно считать завершенным. Таким образом, cформированное множество простых им-пликант:

1*x*1*x*0

*Z = Z0* ⋃ *Z1* ⋃ *Z2 = .*

*xx*11*x*

Следующий этап минимизации – определение *L*-экстрималей. Для этого необходимо прежде всего выявить обязательные простые импликанты. В таблице 3.5 и таблице 3.6 представлены операция вычитания (#) и пересечения (∩) кубов соответственно, которые позволяет определить, который из кубов (и только он) покрывает некоторую вершину из множества единичных кубов *L.*

Таблица 3.5 – Операция вычитания импликант

|  |  |  |
| --- | --- | --- |
| z#(Z-z) | 1*x*1*x*0 | *xx*11*x* |
| 1*x*1*xx* | ----- | 0*zzz*1  0*x*1*xx* *xx*111 |
| *xx*11*x* | *zzz0z*  1*x*100 | ----- |
| Остаток | 1*x*100 | 0*x*1*xx*, *xx*111 |

Таким образом, из таблицы получено множество *L*-экстрималей:

1*x*1*x*0

*E = . .*

*xx*11*x*

Операция пересечения позволяет выявить остатки простые импликанты, которые покрывают только некоторое множество кубов *N.*

Таблица 3.6 – Операция пересечения

|  |  |  |
| --- | --- | --- |
| *Z#(Z-z)*∩*L* | 10100 | 11100 |
| 1*x*100 | 10100 | 11100 |
| 0*x*1*xx* | Ø | Ø |
| *xx*111 | Ø | Ø |

Из таблицы 3.6 видно, что кубы 0*x*1*xx*, *xx*111 не пересекаются с кубами множества *L.* Таким образом, можно сделать вывод, что они образованы на множестве *N*, что позволяет уменьшить цену схемы. Это дает основание удалить куб *xx*11*x* из множества *Е.*

В таблице 3.7 представлено вычитание из каждого куба комплекса *L* элементов множества *E*, что позволяет определить какие кубы множеств *Е* не покрывают вершин множества *L.*

Таблица 3.7 – Операция вычитания из *L* множества *E*

|  |  |  |
| --- | --- | --- |
| *L* # *E* | 10100 | 11100 |
| 1*x*1*x*0 | *zzzzz*  Ø | *zzzzz*  Ø |

Следовательно, все кубы множества *Е* покрывают все вершины множества *L.* Значит, может быть получена одна тупиковая форма:

*P2* min=*x1y1*.

**3.1.2** Минимизация функции *P3*

Множество единичных кубов *L* и множество безразличных кубов *N*:

01100, 11001

10001, 11010

*L = , N* = *xx*11*x* .

10010, 11011

10011, 11100

10101, 11101

Откуда образуется множество *C0 = L* ⋃ *N* :

01100, 11001

10001, 11010

10010, 11011

*C0 = .*

10011, 11100

10101, 11101

*xx*11*x*

*Нахождение простых импликант.* Умножение *С0\*С0*приведено в таблице 3.8.

Таблица 3.8 – Поиск простых импликант (*С0\*С0*)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *С0\*С0* | 01100 | 10001 | 10010 | 10011 | 10101 | 11001 | 11010 | 11011 | 11100 | 11101 |
| 01100 | ----- |  |  |  |  |  |  |  |  |  |
| 10001 |  | ----- |  |  |  |  |  |  |  |  |
| 10010 |  |  | ----- |  |  |  |  |  |  |  |
| 10011 |  | 100*y*1 | 1001*y* | ----- |  |  |  |  |  |  |

*Продолжение таблицы 3.8*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10101 |  | 10*y*01 |  |  | ----- |  |  |  |  |  |
| 11001 |  | 1*y*001 |  |  |  | ----- |  |  |  |  |
| 11010 |  |  | 1*y*010 |  |  |  | ----- |  |  |  |
| 11011 |  |  |  | 1*y*011 |  | 110*y*1 | 1101*y* | ----- |  |  |
| 11100 | *y*1100 |  |  |  |  |  |  |  | ----- |  |
| 11101 |  |  |  |  | 1*y*101 | 11*y*01 |  |  | 1110*y* | ----- |
| *xx*11*x* | 011*y*0 |  | 10*y*10 | 10*y*11 | 101*y*1 |  | 11*y*10 | 11*y*11 | 111*y*0 | 111*y*1 |

В результате умножения образовалось множество кубов первой размер-ности *A1* и пустое множество простых импликант *Z0*.

*x*1100, 1001*x*, 1*x*101, 11*x*10   
011*x*0, 1*x*010, 101*x*1, 11*x*11

*A1= , Z0* = Ø.

100*x*1, 10*x*10, 110*x*1, 1110*x*  
10*x*01, 1*x*011, 11*x*01, 111*x*0  
1*x*001, 10*x*11, 1101*x*, 111*x*1

Отсюда формируется множество *B1* = *C0* – *Z0* = *C0.* После чего можно сформировать множество *C1* = *A1* ⋃ *B1*.

*x*1100, 1001*x*, 1*x*101, 11*x*10  
011*x*0, 1*x*010, 101*x*1, 11*x*11

100*x*1, 10*x*10, 110*x*1, 1110*x*  
10*x*01, 1*x*011, 11*x*01, 111*x*0  
1*x*001, 10*x*11, 1101*x*, 111*x*1

*C1=* .

*xx*11*x,*

Умножение *С1\*С1* приведено в таблице 3.9.

В следствие умножения *С1\*С1* образовались множество кубов второй размерности *A2*, пустое множество *Z1* и множество *B2*:

*x*11*x*0, 1*x*01*x*, 1*x*1*x*1  
10*xx*1, 10*x*1*x*, 11*xx*1  
1*x*0*x*1, 1*xx*10, 11*x*1*x*  
1*xx*01, 1*xx*11, 111*xx*

*A2= , Z1* = Ø, *B2* = *С1.*

В результате образовалось множество *С2*:

*x*11*x*0, 1*x*01*x*, 1*x*1*x*1  
10*xx*1, 10*x*1*x*, 11*xx*1  
1*x*0*x*1, 1*xx*10, 11*x*1*x*  
1*xx*01, 1*xx*11, 111*xx  
xx*11*x,*

*C2= .*

В таблице 3.10 приведен следующий шаг поиска простых импликант – операция *С2\*С2*.

Таблица 3.10 – Поиск простых импликант (*С2\*С2*)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *C2\*C2* | *x*11*x*0 | 10*xx*1 | 1*x*0*x*1 | 1*xx*01 | 1*x*01*x* | 10*x*1*x* | 1*xx*10 | 1*xx*11 | 1*x*1*x*1 | 11*xx*1 | 11*x*1*x* | 111*xx* |
| *x*11*x*0 | ----- |  |  |  |  |  |  |  |  |  |  |  |
| 10*xx*1 |  | ----- |  |  |  |  |  |  |  |  |  |  |
| 1*x*0*x*1 |  |  | ----- |  |  |  |  |  |  |  |  |  |
| 1*xx*01 |  |  |  | ----- |  |  |  |  |  |  |  |  |
| 1*x*01*x* |  |  |  |  | ----- |  |  |  |  |  |  |  |
| 10*x*1*x* |  |  |  |  |  | ----- |  |  |  |  |  |  |
| 1*xx*10 |  |  |  |  |  |  | ----- |  |  |  |  |  |
| 1*xx*11 |  |  |  | 1*xxy*1 |  |  | 1*xx*1*y* | ----- |  |  |  |  |
| 1*x*1*x*1 |  |  | 1*xyx*1 |  |  |  |  |  | ----- |  |  |  |
| 11*xx*1 |  | 1*yxx*1 |  |  |  |  |  |  |  | ----- |  |  |
| 11*x*1*x* |  |  |  |  |  | 1*yx*1*x* |  |  |  |  | ----- |  |
| 111*xx* |  |  |  |  |  |  |  |  |  |  |  | ----- |
| *xx*11*x* |  |  |  |  | 1*xy*1*x* |  |  |  |  |  |  |  |

В следствие умножения *С2\*С2* образовались множество кубов третей размерности *A3*, множество простых импликант *Z2* и множество *B3*:

*A3* = 1*xxx*1, 1*xx*1*x* , *Z2* = *x*11*x*0, 111*xx* ,

10*xx*1, 10*x*1*x*, 11*xx*1  
1*x*0*x*1, 1*xx*10, 11*x*1*x*  
1*xx*01, 1*xx*11, *xx*11*x*  
1*x*01*x*, 1*x*1*x*1,

*В2= .*

В результате образовалось множество кубов *С3*:

*C3=* 1*xxx*1, 1*xx*1*x, xx*11*x .*

В таблице 3.11 приведен следующий шаг поиска простых импликант – операция *С3\*С3*.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *C1\*C1* | x1100 | 011x0 | 100x1 | 10x01 | 1x001 | 1001x | 1x010 | 10x10 | 1x011 | 10x11 | 1x101 | 101x1 | 110x1 | 11x01 | 1101x | 11x10 | 11x11 | 1110x | 111x0 | 111x1 |
| x1100 | ----- |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 011x0 |  | ----- |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 100x1 |  |  | ----- |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10x01 |  |  |  | ----- |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1x001 |  |  |  |  | ----- |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1001x |  |  |  |  |  | ----- |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1x010 |  |  |  |  | 1x0y1 |  | ----- |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10x10 |  |  |  |  |  |  |  | ----- |  |  |  |  |  |  |  |  |  |  |  |  |
| 1x011 |  |  |  |  | 1xy01 |  | 1x01y |  | ----- |  |  |  |  |  |  |  |  |  |  |  |
| 10x11 |  |  |  | 10xy1 |  |  |  | 10x1y |  | ----- |  |  |  |  |  |  |  |  |  |  |
| 1x101 |  |  |  |  |  |  |  |  |  |  | ----- |  |  |  |  |  |  |  |  |  |
| 101x1 |  |  | 10yx1 |  |  |  |  |  |  |  |  | ----- |  |  |  |  |  |  |  |  |
| 110x1 |  |  | 1y0x1 |  |  |  |  |  |  |  |  |  | ----- |  |  |  |  |  |  |  |
| 11x01 |  |  |  | 1yx01 |  |  |  |  |  |  |  |  |  | ----- |  |  |  |  |  |  |
| 1101x |  |  |  |  |  | 1y01x |  |  |  |  |  |  |  |  | ----- |  |  |  |  |  |
| 11x10 |  |  |  |  |  |  |  | 1yx10 |  |  |  |  |  |  |  | ----- |  |  |  |  |
| 11x11 |  |  |  |  |  |  |  |  |  | 1yx11 |  |  |  | 11xy1 |  | 11x1y | ----- |  |  |  |
| 1110x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ----- |  |  |
| 111x0 |  | y11x0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ----- |  |
| 111x1 |  |  |  |  |  |  |  |  |  |  |  | 1y1x1 | 11yx1 |  | 11y1x |  |  |  | 111xy | ----- |
| xx11x | x11y0 |  |  |  |  | 10y1x | 1xy10 |  | 1xy11 |  | 1x1y1 |  |  |  |  |  |  | 111yx |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Таблица 3.9 – Поиск простых импликант (*С1\*С1*)

Таблица 3.11 – Поиск простых импликант (*С3\*С3*)

|  |  |  |
| --- | --- | --- |
| *C3\*C3* | 1*xxx*1 | 1*xx*1*x* |
| 1*xxx*1 | ----- |  |
| 1*xx*1*x* |  | ----- |
| *xx*11*x* |  |  |

В результате выполнения умножения *С3\*С3* не было получено кубов новой размерности, т.е. множество *A4* – пустое множество, а во множество *Z3* пошли все кубы множества *C3*:

*A4* = Ø, *Z3* = 1*xxx*1, 1*xx*1*x*, *xx*11*x .*

Таким образом, cформированное множество простых импликант:

*x*11*x*0

*Z= Z0* U*Z1* U *Z2* U *Z3 = .*

111*xx*

1*xxx*1

1*xx*1*x*

*xx*11*x*

*Определение L-экстрималей*.

В таблице 3.12 и таблице 3.13 представлены операция вычитания (#) и пересечения (∩) кубов соответственно.

Таблица 3.12 – Операция вычитания импликант

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *z*#(*Z-z*) | *x*11*x*0 | 111*xx* | 1*xxx*1 | 1*xx*1*x* | *xx*11*x* |
| *x*11*x*0 | ----- | *zzzz*1 111*x*1 | *z*00*zy* 1*xxx*1 | *z*00*z*1 10*x*1*x* 1*x*01*x* 1*xx*11 | *z*0*zz*1 *x*011*x* *xx*111 |
| 111*xx* | 0*zzzz* 011*x*0 | ----- | *z*00*zz* 10*xx*1 1*x*0*x*1 | *zy*0*zz z*0*yzz z*00*zz* 10*x*1*x* 1*x*01*x* 10*x*11 1*x*011 | 0*yzzz* 00*zzz* *x*011*x* 0*x*111 *x*0111 |
| 1*xxx*1 | *yzzzy* 011*x*0 | *zzzzz* Ø | ----- | *zzzz*0 *zzzz*0 *zzzzz zzzzz*  10*x*10 1*x*010 Ø Ø | 0*zzz0 yzzzz 0zzzz* 0011*x* *x*0110 Ø 0*x*111 |
| 1*xx*1*x* | *yzz*0*z* 011*x*0 | Ø | *zzz*0*z zzz*0*z* 10*x*01 1*x*001 | ----- | *yzzzz* 0*zzzz yzzzz* 0011*x*  00110 0x111 |
| *xx*11*x* | *zzz*0*z* 01100 | Ø | *zz*0*yz zzyyz* 10*x*01 1*x*001 | *zz*0*zz zzyzz* 10010 1*x*010 | ----- |

Таким образом, из таблицы получено множество *L*-экстрималей.

*x*11*x*0

*E= .*

1*xxx*1

1*xx*1*x*

*xx*11*x*

Таблица 3.13 – Операция пересечения

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Z#(Z-z)*∩*L* | 01100 | 10001 | 10010 | 10011 | 10101 | 11001 | 11010 | 11011 | 11100 | 11101 |
| 01100 | 01100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 10*x*01 | Ø | 10001 | Ø | Ø | 10101 | Ø | Ø | Ø | Ø | Ø |
| 1*x*001 | Ø | 10001 | Ø | Ø | Ø | 11001 | Ø | Ø | Ø | Ø |
| 10010 | Ø | Ø | 10010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 1*x*010 | Ø | Ø | 10010 | Ø | Ø | Ø | 11010 | Ø | Ø | Ø |
| 0011*x* | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 00110 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 0*x*111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Из таблицы 3.6 видно, что кубы 0011*x*, 00110, 0x111 не пересекаются с кубами множества *L.* Таким образом, можно сделать вывод, что они образованы на множестве *N*, что позволяет уменьшить цену схемы. Это дает основание удалить куб *xx*11*x* из множества *Е.*

В таблице 3.14 представлено вычитание из каждого куба комплекса *L* элементов множества *E.*

Таблица 3.14 – Операция вычитания из *L* множества *E*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *L*#*E* | 01100 | 10001 | 10010 | 10011 | 10101 | 11001 | 11010 | 11011 | 11100 | 11101 |
| *x*11*x*0 | Ø | 10001 | 10010 | 10011 | 10101 | 11001 | 11010 | 11011 | Ø | 11101 |
| 1*xxx*1 | Ø | Ø | 10010 | Ø | Ø | Ø | 11010 | Ø | Ø | Ø |
| 1*xx*1*x* | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| *L1* | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Следовательно, все кубы множества *Е* покрывают все вершины множества *L.* Значит, может быть получена одна тупиковая форма:

*P3* min = *x2y1* + *x1h* + *x1y2* = *x2y1* + *x1(h* + *y2*).

**3.1.3** Минимизация функции *P4*

Множество единичных кубов *L* и множество безразличных кубов *N*:

01001, 11001

01010, 11010

*L = , N* = *xx*11*x* .

01011, 11011

01101, 11101

Откуда образуется множество *C0 = L* ⋃ *N* :

01001, 11001

01010, 11010

*C0 = .*

01011, 11011

01101, 11101

*xx*11*x*,

*Нахождение простых импликант.*

Умножение *С0\*С0*приведено в таблице 3.15.

Таблица 3.15 – Поиск простых импликант (*С0\*С0*)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *C0\*C0* | 01001 | 01010 | 01011 | 01101 | 11001 | 11010 | 11011 | 11101 |
| 01001 | ----- |  |  |  |  |  |  |  |
| 01010 |  | ----- |  |  |  |  |  |  |
| 01011 | 010*y*1 | 0101*y* | ----- |  |  |  |  |  |
| 01101 | 01*y*01 |  |  | ----- |  |  |  |  |
| 11001 | *y*1001 |  |  |  | ----- |  |  |  |
| 11010 |  | *y*1010 |  |  |  | ----- |  |  |
| 11011 |  |  | *y*1011 |  | 110*y*1 | 1101*y* | ----- |  |
| 11101 |  |  |  | *y*1101 | 11*y*01 |  |  | ----- |
| *xx*11*x* |  | 01*y*10 | 01*y*11 | 011*y*1 |  | 11*y*10 | 11*y*11 | 111*y*1 |

В результате умножения образовалось множество кубов первой размер-ности *A1* и пустое множество простых импликант *Z0*.

010*x*1, *x*1010, *x*1101, 11*x*10  
01*x*01, 01*x*10, 011*x*1, 11*x*11

*A1= , Z0* = Ø.

*x*1001, *x*1011, 110*x*1, 1101*x*  
0101*x*, 01*x*11, 11*x*01, 111*x*1

Отсюда формируется множество *B1* = *C0* – *Z0* = *C0.* После чего можно сформировать множество *C1* = *A1* ⋃ *B1*.

010*x*1, *x*1010, *x*1101, 11*x*10  
01*x*01, 01*x*10, 011*x*1, 11*x*11

*C1=* .

*x*1001, *x*1011, 110*x*1, 1101*x*  
0101*x*, 01*x*11, 11*x*01, 111*x*1

*xx*11*x,*

Умножение *С1\*С1* приведено в таблице 3.16.

В следствие умножения *С1\*С1* образовались множество кубов второй размерности *A2*, пустое множество *Z1* и множество *B2*:

01*xx*1, *x*1*x*10  
*x*10*x*1, *x*1*x*11  
*x*1*x*01, *x*11*x*1  
*x*101*x*, 11*xx*1

*A2= , Z1* = Ø, *B2* = *С1.*

01*x*1*x,* 11*x*1*x*

В результате образовалось множество *С2*:

01*xx*1, 01*x*1*x*, 11*xx*1  
*x*10*x*1, *x*1*x*10, 11*x*1*x*  
*x*1*x*01, *x*1*x*11, *xx*11*x*  
*x*101*x*, *x*11*x*1,

*C2= .*

В таблице 3.17 приведен следующий шаг поиска простых импликант – операция *С2\*С2*.

Таблица 3.17 – Поиск простых импликант (*С2\*С2*)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *C2\*C2* | 01*xx*1 | *x*10*x*1 | *x*1*x*01 | *x*101*x* | 01*x*1*x* | *x*1*x*10 | *x*1*x*11 | *x*11*x*1 | 11*xx*1 | 11*x*1*x* |
| 01*xx*1 | ----- |  |  |  |  |  |  |  |  |  |
| *x*10*x*1 |  | ----- |  |  |  |  |  |  |  |  |
| *x*1*x*01 |  |  | ----- |  |  |  |  |  |  |  |
| *x*101*x* |  |  |  | ----- |  |  |  |  |  |  |
| 01*x*1*x* |  |  |  |  | ----- |  |  |  |  |  |
| *x*1*x*10 |  |  |  |  |  | ----- |  |  |  |  |
| *x*1*x*11 |  |  | *x*1*xy*1 |  |  | *x*1*x*1*y* | ----- |  |  |  |
| *x*11*x*1 |  | *x*1*yx*1 |  |  |  |  |  | ----- |  |  |
| 11*xx*1 | *y*1*xx*1 |  |  |  |  |  |  |  | ----- |  |
| 11*x*1*x* |  |  |  |  | *y*1*x*1*x* |  |  |  |  | ----- |
| *xx*11*x* |  |  |  | *x*1*y*1*x* |  |  |  |  |  |  |

Таблица 3.16 – Поиск простых импликант (*С1\*С1*)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *C1\*C1* | 010*x*1 | 01*x*01 | *x*1001 | 0101*x* | *x*1010 | 01*x*10 | *x*1011 | 01*x*11 | *x*1101 | 011*x*1 | 110*x*1 | 11*x*01 | 1101*x* | 11*x*10 | 11*x*11 | 111*x*1 |
| 010*x*1 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01*x*01 |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *x*1001 |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0101*x* |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |  |
| *x*1010 |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |  |
| 01*x*10 |  |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| *x*1011 |  |  | *x*10*y*1 |  | *x*101*y* |  | - |  |  |  |  |  |  |  |  |  |
| 01*x*11 |  | 01*xy*1 |  |  |  | 01*x*1*y* |  | - |  |  |  |  |  |  |  |  |
| *x*1101 |  |  | *x*1*y*01 |  |  |  |  |  | - |  |  |  |  |  |  |  |
| 011*x*1 | 01*yx*1 |  |  |  |  |  |  |  |  | - |  |  |  |  |  |  |
| 110*x*1 | *y*10*x*1 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |  |
| 11*x*01 |  | *y*1*x*01 |  |  |  |  |  |  |  |  |  | - |  |  |  |  |
| 1101*x* |  |  |  | *y*101*x* |  |  |  |  |  |  |  |  | - |  |  |  |
| 11*x*10 |  |  |  |  |  | *y*1*x*10 |  |  |  |  |  |  |  | - |  |  |
| 11*x*11 |  |  |  |  |  |  |  | *y*1*x*11 |  |  |  | 11*xy*1 |  | 11*x*1*y* | - |  |
| 111*x*1 |  |  |  |  |  |  |  |  |  | *y*11*x*1 | 11*yx*1 |  |  |  |  | - |
| *xx*11*x* |  |  |  | 01*y*1*x* | *x*1*y*10 |  | *x*1*y*11 |  | *x*11*y*1 |  |  |  | 11*y*1*x* |  |  |  |

В следствие умножения *С2\*С2* образовались множество кубов третей размерности *A3*, множество простых импликант *Z2* и множество *B3*:

*A3* = *x*1*x*1*x*, *x*1*xx*1, *Z2* = Ø, *B3* = *С2.*

В результате образовалось множество кубов *С3*:

*C3 = x*1*x*1*x*, *x*1*xx*1, *xx*11*x .*

В таблице 3.18 приведен следующий шаг поиска простых импликант – операция *С3\*С3*.

Таблица 3.18 – Поиск простых импликант (*С3\*С3*)

|  |  |  |
| --- | --- | --- |
| *C3\*C3* | *x*1*x*1*x* | *x*1*xx*1 |
| *x*1*x*1*x* | ----- |  |
| *x*1*xx*1 |  | ----- |
| *xx*11*x* |  |  |

В результате выполнения умножения *С3\*С3* не было получено кубов новой размерности, т.е. множество *A4* – пустое множество, а во множество *Z3* пошли все кубы множества *C3*:

*A4* = Ø, *Z3* = *x*1*x*1*x*, *x*1*xx*1, *xx*11*x .*

Таким образом, cформированное множество простых импликант:

*x*1*x*1*x*    
*x*1*xx*1

*Z= Z0* U*Z1* U *Z2* U *Z2 = .*

*xx*11*x*

*Определение L-экстрималей*.

В таблице 3.19 и таблице 3.20 представлены операция вычитания (#) и пересечения (∩) кубов соответственно.

Таблица 3.19 – Операция вычитания импликант

|  |  |  |  |
| --- | --- | --- | --- |
| *z*#(*Z-z*) | *x*1*x*1*x* | *x*1*xx*1 | *xx*11*x* |
| *x*1*x*1*x* | ----- | *zzz*0*z* *x*1*x*01 | *z*0*zzz* *x*011*x* |
| *x*1*xx*1 | *zzzz*0 *x*1*x*10 | ----- | *zyzz*0 *x*011*x* |

*Продолжение таблицы 3.19*

|  |  |  |  |
| --- | --- | --- | --- |
| *xx*11*x* | *zz*0*zz* *x*1010 | *zz*0*yz* *x*1*x*01 | ----- |

Таким образом, из таблицы получено множество *L*-экстрималей.

*x*1*x*1*x*    
*x*1*xx*1

*E = .*

*xx*11*x*

Таблица 3.20 – Операция пересечения

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Z#(Z-z)*∩*L* | 01001 | 01010 | 01011 | 01101 | 11001 | 11010 | 11011 | 11101 |
| *x*1010 | Ø | 01010 | Ø | Ø | Ø | 11010 | Ø | Ø |
| *x*1*x*01 | 01001 | Ø | Ø | 01101 | 11001 | Ø | Ø | 11101 |
| *x*011*x* | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Из таблицы 3.20 видно, что куб *x*011*x* не пересекается с кубами множества *L.* Это дает основание удалить куб *xx*11*x* из множества *Е.*

В таблице 3.21 представлено вычитание из каждого куба комплекса *L* элементов множества *E.*

Таблица 3.21 – Операция вычитания из *L* множества *E*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *L*#*E* | 01001 | 01010 | 01011 | 01101 | 11001 | 11010 | 11011 | 11101 |
| *x*1*x*1*x* | 01001 | Ø | Ø | 01101 | 11001 | Ø | Ø | 11101 |
| *x*1*xx*1 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| *L1* | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Следовательно, может быть получена одна тупиковая форма:

*P4* min = *x2h* + *x2y2* = *x2(h* + *y2*).

Функциональная электрическая схема одноразрядного четвертичного умножителя представлена в приложении Б.

**3.2** Логический синтез одноразрядного четверичного сумматора

Одноразрядный четверичный сумматор – это комбинационное устрой-ство, 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 3.22).

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

ОЧС синтезируется для схемы первого типа, следовательно в таблице истинности необходимо выделить 16 безразличных наборов, т.к. со старших выходов ОЧУ не могут прийти коды «2» и «3».

Таблица 3.22 – Таблица истинности ОЧС

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***a1*** | ***a2*** | ***b1*** | ***b2*** | ***p*** | ***П*** | ***S1*** | ***S2*** | **Пример операции в четвертичной с/c** |
| **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 | 0 | 1 | 0+0+1=01 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0+1+0=01 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0+1+1=02 |
| 0 | 0 | 1 | 0 | 0 | x | x | x | 0+2+0=02 |
| 0 | 0 | 1 | 0 | 1 | x | x | x | 0+2+1=03 |
| 0 | 0 | 1 | 1 | 0 | x | x | x | 0+3+0=03 |
| 0 | 0 | 1 | 1 | 1 | x | x | x | 0+3+1=10 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1+0+0=01 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1+0+1=02 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1+1+0=02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1+1+1=03 |
| 0 | 1 | 1 | 0 | 0 | x | x | x | 1+2+0=03 |
| 0 | 1 | 1 | 0 | 1 | x | x | x | 1+2+1=10 |
| 0 | 1 | 1 | 1 | 0 | x | x | x | 1+3+0=10 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | 1+3+1=11 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2+0+0=02 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 2+0+1=03 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2+1+0=03 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2+1+1=10 |
| 1 | 0 | 1 | 0 | 0 | x | x | x | 2+2+0=10 |
| 1 | 0 | 1 | 0 | 1 | x | x | x | 2+2+1=11 |
| 1 | 0 | 1 | 1 | 0 | x | x | x | 2+3+0=11 |
| 1 | 0 | 1 | 1 | 1 | x | x | x | 2+3+1=12 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 3+0+0=03 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 3+0+1=10 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 3+1+0=10 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 3+1+1=11 |
| 1 | 1 | 1 | 0 | 0 | x | x | x | 3+2+0=11 |
| 1 | 1 | 1 | 0 | 1 | x | x | x | 3+2+1=12 |
| 1 | 1 | 1 | 1 | 0 | x | x | x | 3+3+0=12 |

*Продолжение таблицы 3.22*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | x | x | x | 3+3+1=13 |

Минимизацию переключательных функций проведём с помощью карт Вейча и Карно. Для функций *П, S2* заполненные карты приведены на рисунке 3.2, а для *S1* – на рисунке 3.3 ( символом «х» отмечены наборы, в которых функции могут принимать произвольное значение, т.е. безличные наборы).

*p*

*b2*

*b1*

*p*

*a1*

*a2*

*p*

*b2*

*b1*

*p*

*a1*

*a2*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | x | x | x |  | 1 |  |  |
| x | x | x | x | 1 | 1 | 1 |  |
| x | x | x | x |  |  |  |  |
| x | x | x | x |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | x | x | x | 1 |  | 1 |  |
| x | x | x | x |  | 1 |  | 1 |
| x | x | x | x |  | 1 |  | 1 |
| x | x | x | x | 1 |  | 1 |  |

*a б*

Рисунок 3.2 – Минимизация с помощью карты Вейча   
*a* – функции *П, б* – функции *S2*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a2\b1b2p | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 |  |  | 1 |  | x | x | x | x |
| 01 |  | 1 | 1 | 1 | x | x | x | x |
| 11 | 1 |  |  |  | x | x | x | x |
| 10 | 1 | 1 |  | 1 | x | x | x | x |

Рисунок 3.3 – Минимизация функции *S1* с помощью карты Карно

Следовательно:

*Пmin = .*

*S1 min =.*

*S2 min =*

Функциональная электрическая схема одноразрядного четвертичного сумматора представлена в приложении В.

**3.3** Логический синтез преобразователя множителя

Преобразователь множителя – это комбинационное устройство, имею-щее 3 двоичных входов (2 преобразуемых разряда и вход переноса) и 4 двоич-ных выхода.

Принцип работы преобразователя множителя представлен с помощью таблицы истинности (таблица 3.23).

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

Таблица 3.23 – Таблица истинности преобразователя множителя

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Анализиру-емая пара разрядов** | | **Перенос из младшей пары** | **Знак преоб-разованной пары** | **Преобразо-ванная пара** | | **Перенос в старшую пару** |
| ***z1*** | ***z2*** | ***p*** | ***Z*** | ***Q1*** | ***Q2*** | ***S*** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

Минимизацию переключательных функций проведём с помощью карт Вейча и Карно. Для функций *Z, S*заполненные карты приведены на рисунке 3.4, а для *Q1*, *Q2* – на рисунке 3.5.

*Z2*

*p*

*Z1*

*Z2*

*p*

*Z1*

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | 1 |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 1 |  |
|  |  |  |  |

*a б*

Рисунок 3.4 – Минимизация с помощью карты Вейча:

*a* – функция *Z, б* – функция *S*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *z1\z2p* | 00 | 01 | 11 | 10 |
| 0 |  |  | 1 |  |
| 1 | 1 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *z1\z2p* | 00 | 01 | 11 | 10 |
| 0 |  | 1 |  | 1 |
| 1 |  | 1 |  | 1 |

*a б*

Рисунок 3.5 – Минимизация с помощью карты Карно

*a* – функции *Q1, б* – функции *Q2*

Следовательно:

*Zmin =z1* (*z2 p*)*;*

*Q1min = z11*+ *z2p;*

*Q2min = + p + z2  = + z2  p;*

*Smin =p + z2 = z2  p.*

Функциональная электрическая схема преобразователя множителя приведена в приложении Г.

**4** СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ 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* .

На основе мультиплексора будет проведен синтез одноразрядного четвертичного сумматора. Таблица истинности ОЧС была взята как основа для составления таблицы истинности ОЧС на основе мультиплексора (таблица 4.1). Было выбрано два управляющих входа (*a1**–* *А0*, *a2 –* *А1*) и соответственно четыре информационных входа (*D0 – D3*).

Таблица 4.1 *–* Таблица истинности ОЧС на основе мультиплексора

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер**  **вход. набора** | **Упр. в.** | |  | | | **Информационные входы** | | | | | |
| *a1* | *a2* | *b1* | *b2* | *p* | ***П*** | **Ф.** | ***S1*** | **Ф.** | ***S2*** | **Ф.** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | «0» | 0 | *b2 ∙ p* | 0 | *b2  p* |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | x | x | x |
| 0 | 0 | 1 | 0 | 1 | x | x | x |
| 0 | 0 | 1 | 1 | 0 | x | x | x |
| 0 | 0 | 1 | 1 | 1 | x | x | x |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | «0» | 0 | *b2 + p* | 1 |  |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | x | x | x |
| 0 | 1 | 1 | 0 | 1 | x | x | x |
| 0 | 1 | 1 | 1 | 0 | x | x | x |
| 0 | 1 | 1 | 1 | 1 | x | x | x |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | *b2 ∙ p* | 1 |  | 0 | *b2  p* |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | x | x | x |
| 1 | 0 | 1 | 0 | 1 | x | x | x |
| 1 | 0 | 1 | 1 | 0 | x | x | x |
| 1 | 0 | 1 | 1 | 1 | x | x | x |

*Продолжение таблицы 4.1*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 3 | 1 | 1 | 0 | 0 | 0 | 0 | *b2 + p* | 1 |  | 1 |  |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | x | x | x |
| 1 | 1 | 1 | 0 | 1 | x | x | x |
| 1 | 1 | 1 | 1 | 0 | x | x | x |
| 1 | 1 | 1 | 1 | 1 | x | x | x |

*Функция П.*

Для нулевого, первого входных наборов (ВН) функция принимает значение «0», а для второго третьего ВН необходимо провести минимизацию с помощью карт Карно (рисунок 4.1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *b1\b2p* | 00 | 01 | 11 | 10 |
| 0 |  |  | 1 |  |
| 1 | x | x | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *b1\b2p* | 00 | 01 | 11 | 10 |
| 0 |  | 1 | 1 | 1 |
| 1 | x | x | x | x |

*a б*

Рисунок 4.1 – Минимизация с помощью карты Карно функции *П*на третьем ВН

Следовательно:

1. *Пmin = b2 ∙ p;*
2. *Пmin = b2 + p.*

*Функция S1.*

Из таблицы истинности (таблица 4.1) видно, что для нулевого и второго ВН функция принимает одинаковые значения для каждого из наборов соответственно, а для первого и третьего ВН необходимо провести минимизацию с помощью карт Карно (рисунок 4.2).

Следовательно:

1. *S1min = b2 ∙ p;*

*б) S1min = b2 + p.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *b1\b2p* | 00 | 01 | 11 | 10 |
| 0 |  | 1 | 1 | 1 |
| 1 | x | x | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *b1\b2p* | 00 | 01 | 11 | 10 |
| 0 |  | 1 | 1 | 1 |
| 1 | x | x | x | x |

*a б*

Рисунок 4.2 – Минимизация с помощью карты Карно функции *S1   
a* – на нулевом ВН, *б* – на первом ВН

*Функция S2.*

Для нулевого и второго ВН функция принимает одинаковые значения, а также для первого и третьего ВН. Следовательно достаточно провести минимизацию один раз для одного из одинаковых наборов нулевого или второго ВН и для первого или третьего ВН(рисунок 4.3).

Следовательно:

*а) S2min = b2  p;*

*б) S2min =.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *b1\b2p* | 00 | 01 | 11 | 10 |
| 0 |  | 1 |  | 1 |
| 1 | x | x | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *b1\b2p* | 00 | 01 | 11 | 10 |
| 0 | 1 |  | 1 |  |
| 1 | x | x | x | x |

*a б*

Рисунок 4.3 – Минимизация с помощью карты Карно функции *S1   
a* – на нулевом и втором ВН, *б* – на первом и третьем ВН

Функциональная электрическая схема одноразрядного четвертичного сумматора на основе мультиплексоров приведена в приложении Д.

**5** ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ

**5.1** Эффективность минимизации *K* функций

Эффективность минимизации *K* функций определяется по формуле:

*K* = , (5.1)

где *Cf* *−* цена схемы до минимизации,

*Cf min −* цена схемы после минимизации.

В таблицах 5.1, 5.2 представлена оценка эффективности минимизации функций ОЧУ и ОЧС соответственно.

Таблица 5.1 *–* Оценка эффективности минимизации функций ОЧУ

|  |  |  |  |
| --- | --- | --- | --- |
| *f* | *Cf* | *Cf min* | *K* |
| *P1* | 1+2 = 3 | 1+2 = 3 | 1 |
| *P2* | 2∙5+3+2 =15 | 3+1 = 4 | 3,75 |
| *P3* | 10∙5+5+10 = 65 | 2∙2+3+2+1=10 | 6,5 |
| *P4* | 8∙5+4+8 = 52 | 2+2 = 4 | 13 |

Таблица 5.2 *–* Оценка эффективности минимизации функций

|  |  |  |  |
| --- | --- | --- | --- |
| *f* | *Cf* | *Cf min* | *K* |
| *П* | 4∙5+4+4 = 28 | 2∙3+2+6+2 =16 | 1,75 |
| *S1* | 8∙5+5+8 = 53 | 3∙4+10+4= 26 | 2,04 |
| *S2* | 8∙5+4+8=52 | 4∙3+7+4=23 | 2,26 |

**5.2** Временные затраты на умножение

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

Формула расчета временных затрат на умножение:

*Tу* = *n*\*( *ТПМ* +*TФДК* + *ТОЧУ + n*\**ТОЧС* + *Тсдвига* ), (5.2)

где *ТПМ −* время преобразования множителя;

*TФДК –* время формирования дополнительного кода множимого;

*ТОЧУ −* время умножения на одном ОЧУ;

*ТОЧС −* время формирования единицы переноса в ОЧС;

*Тсдвига −* время сдвига частичной суммы;

*n –* количество четвертичных разрядов*.*

ЗАКЛЮЧЕНИЕ

В данной курсовой работе был спроектирован сумматор-умножитель. Его работа была оптимизирована при помощи умножения на два разряда (в двоично-четвертичном коде). Умножение производилось на базе алгоритма «А» (умножение начинается с младших разрядом со сдвигом суммы вправо).

Отдельные элементы сумматора-умножителя были оптимизированы при помощи алгоритма Рота (ОЧУ) и карт Карно-Вейча (ОЧС) и приведены к заданному базису.

Была разработана схема умножения двух числе в заданной кодировке. Были спроектированы и изображены: структурная схема устройства; а также функциональные схемы его узлов (в том числе и на базе мультиплексоров).

Также была получена формула расчёта времени работы устройста.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Луцик, Ю. А. Учебное пособие по курсу «Арифметические и логи-ческие основы вычислительной техники» / Ю. А. Луцик, И. В. Лукьянова, М. П. Ожигина. – Минск : МРТИ, 2001. – 77 с.

2 Луцик, Ю. А. Арифметические и логические основы вычислительной техники : метод. пособие / Ю. А. Луцик, И. В. Лукьянова. – Минск : МРТИ, 2004. – 35 с

3 Cавельев, А. Я. Прикладная теория цифровыз автоматов / А. Я. Савельев. – М. : Высшая шк., 1987. – 272 с.

ПРИЛОЖЕНИЕ А

(*обязательное)*

Структурная схема сумматора-умножителя

ПРИЛОЖЕНИЕ Б

(*обязательное)*

Функциональная электрическая схема одноразрядного четвертичного умножителя

ПРИЛОЖЕНИЕ В

(*обязательное)*

Функциональная электрическая схема одноразрядного четвертичного сумматора

ПРИЛОЖЕНИЕ Г

(*обязательное)*

Функциональная электрическая схема преобразователя множителя

ПРИЛОЖЕНИЕ Д

(*обязательное)*

Функциональная электрическая схема одноразрядного четвертичного сумматора на основе мультиплексоров

ПРИЛОЖЕНИЕ Е

(*обязательное)*

Ведомость к курсовой работе