Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы вычислительной техники

К ЗАЩИТЕ ДОПУСТИТЬ

\_\_\_\_\_\_\_\_\_\_И. В. Лукьянова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе  
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 515 ПЗ

Студент

Руководитель

М. Н. Алексеев

И. В. Лукьянова

МИНСК 2019

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Арифметические и логические основы вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_\_Б. В. Никульшин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_20 г.

ЗАДАНИЕ

по курсовой работе студента

Алексеева Матвея Николаевича

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-четверичных чисел».
2. Срок сдачи студентом законченной работы: 1 июня 2019 г.
3. Исходные данные к работе:
   1. исходные сомножители: Мн10 = 36,92; Мт10 = 53,35
   2. алгоритм умножения: Б.
   3. метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах.
   4. коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования; 04 – 00, 14 – 01, 24 –11 , 34 – 10.
   5. тип синтезируемого умножителя: 1.
   6. логический базис для реализации ОЧС: ИЛИ, mod 2, 1; метод минимизации – алгоритм Рота для одного выхода, метод карт Карно-Вейча – для остальных.
   7. логический базис для реализации ОЧУ: И-НЕ; метод минимизации – карты Карно-Вейча.
4. Содержание пояснительной записки (перечень подлежащих разработке вопросов): Введение.

1. Разработка алгоритма умножения.

2. Разработка структурной схемы сумматора-умножителя.

3. Разработка функциональных схем основных узлов сумматора-умножителя.

4. Синтез комбинационных схем устройств на основе мультиплексоров.

5. Оценка результатов разработки. Заключение. Список литературы.

1. Перечень графического материала:
   1. Сумматор-умножитель первого типа. Схема электрическая структурная.
   2. Одноразрядный четверичный сумматор. Схема электрическая функциональная.
   3. Одноразрядный четверичный умножитель. Схема электрическая функциональная.
   4. Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.
   5. Преобразователь множителя. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсовой работы | Объём этапа, % | Срок выполнения этапа | Примечания |
| Разработка алгоритма умножения | 10 | 22.02-13.03 |  |
| Разработка структурной схемы сумматора-умножителя | 10 | 14.03-27.03 | С выполнением чертежа |
| Разработка функциональных схем основных узлов сумматора-умножителя | 50 | 28.03-08.05 | С выполнением чертежей |
| Синтез комбинационных схем устройств на основе мультиплексоров | 10 | 09.05-22.05 | С выполнением чертежа |
| Завершение оформления пояснительной записки | 20 | 23.05-05.06 |  |

Дата выдачи задания: 22 февраля 2019 г.

Руководитель И.В. Лукьянова

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

Введение 5

1. Разработка алгоритма умножения 6

2. Разработка структурной схемы сумматора-умножителя 9

3. Разработка функциональных схем основных узлов сумматора-

умножителя 10

4. Синтез комбинационных схем устройств на основе мультиплексоров 22

5. Логический синтез преобразователя множителя 24

6. Временные затраты на умножение 26

Заключение 27

Список использованных источников 28

Приложение А 29

Приложение Б 30

Приложение В 31

Приложение Г 32

Приложение Д 33

Приложение Е 34

**ВВЕДЕНИЕ**

Курсовое проектирование является обязательным элементом подготовки специалиста с высшим образованием и одной из форм текущей аттестации студента по учебной дисциплине.

Целью данной курсовой работы является разработка сумматора-умножителя первого типа для алгоритма умножения Б в прямом коде на два разряда одновременно. Это и является целью данной работы.

Для достижения цели необходимо решить ряд задач:

1. Разработать алгоритм умножения и оценить погрешности вычислений.

2. Разработать структурную схему сумматора-умножителя первого типа.

3. Разработать функциональные схемы основных узлов сумматора-умножителя в заданных логических базисах.

4. Разработать комбинационную схему на основе мультиплексора.

5. Рассчитать время умножения.

**1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ**

1. Перевод сомножителей из десятичной системы счисления в четверичную.

Мн = 36,92; Мт = 53,35.

**Множимое**

|  |
| --- |
| 0.92  \* 4 |
| 3.68 |
| \* 4 |
| 2.72  \* 4 |
| 2.88 |

|  |  |  |  |
| --- | --- | --- | --- |
| \_36  36 | 4 | | |
| \_ 9  8 | 4  2 | |
| 0 |  |  | |
|  | 1 |  | |
|  | | | |

Мн4 = 210,322.

В соответствии с кодировкой множимого:

Мн2/4 =110100,101111.

**Множитель**

|  |
| --- |
| 0.35  \* 4 |
| 1.40  \* 4 |
| 1.60  \* 4 |
| 2.40 |
|  |

|  |
| --- |
|  |
|
|
|
|

|  |  |  |  |
| --- | --- | --- | --- |
| \_53  52 | 4 | | |
| \_13  12 | 4  3 |
| 1 |
|  | 1 |
|  | | | |

Мт4 = 311,112

В соответствии с обычной весомозначной кодировкой множителя:

Мт2/4 = 110101,010110.

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

Мн = 0.110100101111 Рмн = 0.0010 + 034  – закодировано по заданию

Мт = 0.110101010110 Рмт = 0.0011 + 034 – закодировано традиционно

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

Порядок произведения будет следующим:

Рмн = 0.0010 +034

Рмт = 0.0011 +034

= 0.0111 +124

Результат закодирован в соответствии с заданием на кодировку множимого.

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

Для умножения мантисс необходимо предварительно преобразовать

множитель, чтобы исключить диаду 11 (34), заменив ее на триаду 101.

Преобразованный множитель имеет вид:

Мтп4  = или Мтп2 =

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

Таблица 1.1 - Перемножение мантисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Четверичная с/c** | | **Двоично-четверичная с/с** | | **Комментарий** |
| **1** | | **2** | | **3** |
| 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  3.  3.  0.  0. | 000000000000  000001021310  000001021300  000002103220  000003131130  000021032200  000030223330  000210322000  000301211330  002103220000  003011031330  312301133333  321312231323  210322000000  132300231323 | 00.  00.  00.  00.  00.  00.  00.4 00.  00.4 01.  00.  10.  10.  00.  00. | 000000000000000000000000  000000000001001101100100  000000000001001101100100  000000000011010010111100  000000000010011001011000  000000001101001011110000  000000001000111110101000  000000110100101111000000  000000100001110101101000  000011010010111100000000  000010000101001001101000  100111100001011010101010  101101100111111001101110  110100101111000000000000  011011100000111001101110 | ∑0ч  П1ч = Мн \* 2  ∑1ч  П2ч = Мн \* 1 \* 41  ∑2ч  П3ч = Мн \* 1 \* 42  ∑3ч  П4ч = Мн \* 1 \* 43  ∑4ч  П5ч = Мн \* 1 \* 44  ∑5ч  П6ч = Мн \* -1 \* 45  ∑6ч  П7ч = Мн \* 1 \* 46  ∑7ч |

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

Мн \* Мт4 = 132300,231323

PМн\*Мт = 0

Мн \* Мт10 = 1968,7175292969.

Результат прямого перемножения операндов дает результат:

Мн10 \* Мт10 = 1969,682

Абсолютная погрешность:

∆ = 1969,6820 – 1968,7175 = 0,9645

Относительная погрешность:

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 первого типа для алгоритма умножения «Б» представлена в приложении А.

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

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

Таблица 2.1 – Режимы работы формирователя дополнительного кода

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

**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 – 01; 1 – 11; 2 – 00; 3 – 10.

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 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 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=10 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | Выход – код «00» |
| 0 | 0 | 1 | 1 | 0 | х | х | x | х | 0·3=12 |
| 0 | 0 | 1 | 1 | 1 | х | x | х | х | 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 | 1 | 1·2=02 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Выход – код «01» |
| 0 | 1 | 1 | 1 | 0 | х | x | х | x | 1·3=03 |
| 0 | 1 | 1 | 1 | 1 | x | х | x | х | 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 «03» |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 3·1=03 |

*Продолжение таблицы 3.1*

x1

y1

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

P1(сднф) = x1x2y1y2h + x1x2y1y2h

P3(сднф) = x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h

P4(скнф) = x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h + x1x2y1y2h

Минимизацию переключательный функций P1, P2, P3, P4 проведём с помощью карт Карно. В силу того, что функция P1 не принимает значения равные единице, то P1(мндф) = 0. Заполненные карты для функций P2, P3 и P4 приведены на рисунках 3.1, 3.2 и 3.3 соответственно. В рисунках 3.1.1 – 3.1.3 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

# X1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 a2 | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | 0 | 0 | 0 | 0 | x | x | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 | x | x | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | x | x | 0 | 1 |
| 10 | 0 | 0 | 0 | 0 | x | x | 0 | 1 |

b1 b2 p

Рисунок 3.1 – Минимизация функции *P2* при помощи карты Вейча

# X1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 a2 | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | 0 | 0 | 0 | 0 | x | x | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 | x | x | 0 | 1 |
| 11 | 0 | 1 | 1 | 1 | x | x | 1 | 0 |
| 10 | 0 | 1 | 1 | 1 | x | x | 1 | 1 |

b1 b2 p

Рисунок 3.2 – Минимизация функции *P3* при помощи карты Вейча

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 a2 | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | 0 | 0 | 0 | 0 | x | x | 0 | 0 |
| 01 | 0 | 1 | 1 | 1 | x | x | 1 | 1 |
| 11 | 0 | 1 | 1 | 1 | x | x | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 | x | x | 0 | 1 |

b1 b2 p

Рисунок 3.3 – Минимизация функции *P4* при помощи карты Вейча

Следовательно:

P2(мднф) =

P3(мднф) =

P4(мднф) =

Запишем результат минимизации в логическом базисе А6:

P1 =

P3 =

P4 =

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

*=*

*=*

*=*

*=*

Построение функциональной схемы ОЧУ состоит из объединения функциональных схем для каждого выхода.

Функциональная схема ОЧУ в заданном базисе представлена в приложении Б.

**3.2. Логический синтез одноразрядного четверичного сумматора**

Одноразрядный четверичный сумматор (ОЧС) – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда одного слагаемого, 2 разряда второго слагаемого и вход переноса) и 3 двоичных выхода.

Принцип работы ОЧС представлен с помощью таблицы истинности (таблица 3.2)

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

В таблице 3.2 выделено 16 безразличных наборов, т.к. со старших выходов 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 | 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 | 1 | 0+1+1=02 |
| 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 | x | x | x | 0+2+0=02 |
| 0 | 0 | 1 | 1 | 1 | x | x | x | 0+2+1=03 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1+0+0=01 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1+0+1=02 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1+1+0=02 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1+1+1=03 |
| 0 | 1 | 1 | 0 | 0 | x | x | x | 1+3+0=10 |
| 0 | 1 | 1 | 0 | 1 | x | x | x | 1+3+1=11 |
| 0 | 1 | 1 | 1 | 0 | x | x | x | 1+2+0=03 |
| 0 | 1 | 1 | 1 | 1 | x | x | x | 1+2+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 | 1 | 0 | 0 | 3+1+0=10 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 3+1+1=11 |
| 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 | x | x | x | 3+2+0=11 |
| 1 | 0 | 1 | 1 | 1 | x | x | x | 3+2+1=12 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 2+0+0=02 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 2+0+1=03 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2+1+0=03 |

*Продолжение таблицы 3.2*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 2+1+1=10 |
| 1 | 1 | 1 | 0 | 0 | x | x | x | 2+3+0=11 |
| 1 | 1 | 1 | 0 | 1 | x | x | x | 2+3+1=12 |
| 1 | 1 | 1 | 1 | 0 | x | x | x | 2+2+0=10 |
| 1 | 1 | 1 | 1 | 1 | x | x | x | 2+2+1=11 |

П(cднф) = a1a2b1b2p + a1a2b1b2p + a1a2b1b2p + a1a2b1b2p

S1 (cднф) = a1a2b1b2p + a1a2b1b2p + a1a2b1b2p + a1a2b1b2p + a1a2b1b2p + a1a2b1b2p +

+ a1a2b1b2p + a1a2b1b2p

S2 (cднф) = a1a2b1b2p + a1a2b1b2p + a1a2b1b2p + a1a2b1b2p + a1a2b1b2p + a1a2b1b2p +

+ a1a2b1b2p + a1a2b1b2p

Минимизацию функции 𝑆2 проведем с помощью алгоритма Рота.

Минимизацию функций П, S1 проведём с помощью карт Карно. Заполненные карты для функций П, S1 приведены на рисунках 3.4 и 3.5 соответственно. На картах символом «х» отмечены безразличные наборы, на которых функция может принимать произвольное значение.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 a2 | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | x | x | 1 | 0 | 0 | 0 | x | x |
| 01 | x | x | 0 | 0 | 0 | 0 | x | x |
| 11 | x | x | 0 | 0 | 0 | 0 | x | x |
| 10 | x | x | 1 | 0 | 1 | 1 | x | x |

b1 b2 p

Рисунок 3.4 – Минимизация функции П при помощи карты Карно

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 a2 | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | x | x | 1 | 0 | 1 | 0 | x | x |
| 01 | x | x | 1 | 0 | 1 | 0 | x | x |
| 11 | x | x | 0 | 1 | 0 | 1 | x | x |
| 10 | X | x | 0 | 1 | 0 | 1 | x | x |

b1 b2 p

Рисунок 3.5 – Минимизация функции S1 при помощи карты Карно

Следовательно:

П(мднф) = a2b1p + a1a2b1

S1(мднф) = a1b1p + a1b1p + a1b1p + a1b1p

**Минимизация функции S2:**

Определим множество единичных кубов

*L* =

и множество безразличных кубов

.

Минимизацию безразличных кубов проведём с помощью карты Карно. Для безразличных кубов заполненная карта приведена на рисунке 3.6, где символом «x» отмечены наборы, на которых функция не определена.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | x | x | 0 | 0 | 0 | 0 | x | x |
| 01 | x | x | 0 | 0 | 0 | 0 | X | x |
| 11 | x | x | 0 | 0 | 0 | 0 | X | x |
| 10 | x | x | 0 | 0 | 0 | 0 | X | x |

b1 b2 p

a1 a2

Рисунок 3.6 – Минимизация безразличных кубов с помощью карт Карно

Множество безразличных наборов после минимизации:

.

Сформируем множество *С0* = *L* ⋃ *N*:

.

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 3.3.

Таблица 3.3 – Поиск простых импликант (*С0\*С0*)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *С0\*С0* | 00111 | 01010 | 01011 | 01110 | 10011 | 10110 | 10111 | 11010 | xxx0x |
| 00111 | - |  |  |  |  |  |  |  |  |
| 01010 | 0yy1y | - |  |  |  |  |  |  |  |
| 01011 | 0yy11 | 0101y | - |  |  |  |  |  |  |
| 01110 | 0y11y | 01y10 | 01y1y | - |  |  |  |  |  |
| 10011 | y0y11 | yy01y | yy011 | yyy1y | - |  |  |  |  |
| 10110 | y011y | yyy10 | yyy1y | yy110 | 10y1y | - |  |  |  |
| 10111 | y0111 | yyy1y | yyy11 | yy11y | 10y11 | 1011y | - |  |  |
| 11010 | yyy1y | y1010 | y101y | y1y10 | 1y01y | 1yy10 | 1yy1y | - |  |
| xxx0x | 001y1 | 010y0 | 010y1 | 011y0 | 100y1 | 101y0 | 101y1 | 110y0 | - |
| A1 | x0111  001x1 | 0101x  01x10  x1010  010x0 | 010x1 | 11x0 | 10x11  100x1 | 1011x  101x0 | 101x1 | 110x0 | Ø |

*Продолжение таблицы 3.3*

Множество *Z0* кубов, не участвовавших в образовании новых кубов, пустое.

Сформируем множество *B1 :*

*B1 = C0\ Z0 = C0\* Ø = *C0*.

После этой операции сформируется новое множество кубов *C1=A1⋃ B1*

.

В таблице 3.4 приведён следующий шаг поиска простых импликант с

помощью операции *С1\*С1*.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C1\*C1 | x0111 | 001x1 | 0101x | 01x10 | x1010 | 010x0 | 010x1 | 011x0 | 10x11 | 100x1 | 1011x | 101x0 | 101x1 |  |
| x0111 | - |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 001x1 | 00111 | - |  |  |  |  |  |  |  |  |  |  |  |  |
| 0101x | 0yy11 | 0yy11 | - |  |  |  |  |  |  |  |  |  |  |  |
| 01x10 | 0y11y | 0y11y | 01010 | - |  |  |  |  |  |  |  |  |  |  |
| x1010 | xyy1y | 0yy1y | 01010 | 01010 | - |  |  |  |  |  |  |  |  |  |
| 010x0 | 0yy1y | 0yyxy | 01010 | 01010 | 01010 | - |  |  |  |  |  |  |  |  |
| 010x1 | 0yy11 | 0yyx1 | 01011 | 0101y | 0101y | 010xy | - |  |  |  |  |  |  |  |
| 011x0 | 0y11y | 0y1xy | 01y10 | 01110 | 01y10 | 01yx0 | 01yxy | - |  |  |  |  |  |  |
| 10x11 | 10111 | y0111 | yy011 | yyx1y | 1y01y | yy01y | yy011 | yy11y | - |  |  |  |  |  |
| 100x1 | 10y11 | y0yx1 | yy011 | yy01y | 1y01y | yy0xy | yy0x1 | yyyxy | 10011 | - |  |  |  |  |
| 1011x | 10111 | y0111 | yyy1x | yy110 | 1yy10 | yyy10 | yyy11 | yy110 | 10111 | 10y11 | - |  |  |  |
| 101x0 | 1011y | y01xy | yyy10 | yy110 | 1yy10 | yyyx0 | yyyxy | yy1x0 | 1011y | 10yxy | 10110 | - |  |  |
| 101x1 | 10111 | y01x1 | yyy11 | yy11y | 1yy1y | yyyxy | yyyx1 | yy1xy | 10111 | 10yx1 | 10111 | 101xy | - |  |
| 110x0 | 1yy1y | yyyxy | y1010 | y1010 | 11010 | y10x0 | y10xy | y1yx0 | 1y01y | 1y0xy | 1yy10 | 1yyx0 | 1yyxy | -  11000 |
| xxx0x | x01y1 | 00101 | 010yx | 01xy0 | x10y0 | 01000 | 01001 | 01100 | 10xy1 | 10001 | 101yx | 10100 | 10101 |  |
| A2 | x01x1 | x01x1 | 010xx | 01xx0 | x10x0 | 010xx  01xx0  x10x0 | Ø | Ø | 10xx1 | 10xx1 | 101xx | 101xx | Ø | Ø |

Таблица 3.4 – Поиск простых импликант *С1\*С1.*

110x0

Множество *Z1* кубов, не участвовавших в образовании новых кубов, пустое.

Сформируем множество *B2 = C1\ Z1 = C1\* Ø = *C1*.

После этой операции сформируется новое множество кубов *C2=A2⋃ B2*:

.

В таблице 3.5 приведён следующий шаг поиска простых импликант – операция *С2\*С2*.

Таблица 3.5 – Поиск простых импликант *С2\*С2*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *С2\*С2* | x01x1 | 010xx | 01xx0 | x10x0 | 10xx1 | 101xx |
| x01x1 | - |  |  |  |  |  |
| 010xx | 0yyx1 | - |  |  |  |  |
| 01xx0 | 0y1xy | 010x0 | - |  |  |  |
| x10x0 | xyyxy | 010x0 | 010x0 | - |  |  |
| 10xx1 | 101x1 | yy0x1 | yyxxy | 1y0xy | - |  |
| 101xx | 101x1 | yyyxx | yy1x0 | 1yyx0 | 101x1 | - |
| xxx0x | x0101 | 0100x | 01x00 | x1000 | 10x01 | 1010x |
| A3 | Ø | Ø | Ø | Ø | Ø | Ø |

Новых кубов (третьей размерности) не образовалось.

Получено множество *= .*

На этом заканчивается этап поиска простых импликант, т.к. |*C3*|≤1. Множество простых импликант:

*Z* =*Z1* ⋃ *Z2* ⋃ *Z3* = *.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| z#(Z\z) | x01x1 | 010xx | 01xx0 | x10x0 | 10xx1 | 101xx | xxx0x |
| x01x1 | - | 010xx | 01xx0 | x10x0 | 100x1 | 101x0 | x1x0x xx00x xxx00 |
| 010xx | x01x1 | - | 011x0 | 110x0 | 100x1 | 101x0 | 111x0x x110x 1x00x x000x  1xx00 x0x00 xx100 |
| 01xx0 | x01x1 | 010x1 | - | 110x0 | 100x1 | 101x0 | 11x0x 1110x x1101 1x00x  x000x 1xx00 x0x00 1x100  x0100 |
| x10x0 | x01x1 | 010x1 | 011x0 | - | 100x1 | 101x0 | 1110x 11x01 1110x x1101  1000x 1x001 x000x 10x00 1x100 x0x00 1x100 x0100 |
| 10xx1 | 001x1 | 010x1 | 011x0 | 110x0 | - | 101x0 | 1110x 11x01 1110x x1101  10000 11001 0000x x0000  10x00 1x100 x0x00 1x100  x0100 |
| 101xx | 001x1 | 010x1 | 011x0 | 110x0 | 100x1 | - | 1110x 11x01 1110x x1101  10000 11001 0000x x0000  10000 11100 00x00 x0000  11100 00100 |
| xxx0x | 00111 | 01011 | 01110 | 11010 | 10011 | 10110 | - |

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

Таблица 3.6 – Поиск *L*-экстремалей

В таблице 3.6 из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты *Z#(Z-z)*.

Получили кубы, “подозрительные” на *L*-экстремальность. Проверяем в таблице 3.7.

Таблица 3.7 – Проверка на *L-*экстремальность

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Z*#*(Z-z)* ⋂ *L* | 00111 | 01010 | 01011 | 01110 | 10011 | 10110 | 10111 | 11010 |
| 00111 | 00111 | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 01011 | Ø | Ø | 01011 | Ø | Ø | Ø | Ø | Ø |
| 01110 | Ø | Ø | Ø | 01110 | Ø | Ø | Ø | Ø |
| 11010 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 11010 |
| 10011 | Ø | Ø | Ø | Ø | 10011 | Ø | Ø | Ø |
| 10110 | Ø | Ø | Ø | Ø | Ø | 10110 | Ø | Ø |
| 1110x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 11x01 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 1110x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| x1101 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 10000 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 11001 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 0000x | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| x0000 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 10000 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 11100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 00x00 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| x0000 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 11100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |
| 00100 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Получено множество L-экстремалей:

*E* = .

Далее необходимо проанализировать, какие из исходных единичных кубов не покрыты найденной *L*-экстремалью. Анализ осуществляется с помощью таблицы 3.8.

Таблица 3.8 – Поиск непокрытых наборов

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L#E | 00111 | 01010 | 01011 | 01110 | 10011 | 10110 | 10111 | 11010 |
| x01x1 | Ø | 01010 | 01011 | 01110 | 10011 | 10110 | Ø | 11010 |
| 010xx | Ø | Ø | Ø | 01110 | 10011 | 10110 | Ø | 11010 |
| 01xx0 | Ø | Ø | Ø | Ø | 10011 | 10110 | Ø | 11010 |
| x10x0 | Ø | Ø | Ø | Ø | 10011 | 10110 | Ø | Ø |
| 10xx1 | Ø | Ø | Ø | Ø | Ø | 10110 | Ø | Ø |
| 101xx | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

Минимальное покрытие - множество L-экстремалей:

E = { x01x1, 010xx, 01xx0, x10x0, 10xx1, 101xx }

S2 min = a2b1p + a1a2b1 + a1a2p + a2b1p + a1a2p + a1a2b1.

Запишем результат минимизации в логическом базисе А3:

П = a2 + b1 + p + a1 + a2 + b1

S1 = a1 + b1 + p + a1 + b1 + p + a1 + b1 + p + a1 + b1 + p

S2 = a2 + b1 + p + a1 + a2 + b1 + a1 + a2 + p + a2 + b1 + p + a1+a2 + p + a1 + a2 + b1

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

*=*

*=*

*=*

Построение функциональной схемы ОЧС состоит из объединения функциональных схем для каждого выхода.

Функциональная схема ОЧС в заданном базисе представлена в приложении В.

**4. Синтез комбинационных схем на основе мультиплексоров**

Мультиплексор – это логическая схема, имеющая n информационных входов, m управляющих входов и один выход. При этом должно выполняться условие 𝑛= 2𝑚.Переключательные функции (ПФ) пяти переменных можно реализовать на мультиплексоре «один из восьми». Управляющее поле такого мультиплексора определяется тремя переменными, поэтому число групп с одинаковыми значениями этих переменных будет равно восьми. Реализация нескольких ПФ потребует для каждой ПФ отдельного мультиплексора.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a1** | **a2** | **b1** | **b2** | **p** | **П** | **Функция** | **S1** | **Функция** | **S2** | **Функция** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |  | 1 |  |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |  | 1 |  | 0 |  |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |  | 1 |  | 1 |  |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |

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

Таблица 4.1 – Таблица синтеза дополнительных логических схем для ПФ ОЧС

Функциональная схема ОЧС на базе мультиплексоров приведена в приложении Г.

**5. Логический синтез преобразователя множителя**

Преобразователь множителя (ПМ) служит для исключения из множителя диад 11 и замены их на триады 101.

Таблица 5.1–Таблица истинности ПМ.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Входы ПМ | | | Выходы ПМ | | | |
| p | Q1 | Q2 | П | S1 | S2 | M |
| 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 | 1 | 0 | 0 | 0 |

Минимизируем выходные функции П, S1 и S2 картами Карно.

Таблица 5.2 – Минимизация функции П.

Q1 Q2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |

П(мднф) = pQ2 + pQ1 = p \* (Q2 + Q1)

Таблица 5.2 – Минимизация функции S1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |

Q1 Q2

S1(мднф) = pQ1Q2 + pQ1Q2

Таблица 5.2 – Минимизация функции S2.

Q1 Q2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |

S2(мднф) = Q1Q2 + Q1Q2 = Q1 ⊕ Q2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |

Q1 Q2

M(мднф) = pQ1Q2 + pQ1Q2 = p \* (Q1 ⊕ Q2) = p \* S2(мднф)

Функциональная схема ПМ представлена в приложении Д.

**6. Оценка эффективности минимизации переключательных функций**

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

Таблица 6.1 – Эффективность минимизации ОЧУ

|  |  |  |  |
| --- | --- | --- | --- |
| Вых. схемы | Рассчитанная цена схемы | | Эфф. мин. k |
| До минимизации | После минимизации |
| P1 | C = 2 \* 5 + 2 + 4 = 16 | C = 3 + 2 = 5 | 3,2 |
| P2 | C = 24 \* 5 + 24 + 5 = 149 | C = 1 | 149 |
| P3 | C = 8 \* 5 + 8 + 4 = 52 | C = 10 + 3 + 4 = 17 | 3,06 |
| P4 | C = 10 \* 5 + 10 + 5 = 65 | C = 7 + 3 + 4 = 14 | 4,64 |

Таблица 6.2 – Эффективность минимизации ОЧС

|  |  |  |  |
| --- | --- | --- | --- |
| Вых. схемы | Рассчитанная цена схемы | | Эфф. мин. k |
| До минимизации | После минимизации |
| П | C = 4 \* 5 + 4 + 4 = 28 | C = 6 + 2 + 2 = 10 | 2,8 |
| S1 | C = 8 \* 5 + 8 + 4 = 52 | C = 12 + 4 + 3 = 19 | 2,73 |
| S2 | C = 8 \* 5 + 8 + 4 = 52 | C = 6 \* 3 + 6 + 4 = 28 | 1,85 |

**Временные затраты на умножение**

Формула расчёта временных затрат на умножение:

𝑇у = (n - 1) \* (𝑇сдвига + 𝑇ПМ + 𝑇ФДК + (n – 1) \* 𝑇ОЧС + 𝑇ОЧУ), где

𝑇сдвига – время сдвига частичной суммы;

𝑇ОЧУ – время умножения на ОЧУ;

𝑇ОЧС – время формирования единицы переноса в ОЧС;

𝑇ПМ – время преобразования множителя;

𝑇ФДК – время формирования дополнительного кода множимого.

n – количество разрядов множителя.

**Заключение**

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

В качестве главного достоинства минимизации картами Карно (Вейча)

можно выделить простоту и минимальные затраты времени. Однако применение данного способа для функций многих переменных будет затруднительно. Для минимизации функций многих переменных удобно использовать алгоритм Рота, который полностью формализует алгоритмы минимизации и делает минимизацию доступной для выполнения компьютерной программой. Метод Квайна-Мак-Класки совместил в себе одновременно простоту минимизации и её эффективность. Однако проведение минимизации функции данным методом будет очень долгим, если эта функция принимает единичные и безразличные значения на большом количестве наборов переменных.

Функциональные схемы были построены в различных логических базисах. Это позволило закрепить теоретические знания основных законов булевой алгебры, например, правило де Моргана.

Реализация переключательных функций на основе мультиплексоров позволила облегчить процесс минимизации этих функций и упростить функциональную схему одноразрядного четверичного сумматора.

**Список использованных источников**

Единая система конструкторской документации (ЕСКД) : справ. пособие /

С. С. Борушек [и др.]. – М. : Изд-во стандартов, 1989. – 352 с. Искра, Н. А. Арифметические и логические основы вычислительной техники : пособие / Н. А. Искра, И. В. Лукьянова, Ю. А. Луцик. – Минск : БГУИР, 2016. – 75 с.

Луцик, Ю. А. Учебное пособие по курсу «Арифметические и логические

основы вычислительной техники» / Ю. А. Луцик, И. В. Лукьянова, М. П. Ожигина. – Минск : МРТИ, 2001. – 77 с. Лысиков, Б. Г. Арифметические и логические основы цифровых автоматов / Б. Г. Лысиков. – Минск : Выш. шк., 1980. – 342 с. Лысиков, Б. Г. Цифровая вычислительная техника / Б. Г. Лысиков. – Минск: Выш. шк., 2003. – 242 с. Савельев, А. Я. Прикладная теория цифровых автоматов / А. Я. Савельев. – М. : Высш. шк., 1987. – 272 с.

Усатенко, С. Т. Выполнение электрических схем по ЕСКД : справочник / С.

Т. Усатенко, Т. К. Каченюк, М. В. Терехова. – М. : Изд-во стандартов, 1989. – 325 с.

**Приложение А**

Сумматор-умножитель первого типа. Схема электрическая структурная

**Приложение Б**

Одноразрядный четверичный умножитель. Схема электрическая функциональная

**Приложение В**

Одноразрядный четверичный сумматор. Схема электрическая функциональная

**Приложение Г**

Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная

**Приложение Д**

Преобразователь множителя. Схема электрическая функциональная

**Приложение Е**

Ведомость документов