

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тульский государственный университет»

Политехнический институт  
Кафедра «Промышленная автоматика и робототехника»

Утверждено на заседании кафедры  
«Промышленная автоматика и робототех-  
ника»  
« 17 » января 2023г., протокол № 2

И.о. заведующего кафедрой

 О.А.Ерзин

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**по выполнению курсовой работы**  
**по дисциплине (модулю)**  
**«Цифровое управление технологическими машинами и оборудованием»**

**основной профессиональной образовательной программы**  
**высшего образования – программы бакалавриата**

по направлению подготовки  
15.03.02 Технологические машины и оборудование

с направленностью (профилем)  
Информационно-измерительные и управляющие системы  
технологических машин

Форма обучения: очная

Идентификационный номер образовательной программы: 150302-01-22

Тула 2022 год

## Разработчик(и) методических указаний

Зайчиков Игорь Вячеславович, канд.техн.наук, доц.

*(ФИО, должность, ученая степень, ученое звание)*

*(подпись)*



*(ФИО, должность, ученая степень, ученое звание)*

*(подпись)*

## 1. ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Курсовая работа предназначена для выработки у студентов умений и навыков применения основных теоретических положений курса в практической работе по синтезу цифровых управляющих устройств.

В процессе выполнения курсовой работы студент должен научиться составлять алгоритмы функционирования цифровых управляющих устройств;

разделять цифровое управляющее устройство на операционный и управляющий автоматы;

синтезировать структуру операционного автомата и строить ее из типовых функциональных модулей;

проводить формальный синтез управляющего автомата с кодированием его состояний;

выполнять структурный синтез автомата с жесткой логикой, включая минимизацию логических функций и приведение их к заданному логическому базису.

## 2. ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

### 2.1. Объект синтеза курсовой работы

Объектом синтеза курсовой работы является цифровое управляющее устройство с жесткой логикой, выполняющее функции получения цифровой информации от внешнего, по отношению к устройству, источника, ее предварительную обработку и выдачу результатов обработки в виде цифрового кода. Логика работы цифрового управляющего устройства должна предусматривать выполнение, как минимум, одной арифметической операции, указанной в задании.

В случае, если студент участвует в выполнении научно-исследовательской работы под руководством ведущих преподавателей кафедры «Робототехника и автоматизация производства», возможна выдача ему индивидуального задания с сохранением общей направленности тематики курсовой работы. Индивидуальное задание должно быть согласовано научным руководителем с ведущим преподавателем по дисциплине и утверждено на заседании кафедры. В приеме такой работы участвуют научный руководитель и ведущий преподаватель, причем требования к объему и уровню выполнения задания не должны снижаться.

### 2.2. Исходные данные к курсовой работе

Исходные данные к курсовой работе приведены в таблицах Приложения 1. Номера вариантов выбираются в соответствии с порядковым номером студента в алфавитном списке группы.

Табл. П.1 определяет состав обрабатываемых цифровым управляющим

устройством данных и описание работы устройства.

Табл. П.2 определяет типы логических элементов для реализации комбинационных и последовательностных схем.

### 2.3. Объем курсовой работы

В процессе выполнения курсовой работы студенты должны:

составить функциональную схему операционного автомата, реализующего заданные элементарные операции;

разработать алгоритм функционирования цифрового управляющего устройства;

выбрать тип абстрактного автомата для реализации управляющего автомата;

произвести формальный синтез управляющего автомата;

привести управляющий автомат к заданным базисам комбинационных схем и элементов памяти в виде электрической схемы, правильность которой должна быть проверена в программе моделирования типа Elektronik Work Bench 5.12;

разработать структурную схему управляющего автомата с заданным базисом.

Содержание пояснительной записки должно содержать следующие основные разделы:

1. Задание в соответствии со своим вариантом

2. Содержание

3. Введение

4. Основная часть

4.1. Описание функциональной схемы цифрового управляющего устройства, реализующего заданные элементарные операции;

4.1.1. Описание функций блока 1

4.1.2. Описание функций блока 2

4.1.....

4.1.N.Описание функций блока N

4.2. Описание алгоритма функционирования цифрового управляющего устройства;

4.3. Описание электрической схемы цифрового управляющего устройства;

4.3.1. Описание электрической схемы блока 1

4.3.2. Описание электрической схемы блока 2

4.3.....

4.3.N.Описание электрической схемы блока N

5. Заключение

6. Список использованной литературы

7. Приложения

7.1. Электрическая схема цифрового управляющего устройства

7.2. Временные диаграммы сигналов цифрового управляющего устройства

### 2.4. Требования к пояснительной записке

Результаты синтеза отражаются в пояснительной записке объемом от 15 до 25 страниц машинописного текста без приложений. Титульный лист записки приведен в Приложении 2.

Пояснительная записка содержит текст, таблицы, рисунки и формулы. Текст должен быть предельно кратким. Его основное назначение - связать материал в единую логическую последовательность. Таблицы, формулы, рисунки должны пояснять *только разработанные студентом* структуры, булевы функции, оптимизации булевых выражений и т.п., в соответствии с выданным заданием на курсовую работу. В пояснительной записке запрещается приводить справочные данные, заимствованные из других источников. Используемые при выполнении курсовой работы источники, на которые имеются ссылки в тексте пояснительной записки, должны быть перечислены в списке литературы.

Курсовая работа выполняется на листах писчей белой бумаги формата А4 (210\*297 мм). Текст должен быть набран в текстовом редакторе Word и распечатан на принтере. Математические выражения должны быть набраны во встроенном редакторе Microsoft Equation-3. Рисунки должны быть сделаны во встроенном графическом редакторе Word'a, или нарисованы по тексту карандашом.

## 2.5. Защита курсовой работы

На защиту представляются пояснительная записка к курсовой работе и рецензия. Предварительно материалы курсовой работы и рецензия должны быть представлены на сайте

Курсовая работа защищается перед комиссией, назначаемой распоряжением заведующего кафедрой. На защите студент должен кратко изложить основные результаты каждого из выполненных этапов, после чего ему могут быть заданы вопросы по его работе, на которые должны быть даны полные ответы. Оценка курсовой работы складывается из следующих оценок:

- качество выполнения работы;
- оценка рецензента;
- оценка знаний, обнаруженных в процессе защиты.

Неудовлетворительная оценка по курсовой работе может быть выставлена студенту в следующих случаях:

- пояснительная записка к курсовой работе не представлена на защиту;
  - на защиту представлена пояснительная записка с чужим вариантом курсовой работы;
  - в пояснительной записке отсутствует решение поставленной задачи.
- студент показывает нулевые знания по курсовой работе и по предмету в целом и не может дать пояснения по выполненным этапам курсовой работы.

В случае получения неудовлетворительной оценки по результатам защиты студенту выдается новое задание.

### 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

#### 3.1. Общая структура цифровых устройств

Общая функциональная схема цифрового автомата приведена на рис. 1.

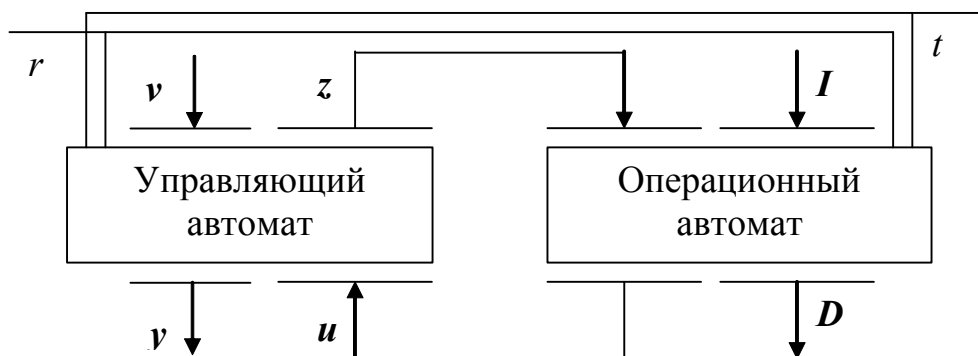


Рис. 1. Общая функциональная схема цифрового автомата

В состав любого цифрового автомата входят операционный автомат и управляющий автомат.

Операционный автомат объединяет функциональные модули, производящие непосредственную обработку поступающей информации. Операционный автомат имеет информационные входы  $I$  и входы  $z$  управления; информационные выходы  $D$ , а также выходы  $u$ , которые сигнализируют о результатах выполнения операций. Сигналы, формируемые на этих выходах, называются осведомительными сигналами или внутренними логическими условиями устройства. Операционный автомат состоит из типовых функциональных модулей, таких, как параллельные и последовательные регистры, счетчики, комбинационные сумматоры, схемы сравнения, мультиплексоры шифраторы, дешифраторы и др. В робототехнических системах и комплексах частью операционного автомата можно считать функциональные блоки, производящие определенные действия, например перемещение манипулятора, опрос датчиков, раскрытие схвата, и т.п. Результатом функционирования указанных блоков должно являться появление сигналов, имеющих два состояния: 0 и 1.

Управляющий автомат вырабатывает символы  $z$  управления операционным автоматом по заданной программе с учетом значений внутренних  $u$  и внешних  $v$  логических условий, которые для него являются входными переменными. Внешние логические условия, задают одну из нескольких возможных в данном устройстве микропрограмм. На выходах  $u$  управляющего автомата могут быть сформированы символы, несущие информацию для внешних устройств о состоянии цифрового устройства. Управляющий автомат определяет логику работы устройства, т.е. последовательность и тип операций, выполняемых операционным автоматом над исходными данными.

Общими управляющими входами автомата являются вход тактирования  $s$  и вход сброса  $r$ . По входу  $s$  осуществляется синхронизация работы операцион-

ного и управляющего автомата. По входу  $r$  производится установки внутренних элементов памяти автоматов в состояние, которое считается исходным (начальным).

Управляющий автомат можно строить двумя способами: как автомат с жесткой логикой, или как автомат со специальной встроенной памятью микропрограмм. Оба принципа формирования УА применяются в робототехнических системах, причем при реализации сложных алгоритмов используется микропрограммное управление, а при реализации простых алгоритмов - схемная логика.

Настоящая курсовая работа предполагает разработку управляющего автомата с жесткой логикой.

### 3.2. Синтез операционного автомата

Общая структура операционного автомата приведена на рис. 2. В состав типового операционного автомата могут входить:

мультиплексоры данных MX1, MX2;

регистры данных RG1, RG2;

комбинационные схемы выполнения математических операций L1, ..., Ln;

комбинационная схема формирования осведомительных символов Lu.

На входы операционного автомата подаются данные  $I_0$ ,  $I_1$  и вектор символов управления  $z$ . Отдельно может подаваться синхронизирующая последовательность импульсов  $c$ , которая может быть общей с синхронизирующей последовательностью управляющего автомата.

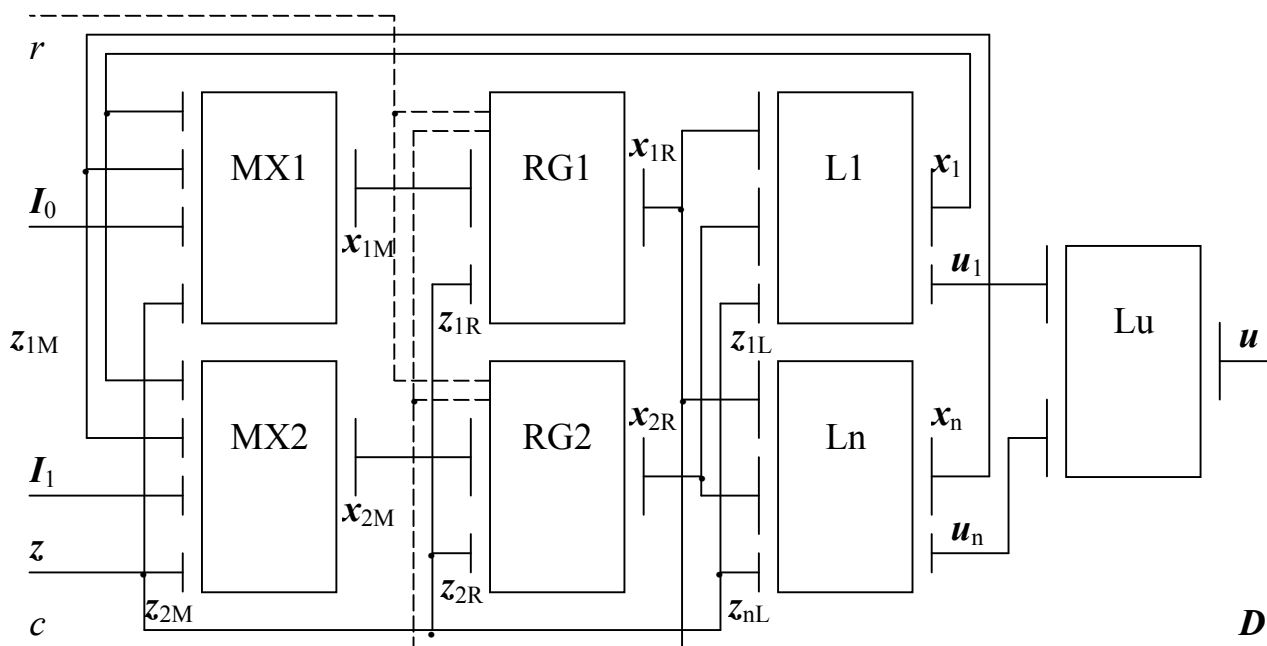


Рис. 2. Типовая схема операционного автомата

По входу  $r$  производится установка регистров в начальное состояние,

например во все разряды регистров записывается логический ноль. По входу  $c$  производится тактирование регистров, т.е., под управлением этого входа триггеры регистров устанавливаются в состояние, которое определяется кодом, поданным на их информационные входы  $x_{1M}, x_{2M}$ .

На выходах  $D$  автомата формируется результат. На выходах  $u$  формируется признак результата.

Операционный автомат функционирует следующим образом.

Мультиплексоры  $MX1$  и  $MX2$  из множества данных, поданных на их информационные входы, выбирают под управлением кодов  $z_{1M}$  и  $z_{2M}$  данные, необходимые в текущий момент для выполнения заданной операции. При этом множества выбираемых кодов могут формироваться из входных данных  $I_0, I_1$  и результатов  $x_1 - x_n$  выполнения элементарных операций комбинационными схемами  $L1 - Ln$ . Выбранный код синхросигналом  $c$  записывается в регистры  $RG1, RG2$ . В результате этого на выходах регистров формируются коды  $x_{1R}$  и  $x_{2R}$ , а на выходах комбинационных схем  $L1 - Ln$  - очередные результаты  $y_1 - y_n$  и признаки результатов  $u_1 - u_n$ . Из признаков  $u_1 - u_n$  на выходе  $u$  с помощью комбинационной схемы  $Lu$  формируется признак результата, после чего на входы могут быть поданы следующие данные и управляющие символы.

В качестве комбинационных схем могут использоваться:

сумматоры, реализующие арифметическую операцию  $x_{1R} + x_{2R}$ ;

схемы сравнения кодов, реализующие одно из отношений:  $x_{1R} > x_{2R}$ ;  $x_{1R} = x_{2R}$ ;  $x_{1R} \neq x_{2R}$ ;  $x_{1R} < x_{2R}$ ;

логические схемы, выполняющие операции отрицания, конъюнкции, дизъюнкции, суммирования по модулю 2 и т.п.;

шифраторы;

другие комбинационные схемы.

Состав комбинационных схем  $L1 - Ln$  определяется номенклатурой операций, определенных заданием на курсовую работу. Операции могут выполняться последовательно, разряд за разрядом, в этом случае проектируется последовательный операционный автомат, или параллельно, над всем кодом сразу, в этом случае проектируется параллельный операционный автомат. В последовательном операционном автомате комбинационные схемы  $L1 - Ln$  содержат структуры, предназначенные для обработки одного разряда данных, в параллельном операционном автомате - однотипные структуры для обработки сразу нескольких разрядов.

Регистры  $RG1, RG2$  являются элементами памяти, предназначенными для запоминания и хранения исходных данных, промежуточных и окончательных результатов обработки данных операционным автоматом. Количество разрядов в регистре определяется разрядностью обрабатываемых данных и количеством разрядов, необходимых для промежуточного хранения признаков результата. Регистры могут быть последовательными/сдвигowymi (в этом случае данные в них вводятся/сдвигаются последовательно, бит за битом, от старшего/младшего бита к младшему/старшему), или параллельными (в этом случае данные в них вводятся параллельно во все разряды регистра). В некоторых



случаях требуется универсальные регистры, в которых при различных кодах  $z_{1R}$  и  $z_{2R}$  данные могут сдвигать последовательно, в одном из направлений, параллельно или сдвиг данных в них может быть блокирован.

Существенным является тип триггера для реализации регистров. В курсовой работе принимается, что триггеры регистров тактируются фронтом, т.е. являются MS-триггерами (см. временную диаграмму, приведенную на рис. 3). Отметим, что регистры на триггерах JK и RS требуют подачи на их входы как самих регистрируемых кодов, так и их инверсных значений, что увеличивает аппаратные затраты на их реализацию.

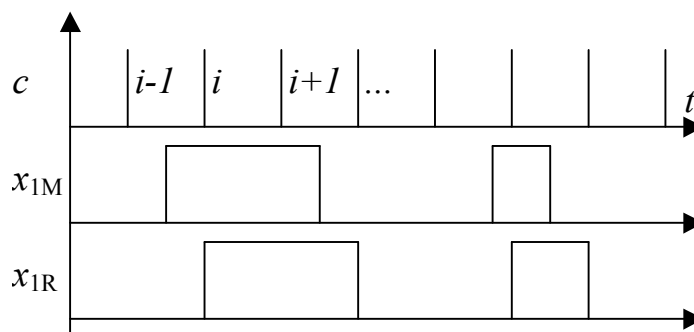


Рис. 3. Процесс смены состояний на выходах регистров в зависимости от состояний входов и сигнала тактирования

При синтезе операционного автомата необходимо:

определить состав комбинационных схем  $L1 - Ln$ , для каждой из комбинационных схем сформировать булево выражение, определяющее логику ее работы;

определить типы регистров для хранения данных;

определить количество направлений, с которых на входы регистров будут поступать данные;

определить разрядность данных каждого из направлений;

сформировать структурную схему операционного автомата;

определить разрядность данных, поступающих на каждый из функциональных блоков;

определить требования к составу и состояниям символов управления  $z$  и осведомительных символов  $u$ .

Результаты синтеза должны быть представлены в виде структурной схемы, таблицы состояний управляющих символов, таблицы состояний осведомительных символов в зависимости от результатов выполнения операций.

### 3.3. Общая методика синтеза управляющего автомата

Управляющий автомат проектируется на основании понятия абстрактного автомата. Существуют следующие схемы абстрактных автоматов.

*Автомат Мили*, или автомат первого рода, приведен на рис. 4. Он описывается следующей системой функций

$$w(t+1) = L_1(u(t), q(t)); z(t) = L_2(u(t), q(t)),$$

где  $u(t)$  - управляющие символы;  $q(t)$  - внутреннее состояние автомата;  $w(t+1)$  - следующее состояние автомата;  $z(t)$  - выходной символ.

В этом абстрактном автомате выдача символа  $z(t)$  происходит сразу, при старом значении внутреннего состояния  $q(t)$ . Поэтому переход в новое состояние отстает по времени на один такт от изменения выходного символа. Это свойство автомата Мили поясняет рис. 4.

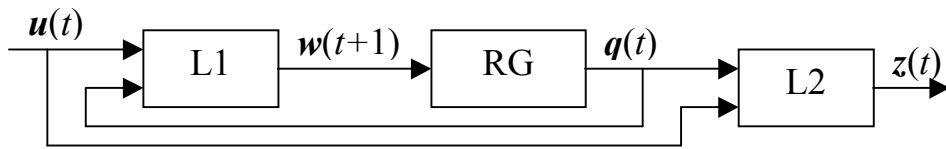


Рис. 4. Автомат Мили

*Автомат Мура*, или автомат второго рода, приведен на рис. 5. Он имеет функцию переходов такую же, как у автомата Мили, а функцию выходов, не зависящую непосредственно от входной переменной  $u(t)$ .

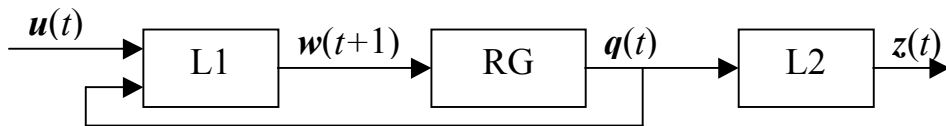


Рис. 5. Автомат Мура

Система функций для автомата Мура имеет вид

$$w(t+1) = L_1(u(t), q(t)); z(t) = L_2(q(t)),$$

Можно сказать, что у автомата Мура сначала меняется внутреннее состояние, и только после этого изменение состояния отражается на состоянии выходного символа. Функция выходов получается сдвинутой на один интервал времени.

При проектировании управляющих блоков на практике чаще применяется более простая модель Мура. Функциональная схема управляющего автомата на базе абстрактного автомата Мура приведена на рис. 6.

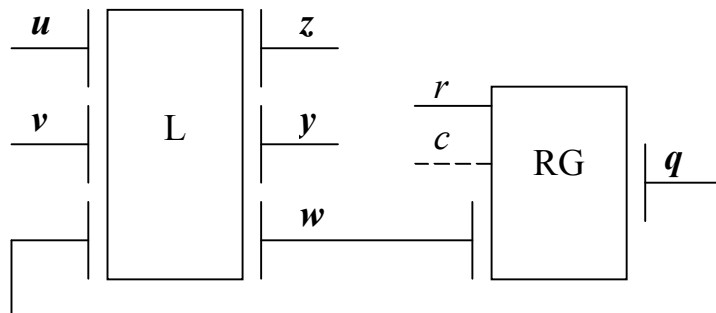


Рис. 6. Функциональная схема управляющего автомата

На рис. 6 приведены следующие обозначения:  
комбинационная схема L;

элемент памяти (регистр) RG;  
вектор  $\mathbf{v}$  внешних управляющих символов;  
вектор  $\mathbf{y}$  символов, несущих информацию о состоянии устройства;  
вектор  $\mathbf{u}$  осведомительных символов, поступающих с операционного автомата;  
вектор управляющих символов  $\mathbf{z}$ , подаваемых на операционный автомат;  
вектор  $\mathbf{q}$  текущих состояний управляющего автомата;  
вектор  $\mathbf{w}$  выходных символов комбинационной схемы (вектор возбуждения), формирующих очередное (следующее) состояние управляющего автомата;  
 $r$  - сигнал сброса, переводящий управляющий автомат в исходное состояние;  
 $s$  - сигнал тактирования управляющего автомата.

Вектор возбуждения  $\mathbf{w}$  представляет собой совокупность структурных переменных, которые должны поступать на входы триггеров регистра RG, чтобы обеспечить их переключение в состояния, определенные алгоритмом функционирования автомата.

Комбинационная схема автомата (элемент L) объединяет все логические элементы, участвующие в формировании вектора управления  $\mathbf{z}$  выходного вектора  $\mathbf{y}$  и вектора возбуждения  $\mathbf{w}$  на основании вектора осведомительных символов  $\mathbf{u}$ , вектора внешнего управления  $\mathbf{v}$  и вектора текущих состояний автомата  $\mathbf{q}$ .

Проектирование управляющего автомата сводится к следующим этапам.

1. Составление алгоритма функционирования устройства управления.
2. По алгоритму функционирования определение количества состояний автомата.
3. Установление связей между состояниями автомата и значениями структурного вектора  $\mathbf{q}$ .
4. Определение количества элементов памяти, необходимое для кодирования состояний автомата; число  $n$  элементов памяти равно количеству компонентов вектора  $\mathbf{q}$  и должно отвечать условию  $n \geq \log_2 m$ , где  $m$  - число состояний заданного абстрактного автомата.

Соответствие между абстрактными обозначениями состояний и их структурными кодами устанавливается с помощью таблицы кодирования состояний. В простейшем случае это соответствие может быть выбрано произвольно.

Если входные и выходные переменные были заданы в абстрактном алфавите, то и для них должны быть решены вопросы кодирования с построением соответствующих таблиц. Практически это означает всего лишь возврат к структурным обозначениям, вытекающим из содержательных схем микропрограмм.

5. Выбор модели автомата, для формирования его структуры (определено заданием на курсовую работу).

6. Выбор базиса для реализации элементов памяти (триггеров регистра RG) и комбинационной схемы L.

7. Определение булевых функций для компонентов вектора  $\mathbf{w}$  и приведение их к заданному базису.

Таким образом, каноническая методика сводит задачу синтеза автомата к синтезу логической схемы, формирующей систему функций возбуждения.

### 3.4. Составление алгоритма функционирования управляющего устройства

Алгоритм функционирования в курсовой работе должен быть представлен в виде блок-схемы. Алгоритм функционирования управляющего устройства, в основном, описан в задании на курсовую работу. Для синтеза управляющего автомата конкретный смысл процессов, следовательно, и конкретное содержание операторов алгоритма, не имеет значения. Поэтому содержательное описание операторов в алгоритме заменяется абстрактными обозначениями управляющих символов  $z$  (см. рис. 7). Следует отметить, что в алгоритме должен иметься оператор начала и оператор окончания. Перед попаданием алгоритма в оператор окончания на его выходе должен формироваться вектор признаков  $\mathbf{y}$  выполнения алгоритма.

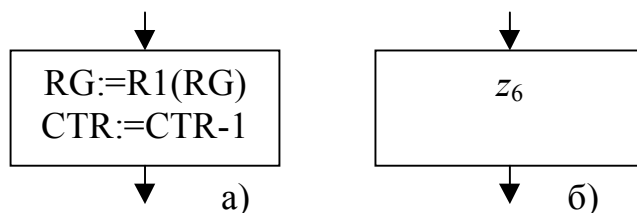


Рис. 7. Замена содержательного описания а) операторов алгоритма абстрактным обозначением б)

Вектор  $\mathbf{z}$  представляется в виде его отдельных составляющих:  $\mathbf{z} = (z_1, \dots, z_i, \dots, z_n)$ . Каждый такой символ  $z_i$  обозначает одну определенную *микрокоманду*, содержащую один или несколько управляющих символов вектора  $\mathbf{z}$ , и задает определенную совокупность одновременно выполняемых *микроопераций*.

Входные структурные переменные вектора  $\mathbf{u}$ ,  $u_1, u_2, \dots$  будем называть логическими условиями (ЛУ). Переход от содержательных обозначений логических условий (например, на языке функционального микропрограммирования) к структурным переменным поясняет рис. 8.

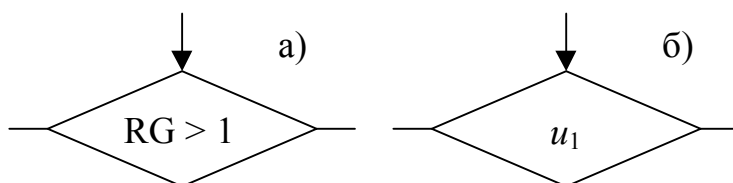


Рис. 8. Замена содержательного обозначения а)

управляющего оператора структурным обозначением б)

Содержательный смысл операторов и логические условия их выполнения устанавливается на подготовительном этапе синтеза на основе выданного задания. Смысл обозначений закрепляется таблицами. Пример результата составления блок-схемы алгоритма приведен на рис. 9.

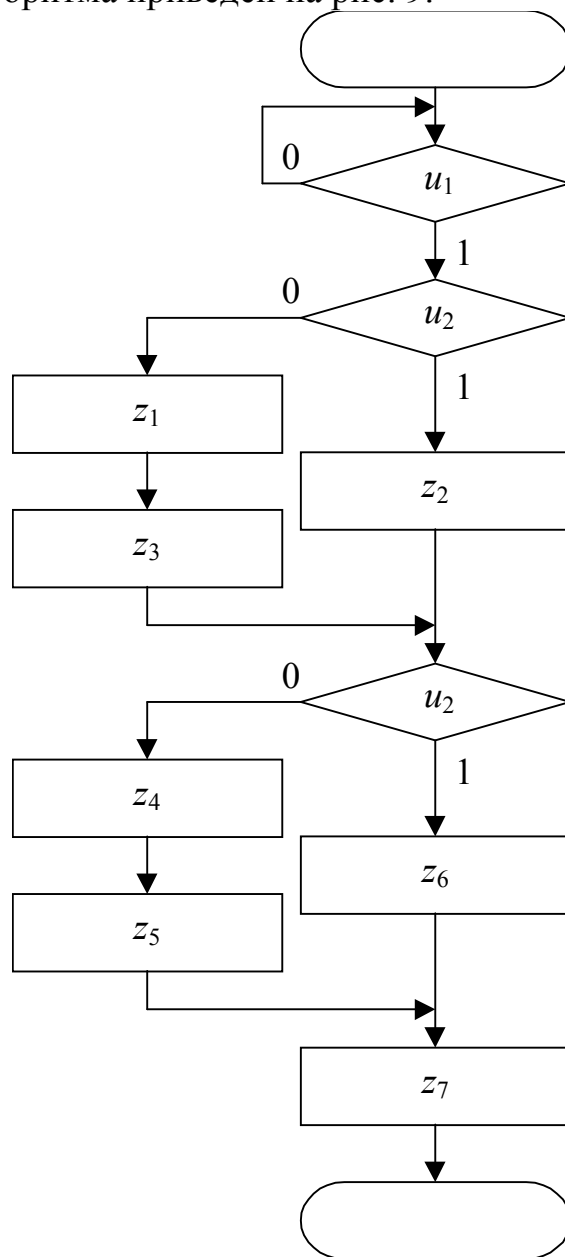


Рис. 9. Блок-схема алгоритма

Следует отметить, что:

1. Управляющий оператор  $u_1$  является ждущим оператором. Алгоритм запускается, когда  $u_1 = 1$ .

2. Оператор  $z_i$ , может означать управление параллельным выполнением нескольких микроопераций, если состав и структура операционного автомата позволяют это сделать. Указанные микрооперации выполняются за один такт,

т.е. при одном состоянии управляющего автомата. Операторы  $z_i, z_j$ , расположенные в последовательно расположенных (сопряженных непосредственно, или через управляющие операторы) вершинах блок-схемы, предполагают выполнение микроопераций последовательно, сначала подмножество микроопераций, заданных оператором  $z_i$ , а затем - подмножество микроопераций, заданных оператором  $z_j$ .

3. Если сопряженные операторы  $z_i, \dots, z_j$  содержат одинаковые микрооперации, то указанные микрооперации должны выполняться столько раз, сколько раз они встречаются в кортеже  $(z_i, \dots, z_j)$ .

4. Одна и та же микрооперация не должна входить в один оператор дважды, при этом одинаковые микрооперации, выполненные одновременно различными компонентами операционного автомата, считаются различными микрооперациями.

В соответствии с блок-схемой алгоритма составляется матричная схема алгоритма, которая представляет квадратную таблицу с горизонтальными входами от  $z_0$  до  $z_n$  со старшим ( $n$ ) из имеющихся порядковых индексов и вертикальными - от  $z_1$  до  $z_{n+1}$ . В клетки таблицы вписываются логические условия соответствующего перехода. Матричная схема алгоритма, приведенного на рис. 9, представлена в табл. 1.

Табл. 1

Матричная схема алгоритма								
	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$z_7$	$z_8$
$z_0$	$u_1 \bar{u}_2 \bar{u}_2$	$u_1 u_2$						
$z_1$			1					
$z_2$				$\bar{u}_2$		$u_2$		
$z_3$				$\bar{u}_2$		$u_2$		
$z_4$					1			
$z_5$							1	
$z_6$							1	
$z_7$								1

Следует отметить также одно важное свойство матричных схем алгоритмов: дизъюнкция содержимого всех клеток одной строки равна единице. Исключением является равенство дизъюнкции булевой переменной ждущего управляющего оператора (как это получилось в верхней строке табл. 1). Логическим условием безусловного перехода является константа, равная единице. Если по блок-схеме прослеживается несколько параллельных путей от оператора  $z_i$  к оператору  $z_j$ , то соответствующие логические условия должны записываться в клетки матричной схемы алгоритма через знак дизъюнкции.

На основании матричной схемы алгоритма составляются формулы перехода:

$$z_0 u_1 \bar{u}_2 \rightarrow z_1; z_0 u_1 u_2 \rightarrow z_2; z_1 \wedge 1 = z_1 \rightarrow z_3; (z_2 \vee z_3) \bar{u}_2 \rightarrow z_4;$$

$$z_4 \wedge 1 = z_4 \rightarrow z_5; (z_2 \vee z_3)u_2 \rightarrow z_6; (z_5 \vee z_6) \wedge 1 = z_5 \vee z_6 \rightarrow z_7; z_7 \wedge 1 = z_7 \rightarrow z_8.$$

Формулы отражают логические условия перехода в состояние, указанное после стрелки из состояний, указанного до стрелки. Если для перехода требуется выполнение нескольких условий, они должны включаться в формулу через конъюнкцию. Если в состояние, указанное после стрелки можно попасть из нескольких состояний, то они должны включаться в формулу через дизъюнкцию.

### 3.5. Кодирование состояний управляющего автомата

В соответствии с формулами перехода должен быть построен граф автомата Мура. Для алгоритма рис. 9 граф имеет вид, приведенный на рис. 10.

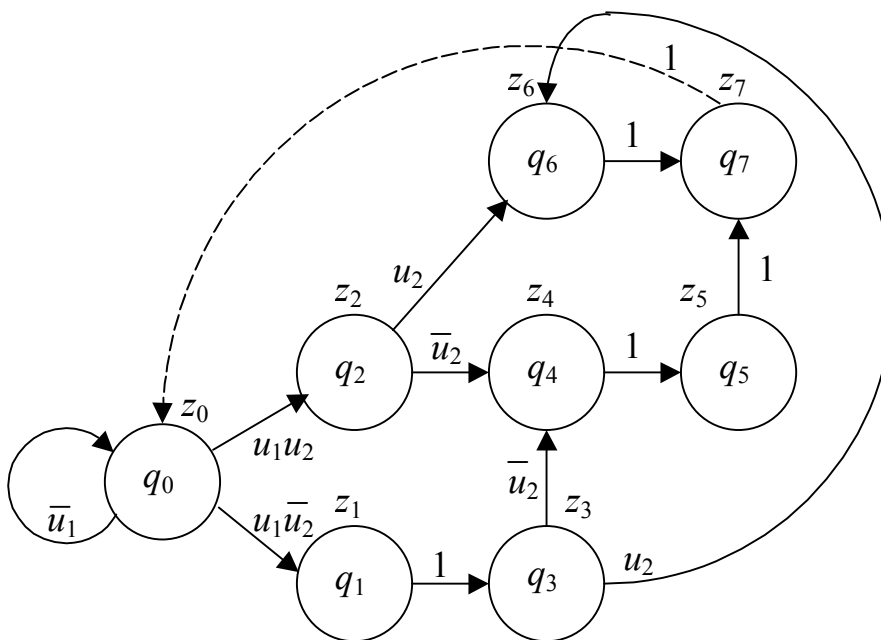


Рис. 10. Граф состояний автомата Мура

Следует различать внутренние состояния автомата Мура (на графе состояний они обозначены через  $q_i$ ) и внешние управляющие символы ( $z_j$ ).

В принципе, не существует правила, согласно которому определяют порядок следования кодов символов  $q_1, \dots, q_n$ . Не существует также правил кодирования векторов  $u, v, w, z, y$ . При решении практических задач коды  $v, y$  могут оказаться заданными в технических требованиях на проектирования цифрового управляющего устройства, а коды  $u, z$  получаться естественным образом на выходах соответствующих блоков управляющего и операционного автомата (например, на выходе переполнения сумматора), или требоваться для управления соответствующим логическим устройством (например, при управлении выбора направления мультиплексора, или направления сдвига данных в регистре). Точно также и связь между кодами  $q$  и  $w$  для регистров, состоящих из разных типов триггеров, различна. Поэтому коды, определяемые в процессе проектирования должны быть такими, чтобы впоследствии можно было мини-

минимизировать комбинационную схему L, (см. рис. 6) преобразующую символы  $q, u, v$  в символы  $w, z, y$ .

В соответствии с выбранной системой кодирования одним из способов задаются булевы формулы, определяющие булевы функции переменных, составляющих символы  $w, z, y$  в зависимости от переменных, составляющих символы  $q, u, v$ . Булевы функции могут быть заданы одним из следующих способов: таблицей истинности; булевыми формулами, картой Карно (диаграммой Вейча).

При описании функций с помощью булевых формул рекомендуется использовать нормальную конъюнктивную (состоящую из элементарных дизъюнкций, соединенных конъюнкциями) или нормальную дизъюнктивную (состоящую из элементарных конъюнкций, соединенных дизъюнкциями) форму представления. Указанные формулы могут быть минимизированы с использованием следующих правил булевой алгебры.

1) Ассоциативность дизъюнкции и конъюнкции

$$a \vee (b \vee c) = a \vee b \vee c = (a \vee b) \vee c; a \wedge (b \wedge c) = a \wedge b \wedge c = (a \wedge b) \wedge c.$$

2) Коммутативность дизъюнкции и конъюнкции

$$a \vee b = b \vee a; a \wedge b = b \wedge a.$$

3) Существование нейтральных элементов для дизъюнкции и конъюнкции

$$a \vee 0 = 0 \vee a = a; a \wedge 1 = 1 \wedge a = a.$$

4) Дистрибутивность конъюнкции относительно дизъюнкции и дизъюнкции относительно конъюнкции

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c); a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

5) Закон исключенного третьего

$$a \vee \bar{a} = 1.$$

6) Закон противоречия

$$a \wedge \bar{a} = 0.$$

7) Замкнутость множества  $\{0, 1\}$  относительно инверсии

$$\bar{0} = 1; \bar{1} = 0.$$

8) Конъюнкция любой переменной с нулем равна нулю

$$0x = 0.$$

9) Идемпотентность дизъюнкции и конъюнкции

$$a \wedge a = a; a \vee a = a.$$

10) Правило поглощения

$$a \vee (a \wedge b) = a; a \wedge (a \vee b) = a.$$

11) Правило склеивания

$$(a \wedge b) \vee (a \wedge \bar{b}) = a; (a \vee b) \wedge (a \vee \bar{b}) = a.$$

12) Правило де-Моргана

$$\overline{a \wedge b} = \bar{a} \vee \bar{b}; \overline{a \vee b} = \bar{a} \wedge \bar{b}$$

В результате минимизации нормальная дизъюнктивная или нормальная конъюнктивная форма булевых выражений должна сохраниться, т.к. указанные



формы наиболее просто сводятся, соответственно к штриху Шеффера и стрелке Пирса.

По указанным выражениям должна быть построена структурная схема управляющего автомата, которая должна быть приведена в пояснительной записке к курсовой работе.

#### 4. ОЦЕНКА КУРСОВОЙ РАБОТЫ

Защита курсовой работы проводится по одноступенчатой схеме, при этом общая оценка 100 баллов складывается из следующих оценок:

качество исполнения курсовой работы – до 35 баллов;

оценка рецензента – до 5 баллов;

качество доклада – до 20 баллов;

защита работы (ответы на вопросы) – до 40 баллов.

#### 5. ЛИТЕРАТУРА

1. Ангер С. Асинхронные последовательностные схемы. - М.: Наука, 1977. - 400 с.
2. Баранов С.И. Синтез микропрограммных автоматов. - Л.: Энергия, 1979. - 231 с.
3. Голдсуорт В. Проектирование цифровых логических устройств. - М.: Машиностроение, 1985. - 288 с.
4. Закревский А.Д. Алгоритмы синтеза дискретных автоматов. - М.: Наука, 1971. - 511 с.
5. Зельдин Е.А. Триггеры. - М.: Энергоатом., 1983. - 95 с.
6. Карцев М.А., Брик В.А. Вычислительные машины и синхронная арифметика. - М.: Радио и связь, 1981. - 238 с.
7. Коршунов Ю.М. Математические основы кибернетики. -М.: Энергия, 1980. - 423 с.
8. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. - М.: Энергия, 1980. - 342 с.
9. Лазарев В.Т., Пийль Е.И. Синтез управляющих автоматов. -М.: Энергоатом., 1988. - 328 с.
10. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. - Минск: Высшая школа, 1980. - 335 с.
11. Мендельсон Э. Введение в математическую логику. -М.: Наука, 1976. -320 с.
12. Потемкин И.С. Функциональные узлы цифровой автоматики. -М.: Энергоатом., 1988. - 380 с.
13. Савельев А.Я. Прикладная теория цифровых автоматов. -М.: Высшая школа, 1987. - 272 с.

14. Филиппов А.Г., Белкин О.С. Проектирование логических узлов ЭВМ. - М.: Советское радио, 1974. - 342 с.
15. Фридман А., Менон П. Теория и проектирование переключательных схем. - М.: Мир, 1978. - 580 с.
16. Фудзисава Т., Касами Т. Математика для радиоинженеров. Теория дискретных структур. - М.: Радио и связь, 1984. - 240 с.

ПРИЛОЖЕНИЕ 1  
ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВУЮ РАБОТУ

Табл. П.1

Логика работы устройства

№ Вар.	Обозначение переменных	Физический смысл	Логика работы устройства
1	2	3	4
1, 15	$I_0, I_1, I_2, I_3$	Натуральный двоичный код ( $I_0$ – младший разряд)	<p>Коды <math>I_0, I_1, I_2, I_3</math> поступают на входы параллельно-последовательно в сопровождении сигналов <math>C</math> и <math>V</math>, всего поступает 8 кодов.</p> <p>В случае, если <math>V = 0</math>, производится суммирование кода, хранящегося в устройстве, со входным кодом. В случае, если <math>V = 1</math>, производится вычитание из кода, хранящегося в устройстве, входного кода.</p> <p>Если код, сформированный устройством после поступления 8 кодов, положительный, то <math>D = 0</math>, если отрицательный – то <math>D = 1</math>.</p>
	$V$	Управляющий вход	
	$C$	Вход тактирования	
	$D$	Выход устройства	
2, 16	$I_{00}, I_{01}, I_{02}, I_{03}$	Дополнительный двоичный код $I_0$ ( $I_{03}$ – знаковый разряд)	<p>Коды (<math>I_{00}, I_{01}, I_{02}, I_{03}</math>) и (<math>I_{10}, I_{11}, I_{12}, I_{13}</math>) поступают на входы последовательно-параллельно (побитно на параллельные входы) в сопровождении сигналов <math>C</math> и <math>V</math>.</p> <p>В случае, если <math>V = 0</math>, производится суммирование кодов. В случае, если <math>V = 1</math>, суммирования не происходит.</p> <p>Если после суммирования сформирован хотя бы один отрицательный результат, то <math>D = 1</math> и устройство прекращает работу.</p>
	$I_{10}, I_{11}, I_{12}, I_{13}$	Дополнительный двоичный код $I_1$ ( $I_{13}$ – знаковый разряд)	
	$V$	Управляющий вход	
	$C$	Вход тактирования	
	$D$	Выход устройства	
3, 17	$I$	Дополнительный код	<p>Дополнительный код <math>I = (I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7)</math>, который может быть как положительным, так и отрицательным, поступает на вход последовательно, разряд за разрядом, начиная с младшего, в сопровождении сиг-</p>
	$C$	Тактирование	
	$D_0, D_1, D_2, D_3, D_4, D_5$	Информационные выходы устройства	

	$Y_1, Y_2$	Управляющие выходы	<p>нала С.</p> <p>Код суммируется с кодом, хранящимся в устройстве.</p> <p>Результат суммирования в параллельном коде выдается на выходы <math>D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7</math>. При переполнении сумматора устройство останавливается.</p>
4, 18	$I_{00}, I_{01}, I_{02}, I_{03}$	Натуральный двоичный код $I_0$ ( $I_{00}$ – младший разряд)	<p>Производится установка <math>D = 0</math>.</p> <p>Коды (<math>I_{00}, I_{01}, I_{02}, I_{03}</math>) и (<math>I_{10}, I_{11}, I_{12}, I_{13}</math>) поступают на входы параллельно-последовательно начиная с младшего разряда в сопровождении сигналов С (тактирует поступление битов) и V.</p> <p>В случае, если <math>V = 0</math>, производится суммирование кодов. В случае, если <math>V = 1</math>, суммирование блокируется.</p> <p>Если после суммирования происходит переполнение, то <math>D = 1</math>, и устройство прекращает работу.</p>
	$I_{10}, I_{11}, I_{12}, I_{13}$	Натуральный двоичный код $I_1$ ( $I_{10}$ – младший разряд)	
	C	Тактирование	
	V	Управление суммированием	
	D	Код результата	
5, 19	$I_{00}, I_{01}, I_{02}, I_{03}$	Натуральный двоичный код $I_0$ ( $I_{00}$ – младший разряд)	<p>Производится начальная установка <math>D = 0, Y = 0</math>.</p> <p>Коды (<math>I_{00}, I_{01}, I_{02}, I_{03}</math>) и (<math>I_{10}, I_{11}, I_{12}, I_{13}</math>) поступают на единственный вход последовательно, начиная со старшего разряда, сначала код <math>I_0</math>, затем код <math>I_1</math> в сопровождении сигнала С, тактирующего поступление битов.</p> <p>Устройство выполняет одно суммирование, после чего выдает на выходы параллельный код <math>D_0, D_1, D_2, D_3</math> и <math>Y = 1</math>. Если после суммирования происходит переполнение, то <math>D_0, D_1, D_2, D_3 = 0000, Y = 0</math>.</p>
	$I_{10}, I_{11}, I_{12}, I_{13}$	Натуральный двоичный код $I_1$ ( $I_{10}$ – младший разряд)	
	C	Тактирование	
	$D_0, D_1, D_2, D_3$	Выходной код	
	Y	Индикатор переполнения	
6, 20	$I_0, I_1, I_2, I_3$	Натуральный двоичный код $I_0$ ( $I_0$ – младший разряд)	<p>Производится установка <math>D = 0, Y = 0</math>.</p> <p>Коды (<math>I_0, I_1, I_2, I_3</math>) поступают на входы последовательно, начиная со старшего разряда, в сопровождении сигнала С, тактирующего поступле-</p>
	C	Тактирование	
	$D_0, D_1, D_2, D_3$	Выходной код	

	Y	Индикатор переполнения	<p>ние битов.</p> <p>Производится суммирование содержимого устройства с кодом, имеющим четный номер и вычитание из содержимого кода, имеющего нечетный номер.</p> <p>Результат выдается на выходы <math>D_0, D_1, D_2, D_3</math> параллельно после поступления на входы 8-ми последовательностей.</p> <p>Если после суммирования происходит хотя бы одно переполнение, положительное или отрицательное, то <math>D_0, D_1, D_2, D_3 = 0000, Y = 1</math>.</p>
7, 21	$I_0, I_1, I_2, I_3$	Натуральный двоичный код ( $I_0$ – младший разряд)	<p>Производится установка <math>D = 0, Y = 0</math>.</p> <p>Коды <math>I_0, I_1, I_2, I_3</math> поступают на входы параллельно-последовательно в сопровождении сигнала С.</p> <p>Каждый последующий код сравнивается с предыдущим. Если результат сравнения больше нуля, то к выходному коду D инкрементируется единица, если меньше, то инкрементирования не происходит.</p> <p>После поступления 16 кодов результат инкрементирования выдается на выходы D, при этом <math>Y = 1</math>.</p>
	C	Тактирование	
	D	Выход устройства	
	Y	Управляющий выход	
8, 22	$I_0, I_1, I_2, I_3$	Натуральный двоичный код ( $I_0$ – младший разряд)	<p>Производится установка <math>D = 0, Y = 0</math>.</p> <p>Коды <math>I_0, I_1, I_2, I_3</math> поступают на вход последовательно в сопровождении сигнала С.</p> <p>Каждый последующий код сравнивается с предыдущим. Если результат сравнения больше нуля, то к выходному коду D инкрементируется единица, если меньше, то инкрементирования не происходит.</p> <p>После поступления 4 кодов результат инкрементирования выдается на выходы D, при этом <math>Y = 1</math>.</p>
	C	Тактирование	
	D	Выход устройства	
	Y	Управляющий выход	

9, 23	$I_0, I_1, I_2, I_3$	Натуральный двоичный код ( $I_0$ – младший разряд)	<p>Коды <math>I_0, I_1, I_2, I_3</math> поступают на вход параллельно-последовательно в сопровождении сигналов <math>C</math> и <math>V</math>.</p> <p>В случае, если <math>V = 0</math>, производится вычитание из кода с четным номером кода с нечетным номером. В случае, если <math>V = 1</math>, производится вычитание из кода с нечетным номером кода с четным номером.</p> <p>Количество отрицательных результатов подсчитывается. На выходы <math>D</math> выводится значение кода после поступления 16 кодов.</p>
	$V$	Управляющий вход	
	$C$	Тактирование	
	$D$	Выход устройства	
10, 24	$I_0, I_1, I_2, I_3$	Натуральный двоичный код ( $I_0$ – младший разряд)	<p>Код <math>I_0, I_1, I_2, I_3</math> поступает на единственный вход последовательно в сопровождении сигналов <math>C</math>, и <math>V</math>.</p> <p>В случае, если <math>V = 0</math>, производится вычитание из кода с нечетным номером кода с четным номером. В случае, если <math>V = 1</math>, производится вычитание из с четным номером кода с нечетным номером.</p> <p>Количество результатов больших, или равных нулю подсчитывается. На выходы <math>D</math> выводится значение после поступления 12 кодов.</p>
	$V$	Управляющий вход	
	$C$	Тактирование	
	$D$	Выход устройства	
11, 25	$I_{00}, I_{01}, I_{02}, I_{03}$	Дополнительный двоичный код $I_0$ ( $I_{03}$ – знаковый разряд)	<p>Производится установка <math>D = 0, V = 0</math>.</p> <p>Коды (<math>I_{00}, I_{01}, I_{02}, I_{03}</math>) и (<math>I_{10}, I_{11}, I_{12}, I_{13}</math>) поступают на два входа последовательно-параллельно, начиная с младших разрядов. Производится их суммирование. Результаты выдаются на выходы <math>D</math> в параллельном коде.</p> <p>Если после суммирования сформировано хотя бы одно отрицательное переполнение, то <math>D = 0000, V = 1</math> и устройство прекращает работу.</p>
	$I_{10}, I_{11}, I_{12}, I_{13}$	Дополнительный двоичный код $I_1$ ( $I_{13}$ – знаковый разряд)	
	$C$	Тактирование	
	$D$	Выход устройства	
	$V$	Управляющий выход	
12, 26	$I_{00}, I_{01}, I_{02}, I_{03}$	Дополнительный двоичный код $I_0$ ( $I_{03}$ – знаковый разряд)	<p>По входу <math>R</math> производится начальная установка схемы, при этом <math>D = 0, V = 0</math>.</p> <p>Коды (<math>I_{00}, I_{01}, I_{02}, I_{03}</math>) и (<math>I_{10}, I_{11}, I_{12},</math></p>

	$I_{10}, I_{11}, I_{12}, I_{13}$	Дополнительный двоичный код $I_1$ ( $I_{13}$ – знаковый разряд)	$I_{13}$ ) поступают на входы последовательно-параллельно, начиная с младших разрядов. Производится их суммирование. Результаты выдаются на выходы $D$ в параллельном коде.
	$C$	Тактирование	Если после суммирования сформировано хотя бы одно положительное переполнение, то $D = 0000$ , $V = 1$ и устройство прекращает работу.
	$R$	Сброс	
	$D$	Выход устройства	
	$V$	Управляющий выход	
13, 27	$I_{00}, I_{01}, I_{02}, I_{03}$	Натуральный двоичный код $I_0$ ( $I_{00}$ – младший разряд)	Производится установка $D = 0$ , $Y_1 = 0$ , $Y_2 = 0$ . Коды ( $I_{00}, I_{01}, I_{02}, I_{03}$ ) и ( $I_{10}, I_{11}, I_{12}, I_{13}$ ) поступают на входы последовательно-параллельно, начиная с младшего разряда, в сопровождении сигнала $C$ , тактирующего поступление битов. Устройство выполняет 16 суммирований, каждый раз выдавая на выходы параллельный код $D_0, D_1, D_2, D_3$ и $Y_1 = 1$ , который переключается в ноль после поступления очередного такта. Если в результате суммирования происходит переполнение, то $D_0, D_1, D_2, D_3 = 0000$ , $Y_2 = 1$ и устройство прекращает работу
	$I_{10}, I_{11}, I_{12}, I_{13}$	Натуральный двоичный код $I_1$ ( $I_{10}$ – младший разряд)	
	$C$	Тактирование	
	$R$	Сброс	
	$D_0, D_1, D_2, D_3$	Выходной код	
	$Y$	Индикатор переполнения	
14, 28	$I_0, I_1, I_2, I_3$	Натуральный двоичный код $I_0$ ( $I_0$ – младший разряд)	Производится установка $D = 0$ , $Y = 0$ . Коды ( $I_0, I_1, I_2, I_3$ ) поступают на вход последовательно, начиная со старшего разряда, в сопровождении сиг-
	$C$	Тактирование	
	$D_0, D_1, D_2, D_3$	Выходной код	

	Y	Индикатор переполнения	<p>нала С, тактирующего поступление битов.</p> <p>Производится суммирование содержимого устройства с кодом, имеющим четный номер, и вычитание из содержимого кода, имеющего нечетный номер.</p> <p>Результат выдается на выходы <math>D_0</math>, <math>D_1</math>, <math>D_2</math>, <math>D_3</math> параллельно после поступления на входы 8-ми последовательностей.</p> <p>Если после суммирования происходит хотя бы одно переполнение, положительное или отрицательное, то <math>D_0, D_1, D_2, D_3 = 0000</math>, <math>Y = 1</math> и устройство прекращает работу</p>
--	---	------------------------	--

Табл. П.2

Типы логических устройств для реализации элементов памяти и комбинационных схем.

№ в.	Базис комбинационных схем	Базис элементов памяти	№ варианта	Базис комбинационных схем	Базис элементов памяти
1, 7, 13, 19	Штрих Шеффера	RSC-триггер	4, 10, 16, 22	Стрелка Пирса	JKC-триггер
2, 8, 14, 20	Штрих Шеффера	DC-триггер	5, 11, 17, 23	Стрелка Пирса	RSC-триггер
3, 9, 15, 21	Штрих Шеффера	JKC-триггер	6, 12, 18, 24	Стрелка Пирса	DC-триггер



**ПРИЛОЖЕНИЕ 2**  
**ТИТУЛЬНЫЙ ЛИСТ К ПОЯСНИТЕЛЬНОЙ ЗАПИСКЕ**  
**КУРСОВОЙ РАБОТЫ ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**  
**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное учреждение высшего образования  
«Тульский государственный университет»  
Кафедра «Робототехника и автоматизация производства»

**Цифровое управление технологическими машинами и оборудованием**

Пояснительная записка к курсовой работе  
для студентов очной формы обучения  
направления 15.03.02 Технологические машины и оборудование

Выполнил студент гр. \_\_\_\_\_  
Принял \_\_\_\_\_  
Должность

И.О. Фамилия  
И.О. Фамилия

Тула: 2022