



**МОСКОВСКИЙ
АВИАЦИОННЫЙ ИНСТИТУТ**

ИЗДАТЕЛЬСТВО

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К КУРСОВОЙ РАБОТЕ
ПО ДИСЦИПЛИНЕ
«ВЫЧИСЛИТЕЛЬНЫЕ
СИСТЕМЫ
И МИКРОПРОЦЕССОРНАЯ
ТЕХНИКА»**

МОСКВА · 1991

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СССР ПО НАРОДНОМУ ОБРАЗОВАНИЮ

МОСКОВСКИЙ ОРДЕНА ЛЕНИНА И ОРДЕНА ОКТЯБРЬСКОЙ РЕВОЛЮЦИИ
АВИАЦИОННЫЙ ИНСТИТУТ имени СЕРГО ОРДЖОНИКИДЗЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ
"ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И МИКРОПРОЦЕССОРНАЯ ТЕХНИКА"

Утверждено
на заседании редсовета
15 октября 1990 г.

Москва
Издательство МАИ
1991

УДК: 68I.325.5:62I.382.049.77+68I.322(075)

Авторы-составители: Б.С. Мельников, А.В. Щеглов

Методические указания к курсовой работе по дисциплине "Вычислительные системы и микропроцессорная техника" / Авт.-сост.: Б.С. Мельников, А.В. Щеглов. - М.: Изд-во МАИ, 1991. - 36 с.: ил.

Методические указания предназначены для студентов факультета радиозлектроники летательных аппаратов, выполняющих курсовую работу по дисциплине "Вычислительные системы и микропроцессорная техника".

Определяются цели и задачи работы, направленные на закрепление у студентов знаний и навыков по построению цифровых устройств на современной элементной базе, порядок выполнения работы, требования к оформлению записки и порядок защиты.

Рассматривается пример построения цифрового устройства, на котором демонстрируется подход к выполнению курсовой работы.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ
ПО ДИСЦИПЛИНЕ "ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
И МИКРОПРОЦЕССОРНАЯ ТЕХНИКА"

Мельников Борис Сергеевич
Щеглов Александр Васильевич

Редактор Г.Н. Борисова
Техн. редактор В.Н. Горячева

Подписано в печать 05.06.91

Бум. офсетная. Формат 60x84 I/I6. Печать офсетная
Усл. печ. л.2,09. Уч.-изд. л. 2,12. Тираж 500
Зак.2431 / I76. Цена 20 к.

Типография издательства МАИ
I2587I, Москва, Волоколамское шоссе, 4

I. ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ И ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

I.1. ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Курсовая работа по дисциплине "Вычислительные системы и микропроцессорная техника" способствует развитию у студентов навыков проектирования цифровых устройств на современной элементной базе (БИС и микропроцессорах). Она помогает закрепить полученные ранее теоретические знания по оптимальному построению разрабатываемых устройств и микропроцессорных вычислителей и применению этих знаний для решения нестандартных задач, ранее не описанных в литературе.

Необходимо не только построить реальное работающее устройство, но и доказать, что это устройство, в некотором смысле, является наилучшим - наиболее просто решающим поставленную задачу, поэтому многовариантность решения является обязательным условием, позволяющим достигнуть поставленной в работе цели.

I.2. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Задание выдается в течение первой недели преподавателем-консультантом на вводной лекции, на которой каждый студент получает индивидуальное задание на курсовую работу. В задании указывается дата выдачи и защиты работы, приводятся исходные данные к разрабатываемому устройству.

Всякое цифровое устройство служит для преобразования входной информации в выходную, поэтому содержанием главной части задания является описание входной, выходной информации и требуемого закона преобразования. Возможны формы задания в виде: словесной формулировки, формулы преобразования, таблиц или графиков, эшор напряжений, схемы алгоритма и др.

Как правило, главная часть задания сопровождается вспомогательными требованиями к проектируемому устройству. К ним можно отнести:

I) быстродействие, время задержки на формирование выходных сигналов и тактовую частоту устройства;

- 2) способ синхронизации с источником и потребителем информации;
- 3) наличие внутренней индикации;
- 4) способы схемного самоконтроля работоспособности устройства;
- 5) потребляемую мощность;
- 6) ограничения в выборе элементной базы;
- 7) наличие элементов ручного управления – кнопок, тумблеров, переключателей;
- 8) характер питания (автономное, общее);
- 9) инициализацию устройства при включении питания и т.д.

Некоторые вспомогательные требования могут не указываться. В этом случае студент либо уточняет эти требования вместе с преподавателем, либо аргументированно выдвигает эти требования самостоятельно.

Основная группа тем, выдаваемых преподавателем, связана с разработкой специализированных устройств обработки информации с различными законами преобразования входных кодов в выходные. Например, темы могут быть сформулированы следующим образом:

1. На вход устройства приходит код $A \{ I:64 \}$. Определить в этом коде комбинации вида $II0II$ и установить номера разрядов, где расположены нули этих комбинаций.

2. Дан параллельный код $A \{ I:64 \}$. Разработать устройство, формирующее разность числа единиц и нулей кода, и выразить эту разность в прямом коде.

Этот список можно продолжить другими темами, в которых задаются различные законы преобразования кодов. Например, возможна разработка устройств для определения числа одинаковых кодов из ста входящих на его вход параллельных кодов, кода с максимальным числовым эквивалентом среди множества двоичных позиционных кодов со знаком или без него и т.д. Все эти темы связаны с задачами специального вида обработки входной информации.

Наряду с ними могут выдаваться темы, связанные с разработкой отдельных узлов ЭВМ: арифметических устройств, множительных устройств с различными способами выполнения умножения, преобразователей десятичной информации в двоичную и обратно, интерфейсных устройств ввода-вывода информации и связи с внешними устройствами (преобразователями аналог-код, код-аналог, исполнительными устройствами и т.д.). Некоторые студенты самостоятельно изготавливают персональные компьютеры. В этом случае тема задания может отра-

жать интересы студента в этой области и содержать разработку узлов этого компьютера.

Тема задания может определяться также и тематикой научно-исследовательской работы студентов на кафедрах или СКБ, если она связана с построением цифровых вычислителей и по своему уровню не ниже стандартного задания на проектирование по дисциплине "Вычислительные системы и микропроцессорная техника".

1.3. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Работа с учетом многовариантности построения устройства включает в себя следующие этапы.

1.3.1. Уточнение задания

На этом этапе работы студент должен проанализировать задание, уточнить сведения о характеристиках и параметрах проектируемого устройства и определить дополнительную информацию, необходимую для реализации устройства в виде технического макета. Эта часть работы является наиболее ответственной, так как в правильной постановке задачи содержится 50% ее решения.

1.3.2. Выбор направления и обсуждение вариантов решения задачи

Прежде всего необходимо решить, к какому классу относится разрабатываемое устройство – к классу комбинационных или к классу регистровых (автоматных) устройств. Если оно может быть построено как в комбинационном, так и регистровом исполнении, то выбор метода реализации должен быть предельно полно обоснован. Допустима и двойная реализация устройства с последующим сравнительным анализом вариантов в курсовой работе.

При регистровом методе реализации цифрового устройства необходимо обосновать выбор одного оптимального варианта из нескольких возможных вариантов. Сравнение вариантов и выбор простейшего решения может производиться на различных этапах: составления схемы алгоритма, операционного описания устройства, разработки функциональной, а также принципиальной схемы устройства. Возможна реализация нескольких вариантов с последующим сравнительным анализом в конце курсовой работы.

Рекомендуется в качестве одного из направлений регистровой реализации использовать стандартную структуру микропроцессорного

вычислителя на КР580ВМ80, для которой в обязательном порядке составляется программа на языке ассемблера [17, 18]. Могут быть использованы микропроцессоры К1810ВМ86, К1810ВМ88, КР1821ВМ85 [19, 22] и др. Программа отлаживается на персональных ЭВМ, микроЭВМ или на лабораторной микроЭВМ "Микролаб". В последнем случае ассемблерная программа вручную транслируется в программу на машинном языке в системе команд микропроцессора КР580ВМ80.

Обсуждение вариантов решения ведется в словесной форме, общей для различных методов технического решения задачи. При необходимости могут привлекаться методы задания комбинационных схем, алгоритмические и языковые средства описания функционирования операционных устройств.

1.3.3. Составление схемы алгоритма

Первоначальное формализованное описание функционирования устройства выполняется с помощью схемы алгоритма. Она включает в себя совокупность операторов преобразования и операторов перехода (рис. 1.1). Схема начинается с оператора "Начало", который может обозначать включение устройства, инициализацию начального состояния, начало обработки входных сигналов и т.д., и завершается оператором "Конец", который может обозначать окончание обработки очередного кода, переход в режим ожидания, выключение устройства и т.д. Рекомендуется использовать операторы переходов на два направления, хотя применение переключателей на большее число переходов также допустимо. На степень детализации алгоритма никаких ограничений не накладывается, необходимо следить лишь за тем, чтобы разветвления и циклы проявляли себя в схеме алгоритма явным образом.

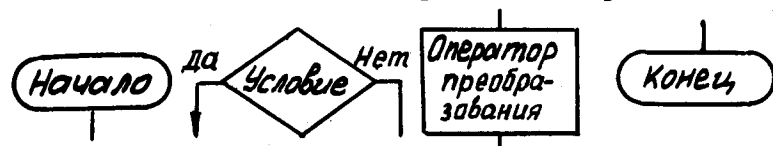


Рис. 1.1

Желательно, чтобы в операторах преобразования и переходов имена шин и операционных элементов использовались явно. Уловия переходов должны записываться содержательно в виде логических или арифметических выражений, а не в виде переменной, обозначаемой P_i . Использование идентификаторов допустимо, если они являются именами разрядов устройств или шин. При составлении схемы алгоритма целе-

сообразно отделять действия, выполняемые человеком-оператором, от действий, выполняемых устройством управления, так как только последние требуют технической реализации.

Если разрабатывается несколько алгоритмов, то их представление должно быть предельно идентичным, поскольку только в этом случае можно качественно и количественно сравнивать предлагаемые решения.

1.3.4. Составление операционного описания

Операционное описание синтезируемого устройства составляется на учебном языке операционного описания (УЮО). Возможно использование языков описания цифровых устройств и языков регистровых передач, применяемых для моделирования сложных цифровых устройств - таких, как языки *CDL*, *DDL* [7, 10].

Операционное описание является выражением алгоритма в терминах действий, выполняемых операционными элементами, - микроопераций. Поэтому при описании внутренних переменных надо учитывать имеющуюся в распоряжении разработчика элементную базу. Для упрощения операционного описания допускается вводить в него нестандартные элементы, например трехходовые сумматоры или регистры, осуществляющие сдвиг на $7z$ разрядов. Такие элементы должны быть описаны в виде программных модулей операционного описания.

При составлении операционного описания могут быть выявлены неточности в решающем алгоритме, которые потребуют либо изменения последовательности микрокоманд, либо включения новых микрокоманд. В этом случае исходный алгоритм обязательно исправляется. Таким образом, должно строго соблюдаться однозначное соответствие схемы алгоритма и операционного описания. Если замечено хоть малейшее отклонение одного от другого, то соответствие должно быть восстановлено изменением либо схемы алгоритма, либо операционного описания.

При переходе от схемы алгоритма к операционному описанию необходимо учитывать параллельность протекающих в устройстве процессов. Это обстоятельство заставляет особенно внимательно подходить к последовательной записи одновременно реализуемых операторов.

Для упрощения управляющего автомата выделяют совместные, условно-эквивалентные и эквивалентные по управлению микрокоманды. Операционное описание переписывают в терминах эквивалентных управляющих сигналов, учитывая, что условно-эквивалентные сигналы формируются непосредственно в операционном автомате.

1.3.5. Построение функциональной схемы операционного автомата

Функциональная схема операционного автомата (ОА) должна включать набор операционных элементов, связи между ними, связи этих элементов с входными и выходными шинами устройства, связи с управляющим автоматом (УА).

1.3.6. Построение функциональной схемы управляющего автомата

На этом этапе работы рекомендуется строить УА в двух вариантах – с жесткой и программируемой логикой.

При построении УА с жесткой логикой можно использовать как идеи классического синтеза конечных автоматов [13, 14], так и идеи реализации УА на счетчиках, распределителях импульсов и дешифраторах [16]. Выбор метода построения определяется вкусами разработчика.

Для построения управляющего автомата с жесткой логикой в виде конечного автомата следует получить граф переходов и выходов. Для этого необходимо построить граф-схему алгоритма. На схеме в операторах преобразования указываются только управляющие сигналы без перечисления выполняемых под их воздействием микроопераций; в операторах перехода указываются только переменные условия перехода, а направления переходов отмечаются значением этой переменной – 0 или 1. Переход от граф-схемы алгоритма к графу переходов и выходов автомата осуществляется в соответствии с методикой, изложенной в [15]. Он составляется как для автомата Мура, так и автомата Мили, после чего выбирается лучший автомат с меньшим числом состояний.

Выбор метода адресации для УА с программируемой логикой определяется характером решающего алгоритма – отношением числа операционных микрокоманд к числу управляющих микрокоманд. Как правило, лучшие результаты дает применение "естественной" адресации. Тогда работа по созданию УА будет заключаться в построении стандартной структуры УА, выборе элементов и написании соответствующей микропрограммы. На этом этапе работ целесообразно рассмотреть и обсудить несколько вариантов форматов микрокоманд и выбрать оптимальный.

1.4. ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ

Курсовая работа оформляется в виде расчетно-пояснительной записки, включающей рукописный текст и графику общим объемом 25 –

30 страниц формата А4. Записка должна включать следующие основные разделы, расположенные в порядке выполнения работы:

- 1) оглавление;
- 2) задание на проектирование;
- 3) выбор направления решения;
- 4) схему алгоритма функционирования устройства;
- 5) операционное описание устройства;
- 6) функциональную схему операционного автомата;
- 7) функциональную схему управляющего автомата;
- 8) принципиальную схему устройства;
- 9) построение устройства в виде микропроцессорного вычислителя;
- 10) заключение;
- 11) список использованной литературы.

Схемы алгоритмов и принципиальные схемы должны выполняться в соответствии с требованиями ГОСТ и ЕСКД [36, 37]. Все рисунки нумеруются и помещаются в записке после первых ссылок на эти рисунки. Если рисунок выполнен на отдельном листе, то он помещается за страницей текста, содержащей ссылку.

В записке не допускается сокращение слов, кроме общепринятых (ЭВМ, МП, ОЗУ, ПЗУ и др.). Все нестандартные аббревиатуры должны быть обязательно расшифрованы. Формулы, приводимые в записке, могут иметь номера, на которые можно сослаться в тексте. Номера размещаются с правой стороны у кромки листа в круглых скобках.

Записка вкладывается в обложку из ватмана и брошюруется. Лицевая сторона обложки является титульным листом и оформляется так, как показано на рис. 1.2. При сквозной нумерации страниц в записке этот лист считается первой страницей. Титульный лист подписывается студентом и преподавателем.

При оформлении работы в записке должен найти отражение не только наилучший вариант, но и материал, демонстрирующий весь путь движения к цели. Оформление записки производится по частям в порядке завершения работы над отдельными ее разделами.

1.5. КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Руководство курсовой работой и ее организацию осуществляет кафедра. Календарный план является графиком работы студентов и кафедра строго следит за его выполнением. Каждый студент обязан один раз в неделю отчитываться перед преподавателем о проделанной рабо-

Кафедра 4 0 3

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине
"Вычислительные системы и микропроцессорная техника"

Выполнял
студент группы

_____ Фамилия И.О.
подпись, дата

Консультировал

_____ Фамилия И.О.
подпись, дата

Москва
199__

те. Преподаватель еженедельно определяет процент выполнения студентом курсовой работы. Эти данные вывешиваются на специальном стенде кафедры.

Примерный календарный план выполнения работы рассчитан на 17 недель и приведен в табл. I:

Т а б л и ц а I

№ п/п	Наименование этапа работы	Неделя	Объем, %
1	Изучение задания	I	I
2	Разработка схемы алгоритма	3	20
3	Составление операционного описания	4	25
4	Построение функциональной схемы устройства	6	35
5	Построение принципиальной схемы устройства	10	50
6	Разработка программы для МП вычислителя	12	60
7	Оформление записки	15	90
8	Защита курсовой работы	17	100

Защита курсовой работы происходит перед комиссией, в которую обязательно входит и преподаватель-консультант. Защита может осуществляться в присутствии всей учебной группы.

2. ПРИМЕР ПОСТРОЕНИЯ ЦИФРОВОГО УСТРОЙСТВА

На примере решения демонстрационной задачи покажем весь путь выполнения курсовой работы от постановки задачи до получения принципиальной схемы реального устройства.

2.1. ЗАДАНИЕ НА ПРОЕКТИРОВАНИЕ

Первоначальная формулировка задания выглядит следующим образом:

на вход устройства приходит 32-разрядный параллельный код $A \{1:32\}$; на выходе устройства сформировать код $B \{1:k\}$, отображающий число единиц кода A .

Такая формулировка является явно недостаточной. Действительно, задание на преобразование кода A в код B требует дополнительной информации:

1. Каков период T поступления входных кодов, сколько времени отводится на преобразование? Техническое решение естественно будет разным в случае разделения входных кодов интервалами в 100 нс, 1 мкс и 10 мкс. Коды могут поступать асинхронно после преобразования предыдущего кода в устройстве. Если значение T не задается, то можно рассмотреть несколько вариантов построения устройства, оценить их быстродействие и определить область их применения.

2. Каким образом синхронизируется работа устройства с источником и потребителем информации? Примем, что источник входного кода гарантирует правильность выставленной информации во время действия стробирующего импульса СТРОБ, а само устройство подтверждает выдачу кода числа единичных символов генерацией импульса считывания УСЧИТ.

Здесь же решается вопрос о выборе тактовой синхронизирующей последовательности импульсов: или она будет общей для источника, устройства преобразования и потребителя информации (внешняя синхронизация), или она будет формироваться с помощью автономного генератора тактовых импульсов (ГТИ), установленного в разрабатываемом устройстве (внутренняя синхронизация). Выберем внешнюю синхронизацию с частотой ГТИ - 5 МГц.

При внешней синхронизации необходимо уточнить привязку входных и выходных сигналов по отношению к тактовой последовательности импульсов. Будем считать, что смена кодов А и В осуществляется по положительному фронту импульсов ГТИ, длительность импульсов СТРОБ и УСЧИТ равна периоду тактовой последовательности и положительные фронты этих импульсов появляются вслед за положительным фронтом импульсов ГТИ.

3. Каково отображение кода А в код В? Для решения задачи необходимо установить однозначное соответствие между кодом В и кодом А. Это соответствие определяется многими факторами: удобством оценки, необходимостью передачи на уровне помех, требованиями использования кодов для обнаружения и исправления ошибок, удобством индикации и многими другими факторами.

Примем, что выходной код $B \{1:k\}$ - двоичный позиционный код, численный эквивалент которого равен числу единичных символов кода $A \{1:32\}$. Легко заметить, что максимальное число "единиц", равное 32, потребует формирования шестиразрядного выходного кода $B \{1:6\}$, так как $(32)_{10} = (100000)_2$ ($32_{10} = 100000_2$).

В качестве сопутствующей можно рассмотреть задачу формирования выходного кода в двоично-десятичном представлении. Разрядность при

этом не изменится: потребуется тетрада для формирования младшей цифры (0 - 9) и два двоичных разряда для представления старшей цифры (0 - 3). Представляет интерес введение контрольного разряда $B \{7\}$ для проверки кода на четность (нечетность) при передаче по каналу связи.

Итак, задание на проектирование может быть выражено в виде функциональной схемы и эпюр напряжения, изображенных на рис. 2.1.

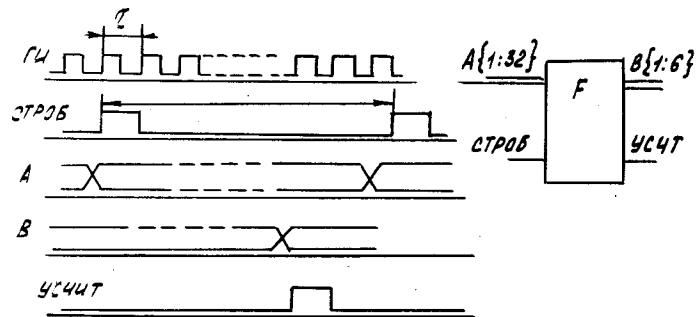


Рис. 2.1

Примем, что в разрабатываемом устройстве не требуется введения внутренней индикации, и не будем пока накладывать никаких ограничений на потребляемую устройством мощность и, следовательно, на выбор оптимальной элементной базы. Эти вопросы могут быть решены при дальнейшей работе над заданием.

Разработка схемы устройства начинается с главного вопроса - определения метода построения устройства как комбинационного или как регистрового. Согласно заданию устройство преобразует входной параллельный код в выходной, поэтому оно может быть построено и как комбинационное, и как регистровое. Рассмотрим оба варианта, сравним сложность их реализации и определим область применения.

2.2. КОМБИНАЦИОННАЯ РЕАЛИЗАЦИЯ УСТРОЙСТВА

При построении устройства в виде комбинационной схемы надо решить задачу синтеза системы шести логических функций 32 переменных. Эта задача достаточно сложна и в условиях ограниченного времени проектирования решена быть не может.

К решению задачи можно подойти с позиций операционного синтеза, построив устройство, реализующее цепочку простых преобразований. Разобьем входной код на такие части, для которых решение за-

дачи преобразования "числа единиц" в позиционный двоичный код является практически реализуемым. Разобьем входной код на четыре части: $A \{1:32\} = A \{1:8\} \cdot A \{9:16\} \cdot A \{17:24\} \cdot A \{25:32\}$.

Каждый байт преобразуется в число единиц с помощью цифрового устройства, которое назовем преобразователем кода и будем обозначать через $PPi \{1:4\}$ ($i = 1, 2, 3, 4$). Тогда

$$B \{1:6\} = PP1 \{1:4\} + PP2 \{1:4\} + PP3 \{1:4\} + PP4 \{1:4\}.$$

Будем формировать сумму в два этапа - суммированием кодов пары преобразователей и суммированием полученных сумм:

$$P1.SM1 \{0:3\} = PP1 \{1:4\} + PP2 \{1:4\},$$

$$P2.SM2 \{0:3\} = PP3 \{1:4\} + PP4 \{1:4\},$$

$$B \{1:6\} = SM \{0:5\} = P1.SM1 \{0:3\} + P2.SM2 \{0:3\}.$$

Здесь $P1$ и $P2$ - сигналы переноса 4-разрядных сумматоров $SM1$ и $SM2$.

Функциональная схема устройства изображена на рис. 2.2. Сигнал УСЧИТ образуется путем задержки сигнала СТРОБ на максимально возможное время формирования выходного кода. Оно зависит от выбранной элементной базы.

Для построения сумматоров $SM1$ и $SM2$ воспользуемся микросхемой К155ИМ3 (4-разрядный сумматор), а для построения сумматора SM - микросхемами К155ИМ3, К155ИМ2 (К155ИМ2 - 2-разрядный сумматор). Задержка в формировании выходного сигнала этими сумматорами составляет не более 40 нс.

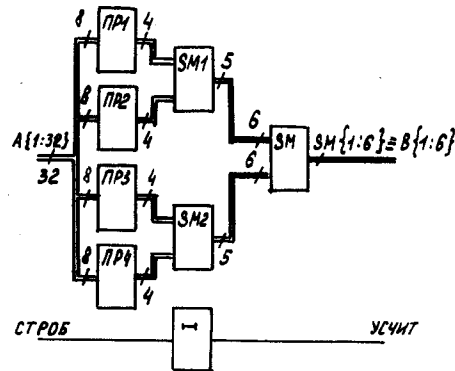


Рис. 2.2

Преобразователи кодов PPi являются нестандартными устройствами. Анализ показывает, что использование логических элементов

(элементов с "жесткой" структурой) приводит к очень объемной реализации. Возникает мысль об использовании программируемых структур - электрически программируемых постоянных запоминающих устройств (ППЗУ) или программируемых логических матриц. Первая структура более доступна; поэтому будем использовать микросхему КР556РТ4, изображенную на рис. 2.3. Эта микросхема совместима с ТТЛ-элементами (ТТЛ - транзисторно-транзисторная логика) и обеспечивает задержку не более 70 нс. Программирование этого ППЗУ будем проводить по таблице истинности функции преобразования 8-разрядного кода в 4-разрядный (табл. 2).

Таблица 2

A0	A1	A2	A3	A4	A5	A6	A7	D0	D1	D2	D3
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	0	0	1	0	0
-	-	-	-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	1	1	0	0	0	1

Проектируемое устройство строится на восьми корпусах микросхем. Оно может использоваться в системах, в которых допустимое время формирования выходного кода не превышает 150 нс: действительно, задержка выходного кода

$$\tau = \tau(\text{ППЗУ}) + \tau(SM1, SM2) + \tau(SM) = \tau(\text{КР556РТ4}) + \tau(\text{К155ИМ3}) + \tau(\text{К155ИМ3} + \text{К155ИМ2}) = 70 + 40 + 40 = 150 \text{ нс}.$$

Преобразователь на стандартных логических элементах может быть построен из двух 4-битовых преобразователей и 3-разрядного сумматора. Если $X0, X1, X2, X3$ - биты 4-разрядного поля входного слова, $Y0, Y1, Y2$ - биты выходного слова 4-битового преобразователя, определяющего двоичный позиционный код числа единиц поля входного слова, то, рассматривая $Y0, Y1, Y2$ как логические функции переменных $X0, X1, X2, X3$, получим следующие логические выражения выходов преобразователя для реализации на элементах И-НЕ и ИСКЛЮЧАЮЩЕЕ ИЛИ ($\text{mod } 2$):

$$Y0 = A0 \text{ mod } 2 \quad A1 \text{ mod } 2 \quad A2 \text{ mod } 2 \quad A3,$$

$$Y1 = (A0/A2/A3) / (A0/A1/A2) / (A0/A2/A3) / (A0/A1/A3),$$

$$Y2 = A0/A1/A2/A3.$$

Преобразователь на восемь разрядов, построенный из двух 4-битовых преобразователей на элементах К155ЛН1, К155ЛП5, К155ЛА4, К155ЛА2, К155ЛА1, К155ИМ3, содержит двенадцать корпусов вместо одного корпуса ПЗУ.

2.3. РЕГИСТРОВАЯ РЕАЛИЗАЦИЯ УСТРОЙСТВА

Комбинационная реализация разрабатываемого устройства имеет крупный недостаток – объем устройства растет пропорционально длине входного кода. Поэтому целесообразно комбинационную реализацию применять при длине кода не более 32 бит. В ряде практических случаев приходится подчитывать число единиц в многоразрядных словах, длина которых заранее не определена. Отсюда следует, что преобразователь "числа единиц" в двоичный позиционный код должен быть более гибким, более универсальным.

В основу построения регистрового варианта устройства положим идею преобразования параллельного кода в последовательный с последующим счетом числа единичных символов. Такое преобразование должно начинаться с момента прихода входного кода и сигнала СТРОБ и заканчиваться моментом окончания счета единичных символов с генерацией импульса считывания. Таким образом, задача преобразования распадается на две подзадачи: преобразование параллельного кода в последовательный и получение выходного кода с формированием УСЧИТ.

Первая подзадача решается двумя путями: первый состоит в использовании мультиплексоара, второй – сдвигового регистра. Первый вариант требует фиксации входного кода на все время преобразования. Второй вариант является более универсальным. Для его реализации достаточно выставить входной код $A \{1:32\}$ на время сигнала СТРОБ. В учебном пособии рассмотрим только второй вариант, хотя первый вариант также представляет интерес и может быть проработан в курсовой работе как альтернативный вариант.

Вторую подзадачу – формирование выходного кода – можно решить либо путем создания детерминированной последовательности управляющих сигналов сдвига всех разрядов кода через выходной разряд регистра, либо формированием признака очищения регистра от единичных символов в процессе сдвига. Первый вариант потребует установки дополнительного счетчика циклов, второй вариант – введения схемы анализа содержимого регистра на нулевое значение. При несколько больших затратах второй вариант обеспечивает в среднем более быстрое преобразование, поскольку последние нули кода преобразованию не подвергаются. Примем второй вариант определения конца преобразова-

ния в качестве рабочего, тогда, естественно, применение сдвигового регистра становится обязательным.

2.3.1. Разработка схемы алгоритма и составление операционного описания

Схема алгоритма преобразования по выбранному варианту изображена на рис. 2.4. Составим соответствующее ей операционное описание синтезируемого устройства (рис. 2.5).

В теле процедуры микропрограммы у переменных РГД и СЧЕ опущено поле разрядности. В этом случае его значение берется из раздела описания переменных микропрограммы. Под воздействием сигнала УСЧИТ не выполняется никакой операции. Этот факт отображается записью символа ";" вслед за символом ":", что равносильно введению пустой микрооперации.

Анализ микропрограммы показывает, что сигналы УЗАП и УН эквивалентны и их можно заменить одним сигналом УЗП: $УЗП = УЗАП = УН$. Проверку на ноль содержимого регистра РГД можно осуществить с помощью операции ИЛИ-НЕ (стрелка Парра):

$$P2 = \bigvee_{i=1}^{32} PГД \{i\} = \downarrow PГД \{1:32\}.$$

Текст микропрограммы, учитывающий вышесказанное, представлен на рис. 2.6.

2.3.2. Разработка функциональной схемы устройства

Функциональная схема операционного автомата, составленная в соответствии с микропрограммой 2, изображена на рис. 2.7.

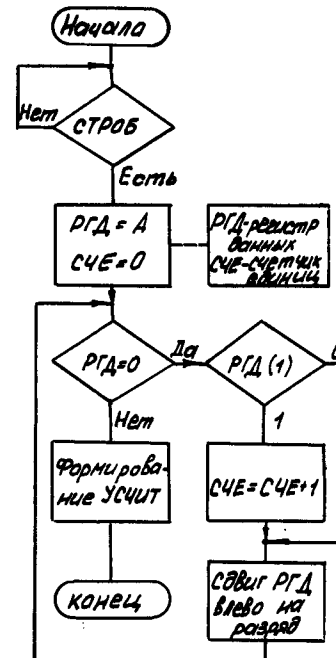


Рис. 2.4

Микропрограмма ЧИСЛО_ЕДИНИЦ;

Переменные

входные: $A\{1:32\}$, СТРОБ;
внутренние: $РГД\{1:32\}$, $СЧЕ\{1:6\}$;
выходные: $B\{1:6\}=СЧЕ\{1:6\}$, УСЧИТ;

Признаки:

$P1 = СТРОБ$;
 $P2 = РГД\{1:32\}=0$;
 $P3 = РГД\{I\}$;
Процедура
M1 если $\neg P1$, то M1;
УЗАП: $РГД = A$;
УН: $СЧЕ = 0$;
M2 если P2, то M4;
если $\neg P3$, то M3;
УСЧ: $СЧЕ = СЧЕ + I$;
M3 УСДВ: $РГД = РГД\{2:32\}, 0$;
идти к M2;
M4 УСЧИТ: ;
конец.

Рис. 2.5

Микропрограмма ЧИСЛО_ЕДИНИЦ;

Переменные

входные: $A\{1:32\}$, СТРОБ;
внутренние: $РГД\{1:32\}$, $СЧЕ\{1:6\}$;
выходные: $B\{1:6\}=СЧЕ\{1:6\}$, УСЧИТ;

Признаки:

$P1=СТРОБ$;
 $P2=\neg РГД\{1:32\}$;
 $P3=\neg РГД\{I\}$;
Процедура
M1 если $\neg P1$, то M1;
УЗП: $РГД = A$, $СЧЕ = 0$;
M2 если P2, то M4;
если $\neg P3$, то M3;
УСЧ: $СЧЕ = СЧЕ + I$;
M3 УСДВ: $РГД = РГД\{2:32\}, 0$;
идти к M2;
M4 УСЧИТ: ;
конец.

Рис. 2.6

Микропрограмма ЧИСЛО_ЕДИНИЦ;

Переменные

входные: $A\{1:32\}$, СТРОБ;
внутренние: $РГД\{1:32\}$, $СЧЕ\{1:6\}$;
выходные: $B\{1:6\}=СЧЕ\{1:6\}$, УСЧИТ;

Признаки:

$P1 = СТРОБ$;
 $P2 = РГД\{1:32\}=0$;
 $P3 = РГД\{I\}$;
Процедура
M1 если $\neg P1$, то M1;
УЗАП: $РГД = A$;
УН: $СЧЕ = 0$;
M2 если P2, то M4;
если $\neg P3$, то M3;
УСЧ: $СЧЕ = СЧЕ + I$;
M3 УСДВ: $РГД = РГД\{2:32\}, 0$;
идти к M2;
M4 УСЧИТ: ;
конец.

Рис. 2.5

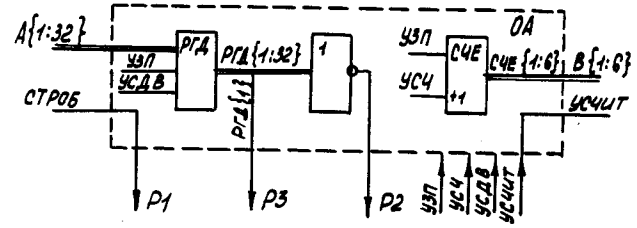


Рис. 2.7

Задачи, решаемые управляющим автоматом, достаточно просты: он генерирует управляющие сигналы сдвига УСДВ до момента появления признака обнуления P2, после чего формирует импульс считывания УСЧИТ. УА может быть построен как с жесткой логикой, так и с программируемой логикой. Рассмотрим оба способа реализации.

Управляющий автомат с жесткой логикой

УА с жесткой логикой реализуется либо в виде классического конечного автомата, либо в виде распределителя импульсов. Конечный автомат может быть построен и как автомат Мура, и как автомат Мили. Построение графов переходов и выходов УА осуществляется на основании отмеченных граф-схем алгоритмов [15]. На рис. 2.8 построены граф-схемы алгоритмов для автоматов Мура (а) и Мили (б), соответствующие микропрограмме 2 на рис.2.6 и учитывающие тот факт, что сигналы УСЧ и УСДВ являются совместными. Метки a_i отображаются в состояния автоматов, условия P1, P2, P3 определяют значения входных сигналов, а управляющие сигналы - значения выходных сигналов автоматов. Графы переходов для автоматов Мура и Мили, полученные из граф-схем алгоритмов, изображены на рис. 2.9, а и б.

Будем строить УА как автомат Мили, поскольку он имеет только два состояния и реализуется на одном триггере. В качестве по-

Таблица 3

$a(t)$			a_0	a_1
$a(t)$			0	1
P1	P2	P3		
0	0	0	0	1 УСДВ
0	0	1	0	1 УСЧ, УСДВ
0	1	1	- (0)	- (0) - (УСЧИТ)
0	1	0	0	0 УСЧИТ
1	1	0	1	0 УСЧИТ
1	1	1	- (1) УЗП	- (0) (УСЧИТ)
1	0	1	1 УЗП	1 УСЧ, УСДВ
1	0	0	1 УЗП	1 УСДВ

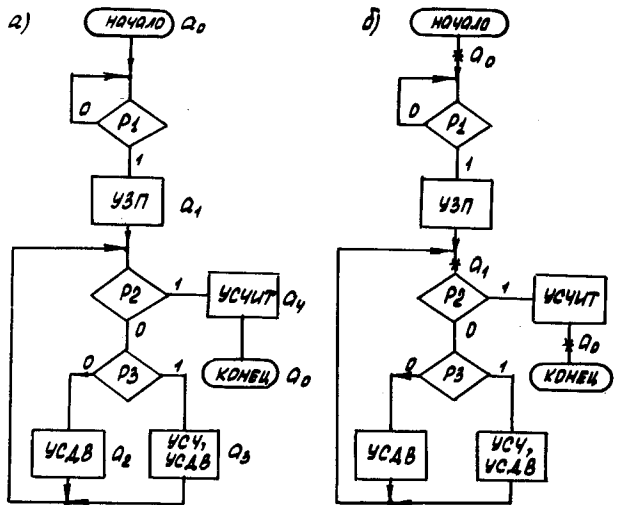


Рис. 2.8

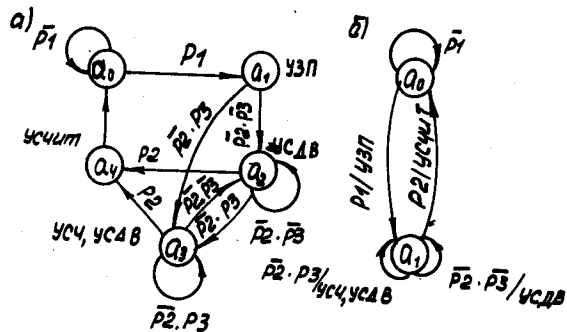


Рис. 2.9

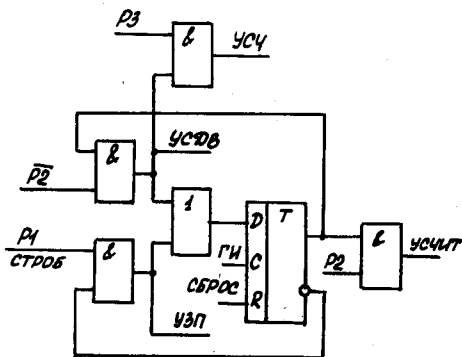


Рис. 2.10

следнего возьмем D-триггер. На основании таблицы переходов и выходов УА (табл. 3) после доопределения запрещенных комбинаций получим функцию возбуждения триггера и выражения для управляющих сигналов:

$$\begin{aligned}
 D &= P1 \cdot \bar{Q} + \bar{P2} \cdot Q, \\
 UZP &= P1 \cdot \bar{Q}, \\
 UCH &= UCDV \cdot P3, \\
 UCHYT &= P2 \cdot Q, \\
 UCDV &= \bar{P2} \cdot Q.
 \end{aligned}
 \tag{2.1}$$

Функциональная схема УА приведена на рис. 2.10, сигнал СБРОС является установочным и формируется при включении питания или от кнопки.

Управляющий автомат с программируемой логикой

Составим каноническую форму микропрограммы синтезируемого операционного устройства с учетом эквивалентности сигналов УЗАП, УН и совместимости сигналов УСЧ, УСДВ в виде табл. 4.

Таблица 4

Номер	Метка	Управляющие сигналы	Переход
0	M1		Если 1 P1, то M1
1	M2	UZP	Если P2, то M4 Если 1 P3, то M3
2	M2		
3	M2		
4	M3	УСЧ, УСДВ	
5	M3		Идти к M2
6	M3	УСДВ	
7	M3		Идти к M2
8	M4	УСЧИТ	
9	M4		Конец

В случае принудительной адресации строки 4 и 5, 6 и 7, 8 и 9 можно объединить. Тогда каноническая форма операционного описания будет на три строки короче (табл. 5).

Таблица 5

Номер	Метка	Управляющие сигналы	Переход
0	M1	УЗП	Если $\overline{P1}$, то M1
1	M2		Если P2, то M4
2			Если $\overline{P3}$, то M3
3	M3	УСЧ, УСДВ	Идти к M2
4		УСДВ	Идти к M2
5		УСЧИТ	Конец
6	M4		

Формат команды с принудительной адресацией имеет вид [I2]

УЗП	УСЧ	УСДВ	УСЧИТ	$\overline{XP1}$	$\overline{XP2}$	$\overline{XP3}$	A2	AI	AO
-----	-----	------	-------	------------------	------------------	------------------	----	----	----

Для естественной адресации имеется два формата микрокоманд [I2]

0	УЗП	УСЧ	УСДВ	УСЧИТ	-	-	-
I	$\overline{XP1}$	$\overline{XP2}$	$\overline{XP3}$	A3	A2	AI	AO

Первый разряд формата микрокоманды UA с естественной адресацией определяет признак микрокоманды: 0 - операционная микрокоманда, I - управляющая микрокоманда.

Учитывая, что табл. 4 дает каноническое описание микропрограммы для UA с естественной адресацией, получаем кодовые выражения микропрограмм для UA с естественной адресацией (табл. 6) и с принудительной адресацией (табл. 7):

Таблица 6

Адрес	Разряды микрокоманд
0000	I I 0 0 0 0 0 0
0001	0 I 0 0 0 0 0 0
0010	I 0 I 0 I 0 0 0
0011	I 0 0 I 0 I I 0
0100	0 0 I I 0 0 0 0
0101	I 0 0 0 0 0 I 0
0110	0 0 0 I 0 0 0 0
0111	I 0 0 0 0 I 0 0
1000	0 0 0 0 I 0 0 0
1000	I 0 0 0 0 0 0 0

Таблица 7

Адрес	Разряды микрокоманд
000	0 0 0 0 I 0 0 0 0
001	I 0 0 0 0 0 0 0 I 0
010	0 0 0 0 0 I 0 I I 0
011	0 0 0 0 0 0 I I 0 I
100	0 I I 0 0 0 0 0 I 0
101	0 0 I 0 0 0 0 0 I 0
110	0 0 0 I 0 0 0 0 0 0

Из сравнения табл. 6 и 7 следует, что для хранения микропрограммы с естественной адресацией требуется 80 бит, а для UA с принудительной адресацией - 70 бит. Таким образом, реализация UA с принудительной адресацией в нашем случае более желательна.

На рис. 2.II изображена функциональная схема UA с принудительной адресацией. Управляющие сигналы формируются регистром микрокоманды (РМК), в который микрокоманда переписывается из ПЗУ микрокоманд. РМК тактируется последовательностью τ_1 , а счетчик адреса СЧА - последовательностью $\tau_2: \tau_1 \& \tau_2 = 0$.

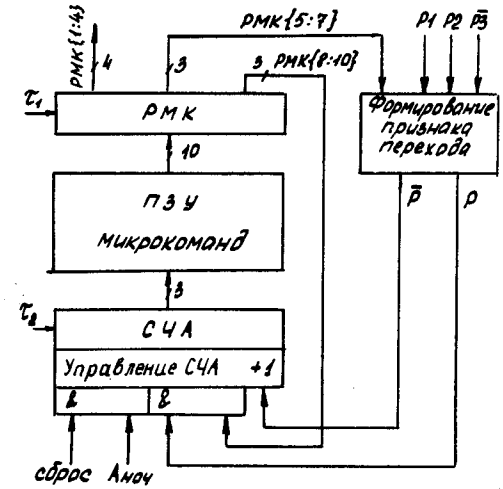


Рис. 2.II

Начальный адрес микропрограммы, равный 000, устанавливается сигналом СБРОС, и UA ждет прихода сигнала СТРОБ для продолжения работы.

Не будем приводить принципиальную схему UA, поскольку она является стандартной. Пусть переход от функциональной схемы UA к принципиальной, выполненной на заданной элементной базе, станет составной частью проекта.

При построении принципиальной схемы выберем UA с жесткой логикой, поскольку он имеет более простую организацию, чем UA с программируемой логикой.

2.3.3. Построение принципиальной схемы

Будем строить наше устройство на элементах серии KI55 и K555, так как они обеспечивают устойчивую работу на заданной тактовой частоте 5 МГц. Принципиальные схемы устройства, построенные по функциональным схемам рис. 2.7 и 2.10, изображены на рис. 2.12 и 2.13. Регистр данных РД строится на четырех микросхемах 8-разрядных сдвигающих регистров KI55IP13. Управляющие сигналы $SE0, SE1$ обеспечивают запись кода A в регистр ($SE0=1, SE1=1$) и сдвиг влево ($SE0=0, SE1=1$). Они формируются из управляющих сигналов UA в соответствии с уравнениями

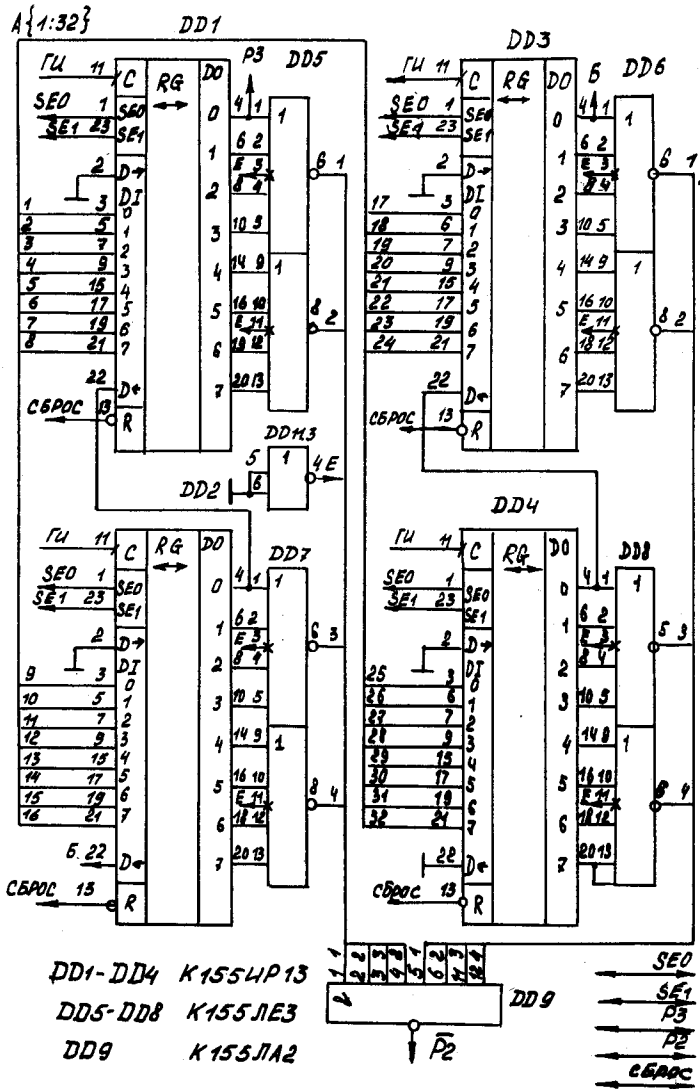


Рис. 2.12

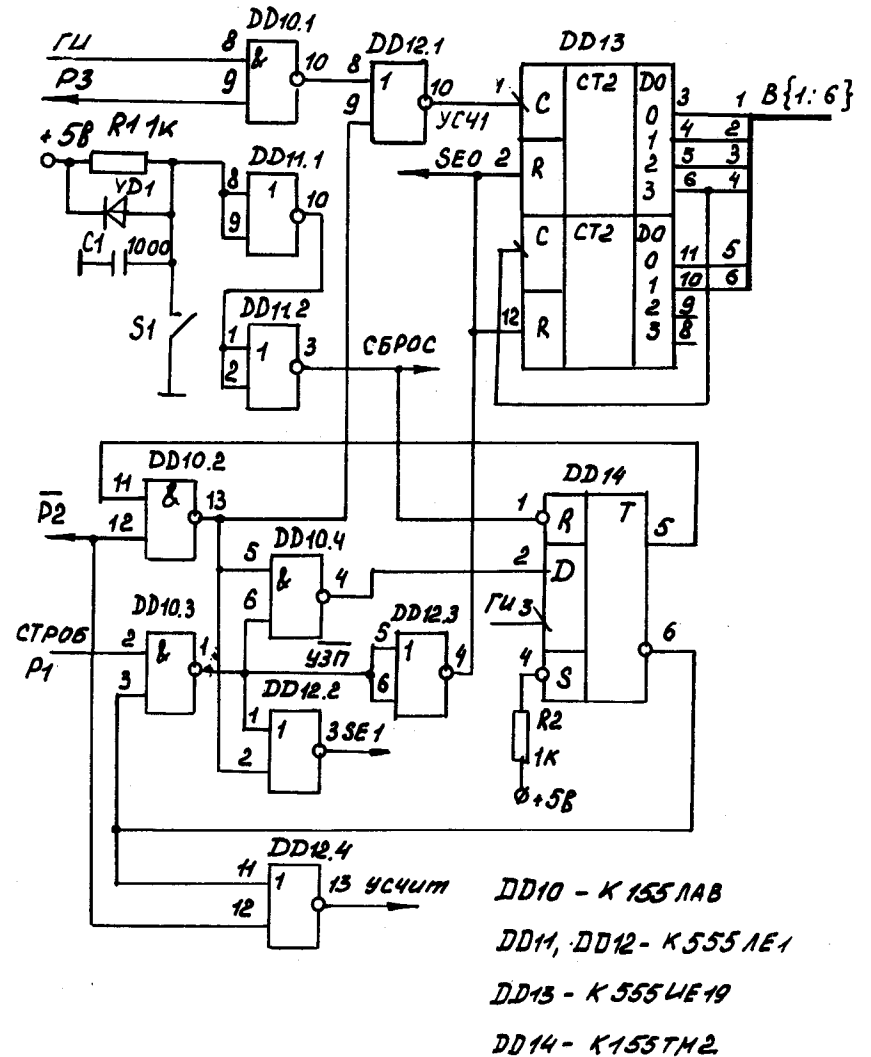


Рис. 2.13

$$SE0 = \overline{УЗП},$$

$$SE1 = \overline{УЗП} + \overline{УСДВ} = \overline{УЗП} \downarrow \overline{УСДВ}.$$

Здесь с учетом логических выражений (2.1) и выбранной элементной базы

$$\frac{\overline{УЗП}}{\overline{УСДВ}} = \frac{P1/\overline{Q}}{\overline{P2}/Q}. \quad (2.2)$$

Сигнал возбуждения Д-триггера на микросхеме DD14 типа KI55TM2, используемого для построения УА, на основании (2.1) и (2.2) запишется следующим образом:

$$D = (P1/Q1) / (\overline{P2}/Q) = \overline{УЗП} / \overline{УСДВ}. \quad (2.3)$$

Счетчик собирается на микросхеме K555IE19, включающей два 4-разрядных асинхронных счетчика с последовательным переносом. Для его работы сигнал УСЧ необходимо стробировать тактовыми импульсами ПИ:

$$\overline{УСЧ1} = \overline{УСЧ} \cdot \tau = (\overline{УСДВ} \cdot P3) \tau = \overline{УСДВ} \downarrow (P3/\tau).$$

Сигнал УСЧИТ в соответствии с (2.1) формируется на элементе DD12.4, выполняющем функцию ИЛИ-НЕ (стрелку Парса):

$$\overline{УСЧИТ} = P2 \cdot Q = \overline{P2} \downarrow \overline{Q}.$$

При его формировании необходимо организовать такой временной интервал между последним импульсом УСЧИ и УСЧИТ, чтобы гарантировать завершение всех переходных процессов в последовательном счетчике на K555IE19 (60 нс на четыре разряда, 90 нс на шесть разрядов). Работа устройства демонстрируется эпюрами напряжений на рис. 2.14. Суммарная задержка фронта импульса УСЧИТ относительно среза импульса УСЧИ – не менее 100 нс. Таким образом, к началу импульса УСЧИТ все переходные процессы в счетчике DD13 завершатся, и выходной код В {1:6} установится.

Устройство требует для своего построения 14 корпусов микросхем, среди них: по четыре микросхемы KI55IP13 и KI55IE3, по одной микросхеме KI55IA2, KI55TM2, K555IE19, KI55IA8 и две микросхемы K555IE1.

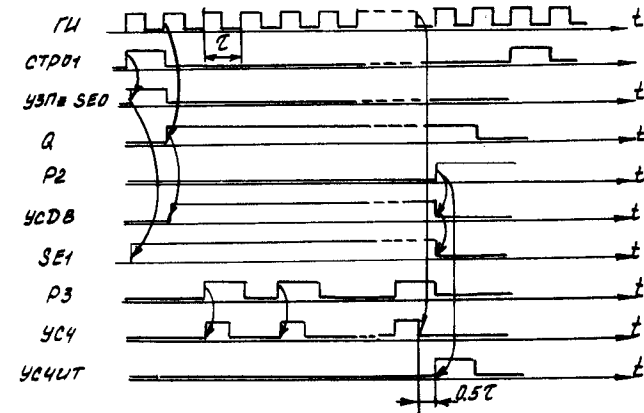


Рис. 2.14

Разработанная схема обеспечивает время преобразования от 0,25 мкс (нулевой код) до 6,25 мкс (код всех единиц) при потребляемой от источника +5 В мощности 3,5 Вт.

2.4. ПОСТРОЕНИЕ УСТРОЙСТВА В ВИДЕ МИКРОПРОЦЕССОРНОГО ВЫЧИСЛИТЕЛЯ НА БИС КР580ВМ80

Микропроцессорный вычислитель имеет стандартную структуру. Поэтому его схему строить не надо. Единственной задачей остается создание программируемого контроллера интерфейса, обеспечивающего подключение к вычислителю источника и потребителя информации, управляющей программы обмена контроллера, называемой драйвером, и обрабатывающей программы.

В отличие от рассмотренных ранее вариантов построения преобразователя в виде самостоятельного устройства для микропроцессорного варианта подсчет числа единиц кода осуществляется с помощью обрабатывающей программы. Поэтому ее написание и отладка является важным этапом в построении устройства, включающем следующие шаги:

- 1) разработку обрабатывающего алгоритма;
- 2) составление программы на языке ассемблера;
- 3) отладку и выполнение программы.

Третий шаг требует дополнительных пояснений. Отладку и выполнение программы можно осуществить либо на микроЭВМ и персональных ЭВМ, имеющих транслятор с языка ассемблера КР580ВМ80 или языков ассемблеров, включающих его как подмножество, либо на лабораторной

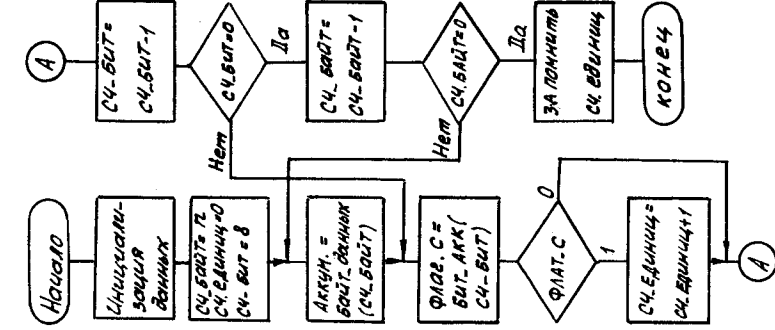


Рис. 2.15

Программа счѣта единиц

```

START 8000H
XRA A
MOV C, A
MVI D, 08H

LXI H, QBITES
MOV B, M
LXI H, DATA
REPEAT: MOV A, M
RLC
JNC NCNT
INR C
DCR D
JNZ LOOP
INX H
DCR B
JNZ REPEAT
LXI H, NONES
MOV M, C
HLT

NONES: DS 1
QBITES: DB 04H
DATA: DB 07H, 07H, 07H, 07H, 07H
END
  
```

: очистка аккумулятора
 : очистка счетчика единиц
 : загрузка в счетчик байтов
 : константы 8
 : загрузка в регистр В числа байтов
 : во входном коде
 : адрес первого байта в регистр (H, L)
 : загрузка текущего байта в аккумулятор
 : счѣт единиц в текущем байте
 : входного кода путем сдвига
 : содержимого аккумулятора
 : подготовка следующего байта
 : :
 : запись числа единиц кода в
 : память
 : :
 : число единиц кода
 : число байтов в коде
 : входной код

Рис. 2.16

микроЭВМ "Микролаб". В последнем случае необходимо выполнить ручное ассемблирование программы.

При составлении алгоритма необходимо учитывать, что микропроцессорный вычислитель имеет байтовую организацию памяти и 8-разрядную шину данных. Обобщим исходную задачу, положив, что входной код, число единиц которого подсчитывается после размещения его в памяти, занимает массив из 72 байтов.

Схема алгоритма изображена на рис. 2.15. Алгоритм включает в себя два цикла: подсчет числа единиц в текущем байте и накопление единиц при просмотре всех байтов. Входной код размещается в массиве БАЙТ_ДАНЫХ. Положение текущего байта в массиве определяется счетчиком байтов СЧ_БАЙТ. Положение текущего бита этого байта, предварительно помещенного в аккумулятор, определяется СЧ_БИТ.

Программа, реализованная по данному алгоритму, представлена на рис. 2.16. Просмотр битов текущего байта осуществляется путем сдвига аккумулятора, при этом анализируемый бит кода формирует значение признака переноса С. Окончание просмотра осуществляется по нулевому содержимому счетчика битов, расположенного в регистре В. Просмотр байтов кода завершается, когда обнуляется счетчик байтов, расположенный в регистре С. Программа загружается в ОЗУ по адресу 8000.

*
*

Для успешного выполнения курсовой работы необходимо, чтобы студент умел пользоваться и ориентироваться в научной и справочной литературе по тематике курсовой работы. Чтобы помочь студенту в этом, ниже приводится указатель литературы, сгруппированной по отдельным разделам работы.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Общие вопросы

1. Каган Б.М. Электронные вычислительные машины и системы. - М.: Энергоатомиздат, 1985.
2. Уокерли Дж. Архитектура и программирование микроЭВМ: В 2-х кн. - М.: Мир, 1984.
3. Бруснецов Н.П. Микрокомпьютеры. - М.: Наука, 1985.
4. Гилмор Ч. Введение в микропроцессорную технику. - М.: Мир, 1984.
5. Микропроцессоры: В 3-х кн. / Под ред. Л.Н. Преснухина. - М.: Высшая школа, 1986.
6. Кондратьев Р.М., Мельников Б.С., Щеглов А.В. Архитектура микропроцессорных вычислителей. - М.: МАИ, 1986.

Языки операционного описания, языки регистровых передач и синтез операционных устройств

7. Автоматизация проектирования вычислительных систем, языки моделирования и базы данных / Под ред. М. Брейера. - М.: Мир, 1979.
8. Проектирование цифровых вычислительных машин / Под ред. С.А. Майорова. - М.: Высшая школа, 1972.
9. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. - Л.: Машиностроение, 1974.
10. Чу Я. Организация ЭВМ и микропрограммирование. - М.: Мир, 1975.
11. Силин В.Б., Мельников Б.С. Электронные вычислительные устройства. Операционный синтез. - М.: МАИ, 1982.
12. Силин В.Б., Мельников Б.С., Кондратьев Р.М. Операционный синтез цифровых устройств. - М.: МАИ, 1985.

Построение управляющих автоматов

13. Силин В.Б., Мельников Б.С. Конечные автоматы: Учебное пособие. - М.: МАИ, 1978.
14. Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов. - М.: Энергия, 1978.
15. Баранов С.И. Синтез микропрограммных автоматов. - Л.: Энергия, 1978.
16. Специализированные ЦВМ / Под ред. В.Б. Смолова. - М.: Высшая школа, 1981.

Программирование микропроцессорных вычислителей

17. Григорьев В.Л. Программное обеспечение микропроцессорных систем. - М.: Энергоатомиздат, 1983.
18. Григорьев В.Л. Программирование однокристалльных микропроцессоров. - М.: Энергоатомиздат, 1987.

Организация и технические средства микропроцессорных систем

19. Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы. - М.: Радио и связь, 1989.
20. Кофферон Дж. Технические средства микропроцессорных систем: Практический курс. - М.: Энергоатомиздат, 1983.
21. Кофферон Дж., Ленг В. Расширение микропроцессорных систем. - М.: Машиностроение, 1987.
22. Микропроцессорный комплект К1810: Структура, программирование, применение: Справочная книга / Под ред. Ю.М. Казаринова. - М.: Радио и связь, 1990.
23. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения. - М.: Радио и связь, 1986.

Схемотехника электронных схем

24. Титце У., Шенк К. Полупроводниковая схемотехника. - М.: Мир, 1982.
25. Хоровиц П., Хилл У. Искусство схемотехники: В 2-х т. - М.: Мир, 1983.

Справочная литература

26. Интегральные микросхемы: Справочник / Под ред. В.В. Табраина. - М.: Радио и связь, 1984.

27. Х в о щ С.Т., В а р л и н с к и й Н.Н., П о п о в Е.А. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник. - Л.: Машиностроение, 1987.

28. Ш и л о В.Л. Популярные цифровые микросхемы: Справочник. - М.: Радио и связь, 1987.

29. Ц и ф р о в ы е и а н а л о г о в ы е и н т е г р а л ь н ы е м и к р о с х е м ы : С п р а в о ч н и к / П о д р е д . С . В . Я к у б о в с к о г о . - М . : Р а д и о и с в я з ь , 1990.

30. Применение интегральных микросхем в электронной вычислительной технике: Справочник / Под ред. Б.Н. Файзулаева, Б.В. Тарабрина. - М.: Радио и связь, 1987.

31. П у х а л ь с к и й Г.И., Н о в о с е л ь ц е в а Т.Я. Проектирование дискретных устройств на интегральных микросхемах: Справочник. - М.: Радио и связь, 1990.

32. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник / Под ред. В.А. Шахнова. - М.: Радио и связь, 1988.

33. Л е б е д е в О.Н. Микросхемы памяти и их применение. - М.: Радио и связь, 1990.

34. Полупроводниковые БИС запоминающих устройств: Справочник / Под ред. А.Ю. Гордонова и Ю.Н. Дьякова. - М.: Радио и связь, 1990.

35. Большие интегральные схемы запоминающих устройств: Справочник / Под ред. А.Ю. Гордонова и Ю.Н. Дьякова. - М.: Радио и связь, 1990.

36. Система документации единой системы ЭВМ / Под общ. ред. А.М. Ларионова. - М.: Статистика, 1976.

37. У с а т е н к о С.Т., К а ч е н ю к Т.К., Т е р е х о в а М.В. Выполнение электрических схем по ЕСКД: Справочник. - М.: Изд-во стандартов, 1989.

ОГЛАВЛЕНИЕ

I. Требования к выполнению и оформлению курсовой работы	3
I.1. Цель и задачи курсовой работы	3
I.2. Задание на курсовую работу	3
I.3. Порядок выполнения курсовой работы	5
I.4. Оформление курсовой работы	8
I.5. Календарный план выполнения курсовой работы	9
2. Пример построения цифрового устройства	II
2.1. Задание на проектирование	II
2.2. Комбинационная реализация устройства	13
2.3. Регистровая реализация устройства	16
2.4. Построение устройства в виде микропроцессорного вычислителя на БИС КР580ВМ80	27
Рекомендуемая литература	30