МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Допущено к защите

Руководитель проекта

\_\_\_\_\_\_\_\_/Исупов К.С./

(подпись) (Ф.И.О)

«\_\_»\_\_\_\_\_\_\_\_\_\_202\_г.

СИНТЕЗ МИКРОПРОГРАММНОГО УПРАВЛЯЮЩЕГО АВТОМАТА

Пояснительная записка курсового проекта по дисциплине

«Теория автоматов»

ТПЖА.09.03.01.723 ПЗ

Разработал студент группы ИВТ-23 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Исупов К.С./

Консультант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мельцов В.Ю./

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_/

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись)

Киров 2021

УДК 004.4

РЕФЕРАТ

Кудяшев Я.Ю. Синтез микропрограммного управляющего автомата: ТПЖА.090301.723 ПЗ: Курс. проект / ВятГУ, каф. ЭВМ; рук. Исупов К.С. - Киров, 2021. – ПЗ 64с, 4 рис., 15 табл., 6 прил.

СИНТЕЗ АВТОМАТА, ДЕЛЕНИЕ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ, ПОРЯДОК, УПРАВЛЯЮЩИЙ АВТОМАТ, ОПЕРАЦИОННЫЙ АВТОМАТ, ДЕЛЕНИЕ БЕЗ ВОССТАНОВЛЕНИЯ ОСТАТКА, ПЕРВЫЙ СПОСОБ ДЕЛЕНИЯ.

Объект исследования и разработки – микропрограммный автомат, выполняющий деление двоичных чисел с плавающей запятой в прямом коде с порядками I способом без восстановления остатков с использованием дополнительного кода при вычитании мантисс.

Цель курсового проекта – синтез микропрограммного автомата, управляющего выполнением деления двоичных чисел с плавающей запятой в прямом коде с порядками I способом без восстановления остатков с использованием дополнительного кода при вычитании мантисс в основном логическом базисе, с минимальными аппаратурными затратами при приемлемом быстродействии.

Результат работы – функциональная (логическая) схема микропрограммного управляющего автомата.

**СОДЕРЖАНИЕ**

[1. ОПИСАНИЕ ИСПОЛЬЗУЕМОГО АЛГОРИТМА ДЕЛЕНИЯ 5](#_Toc71563890)

[2. ЧИСЛЕННЫЕ ПРИМЕРЫ 8](#_Toc71563891)

[**2.1. Пример деления без исключительных ситуаций** 8](#_Toc71563892)

[**2.2. Пример деления с исправимой временной ПМР** 9](#_Toc71563893)

[**2.3. Пример деления с неисправимой временной ПМР** 11](#_Toc71563894)

[**2.4. Пример деления с истинной ПМР** 13](#_Toc71563895)

[**2.5. Пример деления с ПРС при денормализации делимого** 13](#_Toc71563896)

[**2.6. Пример деления при истинном ПРС порядков** 15](#_Toc71563897)

[3. ВЫБОР ФУНКЦИОНАЛЬНОЙ СХЕМЫ ОПЕРАЦИОННОЙ ЧАСТИ УСТРОЙСТВА 16](#_Toc71563898)

[**3.1. Состав операционного автомата** 16](#_Toc71563899)

[**3.2. Описание операционного автомата** 17](#_Toc71563900)

[**3.3. Управляющие и осведомительные сигналы** 18](#_Toc71563901)

[4. РАЗРАБОТКА СОДЕРЖАТЕЛЬНОЙ ГРАФ-СХЕМЫ АЛГОРИТМА 21](#_Toc71563902)

[5. ПОСТРОЕНИЕ ОТМЕЧЕННОЙ ГРАФ-СХЕМЫ АЛГОРИТМА 24](#_Toc71563903)

[6. Построение графов автоматов и выбор структурной схемы управляющего автомата 26](#_Toc71563904)

[**6.1. Кодирование внутренних состояний для модели Мили** 27](#_Toc71563905)

[**6.1.1 Кодирование внутренних состояний для модели Мили на D-триггерах** 27](#_Toc71563906)

[**6.1.2. Кодирование внутренних состояний для модели Мили на RS-триггерах** 33](#_Toc71563907)

[**6.1.3. Кодирование внутренних состояний для модели Мили на счётчике** 44](#_Toc71563908)

[**6.2. Кодирование внутренних состояний для модели Мура на D-триггерах** 49](#_Toc71563909)

[7. ПОСТРОЕНИЕ СХЕМЫ УПРАВЛЯЮЩЕГО МИКРОПРОГРАММНОГО АВТОМАТА 55](#_Toc71563910)

**ВВЕДЕНИЕ**

ТПЖА.09.03.01.723 ПЗ

С каждым годом объемы вычислений возрастают и их все сложнее обрабатывать вручную, поэтому ведутся работы по созданию и применению различных автоматизированных систем для обработки информации. Такие автоматы реализуются в виде самостоятельных устройств или в виде блоков, входящих в системы управления и системы обработки информации. При этом работа ведется с математическими моделями, предназначенными для приближенного отображения физических моделей.

Основной целью данного курсового проекта является получение навыков синтеза управляющего микропрограммного автомата (МПА) с жесткой логикой на основе разработки машинных алгоритмов одной из заданных арифметических операций. Основным требованием является минимизация аппаратурных затрат как управляющего, так и операционного автомата при приемлемом быстродействии.

Постановка задачи. Необходимо синтезировать микропрограммный автомат, управляющий выполнением деления чисел в двоичной системе счисления с плавающей запятой в прямом коде с порядками I способом без восстановления остатков с использованием дополнительного кода при вычитании мантисс, в основном логическом базисе.

1. ОПИСАНИЕ ИСПОЛЬЗУЕМОГО АЛГОРИТМА ДЕЛЕНИЯ

Алгоритм деления чисел I способом, с плавающей запятой, с порядками, в прямом коде, при вычитании использовать дополнительный код без восстановления остатков.

1. Считать делимое
2. Считать делитель
3. Проверить делитель на равенство нулю: если равен нулю, зафиксировать ошибку деления на нуль (ДНН) и перейти к п.19, иначе к п.4
4. Проверить делимое на равенство нулю: если равно нулю, то присвоить результату деления 0 и перейти к п.18, иначе перейти к п.5
5. Определить предварительный порядок частного вычитанием в ДК порядка делителя из порядка делимого
6. Проверить на ПРС. Если в результате вычитания порядков в знаковом разряде единица, но отсутствует единица переноса, то выдать сигнал о ПРС и перейти к п.19, иначе к п.7
7. Проверить ПМР:

* Если в результате вычитания порядков в знаковом разряде 0, но присутствует единица переноса, то произошла ПМР. Сформировать результат 0 и перейти к п.18
* Если в результате вычитания порядков в знаковом разряде 1 и присутствует единица переноса, остальные разряды заполнены нулями, то произошла временная ПМР. Запомнить и перейти к п.8
* Иначе перейти к п.8

1. Определить знак частного путём сложения по модулю два заковых разрядов операндов
2. Выполнить вычитание делителя из остатка
3. Занести инвертированный знак остатка в младший разряд частного
4. Увеличить счётчик тактов на 1
5. Сдвинуть частное и делимое влево
6. Если старший разряд частного равен 1, то к п.14, иначе к п.9
7. Проверить:

* Если счётчик тактов равен 23, то перейти к п.15
* Если счётчик тактов равен 24, то перейти к п.16
* Иначе перейти к п.9

1. Прибавить к порядку 1 и выполнить проверку на ПРС:

* Если в результате увеличения порядка на 1 в знаковом разряде единица, но отсутствует единица переноса, то выдать сигнал о ПРС и перейти к п.19
* Иначе перейти к п.17

1. Проверить на неисправимое временное ПМР:

* Если до этого в п.7 была зафиксирована временная ПМР, то сформировать результат 0 и перейти к п.18
* Иначе перейти к п.17

1. Присвоить результату знак из п.8
2. Выдать результат
3. Завершить операцию деления

Доказательство отсутствия необходимости в нормализации мантиссы частного.

В операциях над числами с плавающей запятой в прямом коде с порядками мантиссы могут принимать значения в полуинтервале [0.5; 1) и ноль. Далее в качестве делимого и делителя будут использоваться их мантиссы. Если один из операндов равен нулю, то такие ситуации обрабатываются отдельно, иначе (значения операндов находятся в полуинтервале [0.5; 1)) необходимо рассмотреть случаи, когда делимое меньше делителя и делимое больше либо равно делителю.

Если делимое меньше делителя, то их частное будет меньше единицы, но не меньше 0.5, т. к. делитель не может превосходить делимое более, чем в два раза, иначе оно бы находилось в полуинтервале [1; 2). Поэтому частное будет принимать значения в том же диапазоне, что и делимое и делитель.

Если делимое не меньше делителя, то их частное будет не меньше единицы, но меньше двух, т. к. делимое не может превосходить делимое более, чем в два раза, иначе оно бы находилось в полуинтервале [1; 2). В таком случае, следуя алгоритму, необходимо уменьшить делимое в два раза (сдвиг на один разряд вправо). Но это также приведет к уменьшению частного в два раза. Следовательно, частное будет принимать значения из полуинтервала [0.5; 1), т. е. такие же, что и делимое и делитель.

Из этого следует, что мантисса частного после выполнения основного цикла деления будет уже нормализована.

2. ЧИСЛЕННЫЕ ПРИМЕРЫ

**2.1. Пример деления без исключительных ситуаций**

Делимое: 14.75 Делитель: 55

Порядок: 0.100 Порядок: 0.110

Вычитание порядков: 0.100 + 1.010 = 1.110

Знак результата: 0 ⊕ 0 = 0

Процесс деления приведен в таблице 1.

Таблица 1 – Пример деления без исключительных ситуаций

|  |  |  |
| --- | --- | --- |
| Частное | Делимое / (Остаток) | Комментарии |
| 0,…… | 0,111011 – делимое (ПК)  0,110111 – делитель (ПК) | Исходные данные, i = 0 |
| 0,…… | 0,111011  1,001001  0,000100 | D – d => △0 ≥ 0 => q6 = 1 |
| 0,…..1 | 0,001000 | Сдвиги, i = 1 |
| 0,…..1 | 0,001000  1,001001  1,010001 | △0 – d => △1 < 0 => q5 = 0 |
| 0,….10 | 0,100010 | Сдвиги, i = 2 |
| 0,….10 | 0,100010  0,110111  1,011001 | △0 + d => △2 < 0 => q4 = 0 |
| 0,…100 | 0,110010 | Сдвиги, i = 3 |
| 0,…100 | 0,110010  0,110111  1,101001 | △0 + d => △3 < 0 => q3 = 0 |
| 0,..1000 | 1,010010 | Сдвиги, i = 4 |
| 0,..1000 | 1,010010  0,110111  0,001001 | △0 + d => △4 ≥ 0 => q2 = 1 |
| 0,.10001 | 0,010010 | Сдвиги, i = 5 |
| 0,.10001 | 0,010010  1,001001  1,011011 | △0 – d => △5 < 0 => q1 = 0 |
| 0,100010 | 0,001000 | Сдвиги, i = 6 |

Т.к. на n-1 в старшем разряде частного 1, то порядок частного: 1.110 + 0.001 = 1.111

Ответ: 0.100010 \* 2-1 = 0.01000102 = 0.26562510

Искомый результат: | 14.75 / 55 | = 0.268181

Абсолютная погрешность: | 0.265625 - 0.268181 | = 0.002556

Относительная погрешность: | 0.002556 / 0.268181 | \* 100% = 0.0095%

**2.2. Пример деления с исправимой временной ПМР**

Делимое: 0.921875 Делитель: 3.4375

Порядок: 1.110 Порядок: 0.010

Вычитание порядков: 1.010 + 1.110 = 1.000

Знак результата: 0 ⊕ 0 = 0

Процесс деления приведён в таблице 2.

Таблица 2 – Пример деления с исправимой временной ПМР

|  |  |  |
| --- | --- | --- |
| Частное | Делимое / (Остаток) | Комментарии |
| 0,…… | 0,111011 – делимое (ПК)  0,110111 – делитель (ПК) | Исходные данные, i = 0 |
| 0,…… | 0,111011  1,001001  0,000100 | D – d => △0 ≥ 0 => q6 = 1 |
| 0,…..1 | 0,001000 | Сдвиги, i = 1 |
| 0,…..1 | 0,001000  1,001001  1,010001 | △0 – d => △1 < 0 => q5 = 0 |
| 0,….10 | 0,100010 | Сдвиги, i = 2 |
| 0,….10 | 0,100010  0,110111  1,011001 | △0 + d => △2 < 0 => q4 = 0 |
| 0,…100 | 0,110010 | Сдвиги, i = 3 |
| 0,…100 | 0,110010  0,110111  1,101001 | △0 + d => △3 < 0 => q3 = 0 |
| 0,..1000 | 1,010010 | Сдвиги, i = 4 |
| 0,..1000 | 1,010010  0,110111  0,001001 | △0 + d => △4 ≥ 0 => q2 = 1 |
| 0,.10001 | 0,010010 | Сдвиги, i = 5 |
| 0,.10001 | 0,010010  1,001001  1,011011 | △0 - d => △5 < 0 => q1 = 0 |
| 0,100010 | 0,110110 | Сдвиги, i = 6 |

Т.к. на n-1 в старшем разряде частного 1, то порядок частного: 1.000 + 0.001 = 1.001

Ответ: 0.100010 \* 2-1 = 0.01000102 = 0.26562510

Искомый результат: | 0.921875 / 3.4375 | = 0.268181

Абсолютная погрешность: | 0.265625 - 0.268181 | = 0.002556

Относительная погрешность: | 0.002556 / 0.268181 | \* 100% = 0.0095%

**2.3. Пример деления с неисправимой временной ПМР**

Делимое: 0.0556640625 Делитель: 3.9375

Порядок: 1.110 Порядок: 0.010

Вычитание порядков: 1.010 + 1.110 = 1.000

Знак результата: 0 ⊕ 0 = 0

Пример деления приведен в таблице 3.

Таблица 3 – Пример деления с неисправимой временной ПМР

|  |  |  |
| --- | --- | --- |
| Частное | Делимое / (Остаток) | Комментарии |
| 0,…… | 0,111001 – делимое (ПК)  0,111111 – делитель (ПК) | Исходные данные, i = 0 |
| 0,…… | 0,111001  1,000001  1,111010 | D – d => △0 < 0 => q6 = 0 |
| 0,…..0 | 1,110100 | Сдвиги, i = 1 |
| 0,…..0 | 1,110100  0,111111  0,110011 | △0 + d => △1 ≥ 0 => q5 = 1 |
| 0,….01 | 1,100110 | Сдвиги, i = 2 |
| 0,….01 | 1,100110  1,000001  0,100111 | △0 - d => △2 ≥ 0 => q4 = 1 |
| 0,…011 | 1,001110 | Сдвиги, i = 3 |
| 0,…011 | 1,001110  1,000001  0,001111 | △0 - d => △3 ≥ 0 => q3 = 1 |
| 0,..0111 | 0,011110 | Сдвиги, i = 4 |
| 0,..0111 | 0,011110  1,000001  1,011111 | △0 - d => △4 < 0 => q2 = 0 |
| 0,.01110 | 0,111110 | Сдвиги, i = 5 |
| 0,.01110 | 0,111110  0,111111  1,111101 | △0 + d => △5 < 0 => q1 = 0 |
| 0,011100 | 1,111010 | Сдвиги, i = 6 |
| 0,011100 | 1,111010  0,111111  0,111001 | △0 + d => △5 ≥ 0 => q1 = 1 |
| 0,111001 | 1,110010 | Сдвиги, i = 7 |

Т.к. ранее была зафиксирована временная ПМР и деление завершилось за n тактов, то временное ПМР является неустранимой.

Ответ: 0.000000

**2.4. Пример деления с истинной ПМР**

Делимое: 0.0430 Делитель: 20

Порядок: 1.100 Порядок: 0.101

Вычитание порядков: 1.100 + 1.011 = 0.111

Т.к. в результате вычитания порядков в знаковом разряде 0 и присутствует единица переноса, то возникло истинное ПМР.

Ответ: 0.000000

**2.5. Пример деления с ПРС при денормализации делимого**

Делимое: 15 Делитель: 0.078125

Порядок: 0.100 Порядок: 1.101

Вычитание порядков: 0.100 + 0.011 = 0.111

Знак результата: 0 ⊕ 0 = 0

Пример деления приведён в таблице 4.

Таблица 4 – Пример деления с ПРС при денормализации делимого

|  |  |  |
| --- | --- | --- |
| Частное | Делимое / (Остаток) | Комментарии |
| 0,…… | 0,111100 – делимое (ПК)  0,101000 – делитель (ПК) | Исходные данные, i = 0 |
| 0,…… | 0,111100  1,011000  0,010100 | D – d => △0 ≥ 0 => q6 = 1 |
| 0,…..1 | 0,101000 | Сдвиги, i = 1 |
| 0,…..1 | 0,101000  1,011000  0,000000 | △0 - d => △1 ≥ 0 => q5 = 1 |
| 0,….11 | 0,000000 | Сдвиги, i = 2 |
| 0,….11 | 0,000000  1,011000  1,011000 | △0 - d => △2 < 0 => q4 = 0 |
| 0,…110 | 0,110000 | Сдвиги, i = 3 |
| 0,…110 | 0,110000  0,101000  1,011000 | △0 + d => △3 ≥ 0 => q3 = 1 |
| 0,..1100 | 0,110000 | Сдвиги, i = 4 |
| 0,..1100 | 0,110000  0,101000  1,011000 | △0 - d => △4 ≥ 0 => q2 = 1 |
| 0,.11000 | 0,110000 | Сдвиги, i = 5 |
| 0,.11000 | 0,110000  0,101000  1,011000 | △0 - d => △5 < 0 => q1 = 0 |
| 0,110000 | 0,110000 | Сдвиги, i = 6 |

Т.к. на n-1 в старшем разряде частного 1, то порядок частного: 0.111 + 0.001 = 1.000. В знаковом разряде частного 1 и отсутствует единица переноса: возникло ПРС.

Ответ: ПРС

**2.6. Пример деления при истинном ПРС порядков**

Делимое: 40 Делитель: 0.199

Порядок: 0.110 Порядок: 1.010

Вычитание порядков: 0.110 + 0.010 = 1.000

Т.к. в знаковом разряде частного 1 и отсутствует единица переноса, то возникло ПРС.

Ответ: ПРС

# 3. ВЫБОР ФУНКЦИОНАЛЬНОЙ СХЕМЫ ОПЕРАЦИОННОЙ ЧАСТИ УСТРОЙСТВА

**3.1. Состав операционного автомата**

Операционный автомат (ОА) должен содержать следующие элементы:

* 24-х разрядный несдвиговый регистр RG1 для хранения делителя и делимого;
* 24-х разрядный сдвиговый влево регистр RG2 для хранения остатка;
* 24-х разрядный сдвиговый влево регистр RG3 для хранения частного;
* 8-разрядный несдвиговый регистр RG4 для хранения порядка делимого и делителя;
* 24-х разрядный управляемый инвертор для инвертирования делителя;
* 8-разрядный управляемый инвертор для инвертирования порядков;
* 7-разрядный управляемый инвертор для инвертирования порядка частного при выдаче на шину;
* 24-х разрядный сумматор SM1 для сложения или вычитания делителя из остатка;
* 8-разрядный сумматор SM2 для сложения или вычитания порядка делителя из порядка делимого;
* 8-разрядный счётчик СТ1 для хранения порядка частного;
* 6-разрядный счётчик СТ2 для подсчёта тактов деления;
* D-триггер T1 для хранения знака остатка;
* D-триггер T2 для хранения знакового разряда делимого;
* D-триггер T3 для хранения знакового разряда порядка делимого;
* RS-триггер T4 для выдачи сигнала о ДНН;
* RS-триггер T5 для выдачи сигнала о ПРС;
* Элемент сложение «по модулю два» для получения знакового разряда результата;
* Элемент «не» для записи знака остатка в частное;
* 4 элемента «не», 5 элементов «и», 7-входовой элемент «или» для проверки на ПМР, ПРС и временное ПРС;
* Усилитель-формирователь для выдачи результата на выходную шину.

**3.2. Описание операционного автомата**

Операнды разрядностью 4 байта поступают по входной шине в прямом коде, результат в прямом коде выводится по выходной шине.

В регистр RG1 поступает мантисса операнда со знаковым разрядом. Поэтому несдвиговый регистр RG1 имеет 24 разряда.

В D-триггер на вход D поступает знак делимого, который в дальнейшем поступит на элемент сложение «по модулю два» вместе со знаком делителя.

В счётчик СТ2 в начале операции деления записывается число «001000», чтобы в дальнейшем можно было отследить такт, на котором закончилась операция деления. С каждым тактом он увеличивается на 1.

Регистр RG2 имеет 24 разряда и является сдвиговым в левую сторону, поскольку в процессе деления выполняются сдвиги влево. В данном регистре хранится остаток.

Сдвиговый регистр RG3 имеет 23 разряда, так как в него записывается частное без знака. В каждом такте деления производится сдвиг данного регистра влево. В начале операции деления в него записываются нули, чтобы отследить такт, на котором придём единица в старший разряд регистра.

Регистр RG4 имеет 8 разрядов, так как в нем сначала хранится порядок делимого, а потом делителя.

Реверсивный счетчик СТ1 имеет 8 разрядов, так как в нем сначала хранится порядок делимого, а потом порядок частного. Необходимость реверсивного счётчика обусловлена нуждой в добавлении единицы к порядку результата при денормализации делимого и вычитании единицы при переводе в ДК.

Сумматор SM1 имеет 24 разряда на входах и выходе, вход CR и выход CRP переноса. На плечо А сумматора поступает содержимое выхода 24-разрядного управляемого инвертора, на плечо В – содержимое регистра RG2, на вход CR подаётся единица, если были инвертированы данные, поступающие из регистра RG1. С выхода S данные записываются в регистр RG2. Старший разряд выхода S записывается в триггер T2 для дальнейшего анализа знака остатка.

Из триггера Т2 данные проходят через инвертор и записываются в младший разряд регистра RG3.

Сумматор SM2 имеет 8 разрядов на входе и выходе, вход переноса CR. На плечо А записываются данные с выхода 8-разрядного управляемого инвертора, на вход В – данные из счетчика СТ1. На вход CR подается единица, если были инвертированы данные, поступающие из регистра RG4. С выхода S данные подаются на счетчик СТ1.

На выходную шину данные из счетчика СТ1 поступают через 7-разрядный управляемый инвертор, который инвертирует данные в случае, если в старшем разряде счетчика СТ1 хранится единица.

Сигналы ПРС и ДНН записываются в триггеры Т4 и Т5 соответственно.

Функциональная схема операционного автомата приведена в приложении А.

**3.3. Управляющие и осведомительные сигналы**

Для выполнения операции деления из управляющего автомата в операционный автомат необходимо подавать управляющие сигналы, реализующие следующие микрооперации:

* y0 – сброс триггеров Т4 и Т5, занесение знака делимого в триггер Т2, сброс регистра RG2, занесение в регистр RG3 нулей, запись в CT2
* y1 – запись в регистры RG1 и RG4
* y2 – очищение счётчиков СТ1, СТ2 и триггера Т2
* y3 – инверсия содержимого регистра RG4, подача единицы на вход CRP сумматора SM2
* y4 – запись в счётчик СТ1
* y5 – запись в D-триггер T3
* y6 – Запись в регистр RG2
* y7 – переключение в 1 триггера T5 для выдачи сигнала о ПРС
* y8 – переключение в 1 триггера T4 для выдачи сигнала о ДНН
* y9 – инверсия содержимого регистра RG1 и подача единицы на вход CR сумматора SM1
* y10 – запись в триггер T1
* y11 – прибавление 1 к счётчику СТ1
* y12 – выдача результата на шину
* y13 – сброс регистров RG1 и RG3
* y14 – сдвиги регистров RG2 и RG3 влево, увеличение CT2 на единицу
* y15 – вычитание единицы из содержимого CT1 и инверсия его содержимого

Из операционного автомата в управляющий автомат необходимо передать осведомительные сигналы о состоянии устройства ОА, определяемые списком следующих логических условий:

* X – ожидание операндов
* P1 – проверка делителя на 0
* P2 – проверка делимого на 0
* P3 – проверка на положительный остаток
* P4 – проверка на положительный порядок
* P5 – проверка на ПРС
* P6 – проверка на истинную ПМР
* P7 – проверка на временную ПМР
* P8 – проверка на равенство счётчика 23
* P9 – проверка старшего разряда частного на 1
* P10 – проверка знака порядка частного
* Z – проверка возможности выдачи результата на выходную шину

Таким образом, управляющий МПА должен вырабатывать 15 управляющих сигналов и посылать их в ОА в нужные такты машинного времени в соответствии с алгоритмом выполнения операции деления, ориентируясь на 11 осведомительных сигналов, поступающих из ОА.

4. РАЗРАБОТКА СОДЕРЖАТЕЛЬНОЙ ГРАФ-СХЕМЫ АЛГОРИТМА

В первом такте производится проверка наличия на входной шине делимого. При поступлении делимого его мантисса заносится в регистр RG1, порядок – в регистр RG4. Знак делимого заносится в триггер Т3. Также в данном такте происходит сброс триггеров Т2, Т4 и Т5, обнуление регистра RG2 и занесение нулей в регистр RG3 и запись числа «001000» в счётчик СТ2.

Во втором такте происходит проверка знака порядка делимого, запись самого порядка в счетчик СТ1 и запись мантиссы делимого в регистр RG2. Если порядок положительный (P4=0), то на плечо А сумматора SM2 подаются неинвертированные данные с RG4, на плечо В – содержимое счетчика СТ1, результат с выхода S сумматора записывается в счетчик СТ1. В триггер Т3 записывается содержимое старшего разряда регистра RG4. Если порядок отрицательный (P4=1), то на плечо А сумматора SМ2 подаются инвертированные данные с RG4, на вход CR подается единица. Остальные действия совпадают с действиями, выполняющимися при положительном порядке.

В третьем такте производится проверка наличия на входной шине делителя. При его поступлении мантисса делителя записывается в младшие разряды RG1, в старший 24 разряд записывается знак делителя. Порядок записывается в регистр RG4.

В четвертом такте производится проверка делителя на равенство нулю. Если он равен нулю (P1=0), то в триггер Т4 записывается единица, символизирующая ДНН, и операция деления прекращается. Если делимое не равно нулю (P1=1), то производится проверка делимого на равенство нулю. Если оно равно нулю (P2=0), то сбрасываются регистры RG1, RG3, счетчик СТ1, триггер Т2, тем самым формируя на выходной шине 0 как результат деления. Если делимое не равно нулю (P2=1), то производится проверка знака порядка делителя. Если знак отрицательный (P4=1), то на плечо А сумматора SM2 подаются неинвертированные данные с регистра RG4, на плечо В – содержимое счетчика СТ1, результат с выхода S сумматора записывается в счетчик СТ1. Если знак положительный (P4=0), то на плечо А сумматора SМ2 подаются инвертированные данные с RG4, на вход CRP подается единица.

В пятом такте производится проверка на ПРС и ПМР. Если произошло ПРС (P5=1), то триггер Т5 переключается в 1 и операция деления прекращается. Если ПРС не произошло, то проверяется ПМР. Если ПМР возникло (P6=1), то сбрасываются регистры RG1, RG3, счетчик СТ1, триггер Т2, тем самым формируя на выходной шине 0 как результат деления. Если ПМР не произошло, то выполняется проверка знака остатка. Если он был отрицательным (P3=0), то на плечо А сумматора SM1 подаются данные с регистра RG1. На плечо B подаются данные с регистра RG2. Данные с 23-го разряда выхода S записываются в триггер Т1 для дальнейшего анализа. Если знак остатка был положительным (P3=1), то выполняются те же действия, но помимо предыдущих действий, ещё происходит инверсия содержимого регистра RG1 и подача единицы на вход CR сумматора SM1. Затем происходят сдвиги регистров RG2 и RG3 влево и увеличение значения счётчика CT2 на единицу.

В шестом такте сначала выполняется проверка окончания цикла деления. Если старший разряд RG3, т.е. частное, равен 0 (P9=0), то переходим вновь к проверке знака остатка и продолжаем цикл деления. Если старший разряд частного равен 1 (P9=1), то выполняем проверку значения счётчика CT2. В случае, когда значение счётчика равно 23 (P8=0), нужно прибавить к порядку частного, который хранится в СТ1, единицу. Если же цикл деления закончился за 24 такта (P8=1), то необходимо перейти к следующему шагу: проверке порядка на временную ПМР. Если в ходе исследования обнаружилось временное ПМР (P7=1), то происходит обнуление счётчика СТ1, триггера Т2 и сброс регистров RG1 и RG3 для выдачи нулевого результата на шину. Если временная ПМР не обнаружилась (P7=0), происходит проверка возможности выдачи результата на шину.

В случае, когда деление закончилось за 23 такта и мы прибавили к порядку частного в СТ1 единицу, нужно ещё проверить новый порядок частного на ПРС. Если при увеличении порядка частного возникло ПРС (P5=1), то нужно подать единичку на триггер Т5 и завершить деление. В обратном же случае, когда ПРС не обнаружилось (P5=0), происходит проверка возможности выдачи результата на шину.

Затем происходит проверка порядка частного. Если он в ДК (P10=1), то происходит вычитание единицы из порядка частного и дальнейшее его инвертирование. В случае, когда порядок в ПК (P10=0), переходим к следующему такту.

Далее происходит проверка возможности выдачи результата на выходную шину. При разрешении выдачи результата на выходную шину подаётся знак результата с выхода элемента сложение «по модулю два», также подаётся знак порядка из старшего разряда счётчика СТ1, инвертированный или неинвертированный (в зависимости от знака порядка) порядок результата с управляемого инвертора КС3 и мантисса результата из регистра RG3.

Содержательная граф-схема алгоритма представлена в приложении Б.

# 5. ПОСТРОЕНИЕ ОТМЕЧЕННОЙ ГРАФ-СХЕМЫ АЛГОРИТМА

Для разметки граф-схемы алгоритма каждой совокупности микроопераций, находящихся в операторных вершинах содержательной граф-схемы, ставятся в соответствие управляющие сигналы Y1…Y14. Эти управляющие сигналы являются выходными сигналами управляющего автомата и обеспечивают выполнение требуемых действий в соответствии со списком микроопераций операционного автомата. Совокупности микроопераций для каждой операторной вершины образуют микрокоманды, список которых представлен в таблице 5.

Таблица 5 – Совокупность микроопераций и соответствующие им микрокоманды

|  |  |
| --- | --- |
| Микрокоманда | Совокупность микроопераций |
| Y1 | y0,y1,y2 |
| Y2 | y3,y4,y5,y6 |
| Y3 | y4,y5,y6 |
| Y4 | y1 |
| Y5 | y8 |
| Y6 | y2,y13 |
| Y7 | y4 |
| Y8 | y3,y4 |
| Y9 | y7 |
| Y10 | y6,y9,y10 |
| Y11 | y6,y10 |
| Y12 | y14 |
| Y13 | y11 |
| Y14 | y12 |

Каждой условной вершине содержательной граф-схемы алгоритма ставится ставится в соответствие один из входных сигналов управляющего автомата X0…X10, список которых представлен в таблице 6.

Таблица 6 – Список входных сигналов для УА

|  |  |
| --- | --- |
| Входной сигнал УА | Логическое условие ОА |
| X0 | X |
| X1 | P4 |
| X2 | P1 |
| X3 | P2 |
| X4 | P5 |
| X5 | P6 |
| X6 | P3 |
| X7 | P9 |
| X8 | P8 |
| X9 | P7 |
| X10 | Z |

Отмеченная граф-схема приведена в приложении В.

# 6. Построение графов автоматов и выбор структурной схемы управляющего автомата

Граф автомата модели Мили имеет 9 вершин, соответствующих состояниям автомата а0…a8. Дуги его отмечены входными сигналами X0...X10, действующими на каждом переходе, и набором выходных сигналов y0…y14, вырабатываемых управляющим автоматом на данном переходе. Граф автомата модели Мили представлен в приложении Г.

Граф автомата Мура имеет 17 вершин, соответствующих состояниям автомата b0…b16. Каждое состояние определяет наборы выходных сигналов y0…y14 управляющего автомата, а дуги графа отмечены входными сигналами X0...X10, действующими на данном переходе. Граф автомата модели Мура представлен в приложении Д.

В управляющем автомате в качестве элементов памяти (ЭП) могут быть использованы D-триггеры, RS-триггеры, счетчик и т.д.

При использовании D-триггеров в качестве ЭП при переходе из одного состояния в другое сигналы возбуждения должны быть поданы только на те триггеры, которые в коде состояния содержат единицу. Отсюда следует, что для получения комбинационной схемы меньшей сложности необходимо соблюдать основное требование: чем больше переходов в какое- либо состояние, тем меньше единиц должен содержать код этого состояния.

Для RS-триггеров лучше использовать соседнее кодирование, так как именно этот способ минимизирует число переключений ЭП. В противном случае, нужно применить метод эвристического кодирования.

При использовании счетчика в качестве элемента памяти необходимо придерживаться последовательного кодирования.

## **6.1. Кодирование внутренних состояний для модели Мили**

### **6.1.1 Кодирование внутренних состояний для модели Мили на D-триггерах**

Для кодирования 9 состояний (a0…a8) графа автомата по модели Мили, минимально необходимо четыре элемента памяти.

С учетом особенностей работы D-триггера для кодирования состояний применяется эвристический метод. Он состоит в следующем:

* Каждому состоянию aj ставится в соответствие целое число Nj, равное числу переходов в данное состояние;
* Числа Nj сортируются в порядке убывания;
* Состоянию, соответствующему первому Nj после сортировки, то есть наибольшему из Nj, присваивается код. Состоящий только из нулей;
* Следующему состоянию в порядке убывания Nj присваивается незанятый код, содержащий наименьшее количество единиц. Данный пункт повторяется до тех пор, пока все состояния не будут закодированы.

Кодирование состояний для модели Мили на D-триггерах представлено в таблице 7.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Состояние | a0 | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 |
| Состояния, из которых есть переход | a0, a3, a4, a7,  a8 | a0 | a1(2), a2 | a2 | a3(2) | a4(2), a6(2) | a5 | a6 | a3, a4, a6(2), a7(2), a8 |
| Сортировка в порядке убывания | | | | | | | | |  |
| Состояние | a8 | a0 | a5 | a2 | a4 | a1 | a3 | a6 | a7 |
| Состояния, из которых есть переход | a3, a4, a6(2), a7(2), a8 | a0, a3, a4, a7,  a8, b | a4(2), a6(2) | a1(2), a2 | a3(2) | a0 | a2 | a5 | a6 |
| Число переходов | 7 | 6 | 4 | 3 | 2 | 1 | 1 | 1 | 1 |
| Код | 0000 | 0001 | 0010 | 0100 | 1000 | 0011 | 1100 | 1001 | 0110 |

Таблица 7 – Коды состояний для модели Мили на D-триггерах

Далее составляется прямая структурная таблица ереходов и выходов автомата для модели Мили, результаты которой приведены в таблице 8, и формируются логические выражения для функций возбуждений.

Таблица 8 – Прямая структурная таблица переходов и выходов автомата модели Мили на D-триггерах

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Исходное состояние am | Код am | Состояние перехода аs | Код as | Входной сигнал (am, as) | Выходной сигнал Y(am, as) | Функции возбуждения  D-триггеров |
| a0 | 0001 | a0 | 0001 | ~x0 | - | D0 |
| a1 | 0011 | x0 | Y1(y0,y1,y2) | D1D0 |
| a1 | 0011 | a2 | 0100 | ~x1 | Y3(y4,y5,  y6) | D2 |
| a2 | 0100 | x1 | Y2(y3,y4,y5,y6) | D2 |
| a2 | 0100 | a2 | 0100 | ~x0 | - | D2 |
| a3 | 1100 | x0 | Y4(y1) | D3 D2 |
| a3 | 1100 | a0 | 0001 | x2 | Y5(y8) | D0 |
| a4 | 1000 | ~x2~x3x1 | Y7(y4) | D3 |
| a4 | 1000 | ~x2~x3~x1 | Y8(y3,y4) | D3 |
| a8 | 0000 | ~x2x3 | Y6(y2,y13) | - |
| a4 | 1000 | a0 | 0001 | x4 | Y9(y7) | D0 |
| a5 | 0010 | ~x4~x5x6 | Y10(y6,y9,y10) | D1 |
| a5 | 0010 | ~x4~x5~x6 | Y11(y6,y10) | D1 |
| a8 | 0000 | ~x4x5 | Y6(y2,y13) | - |
| a5 | 0010 | a6 | 1001 | 1 | Y12(y14) | D3 D0 |
| a6 | 1001 | a5 | 0010 | ~x7x6 | Y10(y6,y9,y10) | D1 |
| a5 | 0010 | ~x7~x6 | Y11(y6,y10) | D1 |
| a7 | 0110 | x7x8 | Y13(y11) | D2D1 |
| a8 | 0000 | x7~x8~x9 | - | - |
| a8 | 0000 | x7~x8x9 | Y6(y2,y13) | - |
| a7 | 0110 | a0 | 0001 | x4 | Y9(y7) | D0 |
| a8 | 0000 | ~x4~x9 | - | - |
| a8 | 0000 | ~x4x9 | Y6(y2,y13) | - |
| a8 | 0000 | a0 | 0001 | x10 | Y14(y12) | D0 |
| a8 | 0000 | ~x10 | - | - |

Логические выражения для каждой функции возбуждения D-триггера получают по таблице, как конъюнкции соответствующих исходных состояний am и входных сигналов, которые объединены знаками дизъюнкции для всех строк, содержащих данную функцию возбуждения:

D0 = a0~x0 ⋁ a0x0~x6x7 ⋁ a3x2 ⋁ a4x4 ⋁ a5 ⋁ a7x4 ⋁ a8x10

D1 = a0x0 ⋁ a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a6~x7x6 ⋁ a6~x7~x6 ⋁ a6x7x8

D2 = a1~x1 ⋁ a1x1 ⋁ a2~x0 ⋁ a2x0 ⋁ a6x7x8

D3 = a2x0 ⋁ a3~x2~x3x1 ⋁ a3~x2~x3~x1 ⋁ a5

Аналогично составляются логические выражения для функций выходов:

y0 = a0x0

y1 = a0x0 ⋁ a2x0

y2 = a0x0 ⋁ a3~x2x3 ⋁ a4~x4x5 ⋁ a6x7~x8x9 ⋁ a7~x4x9

y3 = a1x1 ⋁ a3~x2~x3~x1

y4 = a1~x1 ⋁ a1x1 ⋁ a3~x2~x3x1 ⋁ a3~x2~x3~x1

y5 = a1~x1 ⋁ a1x1

y6 = a1~x1 ⋁ a1x1 ⋁ a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a6~x7x6 ⋁ a6~x7~x6

y7 = a4x4 ⋁ a7x4

y8 = a3x2

y9 = a4~x4~x5x6 ⋁ a6~x7x6

y10 = a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a6~x7x6 ⋁ a6~x7~x6

y11 = a6x7x8

y12 = a8x10

y13 = a3~x2x3 ⋁ a6x7~x8x9 ⋁ a7~x4x9

y14 = a5

Выделение общих частей:

e = a0x0 (2)

f = a1x1 (2)

g = a1~x1 (2)

h = a2x0 (2)

i = a3~x2~x3x1 (4)

j = a3~x2~x3~x1 (4)

k = a3~x2x3 (3)

l = a4~x4~x5x6 (4)

m = a4~x4~x5~x6 (4)

n = a6x7x8 (3)

o = a6~x7x6 (3)

p = a6~x7~x6 (3)

q = a7~x4x9 (3)

r = a8x10 (2)

s = o ⋁ p (2)

t = l ⋁ m (2)

u = s ⋁ t (2)

v = a6x7~x8x9 (4)

w = k ⋁ v ⋁ q (3)

x = a4x4 (2)

y = a7x4 (2)

z = x ⋁ y (2)

ee = f ⋁ g (2)

ff = a3x2 (2)

hh = i ⋁ j (2)

После выделения общих частей в логических выражениях и некоторого их упрощения получаем логические уравнения для построения функциональной схемы управляющего автомата.

Функции возбуждения:

D0 = a0~x0 ⋁ a0x0~x6x7 ⋁ a3x2 ⋁ z ⋁ a5 ⋁ r (14)

D1 = e ⋁ u ⋁ r (3)

D2 = ee ⋁ a2~x0 ⋁ h ⋁ n (6)

D3 = h ⋁ hh ⋁ a5 (3)

y0 = e (0)

y1 = e ⋁ h (2)

y2 = w ⋁ e ⋁ a4~x4x5 (6)

y3 = f ⋁ j (2)

y4 = ee ⋁ hh (2)

y5 = ee (0)

y6 = ee ⋁ u (2)

y7 = z (0)

y8 = ff (0)

y9 = l ⋁ o (2)

y10 = u (0)

y11 = n (0)

y12 = r (0)

y13 = k ⋁ v ⋁ q (3)

y14 = a5 (0)

Инверторы: ~x0, ~x1, ~x2, ~x3, ~x4, ~x5, ~x6, ~x7 ,~x8, ~x9, ~x10 (11)

Цена комбинационной схемы по Квайну автомата по модели Мили при использовании графа, построенного на основе ГСА, который представлен в приложении Д, с использованием в качестве элементов памяти 4 D-триггеров:

Σ = КС + ИНВ + ЭП + НУ + DC = 108 + 11 + 16 + 0 + 4 = 139

Схема начальной установки для D-триггеров в качестве ЭП приведена на рисунке 1, где D0, D1, D2, D3 – функции возбуждения соответствующих ЭП.

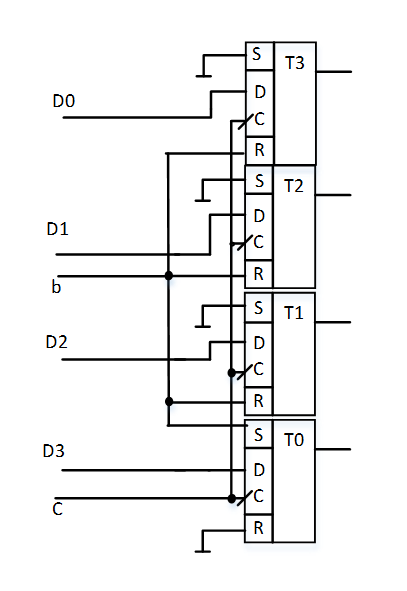


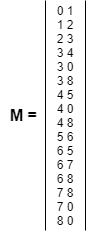
Рисунок 1. – Схема формирования начальной установки для D-триггеров в качестве элементов памяти.

### **6.1.2. Кодирование внутренних состояний для модели Мили на RS-триггерах**

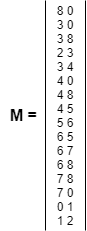
Для кодирования 9 состояний автомата Мили на RS-триггерах так же потребуется 4 триггера. При кодировании следует применить метод соседнего кодирования для минимизации числа переключений триггеров при переходе из одного состояния в другое. Данный граф не получится полностью закодировать по принципу соседнего кодирования, так как в нем присутствуют циклы с нечетным числом вершин. Следовательно, для минимизации числа переключений триггеров при переходе из одного состояния в другое необходимо применить эвристический метод кодирования. Данный метод минимизирует суммарное число переключений элементов памяти на всех переходах автомата. Уменьшение числа переключений триггеров приводит к уменьшению количества единиц соответствующих функций возбуждения, что однозначно приводит к упрощению комбинационной схемы автомата.

Произведём кодирование состояний автомата эвристическим методом кодирования:

1) Строим матрицу *М*, состоящую из всех пар переходов, где переключение триггеров в данном переходе отлично от 0 (числа в матрице соответствуют номеру состояния).



2) Упорядочим строки матрицы *M*. В первую строку матрицы помещаем пару с наибольшим весом. Из всех пар, имеющих общий компонент с парой, помещенной в матрицу 𝑀 выбирается пара с наибольшим весом и заносится в следующую строку матрицы (в случае равенства весов пар вычисляются суммы весов компонентов пар, то есть количество появления элемента в матрице *M*, на основании которых определяется следующая пара, которая будет занесена в таблицу), из всех пар, имеющих общий компонент с парами, помещенными в матрицу 𝑀 выбирается пара с наибольшим весом и заносится в следующую строку матрицы и т.д. Формирование новой матрицы заканчивается, когда все элементы матрицы *M* размещены в новой матрице 𝑀:

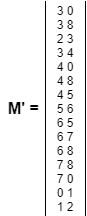


3) Закодируем первые 2 состояния:

a0 = 0000

a8 = 0001

4) Построим матрицу М', исключая уже закодированную строку:



Элемент a3 не закодирован.

Построим матрицу М13:



B3 = {0000, 0001} = {0, 8}

C10 = {0010, 0100, 1000}

C18 = {1001, 0101, 0011}

D13 = {0010, 0100, 1000, 1001, 0101, 0011}

W0010 = |0010 - 0000|2 + |0010 - 0001|2 = 1 + 2 = 3

W0100 = |0100 - 0000|2 + |0100 - 0001|2 = 1 + 2 = 3

W1000 = |1000 - 0000|2 + |1000 - 0001|2 = 1 + 2 = 3

W1001 = |1001 - 0000|2 + |1001 - 0001|2 = 2 + 1 = 3

W0101 = |0101 - 0000|2 + |0101 - 0001|2 = 2 + 1 = 3

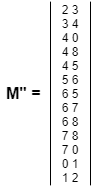
W0011 = |0011 - 0000|2 + |0011 - 0001|2 = 2 + 1 = 3

Закодируем элемент a3 = 0010

5) Построим матрицу М”, исключая уже закодированную строку:

Элемент a4 не закодирован.

Построим матрицу М14:



B4 = {0000, 0001, 0010} = {0, 3, 8}

C10 = {0010, 0100, 1000}

C18 = {1001, 0101, 0011}

C13 = {1010, 0110, 0011}

D14 = {0100, 1000, 1001, 0101, 0011, 1010, 0110}

W0100 = |0100 - 0000|2 + |0100 - 0001|2 + |0100 - 0010|2= 1 + 2 + 2 = 5

W1000 = |1000 - 0000|2 + |1000 - 0001|2 + |1000 - 0010|2= 1 + 2 + 2 = 5

W1001 = |1001 - 0000|2 + |1001 - 0001|2 + |1001 - 0010|2= 2 + 1 + 3 = 6

W0101 = |0101 - 0000|2 + |0101 - 0001|2 + |0101 - 0010|2= 2 + 1 + 3 = 6

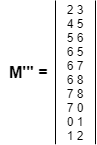
W0011 = |0011 - 0000|2 + |0011 - 0001|2 + |0011 - 0010|2= 2 + 1 + 1 = 4

W1010 = |1010 - 0000|2 + |1010 - 0001|2 + |1010 - 0010|2= 2 + 3 + 1 = 6

W0110 = |0110 - 0000|2 + |0110 - 0001|2 + |0110 - 0010|2= 2 + 3 + 1 = 6

Закодируем элемент a4 = 0011

6) Построим матрицу М”’, исключая уже закодированную строку:



Элемент a5 не закодирован.

Построим матрицу М15:



B5 = {0011} = {4}

C14 = {1011, 0111, 0010}

D15 = {1011, 0111}

W1011 = |1011 - 0011|2 = 1

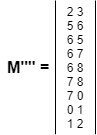
W0111 = |0111 - 0011|2 = 1

Закодируем элемент a5 = 1011

7) Построим матрицу М””, исключая уже закодированную строку:

Элемент a6 не закодирован.

Построим матрицу М16:



B6 = {1011} = {5}

C15 = {1001, 1010, 1111}

D16 = {1001, 1010, 1111}

W1001 = |1001 - 1011|2 = 1 + 1 = 2

W1010 = |1010 - 1011|2 = 1

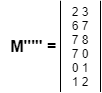
W1111 = |1111 - 1011|2 = 1

Закодируем элемент a6 = 1010

8) Построим матрицу М”’”, исключая уже закодированную строку:

Элемент a7 не закодирован.

Построим матрицу М17:



B7 = {1001, 0000, 0001} = {6, 0, 8}

C16 = {1000, 1101, 1011}

C10 = {0010, 0100, 1000}

C18 = {1001, 0101, 0011}

D17 = {1000, 1101, 0100, 0101}

W1000 = |1000 - 1001|2 + |1000 - 0000|2 + |1000 - 0001|2= 1 + 1 + 2 = 4

W1101 = |1101 - 1001|2 + |1101 - 0000|2 + |1101 - 0001|2= 1 + 3 + 2 = 6

W0100 = |0100 - 1001|2 + |0100 - 0000|2 + |0100 - 0001|2= 3 + 1 + 2 = 6

W0101 = |0101 - 1001|2 + |0101 - 0000|2 + |0101 - 0001|2= 2 + 2 + 1 = 5

Закодируем элемент a7 = 1000

9) Построим матрицу М”””, исключая уже закодированную строку:

Элемент a1 не закодирован.

Построим матрицу М11:



B1 = {0000} = {0}

C10 = {0010, 0100, 1000}

D11 = {0100}

Закодируем элемент a1 = 0100

10) Построим матрицу М”’””, исключая уже закодированную строку:

Элемент a2 не закодирован.

Построим матрицу М12:



B2 = {0010, 0100} = {3, 1}

C11 = {1100, 0110, 0101}

C13 = {0110, 1010, 0011}

D12 = {1100, 0110, 0101, 1010}

W1100 = |1100 - 0010|2 + |1100 - 0100|2 = 3 + 1 = 4

W0110 = |0110 - 0010|2 + |0110 - 0100|2 = 1 + 1 = 2

W0101 = |0101 - 0010|2 + |0101 - 0100|2 = 3 + 1 = 4

W1010 = |1010 - 0010|2 + |1010 - 0100|2 = 1 + 3 = 4

Закодируем элемент a2 = 0110

11) Результаты кодирования внутренних состояний для модели Мили на RS-триггере представлены в таблице 9.

Таблица 9 – Результаты кодирования внутренних состояний для модели Мили на RS-триггера

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Состояние | a0 | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 |
| Код | 0000 | 0100 | 0110 | 0010 | 0011 | 1011 | 1001 | 1000 | 0001 |

Для определения эффективности кодирования применяется коэффициент k, который является отношением минимального количества переключений (если бы состояния были закодированы соседним кодированием) к общему количеству переключений триггеров, где состояния закодированы с помощью эвристического метода кодирования:

W = |0000 - 0100|2 + |0100 - 0110|2 + |0110 - 0010|2 + |0010 - 0000|2 + |0010 - 0001|2 + |0010 - 0011|2 + |0011 - 1011|2 + |0011 - 0000|2 + |0011 - 0001|2 + |1011 - 1001|2 + |1001 - 1011|2 + |1001 - 1000|2 + |1001 - 0001|2 + |1000 - 0001|2 + |0010 - 0000|2 + |0001 - 0000|2  = 19

Среднее число переключений на переход: 19 / 16 = 1.188 – хорошее кодирование.

На основе выбранных кодов составляется прямая структурная таблица переходов и выходов автомата Мили, которая представлена в таблице 10.

Таблица 10 – Прямая структурная таблица переходов и выходов автомата

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Исходное состояние am | Код am | Состояние перехода аs | Код as | Входной сигнал  (am, as) | Выходной сигнал Y(am, as) | Функции возбуждения  RS-триггеров |
| a0 | 0000 | a0 | 0000 | ~x0 | - | - |
| a1 | 0100 | x0 | Y1(y0,y1,y2) | S2 |
| a1 | 0100 | a2 | 0110 | ~x1 | Y3(y4,y5,y6) | S1 |
| a2 | 0110 | x1 | Y2(y3,y4,y5,y6) | S1 |
| a2 | 0110 | a2 | 0110 | ~x0 | - | - |
| a3 | 0010 | x0 | Y4(y1) | S2 |
| a3 | 0010 | a0 | 0000 | x2 | Y5(y8) | R1 |
| a4 | 0011 | ~x2~x3x1 | Y7(y4) | S0 |
| a4 | 0011 | ~x2~x3~x1 | Y8(y3,y4) | S0 |
| a8 | 0001 | ~x2x3 | Y6(y2,y13) | R1 S0 |
| a4 | 0011 | a0 | 0000 | x4 | Y9(y7) | R1 R0 |
| a5 | 1011 | ~x4~x5x6 | Y10(y6,y9,y10) | S3 |
| a5 | 1011 | ~x4~x5~x6 | Y11(y6,y10) | S3 |
| a8 | 0001 | ~x4x5 | Y6(y2,y13) | R1 |
| a5 | 1011 | a6 | 1001 | 1 | Y12(y14) | R1 |
| a6 | 1001 | a5 | 1011 | ~x7x6 | Y10(y6,y9,y10) | S1 |
| a5 | 1011 | ~x7~x6 | Y11(y6,y10) | S1 |
| a7 | 1000 | x7x8 | Y13(y11) | R0 |
| a8 | 0001 | x7~x8~x9 | - | R3 |
| a8 | 0001 | x7~x8x9 | Y6(y2,y13) | R3 |
| a7 | 1000 | a0 | 0000 | x4 | Y9(y7) | R3 |
| a8 | 0001 | ~x4~x9 | - | R3 S0 |
| a8 | 0001 | ~x4x9 | Y6(y2,y13) | R3 S0 |
| a8 | 0001 | a0 | 0000 | x10 | Y14(y12) | R0 |
| a8 | 0001 | ~x10 | - | - |

На основе составленной таблицы формируются функции возбуждения элементов памяти:

R0 = a4x4 ⋁ a6x7x8 ⋁ a8x10

R1 = a3x2 ⋁ a3~x2x3 ⋁ a4x4 ⋁ a4~x4x5 ⋁ a5

R3 = a6x7~x8~x9 ⋁ a6x7~x8x9 ⋁ a7x4 ⋁ a7~x4~x9 ⋁ a7~x4x9

S0 = a3~x2~x3x1 ⋁ a3~x2~x3~x1 ⋁ a3~x2x3 ⋁ a7~x4~x9 ⋁ a7~x4x9

S1 = a1~x1 ⋁ a1x1 ⋁ a6~x7x6 ⋁ a6~x7~x6

S2 = a0x0 ⋁ a2x0

S3 = a4~x4~x5x6 ⋁ a4~x4~x5~x6

Аналогично составляем функции выходов:

y0 = a0x0

y1 = a0x0 ⋁ a2x0

y2 = a0x0 ⋁ a3~x2x3 ⋁ a4~x4x5 ⋁ a6x7~x8x9 ⋁ a7~x4x9

y3 = a1x1 ⋁ a3~x2~x3~x1

y4 = a1~x1 ⋁ a1x1 ⋁ a3~x2~x3x1 ⋁ a3~x2~x3~x1

y5 = a1~x1 ⋁ a1x1

y6 = a1~x1 ⋁ a1x1 ⋁ a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a6~x7x6 ⋁ a6~x7~x6

y7 = a4x4 ⋁ a7x4

y8 = a3x2

y9 = a4~x4~x5x6 ⋁ a6~x7x6

y10 = a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a6~x7x6 ⋁ a6~x7~x6

y11 = a6x7x8

y12 = a8x10

y13 = a3~x2x3 ⋁ a6x7~x8x9 ⋁ a7~x4x9

y14 = a5

Выделяем общие части формул:

e = a0x0 (2)

f = a1x1 (2)

g = a1~x1 (2)

h = a2x0 (2)

i = a3x2 (2)

j = a3~x2x3 (3)

k = a3~x2~x3x1 (4)

l = a3~x2~x3~x1 (4)

m = a4x4 (2)

n = a4~x4x5 (3)

o = a4~x4~x5x6 (4)

p = a4~x4~x5~x6 (4)

q = a6x7x8 (3)

r = a6~x7x6 (3)

s = a6~x7~x6 (3)

t = a6x7~x8x9 (4)

u = a7~x4x9 (3)

v = a7~x4~x9 (3)

w = a7x4 (2)

x = a8x10 (2)

y = g ⋁ f (2)

z = o ⋁ r (2)

ee = e ⋁ h (2)

ff = v ⋁ u (2)

gg = z ⋁ p ⋁ s (3)

hh = j ⋁ t ⋁ u (3)

ii = k ⋁ l (2)

С учётом упрощений:

R0 = m ⋁ q ⋁ x (3)

R1 = i ⋁ j ⋁ m ⋁ n ⋁ a5 (5)

R3 = a6x7~x8~x9 ⋁ t ⋁ w ⋁ ff (8)

S0 = ii ⋁ j ⋁ ff (3)

S1 = y ⋁ r ⋁ s (3)

S2 = ee (0)

S3 = o ⋁ p (2)

y0 = e (0)

y1 = ee (0)

y2 = e ⋁ hh ⋁ n (3)

y3 = f ⋁ l (2)

y4 = y ⋁ ii (2)

y5 = y (0)

y6 = y ⋁ gg (2)

y7 = m ⋁ w (2)

y8 = i (0)

y9 = z (0)

y10 = gg (0)

y11 = q (0)

y12 = x (0)

y13 = hh (0)

y14 = a5 (0)

Инверторы: ~x0, ~x1, ~x2, ~x3, ~x4, ~x5, ~x6, ~x7, ~x8, ~x9, ~x10 (11)

Цена комбинационной схемы по Квайну автомата по модели Мили при использовании графа, построенного на основе ГСА, который представлен в приложении Д, с использованием в качестве элементов памяти 4 RS-триггеров:

Σ = КС + ИНВ + ЭП + НУ + DC = 108 + 11 + 12 + 17 + 4 = 152

Схема формирования начальной установки на RS-триггерах представлена на рисунке 2.

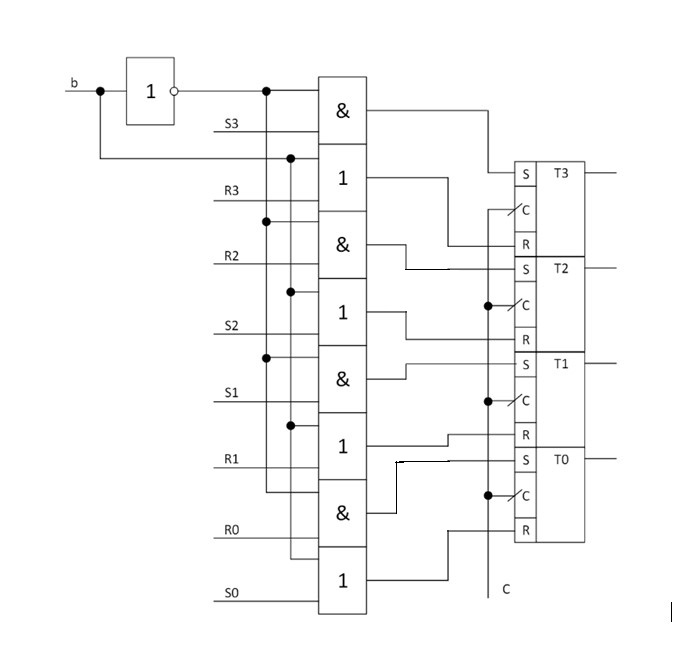


Рисунок 2 – Схема формирования начальной установки на RS-триггерах

### **6.1.3. Кодирование внутренних состояний для модели Мили на счётчике**

При кодировании состояний на счетчике необходимо стараться закодировать состояния таким образом, чтобы код состояния, отличающегося от предыдущего порядковым номером на единицу, был больше на единицу, так как счетчик имеет входные управляющие сигналы «+1», «-1», «Reset». Если при кодировании возникает ситуация, когда происходит переход в состояние отличное от предыдущего состояния более чем на единицу, необходимо заносить нестандартное состояние сразу же на элементы памяти счетчика. Для кодирования 13 состояний автомата по модели Мили потребуется 4-х разрядный счетчик. Получившиеся коды состояний представлены в таблице 11.

Таблица 11 – Состояния и коды состояний

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Состояние | а0 | а1 | а2 | а3 | а4 | а5 | а6 | а7 | a8 |
| Код | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 0000 |

На основе выбранных кодов состояний составляется прямая структурная таблица переходов и выходов автомата Мили, которая представлена в таблице 12.

Таблица 12 – Структурная таблица переходов и выходов автомата Мили

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Исходное состояние am | Код am | Состояние перехода аs | Код as | Входной сигнал  (am, as) | Выходной сигнал Y(am, as) | Функции возбуждения  D-триггеров |
| a0 | 0001 | a0 | 0001 | ~x0 | - | - |
| a1 | 0010 | x0 | Y1(y0,y1,y2) | +1 |
| a1 | 0010 | a2 | 0011 | ~x1 | Y3(y4,y5,y6) | +1 |
| a2 | 0011 | x1 | Y2(y3,y4,y5,y6) | +1 |
| a2 | 0011 | a2 | 0011 | ~x0 | - | - |
| a3 | 0100 | x0 | Y4(y1) | +1 |
| a3 | 0100 | a0 | 0001 | x2 | Y5(y8) | D0, EWR |
| a4 | 0101 | ~x2~x3x1 | Y7(y4) | +1 |
| a4 | 0101 | ~x2~x3~x1 | Y8(y3,y4) | +1 |
| a8 | 0000 | ~x2x3 | Y6(y2,y13) | R |
| a4 | 0101 | a0 | 0001 | x4 | Y9(y7) | D0, EWR |
| a5 | 0110 | ~x4~x5x6 | Y10(y6,y9,y10) | +1 |
| a5 | 0110 | ~x4~x5~x6 | Y11(y6,y10) | +1 |
| a8 | 0000 | ~x4x5 | Y6(y2,y13) | R |
| a5 | 0110 | a6 | 0111 | 1 | Y12(y14) | +1 |
| a6 | 0111 | a5 | 0110 | ~x7x6 | Y10(y6,y9,y10) | -1 |
| a5 | 0110 | ~x7~x6 | Y11(y6,y10) | -1 |
| a7 | 1000 | x7x8 | Y13(y11) | +1 |
| a8 | 0000 | x7~x8~x9 | - | R |
| a8 | 0000 | x7~x8x9 | Y6(y2,y13) | R |
| a7 | 1000 | a0 | 0001 | x4 | Y9(y7) | D0, EWR |
| a8 | 0000 | ~x4~x9 | - | R |
| a8 | 0000 | ~x4x9 | Y6(y2,y13) | R |
| a8 | 0000 | a0 | 0001 | x10 | Y14(y12) | D0, EWR |
| a8 | 0000 | ~x10 | - | - |

Логические выражения для каждой функции возбуждения счетчика получают по таблице, как конъюнкции соответствующих исходных состояний am и входных сигналов, которые объединены знаками дизъюнкции для всех строк, содержащих данную функцию возбуждения:

+1 = a0x0 ⋁ a1~x1 ⋁ a1x1 ⋁ a2x0 ⋁ a3~x2~x3x1 ⋁ a3~x2~x3~x1 ⋁ a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a5 ⋁ a6x7x8

-1 = a6~x7x6 ⋁ a6~x7~x6

R = a3~x2x3 ⋁ a4~x4x5 ⋁ a6x7~x8~x9 ⋁ a6x7~x8x9 ⋁ a7~x4~x9 ⋁ a7~x4x9

D1 = D2 = D3 = ⊘

D0 = EWR = a3x2 ⋁ a4x4 ⋁ a7x4 ⋁ a8x10

y0 = a0x0

y1 = a0x0 ⋁ a2x0

y2 = a0x0 ⋁ a3~x2x3 ⋁ a4~x4x5 ⋁ a6x7~x8x9 ⋁ a7~x4x9

y3 = a1x1 ⋁ a3~x2~x3~x1

y4 = a1~x1 ⋁ a1x1 ⋁ a3~x2~x3x1 ⋁ a3~x2~x3~x1

y5 = a1~x1 ⋁ a1x1

y6 = a1~x1 ⋁ a1x1 ⋁ a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a6~x7x6 ⋁ a6~x7~x6

y7 = a4x4 ⋁ a7x4

y8 = a3x2

y9 = a4~x4~x5x6 ⋁ a6~x7x6

y10 = a4~x4~x5x6 ⋁ a4~x4~x5~x6 ⋁ a6~x7x6 ⋁ a6~x7~x6

y11 = a6x7x8

y12 = a8x10

y13 = a3~x2x3 ⋁ a6x7~x8x9 ⋁ a7~x4x9

y14 = a5

Выделение общих частей:

e = a0x0 (2)

f = a1x1 (2)

g = a1~x1 (2)

h = a2x0 (2)

i = a3~x2~x3x1 (4)

j = a3~x2~x3~x1 (4)

k = a3~x2x3 (3)

l = a4~x4~x5x6 (4)

m = a4~x4~x5~x6 (4)

n = a6x7x8 (3)

o = a6~x7x6 (3)

p = a6~x7~x6 (3)

q = a7~x4x9 (3)

r = a8x10 (2)

s = o ⋁ p (2)

t = l ⋁ m (2)

u = s ⋁ t (2)

v = a6x7~x8x9 (4)

w = k ⋁ v ⋁ q (3)

x = a4x4 (2)

y = a7x4 (2)

z = x ⋁ y (2)

ee = f ⋁ g (2)

ff = a3x2 (2)

hh = i ⋁ j (2)

ii = e ⋁ h (2)

После выделения общих частей в логических выражениях и некоторого их упрощения получаем логические уравнения для построения функциональной схемы управляющего автомата.

Функции возбуждения:

+1 = ii ⋁ ee ⋁ hh ⋁ tt ⋁ a5 ⋁ n (6)

-1 = s (0)

R = w ⋁ s ⋁ a7~x4~x9 (6)

D1 = D2 = D3 = 0 (0)

WR = ff ⋁ z ⋁ r (3)

D0 = 1 (0)

y0 = e (0)

y1 = ii (0)

y2 = w ⋁ e ⋁ a4~x4x5 (6)

y3 = f ⋁ j (2)

y4 = ee ⋁ hh (2)

y5 = ee (0)

y6 = ee ⋁ u (2)

y7 = z (0)

y8 = ff (0)

y9 = l ⋁ o (2)

y10 = u (0)

y11 = n (0)

y12 = r (0)

y13 = w (0)

y14 = a5 (0)

Инверторы: ~x0, ~x1, ~x2, ~x3, ~x4, ~x5, ~x6, ~x7 ,~x8, ~x9, ~x10 (11)

Цена комбинационной схемы по Квайну автомата по модели Мили при использовании графа, построенного на основе ГСА, который представлен в приложении Д, с использованием в качестве элементов памяти 4-х разрядного счётчика:

Σ = КС + ИНВ + ЭП + НУ + DC = 95 + 11 + 9 + 1 + 4 = 120

Схема формирования начальной установки на счётчике представлена на рисунке 3.

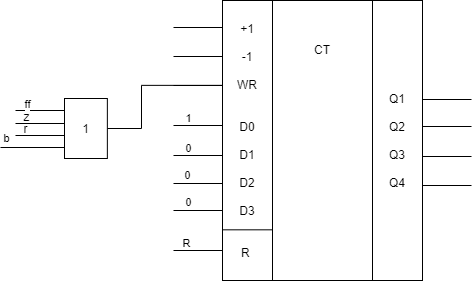


Рисунок 3 – Схема формирования начальной установки на счётчике

## **6.2. Кодирование внутренних состояний для модели Мура на D-триггерах**

Для кодирования 17 состояний (b0…b16) графа автомата Мура, представленного в приложении Е, минимально необходимо пять элементов памяти.

С учетом особенностей работы D-триггера для кодирования состояний применяется эвристический метод. Он состоит в следующем:

1) Каждому состоянию bi ставится в соответствие целое число Ni, равное числу переходов в данное состояние;

2) Числа Ni сортируются в порядке убывания;

3) Состоянию, соответствующему первому Ni после сортировки, то есть наибольшему из Ni, присваивается код, состоящий только из нулей;

4) Следующему состоянию в порядке убывания Ni присваивается незанятый код, содержащий наименьшее количество единиц. Данный пункт повторяется до тех пор, пока все состояния не будут закодированы.

Кодирование состояний для модели Мура на D-триггерах представлено в таблице 13.

Сортировка состояний для модели Мура на D-триггерах представлена в таблице 14.

Структурная таблица переходов и выходов для автомата Мура представлена в таблице 15.

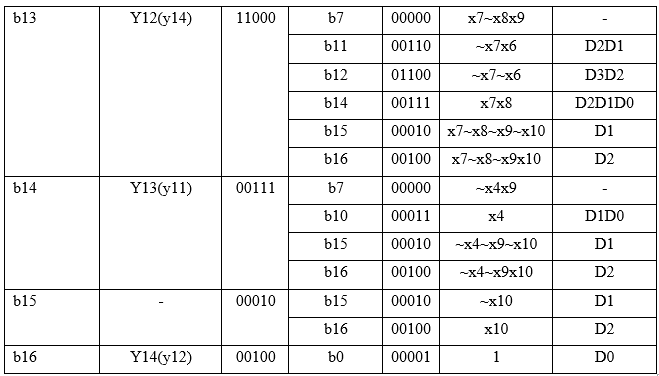
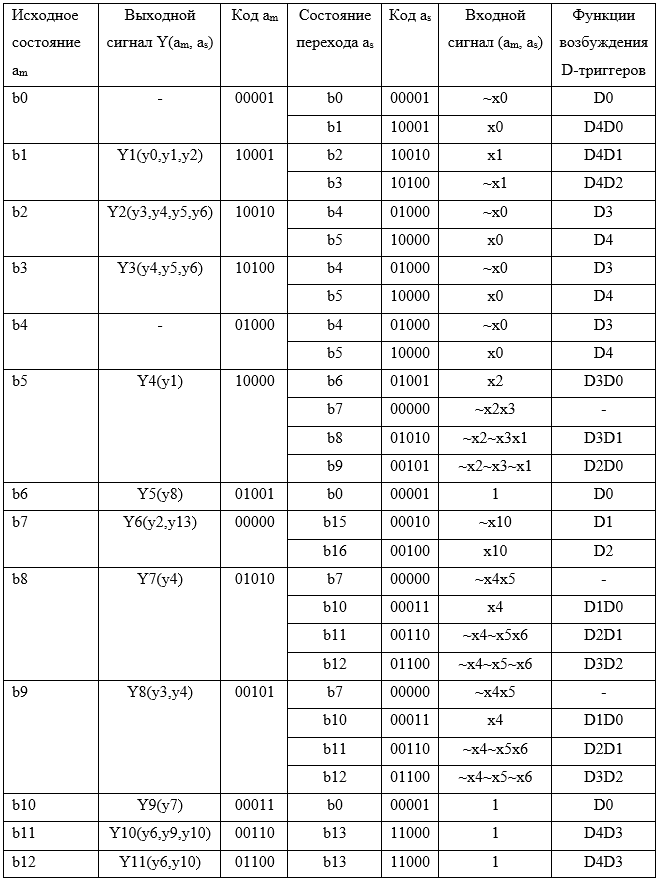
Таблица 13 – Таблица переходов автомата Мура

|  |  |
| --- | --- |
| Состояние | Состояния, из которых есть переход |
| b0 | b0, b6, b10, b16 |
| b1 | b0 |
| b2 | b1 |
| b3 | b1 |
| b4 | b2, b3, b4 |
| b5 | b2, b3, b4 |
| b6 | b5 |
| b7 | b5, b8, b9, b13, b14 |
| b8 | b5 |
| b9 | b5 |
| b10 | b8, b9, b14 |
| b11 | b8, b9, b13 |
| b12 | b8, b9, b13 |
| b13 | b11, b12 |
| b14 | b13 |
| b15 | b7, b13, b14, b15 |
| b16 | b7, b13, b14, b15 |

Таблица 14 – Отсортированная таблица переходов автомата Мура

|  |  |  |  |
| --- | --- | --- | --- |
| Сортировка в порядке убывания | | | |
| Состояние | Состояния, из которых есть переход | Число переходов | Код |
| b7 | b5, b8, b9, b13, b14 | 5 | 00000 |
| b0 | b0, b6, b10, b16 | 4 | 00001 |
| b15 | b7, b13, b14, b15 | 4 | 00010 |
| b16 | b7, b13, b14, b15 | 4 | 00100 |
| b4 | b2, b3, b4 | 3 | 01000 |
| b5 | b2, b3, b4 | 3 | 10000 |
| b10 | b8, b9, b14 | 3 | 00011 |
| b11 | b8, b9, b13 | 3 | 00110 |
| b12 | b8, b9, b13 | 3 | 01100 |
| b13 | b11, b12 | 2 | 11000 |
| b1 | b0 | 1 | 10001 |
| b2 | b1 | 1 | 10010 |
| b3 | b1 | 1 | 10100 |
| b6 | b5 | 1 | 01001 |
| b8 | b5 | 1 | 01010 |
| b9 | b5 | 1 | 00101 |
| b14 | b13 | 1 | 00111 |

Таблица 15 - Структурная таблица переходов и выходов автомата Мура



Логические выражения для каждой функции возбуждения D-триггера получают по таблице как конъюнкции соответствующих исходных состояний bm и входных сигналов, которые объединены знаками дизъюнкции для всех строк, содержащих данную функцию возбуждения:

D4 = b0x0 ⋁ b1x1 ⋁ b1~x1 ⋁ b2x0 ⋁ b3x0 ⋁ b4x0 ⋁ b11 ⋁ b12

D3 = b2~x0 ⋁ b3~x0 ⋁ b4~x0 ⋁ b5x2 ⋁ b5~x2~x3x1 ⋁ b8~x4~x5~x6 ⋁ b9~x4~x5~x6 ⋁ b11 ⋁ b12 ⋁b13~x7~x6

D2 = b1~x1 ⋁ b5~x2~x3~x1 ⋁ b7x10 ⋁ b8~x4~x5x6 ⋁ b8~x4~x5~x6 ⋁ b9~x4~x5x6 ⋁ b9~x4~x5~x6 ⋁ b13~x7x6 ⋁ b13~x7~x6 ⋁ b13x7x8 ⋁ b13x7~x8~x9x10 ⋁ b14~x4~x9x10 ⋁ b15x10

D1 = b1x1 ⋁ b5~x2~x3x1 ⋁ b7~x10 ⋁ b8x4 ⋁ b8~x4~x5x6 ⋁ b9x4 ⋁ b9~x4~x5x6 ⋁ b13~x7x6 ⋁ b13x7x8 ⋁ b13x7~x8~x9~x10 ⋁ b14x4 ⋁ b14~x4~x9~x10 ⋁ b15~x10

D0 = b0~x0 ⋁ b0x0 ⋁ b5x2 ⋁ b5~x2~x3~x1 ⋁ b6 ⋁ b8x4 ⋁ b9x4 ⋁ b10 ⋁ b13x7x8 ⋁ b14x4 ⋁ b16

Аналогично составляем функции выходов:

y0 = b1 (0)

y1 = b1 ⋁ b5 (2)

y2 = b1 ⋁ b7 (2)

y3 = b2 ⋁ b9 (2)

y4 = b2 ⋁ b3 ⋁ b8 ⋁ b9 (4)

y5 = b2 ⋁ b3 (2)

y6 = b2 ⋁ b3 ⋁ b11 ⋁ b12 (4)

y7 = b10 (0)

y8 = b6 (0)

y9 = b11 (0)

y10 = b11 ⋁ b12 (2)

y11 = b14 (0)

y12 = b16 (0)

y13 = b7 (0)

y14 = b13 (0)

Выделение общих частей:

e = b0x0 (2)

f = b1x1 (2)

g = b1~x1 (2)

h = b5x2 (2)

i = b5~x2~x3~x1 (4)

j = b5~x2~x3x1 (4)

k = b8x4 (2)

l = b8~x4~x5x6 (4)

m = b8~x4~x5~x6 (4)

n = b9x4 (2)

o = b9~x4~x5x6 (4)

p = b9~x4~x5~x6 (4)

q = b13~x7x6 (3)

r = b13~x7~x6 (3)

s = b13x7x8 (3)

t = b14x4 (2)

u = i ⋁ s (2)

v = l ⋁ o ⋁ q (3)

w = m ⋁ p ⋁ r (3)

x = n ⋁ t ⋁ k (3)

Упрощение функций переходов:

D4 = e ⋁ f ⋁ g ⋁ b2x0 ⋁ b3x0 ⋁ b4x0 ⋁ b11 ⋁ b12 (14)

D3 = b2~x0 ⋁ b3~x0 ⋁ b4~x0 ⋁ h ⋁ j ⋁ w ⋁ b11 ⋁ b12 (14)

D2 = g ⋁ u ⋁ b7x10 ⋁ v ⋁ w ⋁ b13x7~x8~x9x10 ⋁ b14~x4~x9x10 ⋁ b15x10 (21)

D1 = f ⋁ j ⋁ b7~x10 ⋁ x ⋁ v ⋁ b13x7x8 ⋁ b13x7~x8~x9~x10 ⋁ b14~x4~x9~x10 ⋁ b15~x10 (25)

D0 = b0~x0 ⋁ e ⋁ h ⋁ b6 ⋁ b10 ⋁ x ⋁ b16 ⋁ u (10)

Инверторы: ~x0, ~x1, ~x2, ~x3, ~x4, ~x5, ~x6, ~x7 ,~x8, ~x9, ~x10 (11)

Цена комбинационной схемы по Квайну автомата по модели Мили при использовании графа, построенного на основе ГСА, который представлен в приложении Д, с использованием в качестве элементов памяти 5-х разрядного счётчика:

Σ = КС + ИНВ + ЭП + НУ + DC = 160 + 11 + 20 + 0 + 5 = 196

Схема формирования начальной установки на RS-триггерах представлена на рисунке 4.

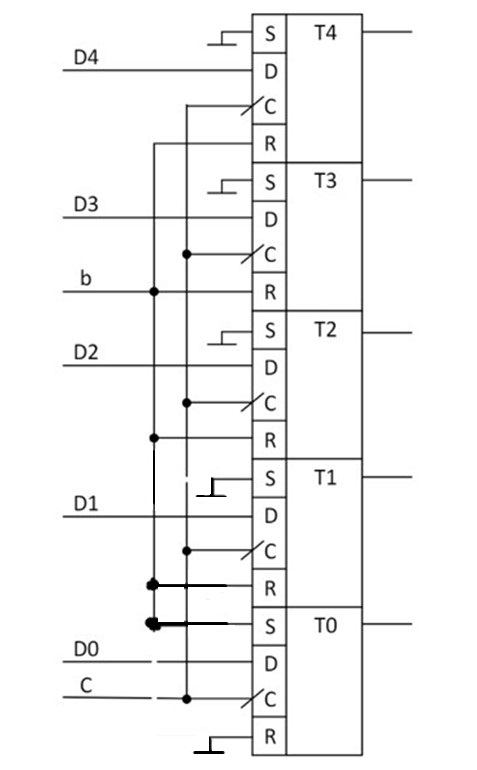


Рисунок 4 – Схема формирования начальной установки на D-триггерах для автомата Мура

Цена по Квайну автомата модели Мура на D-триггерах получилась значительно больше, чем цена по Квайну автомата модели Мили на D- триггерах. Отсюда можно сделать вывод, что цена по Квайну автомата модели Мура на RS-триггерах не будет минимальной.

# 7. ПОСТРОЕНИЕ СХЕМЫ УПРАВЛЯЮЩЕГО МИКРОПРОГРАММНОГО АВТОМАТА

Наименьшей ценой по Квайну обладает модель автомата модели Мили на счетчике, она равна 111, поэтому микропрограммный автомат будет строиться для этой модели. В приложении Ж приведена функциональная схема проектируемого управляющего автомата, выполняющего операцию деления чисел в двоичной системе счисления с плавающей запятой в прямом коде с порядками первым способом с восстановлением остатков с использованием дополнительного кода при вычитании мантисс. Функциональная схема построена в основном логическом базисе «И, ИЛИ, НЕ» в полном соответствии с приведенной для модели Мили системой логических уравнений для функций возбуждения элемента памяти и функций получения выходных сигналов.

ЗАКЛЮЧЕНИЕ

В ходе курсового проекта был синтезирован автомат, выполняющий деление первым способом в двоичной системе счисления с плавающей запятой с порядками с использованием дополнительного кода при вычитании без восстановления остатков в основном логическом базисе. Управляющий автомат был синтезирован по модели Мили с использованием 4-разрядного счетчика в качестве элемента памяти, так как цена по Квайну данного автомата получилась наименьшей и равной 122. Автомат, полученный в ходе выполнения курсового проекта, задан множеством из 9 внутренних состояний a0- a8, множеством из 11 входных сигналов X0-X10, множеством из 15 выходных сигналов y0-y14, функциями переходов и выходов, заданной в таблице 13, начальным состоянием a0.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ДК – дополнительный код

ПК – прямой код

ГСА – граф-схема алгоритма

УА – управляющий автомат

ОА – операционный автомат

МПА – микропрограммный автомат

ЭП – элемент памяти

ПРС – переполнение разрядной сетки

ПМР – потеря младших разрядов

ВрПМР – временная потеря младших разрядов

ИстПМР – истинная потеря младших разрядов

ДНН – деление на ноль

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

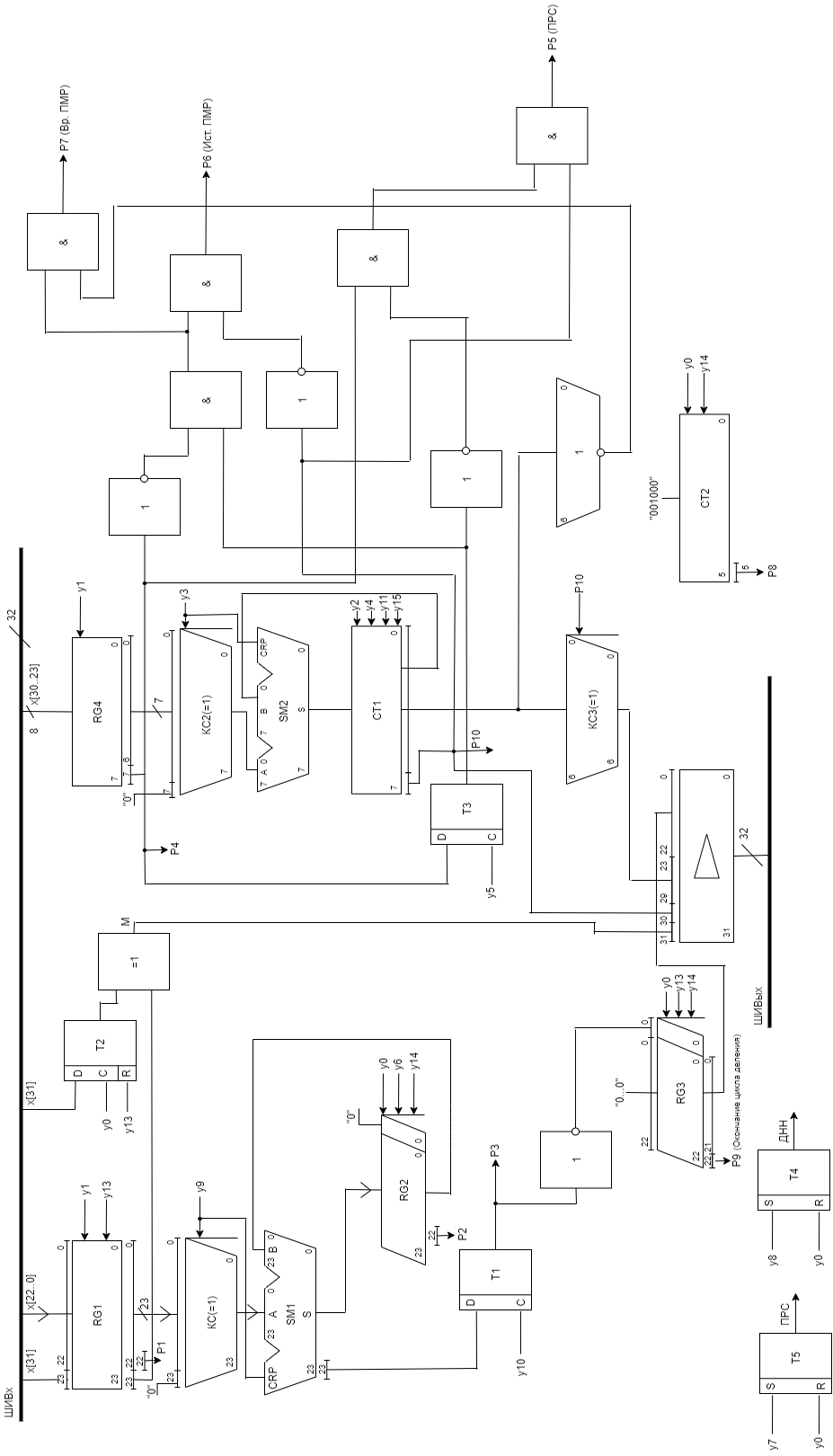
1. Мельцов, В.Ю. Синтез микропрограммных управляющих автоматов. Учебное пособие [Текст] / Мельцов, В. Ю., Фадеева, Т.Р. – Киров: Вятский государственный технический университет, 2000. – 69с.

2. Мельцов, В.Ю. Применение САПР Quartus для синтеза абстрактных и структурных автоматов. Учебное пособие [Текст] – Киров: ГОУ ВПО ВятГУ, 2011. – 86с.

3.Фадеева, Т.Р. Арифметические основы ЭВМ. Методические указания к курсовой работе [Текст] / Фадеева. Т.Р., Матвеева Л.И., Долженкова М.Л. – Киров, 2007. – 80с.

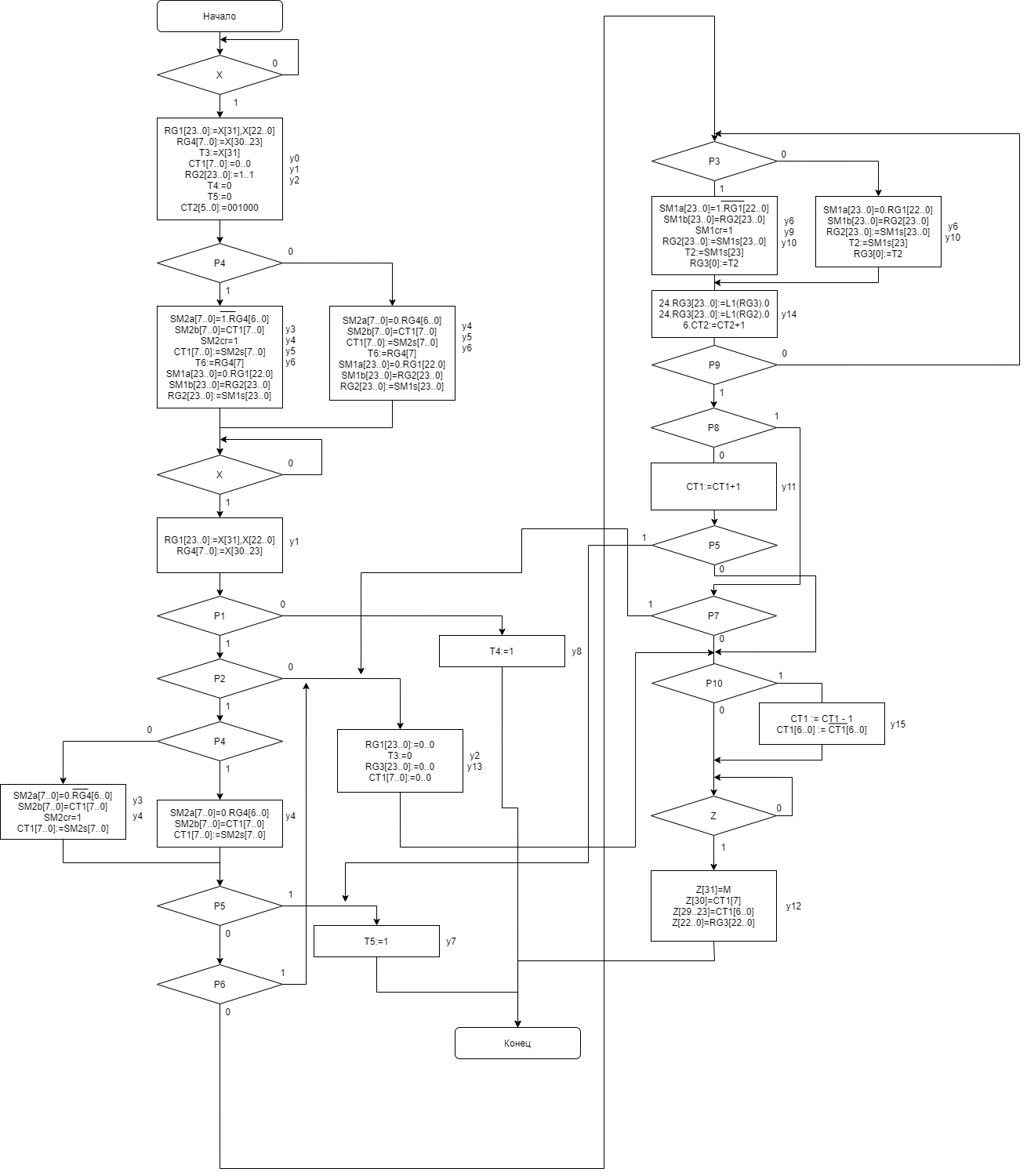
Приложение А. (обязательное)

Функциональная схема операционного автомата



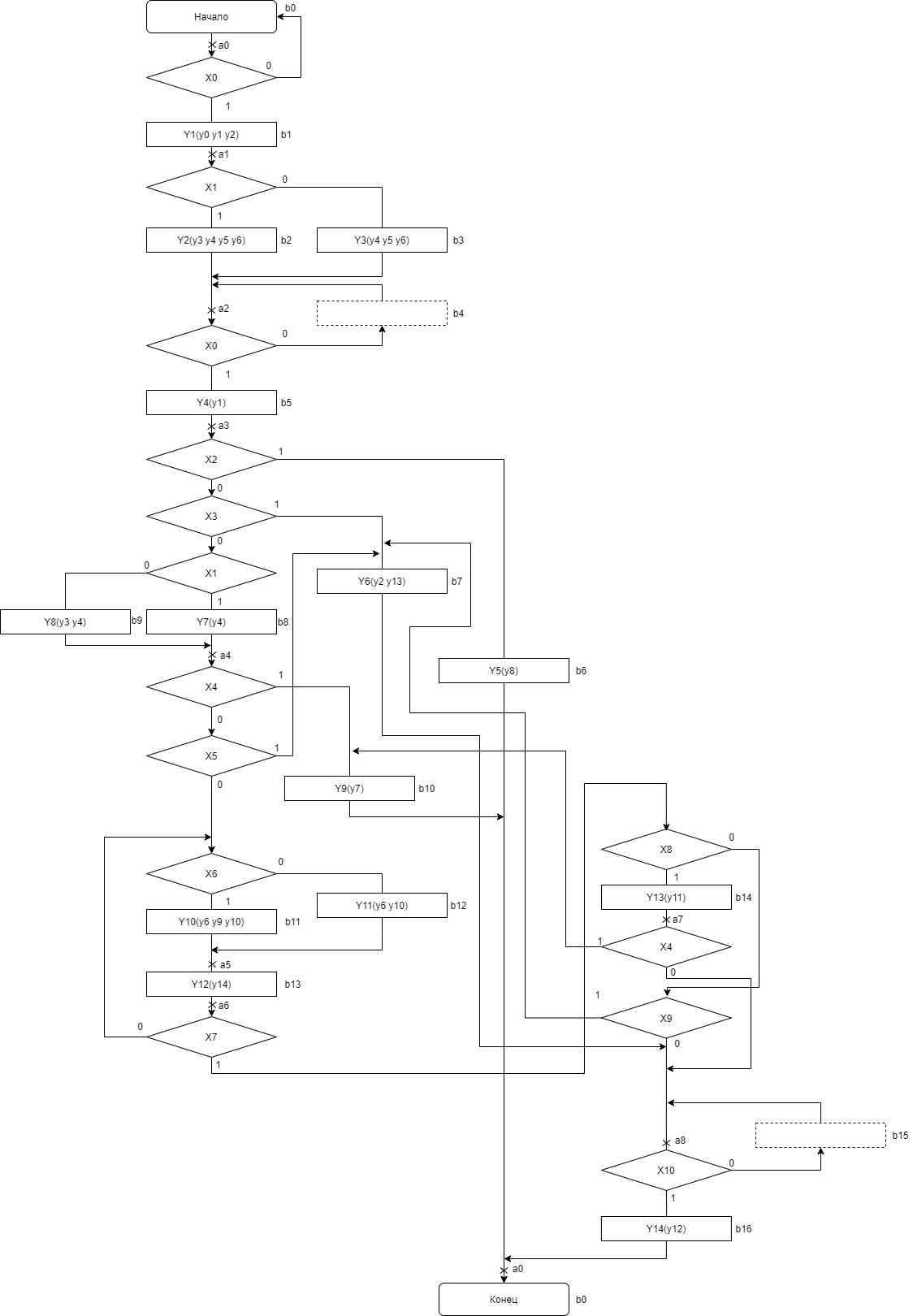
Приложение Б. (обязательное)

Содержательная граф-схема алгоритма



Приложение В. (обязательное)

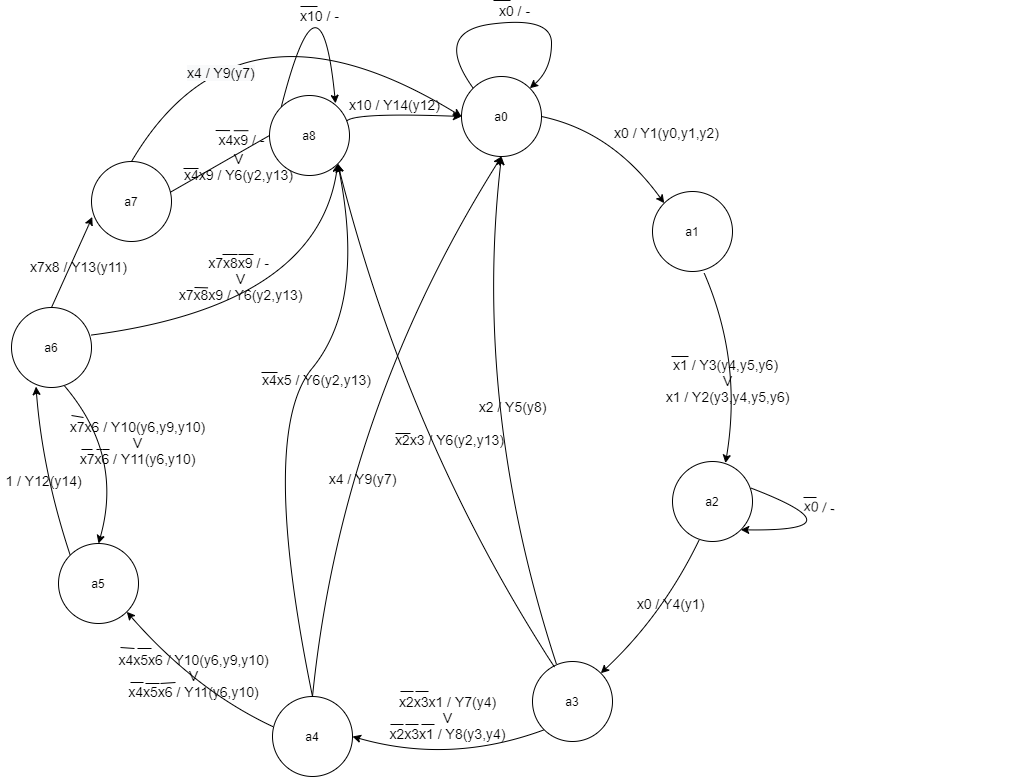
Отмеченная граф-схема алгоритма



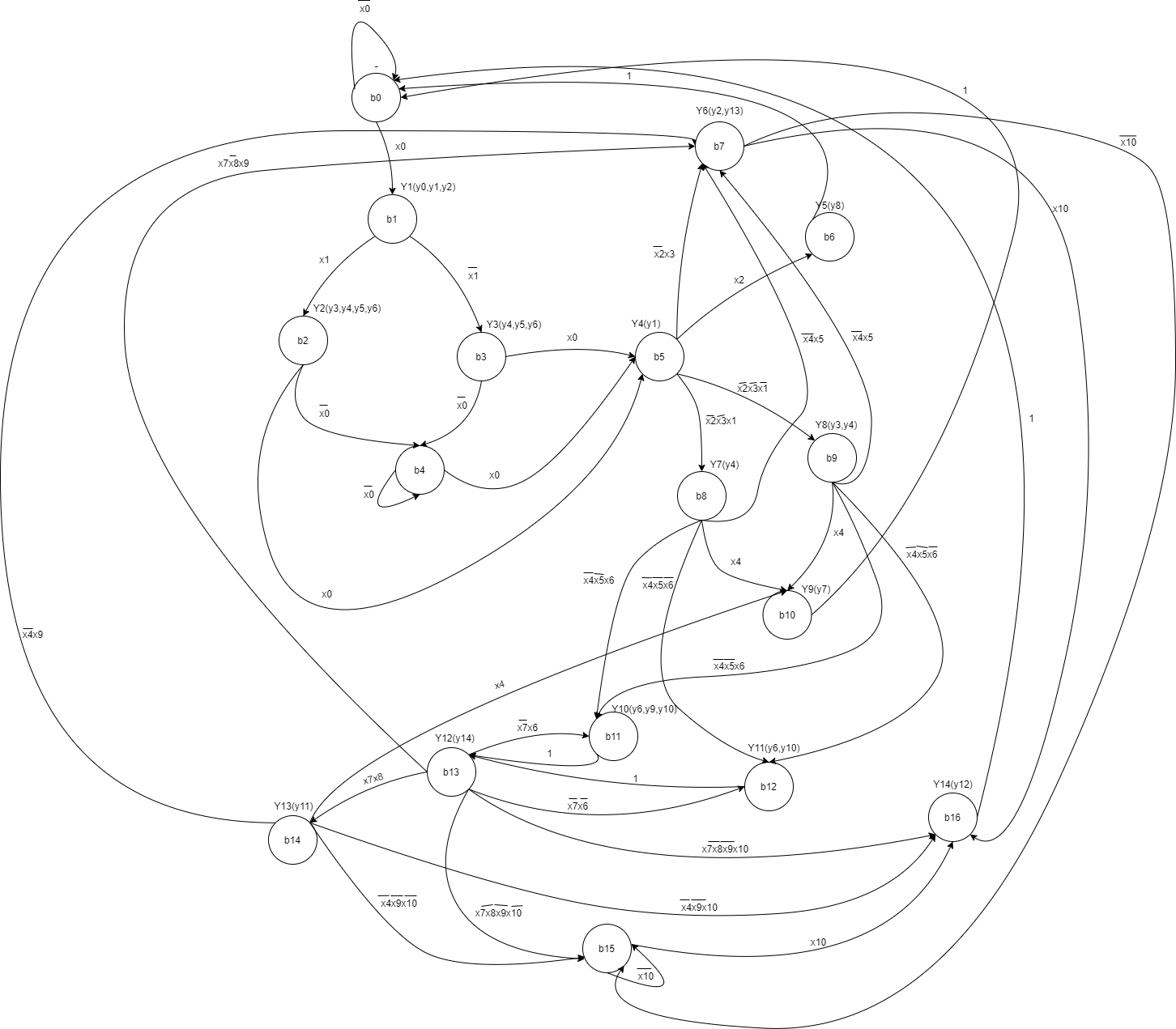
Приложение Г. (обязательное)

Граф автомата модели Мили

Приложение Е. (обязательное)



Граф автомата модели Мура



Приложение Ж ...(обязательное)

Функциональная схема управляющего автомата

