Минобрнауки РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Самарский государственный технический университет»

Кафедра ***"Вычислительная техника"***

**КУРСОВАЯ РАБОТА**

по дисциплине ***"*Моделирование*"***

на тему " **Разработка среды для имитационного моделирования операционного устройства**".

Выполнил студент 3-ИАИТ-119 Коровин Е.А.

Проверил профессор, д.т.н. Мартемьянов Б.В.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Самара 2024***

**СОДЕРЖАНИЕ**

[1 ЗАДАНИЕ 3](#_Toc166457953)

[1.1 Индивидуальный вариант микропрограммы 3](#_Toc166457954)

[1.2 Требования к реализуемым уровням и режимам моделирования объекта 4](#_Toc166457955)

[1.3 Требования к обязательным возможностям интерфейсных средств среды моделирования 5](#_Toc166457956)

[1.4 Требования к оформлению пояснительной записки и особенностям программной реализации 6](#_Toc166457957)

[1.4.1 Язык программирования и графический режим 6](#_Toc166457958)

[1.4.2 Отчетные материалы по курсовому проекту 6](#_Toc166457959)

[1.4.3 Основные требования гост к оформлению текста 7](#_Toc166457960)

[1.4.4 Основные требования к программной реализации 9](#_Toc166457961)

[2 ПРОЕКТНАЯ ЧАСТЬ 10](#_Toc166457962)

[2.1 Авторская оценка соответствия качества проекта объявленным требованиям 10](#_Toc166457963)

[2.3 Математическая постановка задачи 10](#_Toc166457964)

[2.3.1 Описание структур разрядных сеток 10](#_Toc166457965)

[2.3.2 Описание особенностей алгоритма 11](#_Toc166457966)

[2.4 Структурная схема ОУ 11](#_Toc166457967)

[2.5 Описание процесса проектирования 13](#_Toc166457968)

[2.5.1 Коды состояний, микроопераций и условий 13](#_Toc166457969)

[2.5.2 Таблицы переходов УА 15](#_Toc166457970)

[2.5.3 Модель ОУ 18](#_Toc166457971)

[2.5.4 Модель структурной схемы ОУ 18](#_Toc166457972)

[2.6 Описание программной реализации 19](#_Toc166457973)

[2.6.1 Требования к программному и аппаратному обеспечению 19](#_Toc166457974)

[2.6.2 Использованный язык программирования и описание комплекта файлов для запуска программы 19](#_Toc166457975)

[2.6.3 Структура программы 20](#_Toc166457976)

[3 ИНСТРУКЦИИ ПОЛЬЗОВАТЕЛЮ 20](#_Toc166457977)

[3.1 Назначение программы 20](#_Toc166457978)

[3.2 Программные и аппаратные требования 20](#_Toc166457979)

[3.3 Описание интерфейса 20](#_Toc166457980)

[**БИБЛИОГРАФИЧЕСКИЙ СПИСОК** 20](#_Toc166457981)

# 1 ЗАДАНИЕ

## 1.1 Индивидуальный вариант микропрограммы

Индивидуальный вариант ГСА – 9 вариант:

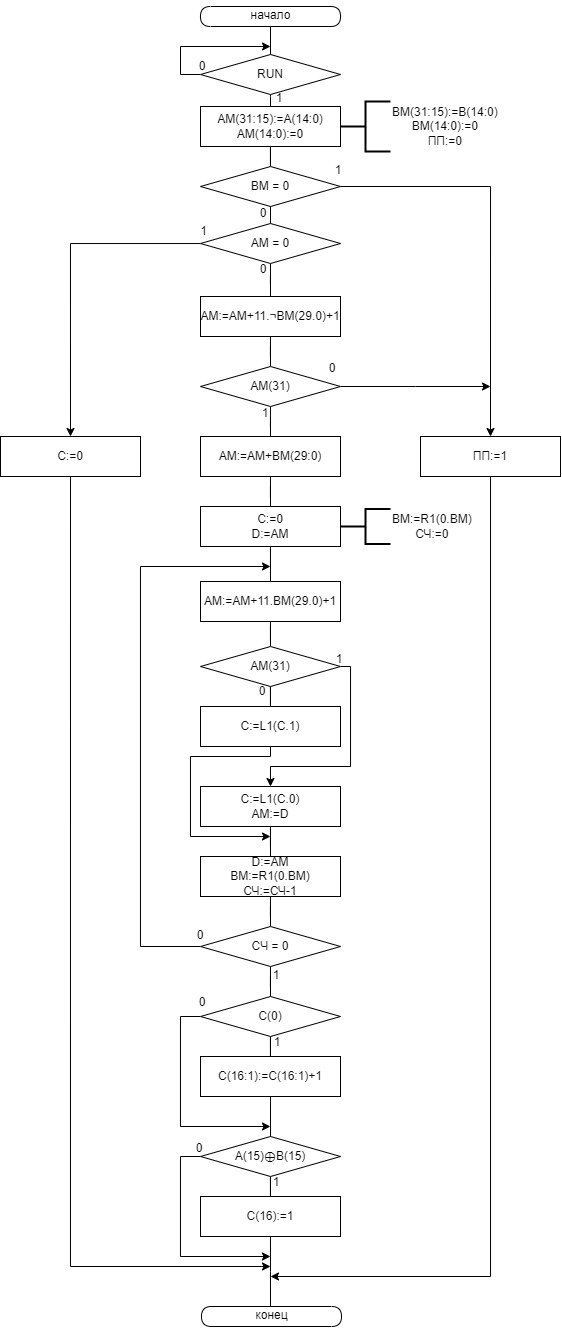


Рисунок 1.1 – ГСА

## 1.2 Требования к реализуемым уровням и режимам моделирования объекта

В интерактивной среде моделирования реализуются два уровня моделирования ОУ:

– моделирование на уровне микропрограммы;

– моделирование на уровне взаимодействия УА и ОА;

***Моделирование ОУ на уровне микропрограммы.*** Ветвления в алгоритме программируются средствами условного оператора if. Модель отдельной микрокоманды представляет собой последовательный вызов соответствующих процедур, моделирующих выполнения микроопераций. Эта последовательность закладывается в разрабатываемый программный код модели.

В результате такого моделирования ОУ можно проверить правильность микропрограммы выполнения арифметической операции.

***Моделирование ОУ на уровне взаимодействия УА и ОА.*** Модель УА строится на основании результатов проектирования УА как автомата Мили на жесткой логике.

***Модель УА формирует вектор Y управляющих сигналов, детализирующий команды для ОА до уровня микроопераций*.** Модель ОА в результате анализа состояния вектора Y вызывает процедуры выполнения указанных вектором микроопераций и затем вычисляет вектор X значений выходных сигналов, поступающих на вход УА.

В модели этого режима запрещается закладывать в программу какую-либо последовательность вызовов процедур моделирования микроопераций. Каждая такая процедура может быть вызвана только как следствие обнаружения конкретного (например единичного) состояния определенной компоненты вектора Y.

Модель УА должна выделять следующие его компоненты:

– память состояний (ПС) УА на D триггерах;

– дешифратор кодов (ДК) состояний УА (это необязательный структурный компонент УА);

– комбинационные схемы (КС), формирующие вектор выходных сигналов Y, и вектор D сигналов управления состояниями разрядов ПС УА;

– память на D триггерах для запоминания некоторых компонент вектора логических условий.

Модели КС строятся на функциональном уровне, т.е. как последовательность булевых выражений, вычисляющих компоненты векторов Y и D.

В интерактивной среде на каждом из уровней моделирования реализуются два режима моделирования ОУ:

– пошаговый режим выполнения микропрограммы;

– автоматический режим выполнения микропрограммы.

В первом из режимов предполагается наличие на форме кнопки с надписью типа «Такт». Каждое нажатие такой кнопки должно вызывать продвижение по микропрограмме на 1 такт. Причем этот такт должен соответствовать понятию такта работы ОУ при проектировании УА на основе модели цифрового управляющего автомата типа Мили.

В автоматическом режиме предполагается наличие на форме кнопки с надписью типа «Выполнить МП». При нажатии на эту кнопку микропрограмма выполняется с начала до конца без вмешательства со стороны пользователя, т.е. автоматически.

## 1.3 Требования к обязательным возможностям интерфейсных средств среды моделирования

***Интерфейсные средства*** среды моделирования должны позволять:

– наблюдать на экране ГСА, размеченную состояниями автомата модели Мили;

– выбирать любой из описанных выше уровней и режимов моделирования:

– задавать начальные значения операндов операции;

– просматривать процесс выполнения микропрограммы по шагам (по тактам) с отображением на форме всех используемых в микропрограмме кодов;

– выполнять микропрограмму автоматически;

– отображать все изменения кодов переменных, участвующих в микропрограмме;

– при моделировании на уровне взаимодействия УА и ОА отображать на форме состояния векторов, задающих все входные, выходные и внутренние векторы УА в предположении, что УА спроектирован как автомат модели Мили на жесткой логике с элементами памяти на D триггерах, а также отображать на ГСА графическую метку, задающую текущее состояние УА.

## 1.4 Требования к оформлению пояснительной записки и особенностям программной реализации

### 1.4.1 Язык программирования и графический режим

Программная часть курсовой работы выполняется с использованием среды ***по выбору студента***, предназначенной для создания Windows приложений. Использованный графический режим видеосистемы должен быть не хуже, чем TrueColor с пространственным разрешением 1024\*768 пикселей.

### 1.4.2 Отчетные материалы по курсовому проекту

В качестве результата выполнения курсового проекта *представляются*:

– какой-либо носитель информации;

– пояснительная записка.

Носитель информации должен содержать:

– исходные тексты всех разработанных программных единиц;

– комплект файлов, достаточный для запуска программы из соответствующей операционной системы (ОС) и среды разработки;

– «бумажную» и электронную версии пояснительной записки, выполненную в текстовом процессоре Word с соблюдением требований ГОСТов на оформление технических текстов;

– файл readme с информацией об авторе, языке программирования и об ОС.

Пояснительная записка должна:

– отображать процесс проектирования программного продукта;

– содержать описание программной реализации;

– содержать инструкции пользователю с описанием интерфейсных средств.

При описании положения на форме интерфейсных средств не допускаются фразы со словами «выше/ниже», «правее/левее». Необходимо использовать нумерованные выноски за границы описываемой формы. Выноски нумеровать по часовой стрелке, начиная с самой левой и верхней. Затем под рисунком формы содержание интерфейсных средств поясняется через номера выносок. При этом желательно такое описание представить в форме таблицы с колонками «№», «Назначение». При этом в колонке «Назначение» не должно быть фраз типа «предназначена для …». Название колонки заменяет подобные фразы.

### 1.4.3 Основные требования гост к оформлению текста

Перечисляемые далее требования соответствуют правилам оформления выпускной квалификационной работы.

1. В качестве основного шрифта используется шрифт Times New Roman. Размер шрифта зависит от формата листа и требований издательств к форматированию текста. Размер может быть 14 пт., 12 пт. и даже 10 пт. В отчетах в качестве основного шрифта Вы должны **использовать шрифт 14 пт**. При этом **названия таблиц и рисунков (они обязательны!)** пишутся шрифтом на 2 пт меньше основного, то есть шрифтом 12 пт. Таким же шрифтом (12 пт) заполняется содержимое таблиц. **Межстрочный интервал** должен быть равным **1,5 строки**. Текст абзацев выравнивать «**по ширине**».

2. **Латинские буквы**, обозначающие переменные, пишутся *курсивным шрифтом*. **Латинские буквы**, обозначающие векторы, пишутся ***полужирным курсивным шрифтом***. **Латинские буквы**, обозначающие матрицы, множества, пишутся **ПРЯМЫМ ПОЛУЖИРНЫМ ШРИФТОМ ПРОПИСНЫМИ БУКВАМИ**. **Греческие буквы**, символы операций и отношений, а также **цифры** пишутся прямым шрифтом (как основной текст документа).

3. Названия таблиц пишутся над таблицами. Строго над таблицами, то есть текст названия не должен выходить за габариты таблицы! При этом правая граница названия должна быть выровнена по правой границе таблицы! В конце названия **точка не ставится**.

4. По поводу рисунков используется понятие «подрисуночная надпись». То есть, название рисунка пишется под рисунком. Строго под рисунком! При этом слово «Рисунок» пишется полностью. В конце названия **точка не ставится**. Название рисунка можно центрировать в строке, можно писать выровненным по левой границе рисунка. Длинные названия пишутся в несколько строк. Примеры смотрите в этом документе.

5. Рисунки и таблицы размещаются по центру строк. Каждая таблица и рисунок «имеют право» на существование в тексте только тогда, когда им предшествует прямая ссылка (упоминание) в тексте. В задании рисунку 2 не предшествовало его упоминание и это плохо! Последнее предложение перед этим рисунком следовало написать в виде, например, «То есть, студент *Иванов Пётр Сергеевич* получает персональный вариант исходного графа, удаляя из него ребра с номерами 10, 12, 13, 15, 16 (рисунок 2).»

6. Таблицы и рисунки отделяются от текста сверху и снизу пустой строкой.

7. Пункты оглавления (названия разделов, подразделов, параграфов …) пишутся без точки в конце. Они могут размещаться по центру либо с абзацным отступом.

8. Пункты оглавления отделяются как от предшествующего, так и от озаглавливаемого текста интервалом большим, чем межстрочный. При этом интервал от предшествующего текста должен быть чуть больше, чем интервал перед озаглавливаемым текстом.

### 1.4.4 Основные требования к программной реализации

1. Необходимо полностью учесть содержание задания

2. Выбрать среду реализации в соответствии с п. 1.4.1

3. Программа не должна завершаться аварийно при любых действиях пользователя и при любых исходных данных.

4. Текст программы должен содержать комментарии к используемым модулям, процедурам, структурам данных, основным переменным и шагам алгоритма. Строки текста писать с отступом от начала строки, подчеркивая подчиненность процедурам, функциям, операторам цикла и операторным скобкам.

5. Модели микропрограммы во всех случаях должны давать правильный результат арифметической операции. Если в исходной ГСА имеются какие-либо логические ошибки, то обнаружить их и исправить – задача автора курсовой работы.

# 2 ПРОЕКТНАЯ ЧАСТЬ

## 2.1 Авторская оценка соответствия качества проекта объявленным требованиям

Проект выполнен со всеми указанными выше требованиями.

## 2.3 Математическая постановка задачи

### 2.3.1 Описание структур разрядных сеток

В проекте были использованы следующие разрядные сетки:

– исходный код регистра A (делимое) – размер сетки 16 бит, позиционная точка расположена после бита под номером 15, знаковый бит под номером 15, тип кода – прямой;

– исходный код регистра B (делитель) – размер сетки 16 бит, позиционная точка расположена после бита под номером 15, знаковый бит под номером 15, тип кода – прямой;

– вспомогательный код регистра AM – размер сетки 32 бита, позиционная точка расположена после бита под номером 31, знаковый бит под номером 31, тип кода – прямой;

– вспомогательный код регистра BM – размер сетки 32 бита, позиционная точка расположена после бита под номером 31, знаковый бит под номером 31, тип кода – прямой;

– вспомогательный код регистра D – размер сетки 32 бита, позиционная точка расположена после бита под номером 31, знаковый бит под номером 31, тип кода – прямой;

– вспомогательный код регистра СЧ (счётчик) – размер сетки 4 бита, все биты выделены под целую часть числа, знаковый бит отсутствует, тип кода – прямой;

– результирующий код регистра C (частное) – размер сетки 17 бит, позиционная точка расположена после бита под номером 16, знаковый бит под номером 16, тип кода – прямой;

### 2.3.2 Описание особенностей алгоритма

При изучении заданного ГСА алгоритма были выделены следующие особенности:

– МП вычисляет частное C:=A/B;

– В МП предусмотрен исход «Переполнение разрядной сетки». Переполнение фиксируется, если |A| ≥ |B|. Это означает, что частное должно быть по модулю строго меньше 1. Поскольку вычисленное частное в следующей команде программы может оказаться делимым или делителем, то естественно принять, что операнды A и B также должны быть по модулю строго меньше 1. Последнее замечание определяет структуру разрядной сетки для представления операндов и результата операции;

– Исход «Переполнение разрядной сетки» фиксируется также в случае, когда B равен нулю, то при попытке деления на 0. Этот случай возможен только тогда, когда операнд А не равен нулю;

– В МП предусмотрен ускоренный вариант получения частного в случае, когда делимое равно 0;

– В МП операнды предполагаются представленными в прямых кодах;

– В МП количество повторений цикла всегда на 1 больше количества вычисляемых разрядов частного, поэтому при завершении циклической части выполняются МО по округлению вычисленного частного;

– В МП реализован алгоритм деления с восстановлением остатка. Но восстановление остатка выполняется не за счет МО вида A:=A+B («Остаток»:= «Остаток» + «Делитель»), а за счет запоминания текущего значения остатка во вспомогательной переменной D и использования на следующем цикле в МО сдвига кода остатка либо только что вычисленного нового остатка, хранящегося в переменной C, либо предыдущего остатка, запомненного в переменной D;

## 2.4 Структурная схема ОУ

Структурная схем ОУ:

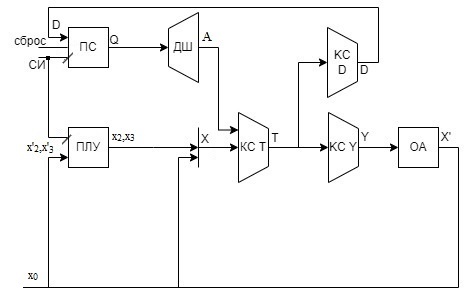


Рисунок 2.1 – Схема структурная ОУ

Данный проект ориентирован на разработку модели описанную структурной схемой выше, беря во внимание существование альтернативных моделей и вариантов структурной схемы.

Назначение структурных элементов ОУ приведены в таблице 2.1:

Таблица 2.1 – Назначение структурных элементов ОУ

|  |  |
| --- | --- |
| Название элемента | Комментарии |
| ПС – регистр памяти кодов состояний | Хранит коды состояний УА |
| ПЛУ – регистр логических условий | Хранит значения логических условий, которые могут привести к сбою работы автомата Мили |
| ДШ – дешифратор кодов состояний | Преобразует коды состояний хранящиеся в ПС |
| КС Т – комбинационная схема термов | Рассчитывает вектор Т логических значений, упрощающих КС Y и КС D |
| КС Y – комбинационная схема Y | Рассчитывает вектор Y логических значений, определяющих выполнение микроопераций |
| КС D – комбинационная схема D | Рассчитывает вектор D логических значений, определяющих состояние ПС на след. такте |
| ОА – операционный автомат | Рассчитывает вектор X значений логических условий, выполняет микрооперации |

При исследовании алгоритма были выявлены функционально зависимые пары. Эти пары приведены в таблице 2.2:

Таблица 2.2 – Функционально связанные пары

|  |  |
| --- | --- |
| Условие | Микрооперация |
| АМ = 0 | AM:=AM+11.¬BM(29.0)+1 |
| АМ(31) | AM:=AM+BM(29:0) |
| AM:=D |

## 2.5 Описание процесса проектирования

### 2.5.1 Коды состояний, микроопераций и условий

Коды условий и коды микроопераций приведены в таблицах 2.3 и 2.4 соответственно:

|  |  |
| --- | --- |
| Условие | Код |
| RUN | x0 |
| BM = 0 | x1 |
| AM = 0 | x2 |
| AM(31) | x3 |
| СЧ = 0 | x4 |
| C(0) | x5 |
| A(0) **⊕** B(0) | x6 |

|  |  |
| --- | --- |
| Микрооперация | Код |
| AM(31:15):=A(14:0) | y0 |
| AM(14:0):=0 | y1 |
| BM(31:15):=B(14:0) | y2 |
| BM(14:0):=0 | y3 |
| ПП:=0 | y4 |
| C:=0 | y5 |
| ПП:=1 | y6 |
| AM:=AM+11.BM(29:0) + 1 | y7 |
| AM:=AM+BM(29:0) | y8 |
| D:=AM | y9 |
| BM:=R1(0.BM) | y10 |
| СЧ:=0 | y11 |
| C:=L1(C.1) | y12 |
| C:=L1(C.0) | y13 |
| AM:=D | y14 |
| СЧ:=СЧ-1 | y15 |
| C(16:1):=C(16:1)+1 | y16 |
| C(16):=1 | y17 |

Таблица 2.3 – Коды условий Таблица 2.4 – Коды микроопераций

Полученная закодированная и размеченная ГСА:

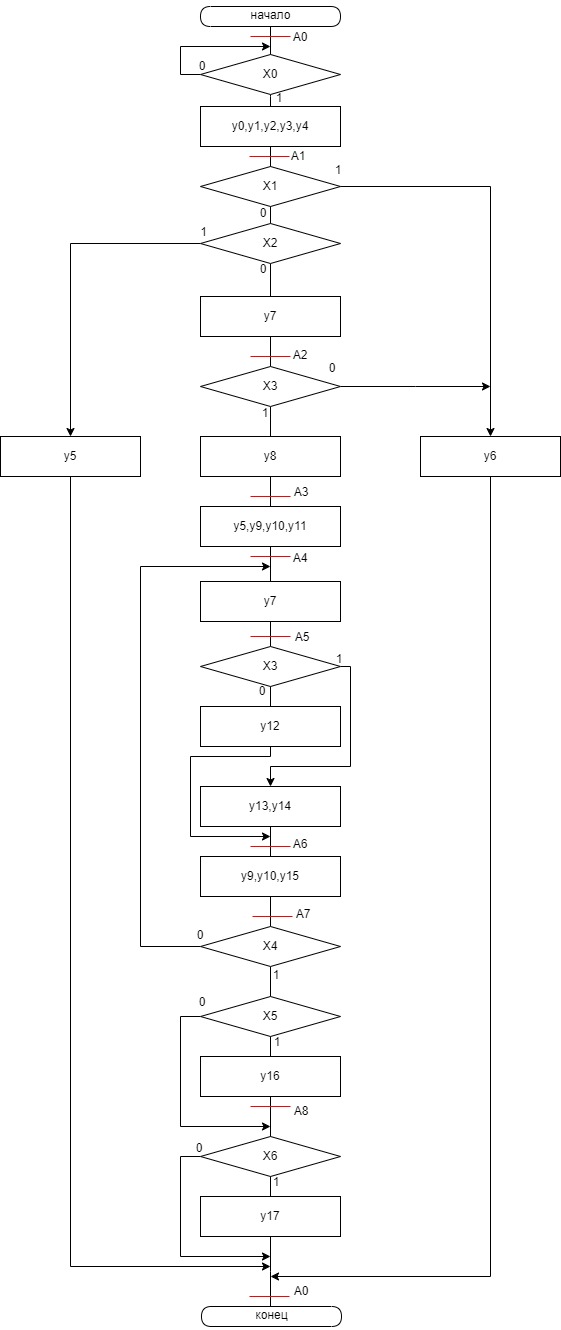


Рисунок 2.2 – ГСА размеченная

При кодировании состояний использовался метод минимального кодирования. Назначенные коды состояний приведены в таблице 2.5:

|  |  |
| --- | --- |
| Метка | Код состояния |
| A0 | 0000 |
| A1 | 0001 |
| A2 | 0010 |
| A3 | 0011 |
| A4 | 0100 |
| A5 | 0101 |
| A6 | 0110 |
| A7 | 0111 |
| A8 | 1000 |

Таблица 1.3 – Коды состояний

### 2.5.2 Таблицы переходов УА

Для контроля переноса всех переходов с ГСА составлена прямая таблица переходов (Таблица 2.6):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Am | As | Xms | Yms |
| 1 | A0 | A0 | *x̅*0 | - |
| 2 | A1 | *x*0 | *y*0,*y*1,*y*2,*y*3,*y*4 |
| 3 | A1 | A0 | *x*1 | *y*6 |
| 4 | A0 | *x̅*1*x*2 | *y*5 |
| 5 | A2 | *x̅* 1 *x̅* 2 | *y*7 |
| 6 | A2 | A0 | *x̅*3 | *y*6 |
| 7 | A3 | *x*3 | *y*8 |
| 8 | A3 | A4 | 1 | *y*5,*y*9,*y*10,*y*11 |
| 9 | A4 | A5 | 1 | *y*7 |
| 10 | A5 | A6 | *x̅*3 | *y*12 |
| 11 | A6 | *x*3 | *y*13,*y*14 |
| 12 | A6 | A7 | 1 | *y*9,*y*10,*y*15 |
| 13 | A7 | A5 | *x̅*4 | *y*7 |
| 14 | A8 | *x*4*x*5 | *y*16 |
| 15 | A0 | *x*4 *x̅*5 *x̅*6 | - |
| 16 | A0 | *x*4 *x̅*5 *x*6 | *y*17 |
| 17 | A8 | A0 | *x̅*6 | - |
| 18 | A0 | *x*6 | *y*17 |

Таблица 2.6 – Прямая таблица переходов

Контроль полноты переходов для нетривиальных случаев описан в таблице 2.7:

Таблица 2.7 – Проверка полноты переходов

|  |  |
| --- | --- |
| Am | Проверка |
| A1 | *x*1 v *x̅*1*x*2 v *x̅* 1 *x̅* 2 = *x̅*1 (*x*2 v *x̅* 2) v *x*1 = *x̅*1 (1) v *x*1 = 1 |
| A7 | *x̅*4 v *x*4*x*5 v *x*4 *x̅*5 *x̅*6 v *x*4 *x̅*5 *x*6 = *x̅*4 v *x*4 (*x*5 v *x̅*5 *x̅*6 v *x̅*5 *x*6) = *x̅*4 v *x*4 (*x*5 v *x̅*5(*x̅*6 v *x*6)) =  = 1 ( 1 (1)) = 1 |

Для проектирования автомата составлена обратная структурная таблица (Таблица 2.8):

Таблица 2.8 – Обратная структурная таблица

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Am | Km | As | Ks | Xms | Yms | Tms | Fms |
| 1 | A0 | 0000 | A0 | 0000 | *x̅*0 | - | *t*0=*a*0 *x̅*0 | - |
| 2 | A1 | 0001 | *x*1 | *y*6 | *t*1=*a*1 *x*1 |
| 3 | A1 | 0001 | *x̅*1*x*2 | *y*5 | *t*2=*a*1 *x̅*1*x*2 |
| 4 | A2 | 0010 | *x̅*3 | *y*6 | *t*3=*a*2 *x̅*3 |
| 5 | A7 | 0111 | *x*4 *x̅*5 *x̅*6 | - | *t*4=*a*7 *x*4 *x̅*5 *x̅*6 |
| 6 | A7 | 0111 | *x*4 *x̅*5*x*6 | *y*17 | *t*5=*a*7 *x*4 *x̅*5*x*6 |
| 7 | A8 | 1000 | *x̅*6 | - | *t*6=*a*8 *x̅*6 |
| 8 | A8 | 1000 | *x̅*6 | *y*17 | *t*7=*a*8 *x̅*6 |
| 9 | A0 | 0000 | A1 | 0001 | *x*0 | *y*0,*y*1,*y*2,*y*3,*y*4 | *t*8=*a*0 *x*0 | D0 |
| 10 | A1 | 0001 | A2 | 0010 | *x̅*1 *x̅*2 | *y*7 | *t*9=*a*1 *x̅*1 *x̅*2 | D1 |
| 11 | A2 | 0010 | A3 | 0011 | *x*3 | *y*8 | *t*10=*a*2 *x*3 | D0,D1 |
| 12 | A3 | 0011 | A4 | 0100 | 1 | *y*5,*y*9,*y*10,*y*11 | *t*11=*a*3 | D2 |
| 13 | A4 | 0100 | A5 | 0101 | 1 | *y*7 | *t*12=*a*4 | D0,D2 |
| 14 | A7 | 0111 | *x̅*4 | *y*7 | *t*13=*a*7 *x̅*4 |
| 15 | A5 | 0101 | A6 | 0110 | *x̅*3 | *y*12 | *t*14=*a*5 *x̅*3 | D1,D2 |
| 16 | A5 | 0101 | *x*3 | *y*13,*y*14 | *t*15=*a*5 *x*3 |
| 17 | A6 | 0110 | A7 | 0111 | 1 | *y*9,*y*10,*y*15 | *t*16=*a*6 | D0,D1,D2 |
| 18 | A7 | 0111 | A8 | 1000 | *x*4*x*5 | *y*16 | *t*17=*a*7 *x*4*x*5 | D3 |

### 2.5.3 Модель ОУ

Модель ОУ строится как совокупность структурных моделей блоков, составляющих ОУ. КС Y, КС D, КС Т – система булевых выражений. Функции для КС Y приведены в таблице 2.9, а для КС D в таблице 2.10:

Таблица 2.9 – Функции для КС Y Таблица 2.10 – Функции для КС D

|  |  |
| --- | --- |
| *y*i | Fi |
| *y*0 | *t*8 |
| *y*1 | *t*8 |
| *y*2 | *t*8 |
| *y*3 | *t*8 |
| *y*4 | *t*8 |
| *y*5 | *t*2 v *t*11 |
| *y*6 | *t*1 v *t*3 |
| *y*7 | *t*9 v *t*12 v *t*13 |
| *y*8 | *t*10 |
| *y*9 | *t*11 v *t*16 |
| *y*10 | *t*11 v *t*16 |
| y11 | *t*11 |
| y12 | *t*14 |
| y13 | *t*15 |
| *y*14 | *t*15 |
| *y*15 | *t*16 |
| *y*16 | *t*17 |
| *y*17 | *t*5 v *t*7 |

|  |  |
| --- | --- |
| Di | Fi |
| D0 | *t*8 v *t*10 v *t*12 v *t*13 v *t*16 |
| D1 | *t*9 v *t*10 v *t*14 v *t*15 v *t*16 |
| D2 | *t*11 v *t*12 v *t*13 v *t*14 v *t*15 v *t*16 |
| D3 | *t*17 |

### 2.5.4 Модель структурной схемы ОУ

Порядок обращений к модели структурой схемы ОУ представлен на рисунке

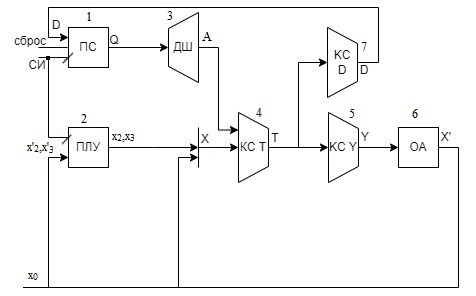


Рисунок 1.3 – Порядок обращений к компонентам ОУ

## 2.6 Описание программной реализации

### 2.6.1 Требования к программному и аппаратному обеспечению

Операционная система – Windows 10 и выше;

Аппаратное обеспечение:

– процессор мощностью 1,6 ГГц или выше;

– 2 ГБ ОЗУ;

– 420 КБ свободного дискового пространства;

– видеоадаптер, совместимый с DirectX 9 и поддерживающий разрешение экрана 1024 x 768 точек и выше.

### 2.6.2 Использованный язык программирования и описание комплекта файлов для запуска программы

В данном проекте был использован язык программирования C#. Для запуска программы необходим следующий комплект файлов:

– modeling.exe

– modeling.exe.config

– modeling.pdb

Все файлы должны быть расположены на одном уровне для успешного запуска программы.

### 2.6.3 Структура программы

Программы состоит из следующих модулей:

|  |  |
| --- | --- |
| Модуль | Комментарии |
| Класс **OperatingMachine** | Выполняет микроопераций, вырабатывает значения логических условий |
| Класс **ControlMachine** | Выполняет генерацию значений КС Т, КС Y, КС D |
| Класс **OperatingDevice** | Моделирует работу такта на уровнях микропрограммы и взаимодействия УА и ОА |
| Класс **OperatingDeviceDetails** | Инкапсулирует данные ОУ и представляет их в удобном формате |
| Класс **Form1** | Создаёт форму с пользовательским интерфейсом |

# 3 ИНСТРУКЦИИ ПОЛЬЗОВАТЕЛЮ

## 3.1 Назначение программы

## 3.2 Программные и аппаратные требования

## 3.3 Описание интерфейса

# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**