


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

Заведующий кафедрой

 О.А. Ерзин

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ  
РАБОТЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)  
«Основы дискретной математики»**

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

по направлению подготовки  
**09.03.02 Информационные системы и технологии**

с профилем  
**Информационные системы и технологии в робототехники**

Форма обучения: очная

Идентификационный номер образовательной программы: 090302-02-22

Тула 2023 год

**Разработчик:**

Ларкин Евгений Васильевич, профессор,  
доктор тех. наук, профессор

(ФИО, должность, ученая степень, ученое звание)



---

(подпись)

## 1. ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Курсовая работа по дисциплине «Основы дискретной математики» 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 проектирования курсовой работы

Объектом проектирования 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.

### 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, реализую-

щего 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 заданному базису комбинационных схем и элементов памяти;

разработать структурную схему управляющего автомата с заданным базисом.

## 2.4. Требования к пояснительной записке

Результаты синтеза отражаются в пояснительной записке объемом до 10 страниц машинописного текста. 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 набираются во встроенном редакторе MICROSOFT EQUATION. Рисунки должны быть набраны в любом графическом редакторе и вставлены в текст.

## 2.5. Защита курсовой работы

Курсовая работа защищается перед преподавателем, ведущим предмет «Основы дискретной математики». 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 состоит из оценки качества выполнения работы в соответствии с выданным заданием и от знаний, обнаруженных в процессе защиты. Результаты защиты курсовой работы могут быть учтены при оценке итоговых знаний по предмету (при сдаче экзамена по лекционному курсу).

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

пояснительная записка к курсовой работе не представлена на защиту;  
В пояснительной записке отсутствует решение поставленной задачи;  
на защиту представлена пояснительная записка с чужим вариантом курсовой работы;

студент 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 устройства приведена на рис.

1.

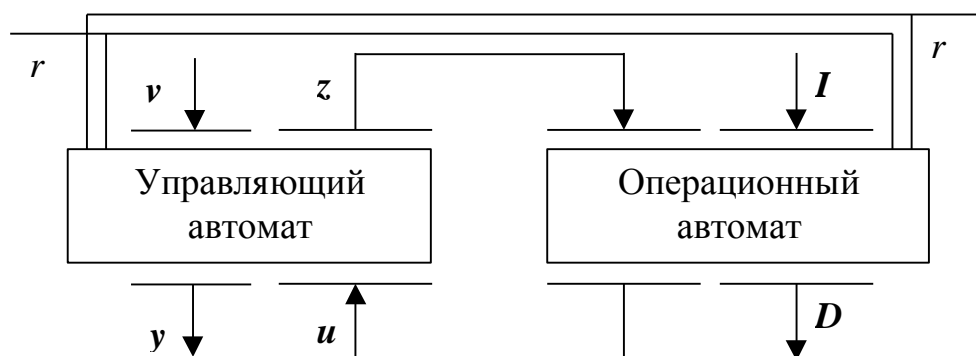


Рис. 1. Общая функциональная схема цифрового устройства

В состав любого цифрового устройства 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  $y$  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 операционного 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  $MX_0, MX_1$ ;
- регистры данных  $RG_0, RG_1$ ;
- комбинационные схемы выполнения математических операций  $L_0 - L_{n-1}$ ;
- комбинационная схема формирования признака результата  $L_n$ .

На входы операционного автомата подаются данные  $I_0, I_1$  и управляющее воздействие  $z$ , которое разделяется на управляющее воздействие мультиплексоров ( $z_{0M}, z_{1M}$ ), управляющее воздействие регистров, ( $z_{0R}, z_{1R}$ ), управляющее воздействие комбинационных схем выполнения математических операций ( $z_{0L} - z_{nL}$ ). Отдельно подается синхронизирующая последовательность импульсов  $s$ , которая может быть общей с синхронизирующей последовательностью управляющего автомата.

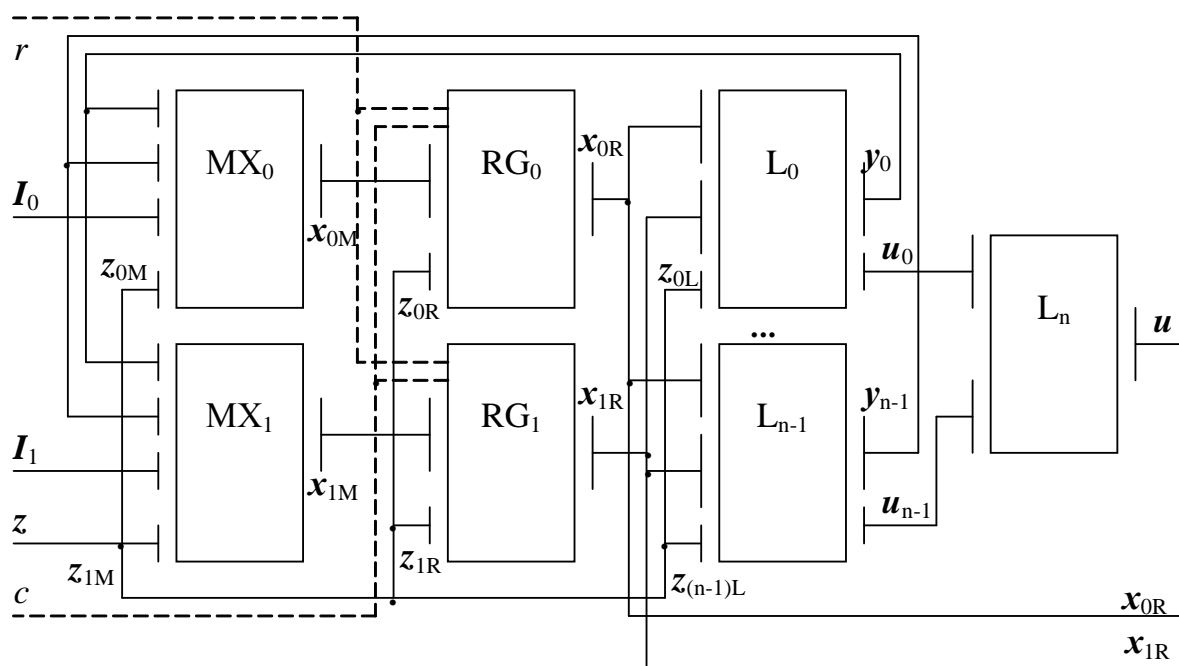


Рис. 2. Типовая схема операционного автомата

По входу  $r$  может производиться установка регистров в начальное состояние, например во все разряды регистров может записываться логический ноль.

На выходах  $x_{0R}$ ,  $x_{1R}$ , автомата формируется результат. На выходах  $u$  формируется признак результата.

Операционный автомат функционирует следующим образом.

Мультиплексоры  $MX_0$  и  $MX_1$  из множества данных, поданных на их информационные входы, выбирают в соответствии с управляющими воздействиями  $z_{0M}$  и  $z_{1M}$  данные, необходимые в текущий момент для выполнения заданной операции. При этом для обработки могут быть выбраны данные  $I_0$ ,  $I_1$  или  $x_{0R}$  или  $x_{1R}$ . Код, выбранный для выполнения операции, синхросигналом  $c$  записывается в регистры  $RG_0$ ,  $RG_1$ , в результате чего на выходах указанных регистров формируются коды  $x_{1R}$  и  $x_{2R}$ , поступающие на обработку в комбинационные схемы  $L_0 - L_{n-1}$ . В момент поступления кодов на входы комбинационных схем  $L_0 - L_{n-1}$  на их выходах формируются результаты  $y_0 - y_{n-1}$ , а также признаки результатов  $u_0 - u_{n-1}$ . Из признаков  $u_0 - u_{n-1}$  на выходе  $u$  комбинационной схемы  $L_u$  формируется общий признак результата.

В качестве комбинационных схем  $L_0 - L_{n-1}$  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.

Обычно комбинационные схемы интегрируются в единое арифметико-

логическое устройство (АЛУ).

Состав комбинационных схем  $L_0 - L_{n-1}$  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  $L_0 - L_{n-1}$  содержат структуры, предназначенные для обработки одного разряда данных, в параллельном операционном автомате - однотипные структуры для обработки сразу нескольких разрядов.

Регистры  $RG_0, RG_1$  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/сдвиговыми (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.

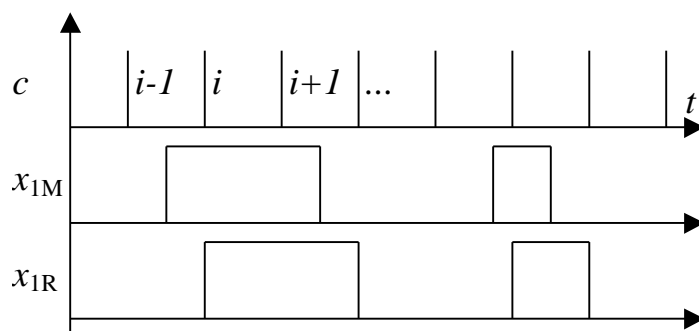


Рис. 3. Процесс смены состояний на выходах регистров в зависимости от состояний входов и сигнала тактирования

При синтезе операционного автомата необходимо:

определить состав комбинационных схем  $L_0 - L_{n-1}$ , 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 управляющего воздействия  $z$  и признаков результата  $u$ .

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. 5. 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;  $w(t+1)$  - вектор возбуждения, формирующий следующее состояние автомата;  $z(t)$  - выходной символ.

На входы  $c$  и  $r$  автомата Мили подается тактирующий сигнал  $c$ , формирующий машинное дискретное время  $t$  и сигнал  $r$  установки автомата в исходное состояние.

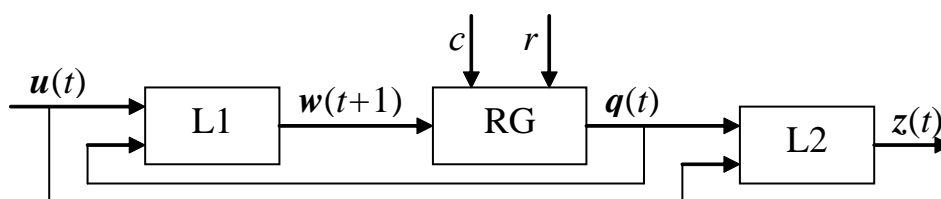


Рис. 4. Автомат Мили

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 машинного времени от изменения выходного символа.

*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  $u(t)$ .

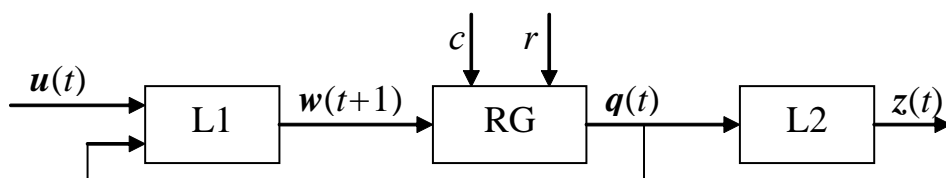


Рис. 5. Автомат Мура

Система функций для автомата Мура имеет вид

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

На входы  $c$  и  $r$  автомата Мура подается тактирующий сигнал  $c$ , формирующий машинное дискретное время  $t$  и сигнал  $r$  установки автомата в исходное состояние.

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.

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 вектора внутренних состояний автомата  $q(t)$ .
4. Определение количества элементов памяти, необходимое для кодирования состояний автомата  $q(t)$ ; число  $n$  элементов памяти равно количеству компонентов вектора  $q$  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  $w$  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, 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. 9). 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  $u$  результата выполнения алгоритма.

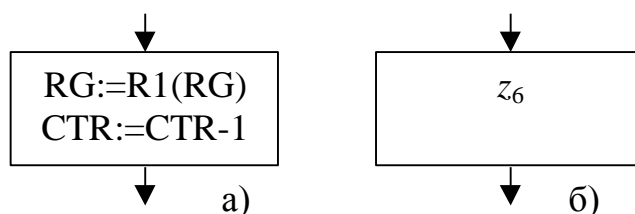


Рис. 9. Замена содержательного описания а) операторов алгоритма абстрактным обозначением б)

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

Входные структурные переменные вектора  $u$ ,  $u_1$ ,  $u_2$ , ... будем называть логическими условиями (ЛУ). Переход от содержательных обозначений логических условий (например, на языке функционального микропрограммирования) к структурным переменным поясняет рис. 10.

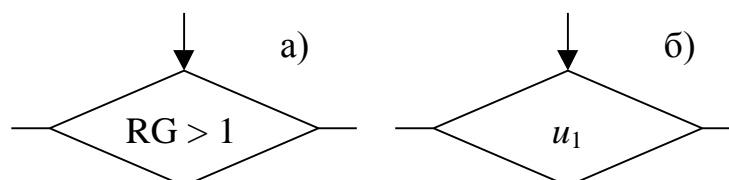


Рис. 10. Замена содержательного обозначения а) управляющего оператора структурным обозначением б)

Содержательный смысл операторов и логические условия их выполнения устанавливается на подготовительном этапе синтеза на основе выданно-

го задания. Смысл обозначений закрепляется таблицами. Пример результата составления блок-схемы алгоритма приведен на рис. 11.

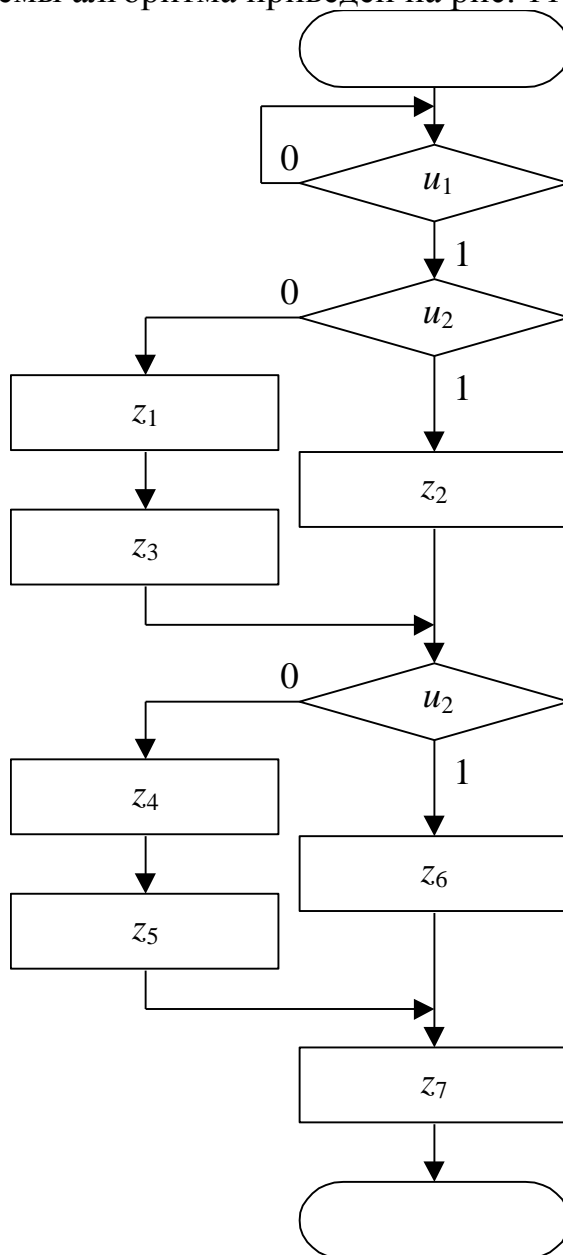


Рис. 11. Блок-схема алгоритма

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. 11, 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 фор-

мулу 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. 11 граф имеет вид, приведенный на рис. 12.

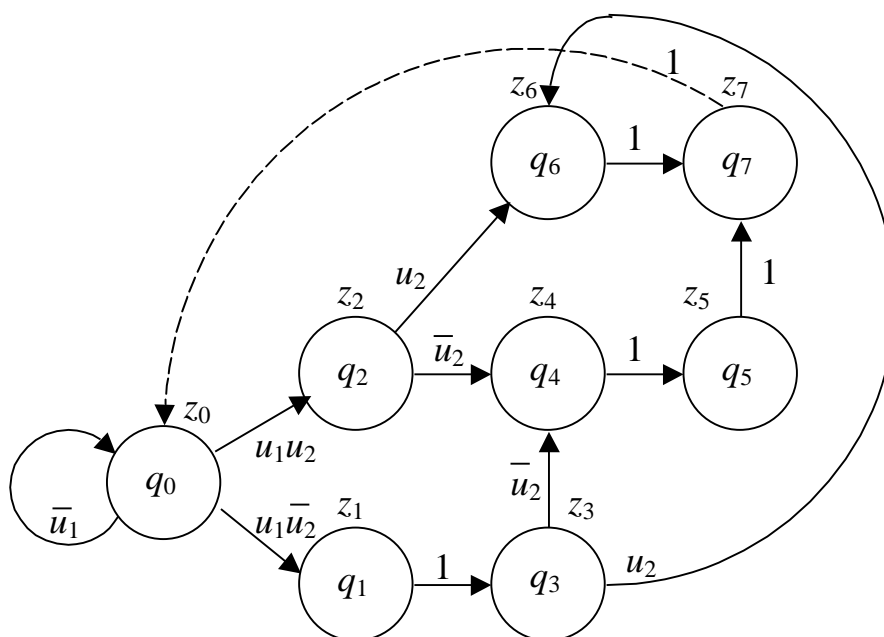


Рис. 12. Граф состояний автомата Мура

Следует различать внутренние состояния автомата Мура (на графе состояний они обозначены через  $q_i$ ) и внешние управляющие символы ( $z_j$ ). В общем случае  $z_i \neq q_i$ , в том смысле, что одни и те же управляющие символы могут формироваться при различных состояниях управляющего автомата. Пример несовпадения состояний автомата и управляющих символов приведен на рис. 13. В алгоритме оператор  $z_i$  устанавливает исходное состояние  $q$  автомата, а оператор  $z_j$  вызывает его инкремент. При четных вхождениях в цикл формируется управляющий символ  $z_0$ , при нечетных – символ  $z_1$ . Выход из цикла происходит при проверке условия  $q > a$ . Несмотря на то, что выполняется один и тот же оператор, например  $z_0$  или  $z_1$ , внутреннее состояние автомата  $q$  меняется, что должно быть отражено на граф схеме алгоритма, и в данном случае должно быть реализовано в виде счетчика состояний.

В соответствии с графом состояний алгоритма составляется список состояний автомата Мура, имеющий три столбца. В первом столбце перечисляются исходные состояния, во втором – состояния, сопряженные с исход-

ными, и в третьем – логические условия переходов.

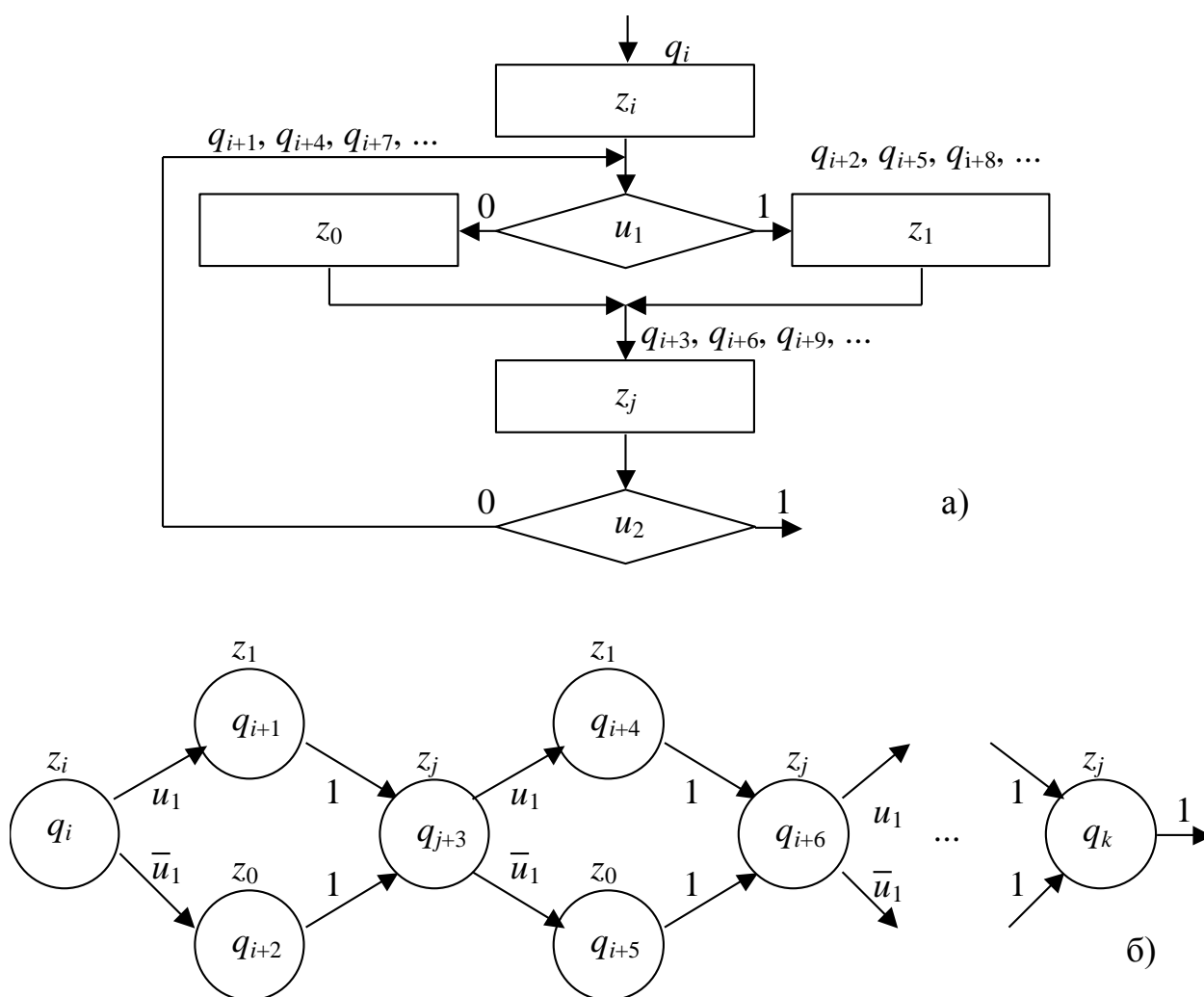


Рис. 13. Фрагмент алгоритма а), в котором состояния автомата не совпадают с состояниями управляющих символов и граф состояний б)

Табл. 2

Список состояний автомата Мура		
$q(t)/z$	$w(t)=q(t+1)$	$u$
$q_0/b$	$q_0$	$\bar{u}_1$
	$q_1$	$u_1 \bar{u}_2$
$q_2$	$u_1 u_2$	
	$q_1/z_1$	$q_3$
$q_2/z_2$	$q_6$	$u_2$
	$q_4$	$\bar{u}_2$
$q_3/z_3$	$q_4$	$\bar{u}_2$
$q_6$	$u_2$	
$q_4/z_4$	$q_5$	1
$q_5/z_5$	$q_7$	1

$q_6/z_6$	$q_7$	$1$
$q_7/e \quad \dots \quad q_0$		$1$

В табл. 2  $b$  и  $e$  - соответственно, начальное и конечное состояния управляющего автомата, которые в общем случае не совпадают. Начальное состояние автомата формируется при запуске управляющего устройства, который может быть реализован при подаче сигнала  $r$  (reset - сброс) на его вход. Конечное состояние формируется из начального в результате интерпретации управляющим устройством заложенного в него алгоритма.

Для проведения структурного синтеза управляющего автомата по автомату Мура символы  $q_1, \dots, q_n$ , определяющие внутренние состояния автомата должны быть представлены в виде кортежа булевых переменных

$$q_i = (q_i^0, \dots, q_i^j, \dots, q_i^{n-1}),$$

где  $q_i^j$  - логическая переменная, принимающая одно из двух значений  $\{0, 1\}$ .

Точно также представляются и векторы  $u, v, w, z, y$ :

$$u_i = (u_i^0, \dots, u_i^j, \dots, u_i^{m-1}); v_i = (v_i^0, \dots, v_i^j, \dots, v_i^{p-1}); w_i = (w_i^0, \dots, w_i^j, \dots, w_i^{s-1});$$

$$z_i = (z_i^0, \dots, z_i^j, \dots, z_i^{f-1}); y_i = (y_i^0, \dots, y_i^j, \dots, y_i^{h-1}).$$

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 минимизировать комбинационную схему  $L$ , (см. рис. 7) 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).

Таблицей истинности называется таблица, в первом столбце которой указывается номер строки, во втором - всевозможные комбинации значений независимых переменных (в данном случае определяющих векторы  $q, u, v$ ), а в третьем - значения переменных булевой функции (в данном случае формируемой векторами  $w, z, y$ ). Если при какой-либо комбинации независимых переменных значение функции не определено, то вместо значения функции

ставится один из символов: «-», «н», «х».

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.

Карта Карно (диаграмма Вейча) представляет собой таблицу (см. табл. 3), в которой строки и столбцы определяют коды переменных, а в клетки столбцов вписывается значение определяемой функции. Коды по осям располагаются таким образом, что каждый соседний код отличается от преды-

дущего только в одном разряде. На графе состояний такие коды имеют минимальное расстояние Хемминга, равное единице.

Табл. 3

Карта Карно для 6-ти разрядов

a <sub>4</sub> , a <sub>5</sub> , a <sub>6</sub>	a <sub>1</sub> , a <sub>2</sub> , a <sub>3</sub>							
	000	001	011	010	110	111	101	100
000								
001								
011								
010								
110								
111								
101								
100								

На карте Карно формы вида:  $xa_i \vee \overline{xa_i}$ . Они располагаются в соседних столбцах одной строки, либо в соседних строках одного столбца. К указанным формам применимо правило склеивания

$$xa_i \vee \overline{xa_i} = x,$$

что позволяет использовать карту Карно для минимизации булевых выражений.

В конечном итоге булевы выражения из нормальной дизъюнктивной (конъюнктивной) формы с помощью правила Де-Моргана должны быть сведены к штриху Шеффера (стрелке Пирса), т.е. к виду, указанному в варианте задания на курсовую работу.

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

## 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, 14	$I_0, \dots I_{15}$	Натуральный двоичный код ( $I_{15}$ – старший разряд)	По входу $r$ производится начальная установка устройства, при этом $u = 00$ .
	$z$	Управляющий вход	Код $I_0, \dots I_{15}$ поступает на входы $I$ параллельно в сопровождении сигналов $c$ и $z$ .
	$c$	Вход тактирования	

	<b><i>u</i></b>	Признак результата	<p>В случае, если <math>z = 0</math>, производится суммирование кода, хранящегося в устройстве, с входным кодом. В случае, если <math>z = 1</math>, производится вычитание из кода, хранящегося в устройстве, входного кода.</p> <p>На входы <b><i>I</i></b> устройства поступает 8 параллельных кодов.</p> <p>Устройство останавливается, если возникает переполнение, при этом на выходах <b><i>u</i></b> признака результата выставляется 01, или 10, в зависимости от знака переполнения.</p> <p>Если в процессе работы не возникает переполнения, то устройство останавливается после поступления 8 кодов. При этом на выходах <b><i>u</i></b> признака результата выставляется 00, или 11.</p>
2, 15	$I_{0,15}, \dots, I_{0,00}$	Дополнительный двоичный код ( $I_{0,15}$ – знаковый разряд)	По входу <b><i>r</i></b> производится начальная установка устройства, при этом <b><i>u</i></b> = 0.
	$I_{1,15}, \dots, I_{1,00}$	Дополнительный двоичный код ( $I_{1,15}$ – знаковый разряд)	Коды $I_{0,15}, \dots, I_{0,00}$ и $I_{1,15}, \dots, I_{1,00}$ поступают на входы <b><i>I</i><sub>0</sub></b> и <b><i>I</i><sub>1</sub></b> параллельно, побитно, начиная с младших разрядов, в сопровождении сигналов <b><i>c</i></b> и <b><i>z</i></b> .
	<b><i>z</i></b>	Управляющий вход	<p>В случае, если <math>z = 0</math>, производится суммирование кодов. В случае, если <math>z = 1</math>, суммирование блокируется.</p> <p>Если после суммирования сформирован хотя бы один отрицательный результат, то устройство прекращает работу.</p> <p>Устройство прекращает работу также, после поступления 32 пар кодов. По окончании работы на выходах <b><i>u</i></b> признака результата выставляется 00, или 11, если результат положительный или отрицательный без переполнения, и 01, или 10, если результат положительный или отрицательный с переполнением.</p>
	<b><i>c</i></b>	Вход тактирования	
	<b><i>u</i></b>	Признак результата	

3, 16	$I_7, \dots, I_0$	Натуральный двоичный код	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 0</math>. Код <math>I_0, \dots, I_8</math> поступает на вход последовательно, разряд за разрядом, начиная со старшего, в сопровождении сигнала <math>C</math>.</p> <p>Код суммируется с кодом, хранящимся в устройстве.</p> <p>Если в результате суммирования возникает переполнение, то все следующие коды вычитаются из содержимого устройства.</p> <p>Устройство прекращает работу после того, как возникает отрицательное переполнение, при этом <math>u = 0</math>, если результат четный и <math>u = 1</math>, если результат нечетный.</p>
	$c$	Тактирование	
	$u$	Признак результата	
4, 17	$I_{0,07}, \dots, I_{0,00}$	Натуральный двоичный код $I_0$ ( $I_{0,07}$ – старший разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 0</math>. Коды <math>I_{0,07}, \dots, I_{0,00}</math> и <math>I_{1,07}, \dots, I_{1,00}</math> поступают на группу из 8 входов параллельно-последовательно, сначала код <math>I_0</math>, затем код <math>I_1</math> в сопровождении сигналов <math>c</math> (тактирует поступление битов) и <math>z</math>.</p> <p>В случае, если <math>z = 00</math>, производится суммирование кодов. В случае, если <math>z = 10</math>, производится вычитание из кода <math>I_0</math> кода <math>I_1</math>. В случае, если <math>z = 01</math>, производится вычитание из кода <math>I_1</math> кода <math>I_0</math>. В случае, <math>z = 01</math> операции суммирования и вычитания блокируются.</p> <p>Если в результате операции происходит переполнение, то устройство прекращает работу, при этом <math>u = 0</math>, если переполнение положительное, и <math>u = 1</math>, если переполнение отрицательное.</p>
	$I_{1,07}, \dots, I_{1,00}$	Натуральный двоичный код $I_1$ ( $I_{1,07}$ – старший разряд)	
	$c$	Тактирование	
	$z$	Управление суммированием	
	$u$	Признак результата	
5, 18	$I_{0,15}, \dots, I_{0,0}$	Натуральный двоичный код $I_0$ ( $I_{0,15}$ – старший разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>. Коды <math>I_{0,15}, \dots, I_{0,0}</math> и <math>I_{1,15}, \dots, I_{1,0}</math> посту-</p>

	$I_{1,15}, \dots, I_{1,0}$	Натуральный двоичный код $I_1$ ( $I_{1,15}$ – старший разряд)	<p>паяют на входы последовательно, начиная со старшего разряда, сначала код <math>I_0</math>, затем код <math>I_1</math> в сопровождении сигнала <math>c</math>, тактирующего поступление битов.</p> <p>Если <math>z = 0</math>, то происходит вычитание из кода <math>I_1</math> кода <math>I_0</math>. Если <math>z = 1</math>, то происходит вычитание из кода <math>I_0</math> кода <math>I_1</math>.</p> <p>Устройство выполняет восемь операций, после чего останавливается. Если после вычитания формируется нулевой результат, то устройство также останавливается.</p> <p>По окончании работы выставляется следующие признаки результата:  <math>u = 00</math> - нулевой результат;  <math>u = 01</math> - положительный результат;  <math>u = 10</math> - отрицательный результат.</p>
	$c$	Тактирование	
	$z$	Управление суммированием	
	$u$	Признак результата	
6, 19	$I_{15}, \dots, I_0$	Натуральный двоичный код ( $I_{15}$ – старший разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>.</p> <p>Код <math>I_{15}, \dots, I_0</math> поступают на единственный вход последовательно, начиная со старшего разряда, в сопровождении сигнала <math>c</math>, тактирующего поступление битов.</p> <p>Производится суммирование содержимого устройства с кодом с четным номером и вычитание из содержимого кода с нечетным номером.</p> <p>Устройство останавливается после поступления на вход 8 последова-</p>
	$c$	Тактирование	

	$u$	Признак результата	<p>тельностью.</p> <p>Если после суммирования происходит хотя бы одно переполнение, положительное или отрицательное, то устройство также останавливается. При этом <math>u = 00</math> при нормальном положительном результате, <math>u = 11</math> при нормальном отрицательном результате, <math>u = 01</math> при положительном переполнении, <math>u = 10</math> при отрицательном переполнении.</p>
7, 20	$I_7, \dots, I_0$	Натуральный двоичный код ( $I_7$ – старший разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>. Коды <math>I_7, \dots, I_0</math> поступают на входы параллельно в сопровождении сигнала <math>z</math>.</p> <p>Код с нечетным номером сравнивается с предыдущим кодом с четным номером. Если код с нечетным номером больше кода с четным номером и <math>z = 0</math>, то к выходному коду <math>D</math> прибавляется единица, если меньше, то суммирования не происходит.</p> <p>Если код с нечетным номером меньше кода с четным номером и <math>z = 1</math>, то из выходного кода <math>D</math> вычитается единица, если больше, то вычитания не происходит.</p> <p>Устройство останавливается после поступления 128 кодов. Устройство также останавливается, если происходит переполнение <math>D</math>. При этом выставляется признак результата: <math>u = 11</math> при нормальном отрицательном результате, <math>u = 01</math> при положительном переполнении, <math>u = 10</math> при отрицательном переполнении.</p>
	$c$	Тактирование	
	$z$	Управляющий вход	
	$D_4, \dots, D_0$	Выходной код ( $D_4, D_3$ – знаковые разряды)	
	$u$	Признак результата	
8, 21	$I_{15}, \dots, I_0$	Дополнительный двоичный код ( $I_{15}$ – знаковый разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>. Коды <math>I_{15}, \dots, I_0</math> поступают на единственный вход поразрядно, начиная со старшего разряда.</p> <p>Находится модуль каждого посту-</p>
	$c$	Тактирование	
	$u$	Признак результата	

			<p>пившего кода и этот модуль суммируется с кодом, хранящимся в устройстве, если <math>z = 0</math> и вычитается, если <math>z = 1</math>.</p> <p>После поступления 8 кодов устройство останавливается, при этом выставляется признак результата:</p> <p><math>u = 11</math> при нормальном отрицательном результате, <math>u = 01</math> при положительном переполнении, <math>u = 10</math> при отрицательном переполнении, <math>u = 11</math> при нормальном отрицательном результате,</p>
	$z$	Управляющий выход	
9, 22	$I_{15}, \dots, I_0$	Натуральный двоичный код ( $I_{15}$ – старший разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>.</p> <p>Коды <math>I_{15}, \dots, I_0</math> поступают на входы параллельно в сопровождении сигнала <math>z</math>.</p> <p>В случае, если <math>z = 0</math>, производится вычитание из кода с четным номером кода с нечетным номером. В случае, если <math>z = 1</math>, производится вычитание из кода с нечетным номером кода с четным номером.</p> <p>Количество отрицательных результатов подсчитывается.</p> <p>Устройство кончает работу после поступления на вход двух одинаковых сравниваемых кодов (результат вычитания равен нулю), или после поступления 32 кодов.</p> <p>При этом, если был нулевой результат сравнения, то <math>u = 0</math>, и <math>D_3, \dots, D_0 = 0000</math>, если не было нулевого результата сравнения, то <math>u = 1</math>, и на выходы <math>D_3, \dots, D_0</math> выдается количество отрицательных результатов.</p>
	$z$	Управляющий вход	
	$c$	Тактирование	
	$u$	Признак результата	
	$D_3, \dots, D_0$	Выход устройства	
10, 23	$I_{15}, \dots, I_0$	Натуральный двоичный код ( $I_{15}$ – старший разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>, <math>D_3, \dots, D_0 = 0000</math>.</p> <p>Коды <math>I_0, I_1, I_2, I_3</math> поступает на входы параллельно в сопровождении сигналов <math>z</math>.</p>
	$z$	Управляющий вход	
	$c$	Тактирование	

			<p>В случае, если <math>z = 0</math>, производится вычитание из кода с нечетным номером кода с четным номером. В случае, если <math>z = 1</math>, производится вычитание из кода с четным номером кода с нечетным номером.</p> <p>Количество четных результатов фиксируется в коде <math>D_3, \dots, D_0</math>, причем, если был положительный четный результат, то к коду <math>D_3, \dots, D_0</math> добавляется единица, если был отрицательный четный результат, то из кода <math>D_3, \dots, D_0</math> вычитается единица, если результат был нечетный, то изменение кода <math>D_3, \dots, D_0</math> блокируется.</p> <p>Устройство кончает работу после поступления 32 кодов.</p> <p>При этом, если результат, получившийся на выходах <math>D_3, \dots, D_0</math> отрицательный, то <math>u = 1</math>, если положительный, то <math>u = 0</math>.</p>
	$D_3, \dots, D_0$	Выход устройства	
11, 24	$I_{0,15}, \dots, I_{0,0}$	Дополнительный двоичный код $I_0$ ( $I_{0,15}$ – знаковый разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>, <math>D_3, \dots, D_0 = 0000</math>.</p> <p>Коды <math>I_{0,15}, \dots, I_{0,0}</math> и <math>I_{1,15}, \dots, I_{1,0}</math> поступают на входы последовательно-параллельно, начиная с младших разрядов. Производится их суммирование и фиксация результата. Если результат суммирования положительный и <math>z = 1</math> то к коду <math>D_3, \dots, D_0</math> добавляется единица, если результат суммирования отрицатель-</p>
	$I_{1,15}, \dots, I_{1,0}$	Дополнительный двоичный код $I_1$ ( $I_{1,15}$ – знаковый разряд)	
	$c$	Тактирование	
	$u$	Признак результата	
	$D_3, \dots, D_0$	Выход устройства	

	$z$	Управляющий вход	<p>ный <math>z = 1</math> то из кода <math>D_3, \dots, D_0</math> вычитается единица. Если результат суммирования положительный и <math>z = 0</math> то из кода <math>D_3, \dots, D_0</math> вычитается единица, если результат суммирования отрицательный <math>z = 0</math> то к коду <math>D_3, \dots, D_0</math> добавляется единица.</p> <p>Результаты выдаются на выходы <math>D</math> в параллельном коде после поступления 32 кодов, или если после суммирования получилось хотя бы одно переполнение, положительное или отрицательное.</p> <p>При этом, если было переполнение при суммировании кодов <math>I_{0,15}, \dots, I_{0,0}</math> и <math>I_{1,15}, \dots, I_{1,0}</math>, то <math>u = 01</math> при положительном переполнении, <math>u = 10</math> при отрицательном переполнении. Если переполнения не было, то <math>u = 00</math>, если код <math>D_3, \dots, D_0</math> больше нуля <math>u = 1</math>, если код <math>D_3, \dots, D_0</math> меньше нуля.</p>
12, 25	$I_{0,15}, \dots, I_{0,0}$	Дополнительный двоичный код $I_0$ ( $I_{0,15}$ – знаковый разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 00</math>, <math>D_3, \dots, D_0 = 0000</math>.</p> <p>Коды <math>I_{0,15}, \dots, I_{0,0}</math> и <math>I_{1,15}, \dots, I_{1,0}</math> поступают на входы последовательно-параллельно, начиная со старших разрядов. После поступления пары кодов производится их суммирование с кодом, хранящемся в устройстве, сначала кода <math>I_{0,15}, \dots, I_{0,0}</math>, затем кода <math>I_{1,15}, \dots, I_{1,0}</math>.</p>
	$I_{1,15}, \dots, I_{1,0}$	Дополнительный двоичный код $I_1$ ( $I_{1,15}$ – знаковый разряд)	
	$c$	Тактирование	
	$D_{0,15}, \dots, D_{0,0}$	Выход устройства	

			<p>Устройство останавливается и код, хранящийся в устройстве, выдается на выходы <math>D_{0,15}, \dots, D_{0,0}</math> после поступления 16 пар кодов, или после переполнения.</p> <p>При этом, <math>u = 01</math> при положительном переполнении, <math>u = 10</math> при отрицательном переполнении, <math>u = 00</math>, если код, сформированный в устройстве положительный без переполнения и код <math>D_3, \dots, D_0</math> больше нуля <math>u = 11</math>, если код отрицательный без переполнения.</p>
13, 26	$I_{0,15}, \dots, I_{0,0}$	Натуральный двоичный код $I_0$ ( $I_{00}$ – старший разряд)	<p>По входу <math>r</math> производится начальная установка схемы, при этом <math>u = 0</math>, <math>D_3, \dots, D_0 = 0000</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 последовательно-параллельно, начиная с младших разрядов в сопровождении сигнала <math>s</math>, тактирующего поступление битов.</p> <p>Если <math>z = 01</math>, то с кодом, хранящимся в устройстве, суммируется код <math>I_{0,15}, \dots, I_{0,0}</math>. Если <math>z = 10</math>, то с кодом, хранящимся в устройстве, суммируется код <math>I_{1,15}, \dots, I_{1,0}</math>. Если <math>z = 11</math>, то суммирование блокируется. Если <math>z = 00</math>, то устройство обнуляется.</p> <p>Устройство выполняет 16 операций и останавливается. Остановка также происходит при переполнении сумматора.</p> <p>Результат выдается на выходе в коде <math>D_{19}, \dots, D_0</math>. При этом, <math>u = 0</math>, если переполнения не было, <math>u = 1</math> при переполнении,</p>
	$I_{0,15}, \dots, I_{0,0}$	Натуральный двоичный код $I_1$ ( $I_{10}$ – старший разряд)	
	$C$	Тактирование	
	$z$	Управляющие входы	
	$D_{19}, \dots, D_0$	Выходной код	
	$u$	Признак результата	

Табл. П.2

Типы логических устройств для реализации элементов памяти и комбинационных схем.

№ в.	Базис комбинацион-	Базис элементов па-	№ варианта	Базис комбинацион-	Базис элементов па-
------	--------------------	---------------------	------------	--------------------	---------------------

	ных схем	мяти		ных схем	мяти
1, 7, 13, 21	Штрих Шеффера	RS-триггер	4, 10, 16, 24	Стрелка Пирса	RS-триггер
2, 8, 14, 22	Штрих Шеффера	D-триггер	5, 11, 17, 23	Стрелка Пирса	D-триггер
3, 9, 15, 23	Штрих Шеффера	JK-триггер	6, 12, 18, 24	Стрелка Пирса	JK-триггер

ТИТУЛЬНЫЙ ЛИСТ К ПОЯСНИТЕЛЬНОЙ ЗАПИСКЕ  
КУРСОВОЙ РАБОТЫ ПО МАТЕМАТИЧЕСКОЙ ЛОГИКЕ И КОНЕЧНЫМ  
АВТОМАТАМ

МИНОБРНАУКИ РОССИИ

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

Политехнический институт  
Кафедра «Робототехника и автоматизация производства»

ОСНОВЫ ДИСКРЕТНОЙ МАТЕМАТИКИ

Пояснительная записка к курсовой работе

Направление подготовки  
**09.03.02 Информационные систем и технологии**

Профиль  
**Информационные систем и технологии в робототехнике**

Форма обучения: очная

Идентификационный номер образовательной программы:  
090302-02-21

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

И.О. Фамилия  
И.О. Фамилия

Тула 2021 год