

Министерство образования Российской Федерации  
Пензенский государственный университет

**ПРОЕКТИРОВАНИЕ ПРОЦЕССОРА ЭВМ  
НА СЕКЦИОНИРОВАННЫХ  
МИКРОПРОЦЕССОРНЫХ БИС**

*Методические указания  
к курсовому проектированию*

Пенза 2001

---

УДК 681.33,001.63

П 79

Даны указания к курсовому проектированию по дисциплине "Организация ЭВМ и систем", изложены принципы работы и методы проектирования процессоров универсальных ЭВМ на микропроцессорных секционированных БИС серии К1804 с помощью учебной кросс- системы. Издание второе, переработанное и дополненное.

Методические указания подготовлены на кафедре "Вычислительная техника" и предназначены для студентов специальности 22.01.

Ил.9, табл.1, библиогр. 16 назв.

**С о с т а в и т е л и: Е.И.Гурин, Н.Н.Коннов, В.Б.Механов**

**П о д р е д а к ц и е й Н.П. Вашкевича**

**Р е ц е н з е н т: С.В. Печерский**, кандидат технических наук, доцент кафедры "Вычислительные машины и системы" Пензенского технологического института.

---

## **Основные сведения**

При проектировании процессоров ЭВМ могут использоваться микропроцессоры двух типов: однокристалльные микропроцессоры и секционированные микропроцессорные БИС. Микропроцессоры первого типа имеют свою неизменяемую систему команд и позволяют проектировать процессоры с малой и средней производительностью. Принципы построения процессоров на микропроцессорах этого типа достаточно широко описаны в литературе.

В микропроцессорах второго типа проектирование ведётся на микропрограммном уровне, что позволяет реализовать любой набор команд, а также при необходимости видоизменять его. Они обеспечивают большую гибкость в достижении нужных характеристик процессора, позволяют наращивать разрядность обрабатываемых слов. Использование микропроцессоров с разрядно-модульной организацией позволяет создавать процессоры, обладающие высоким быстродействием. В то же время принципы построения процессоров на микропроцессорах этого типа недостаточно подробно описаны в литературе.

---

## Задание на курсовое проектирование

Целью курсового проектирования является:

- изучение принципов работы и методов проектирования процессоров;
- изучение микропроцессорных БИС конкретных серий и выработка навыков практического проектирования микропроцессорных систем.

Содержанием курсового проектирования является разработка центрального процессора универсальной ЭВМ на схемотехнической базе микропроцессорного комплекта серии K1804.

Технические характеристики проектируемого процессора:

- разрядность - 16;
- адресное пространство - 32К слов;
- формат данных - шестнадцатиразрядные целые числа, представленные в дополнительном коде;
- система команд - программная совместимость с ЭВМ типа СМЗ ("Электроника- 60"), кроме команд обработки данных в байтовом формате;
- система элементов - серии K1804, K556, K53Г, K555;
- интерфейс типа "Общая шина" или МПИ.

Примечание:

- 1)возможна по заданию преподавателя реализация системы команд, отличной от "Электроники - 60";
- 2)конкретный состав реализуемого набора команд и способов адресации к памяти уточняется преподавателем;
- 3)такие характеристики процессора, как быстродействие, потребляемая мощность, устанавливаются проектантом.

При выполнении проекта необходимо отработать структурную схему процессора; функциональную организацию его блоков; форматы микрокоманд; схемы алгоритмов выполнения заданного набора команд и диаграммы микропрограммной логики, реализующие определенную часть разработанного алгоритма; таблицы "прошивки" памяти микропрограмм; принципиальные схемы одного из блоков; расчет времени выполнения отдельных операций в процессоре.

---

## **Этапы и порядок выполнения проекта**

Разработка процессора ведется в следующем порядке:

1. Изучается состав программно-доступных регистров реализуемого процессора, форматы и содержание заданного набора команд, особенности адресации к памяти. Для этого необходимо использовать литературу, где описаны система команд и архитектура мини - и микроЭВМ (СМЗ, "Электроника- 60" и т.п.) [1,2,3. 9,14].

2. Изучается архитектура базовой микроЭВМ на БИС серии K1804, на которой эмулируется заданная система команд, уточняются формат микрокоманд, принципы работы БИС. Для этого следует использовать литературу [5(кн.1),6,8,12,16]. Определяется состав регистров базовой микроЭВМ, используемых в качестве программно-доступных, регистров эмулируемой ЭВМ, назначаются буферные и вспомогательные регистры.

3. Разрабатывается алгоритм выполнения операций в процессоре, для чего сначала составляется укрупненная схема алгоритма, позволяющая учитывать последовательность основных фаз исполнения команд (выборка команды, вычисление адресов операндов, выборка операндов и т.д.). Укрупненная схема алгоритма выполняется для полного набора операций (всей системы команд) проектируемого процессора. Разрабатывается схема алгоритма на уровне межрегистровых передач для заданного набора команд и способов адресации. При этом рекомендуется повторявшиеся операции выделять в виде подпрограмм с последующей проработкой каждой подпрограммы.

4. По разработанной схеме алгоритма производится составление диаграмм микропрограммной логики, показывающих, как конкретно описанные последовательности микроопераций реализуются в среде базовой микроЭВМ. Диаграммы служат исходным материалом для кодирования микрокоманд и составления таблицы "прошивки" памяти микропрограмм.

5. Кодированная микропрограмма составляется и отлаживается с помощью учебной кросс-системы проектирования микропроцессорных устройств. Отладка микропрограммы должна выполняться на тестовых комбинациях данных в соответствии с рекомендациями.

---

6. Разрабатываются структурная схема процессора, функциональные схемы его блоков и принципиальная схема одного из блоков или его части, конструктивно реализуемой как типовой элемент замены (ТЭЗ). При разработке схемной документации можно пользоваться ГОСТом, приведенным в [13].

7. Анализируется быстродействие спроектированной схемы. Рассчитываются длительность машинного такта и время выполнения каждой реализуемой команды. Методика расчёта быстродействия микроЭВМ изложена в [5,7,12], исходные данные по задержкам сигналов в интегральных схемах приведены в [11, 12, 16].

8. Оформляется отчетная техническая документация по курсовому проекту.

Подробные рекомендации по выполнению отдельных из перечисленных этапов приведены ниже.

### **Описание структуры базовой микроЭВМ**

Для реализации процессора с заданной системой команд предлагается использовать в качестве базовой микроЭВМ с разрядно модульной организацией на БИС серии K1804, построенную по типовой схеме, рекомендованной в [5, 6]. Программная модель такой микро-ЭВМ включена в учебную кросс-систему проектирования микропроцессорных устройств. В состав процессора базовой микроЭВМ входят следующие блоки (рис.1): блок микропрограммного управления (БМУ), блок обработки данных (БОД) и интерфейсный блок (ИБ).

БМУ служит для генерации кодов микрокоманд, содержащих поля M1,...M15 и управляющих работой БОД и ИБ. БОД осуществляет обработку информации под управлением полей M1, M5,...M14, вырабатываемых в БМУ кодов микрокоманд. По результатам выполнения операций в БОД формируется слово состояния процессора PSW, поступавшее в БМУ. Информация в БМУ и БОД поступает из ИБ по шине данных VI, а с выхода БОД в ИБ поступают данные по шине ВО и адрес по шине ADR. Поле M15, код которого поступает из БМУ в ИБ, служит для управления вводом-выводом, при этом осуществляется анализ сигналов FLG, вырабатываемых в ИБ.

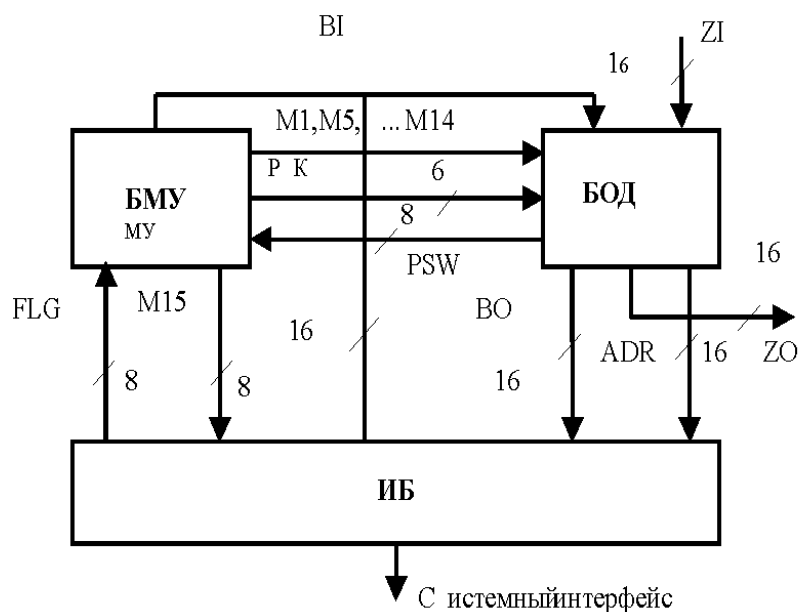


Рис.1. Структура базовой микроЭВМ.

Функциональная схема БОД приведена на рис.2. В состав БОД входят обрабатывающий узел (ОУ), вентили В1 и В2, входной регистр данных RDI, схема формирования переноса (СФП) - C0 в младший разряд, регистр состояния PSW, выходной регистр, данных RDO, регистр адреса MAR, мультиплексоры, адреса MSA и MSB, дешифратор DC. Обрабатывающий узел выполнен на микросхемах K1804BCI и может выполнять все арифметические и логические операции, предусмотренные в данной микросхеме. Выполняемая операция задается девятиразрядным кодом 18-10, который образуется кодами M9, M10, M11. Адреса A и B регистров во внутреннем регистровом ЗУ операционного блока ОУ задаются соответственно кодами M5 и M6, при этом на входы A и B могут поступать как коды M5 и M6, так и разряды 8...6 или 2...0 регистра команд.

На вход D операционного узла может подаваться информация с вентилей В1, В2 и регистра RDI, подключение которых к шине D управляется кодом M12. С выхода Y информация может заноситься в RDO или в MAR в соответствии с заданным кодом M14.





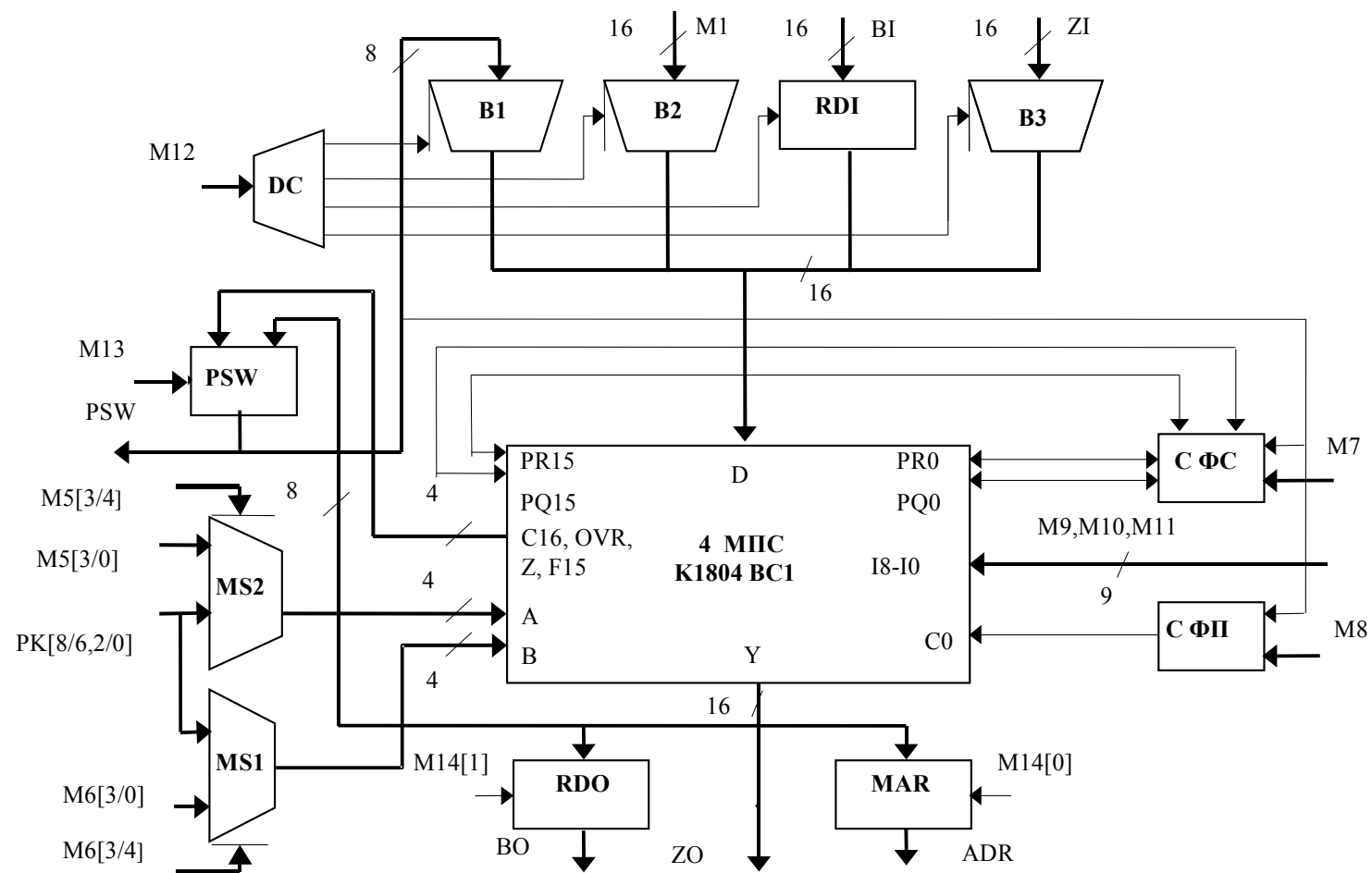


Рис. 2. Функциональная схема БОД



Код M13 управляет занесением информации на PSW, а коды M7 и M8 - соответственно сдвигами и формированием переноса в младший разряд.

На рис. 3 приведена функциональная схема БМУ без аппаратного формирования начального адреса. В состав которого входят регистр команд (РК), схема формирования сигналов уровней команд (СФУР), постоянное запоминающее устройство (ПЗУ), мультиплексор условий (MS), схема управления адресом микрокоманды (СУАМ), схема управления следующим адресом (СУСА), регистр микрокоманд (РМК), счетчик тактов (СТ).

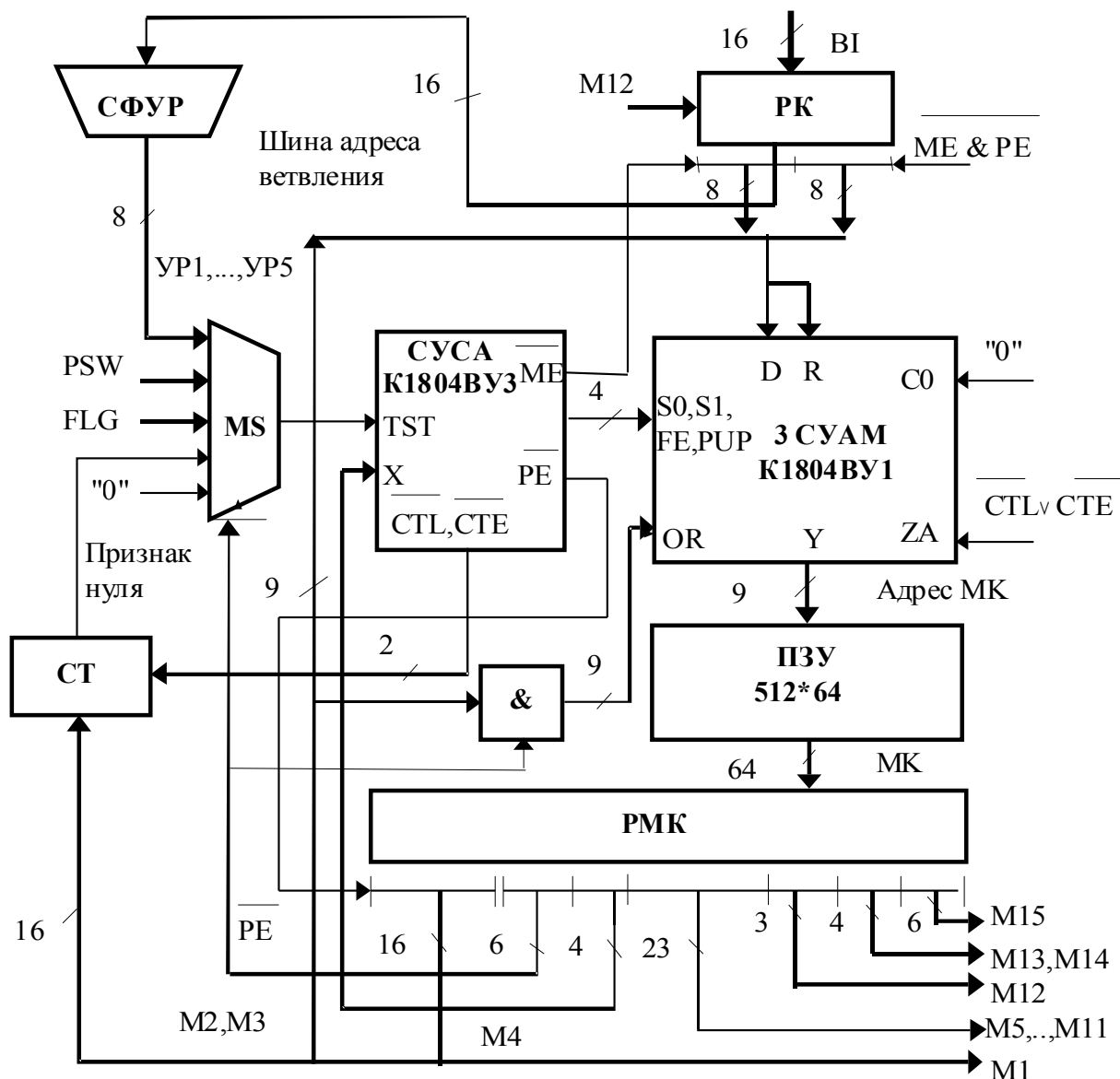


Рис.3. Функциональная схема БМУ

---

РК служит для хранения кода выполняемой команды, СФУР - для выработки сигналов УР1,...УР5, определяющих, к какому уровню относится формат выполняемой команды, MS выбирает нужный сигнал условия из всех поступающих на его вход и под действием кодов М2 и М3 формирует проверяемый сигнал ТST, который передается в СУАМ. СУАМ совместно с СУСА определяют следующий адрес микрокоманды в соответствии с кодом М4, проверяемым условием ТST, а также кодом, приходящим на вход D СУАМ.

На вход D СУАМ могут поступать коды М1, младшие или старшие разряды РК. Подключением того или иного кода на шину D управляют сигналы ME и PE. Если ME = 1, PE = 0, то к шине адреса ветвления подключается поле М1, если ME = 0, PE = 1, то подключаются старшие разряды РК; если ME = 1, PE = 1, то подключаются младшие разряды РК. Сигналы CTL, CTE управляют работой счетчика, а также при CTE=CTL=0 задается нулевой адрес на выходе Y схемы СУАМ (переход к микрокоманде с нулевым адресом).

БМУ работает следующим образом. С выхода ПЗУ коды микрокоманд записываются на РМК, где хранятся в течение одного такта. Каждая микрокоманда разбивается на 15 полей, назначение которых подробно будет описано ниже. Поля М1...М4 определяют следующий адрес, а поля М5...М15 задают микрооперацию. Поля М2 и М3 управляют мультиплексором условий, а поле М4 определяет тип перехода к следующей микрокоманде. В течение очередного такта производятся следующие действия: по коду М4 определяется тип перехода, а по кодам М2, М3 - значение сигнала ТST; в СУСА формируются сигналы, управляющие схемой СУАМ, на выходе Y которой формируется адрес следующей микрокоманды; код следующей микрокоманды вырабатывается на выходе ПЗУ. Этот код будет занесен на РМК в начале следующего такта.

Особенностью данной схемы является то, что все команды разбиты на уровни в соответствии с [4]. Поэтому в микропрограмме необходимо предусмотреть определение, к какому уровню относится данная команда.

## Разработка алгоритма функционирования процессора

Разработке алгоритма работы процессора, эмулирующего какую-либо систему команд, должна предшествовать процедура установления соответствия между программно-доступной аппаратурой базовой и эмулируемой ЭВМ. Например, при эмуляции системы команд ЭВМ типа PDP-11 ("Электроника-60") на микропроцессоре K1804 необходимо установить однозначное соответствие между регистрами общего назначения (РОН) и оговорить правила формирования и хранения признаков в слове состояния процессора. ЭВМ "Электроника-60" имеет восемь программно-доступных РОНов, два из которых имеют целевое назначение: R6 - указатель стека, R7 - счетчик команд. Микропроцессор K1804 имеет 16 РОНов той же разрядности, что и "Электроника-60". "Электроника-60" имеет 16-разрядный регистр состояния процессора (РСП), тогда как K1804 имеет 8-разрядный РСП. На основании этой информации разработчик может, например, установить следующее соответствие:

<b>PDP-11</b>		<b>K1804</b>
R0	<--	R0
R1	<--	R1
R2	<--	R2
R3	<--	R3
R4	<--	R4
R5	<--	R5
R6	<--	R6
R7	<--	R7
PSW	<--	R15

Остальные регистры БИС K1804BC1 (R8-R14) могут быть использованы в этом случае для хранения промежуточных результатов.

Укрупненный алгоритм работы процессора показан на рис.4. Этап начальной инициализации заключается в формировании конкретной конфигурации вычислительной системы, подключении и переводе в активный режим всех устройств, обеспечивающих работу процессора (ввод-вывод данных, связь с оператором и т.п.).

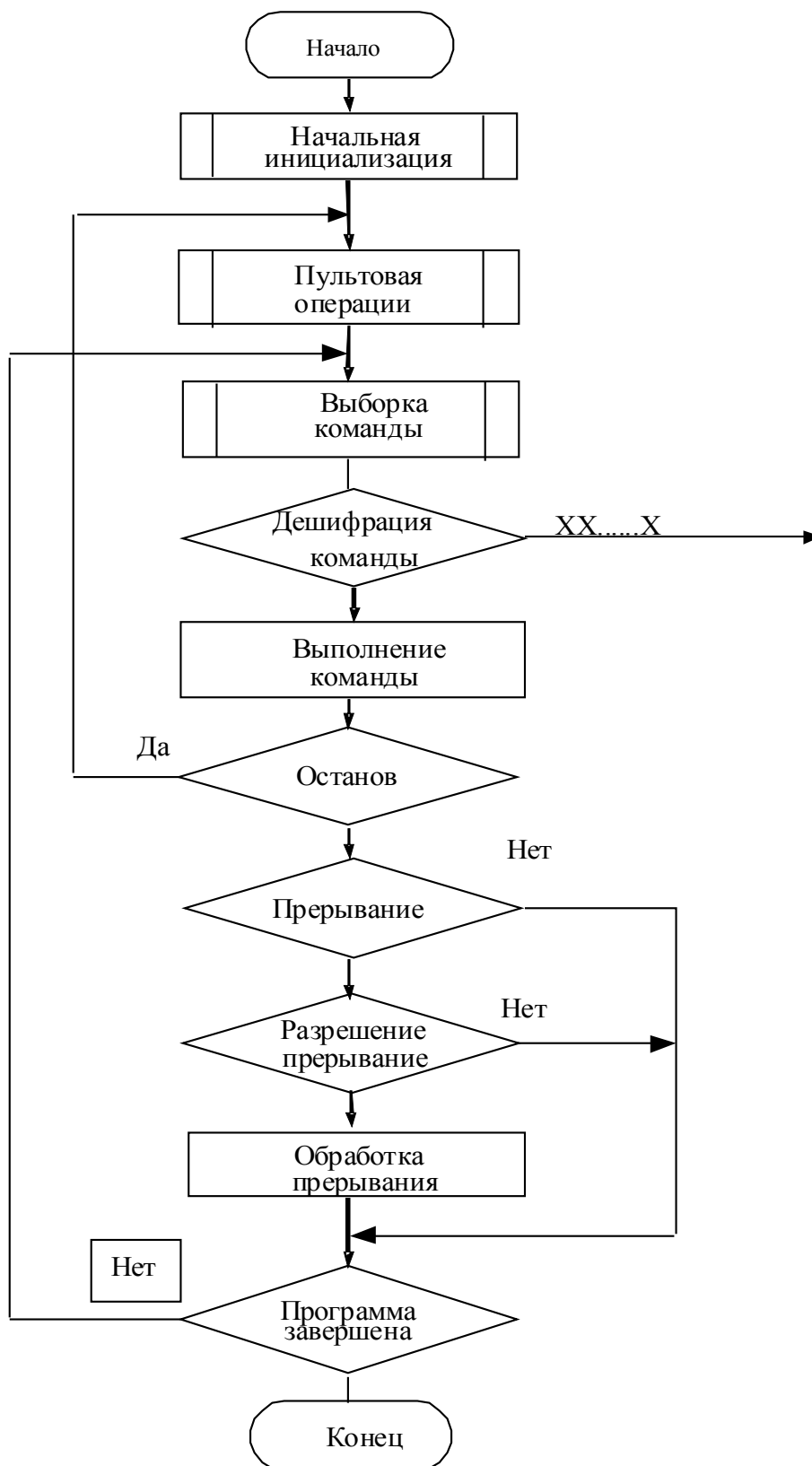


Рис.4.Укрупненный алгоритм работы процессора

---

Пультовые операции позволяют выполнить установку начальных значений регистров процессора, определить режимы работы ЭВМ, реализовать диалог оператора с ЭВМ.

Указанные два этапа являются стандартными и их разработка не входит в задачи курсового проекта.

Выборка команды представляет собой последовательность действий по пересылке адреса команды со счетчика команд на регистр адреса памяти, обращения к ЗУ, приема кода команды из ЗУ в регистр команд и формирования адреса следующей команды.

Дешифрация команды представляет собой, фактически, две процедуры: первичная и вторичная дешифрации. Первичная дешифрация предполагает определение принадлежности команды к какой-либо группе в эмулируемой системе команд с дальнейшей выборкой операндов. Вторичная дешифрация позволяет окончательно определить команду, находящуюся в данный момент в регистре команд, с дальнейшим выполнением действий, предписанных данной командой.

Дешифрация и выполнение команды обязательно учитывают специфику реализуемой системы команд и особенности базовой ЭВМ. Рассмотрим эти этапы на примере реализации системы команд ЭВМ типа "Электроника - 60". 2.14

В соответствии с [4] всю систему команд эмулируемой ЭВМ можно разбить на следующие подгруппы: команды пересылок, арифметические и логические операции, команды вызова и возврата из подпрограмм, команды работы с флагами, прочие команды. Команды кодируются по схеме, показанной на рис.5.

Имеется пять основных уровней кодов. Уровень 1 задается трехбитовым полем (биты 14,13 и 12); уровень 2 - битом 11 кода команды; уровень 3 - четырехбитовым полем (биты 15,10,9 и 8); уровень 4 - двухбитовым полем (биты 7 и 6); уровень 5 - трехбитовым полем (биты 2, 1 и 0).

Дешифрация выполняется последовательно, начиная с уровня 1. Признаком перехода с одного уровня на другой является нулевая комбинация разрядов в соответствующем поле.

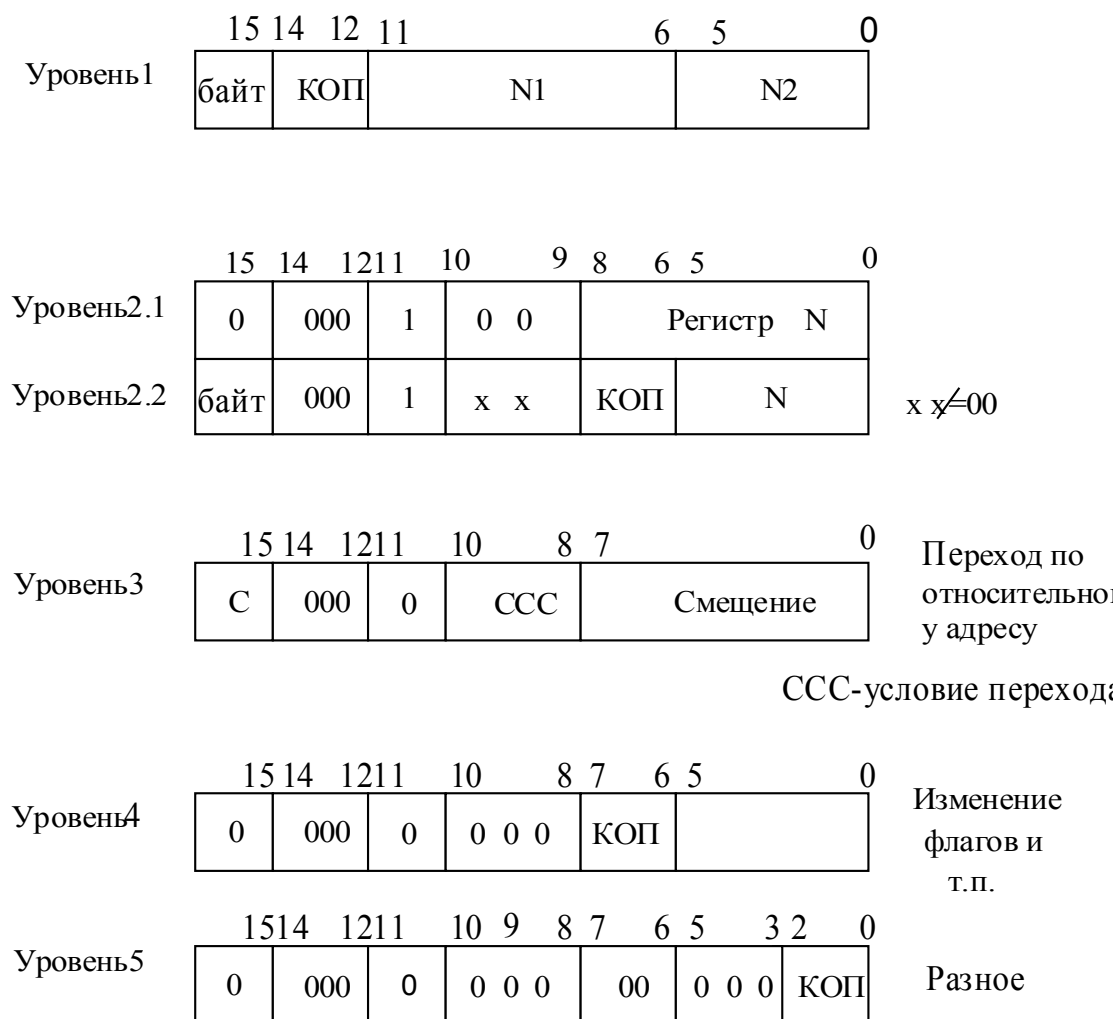


Рис.5. Кодировка команд микро-ЭВМ "Электроника-60

Команды **уровня 1** - двухадресные. В большинстве из них бит 15 указывает, на какую границу настроен адрес (слова или 8-битового байта в слове). Исключением для уровня 1 является команда SOB, предназначенная для организации циклов. Процедура выборки операндов команды SOB отличается от выборки операндов всех остальных команд уровня 1, поэтому разработчику необходимо предусмотреть на этапе первичной дешифрации выделение этой команды с целью ее выполнения по отдельному алгоритму.



---

Команды **уровня 2**, в свою очередь, подразделяются на два подуровня. Команды уровня 2.1 имеют формат "полтора адреса" например, команда JSR), т.е. под адрес первого операнда в формате отводится всего три разряда (для задания номера регистра), а для адреса второго операнда выделяется, как обычно, шесть разрядов. Команды уровня 2.2 - одноадресные команды арифметико-логической группы.

**Уровень 3** - это 15 команд условного перехода.

**Уровень 4** - команды работы с флагами и возврата из подпрограммы.

**Уровень 5** - команды общего назначения.

Первичная дешифрация команд в кросс-системе K1804 (см. Приложение 2) выполняется в соответствии с рассмотренной системой кодирования.

По завершении первичной дешифрации кода команды выполняется дешифрация способов адресации, выборка операндов и фиксация их в отведенных регистрах базовой ЭВМ. Затем происходит выполнение команды и формирование всех необходимых признаков результата по правилам эмулируемой ЭВМ.

На рис.6 показан пример выполнения команды DEC, которая относится к одноадресным командам уровня 2.2. Считается, что код команды фиксируется в регистре команд РК. Регистр R7 выполняет функции счетчика команд. Регистр RN является регистром общего назначения, содержимое которого является операндом.

Дешифрация способа адресации реализуется в подпрограммах выборки операнда и записи результата.

Заполнение любой команда завершается стандартной процедурой анализа состояния системы, которая заключается в последовательном просмотре трех сигналов: останова (аппаратного или программного), прерывания (аппаратного или программного) и разрешения прерывания (слова состояния текущей программы). При этом завершение программы фактически не анализируется, а выполняется по команде останова.

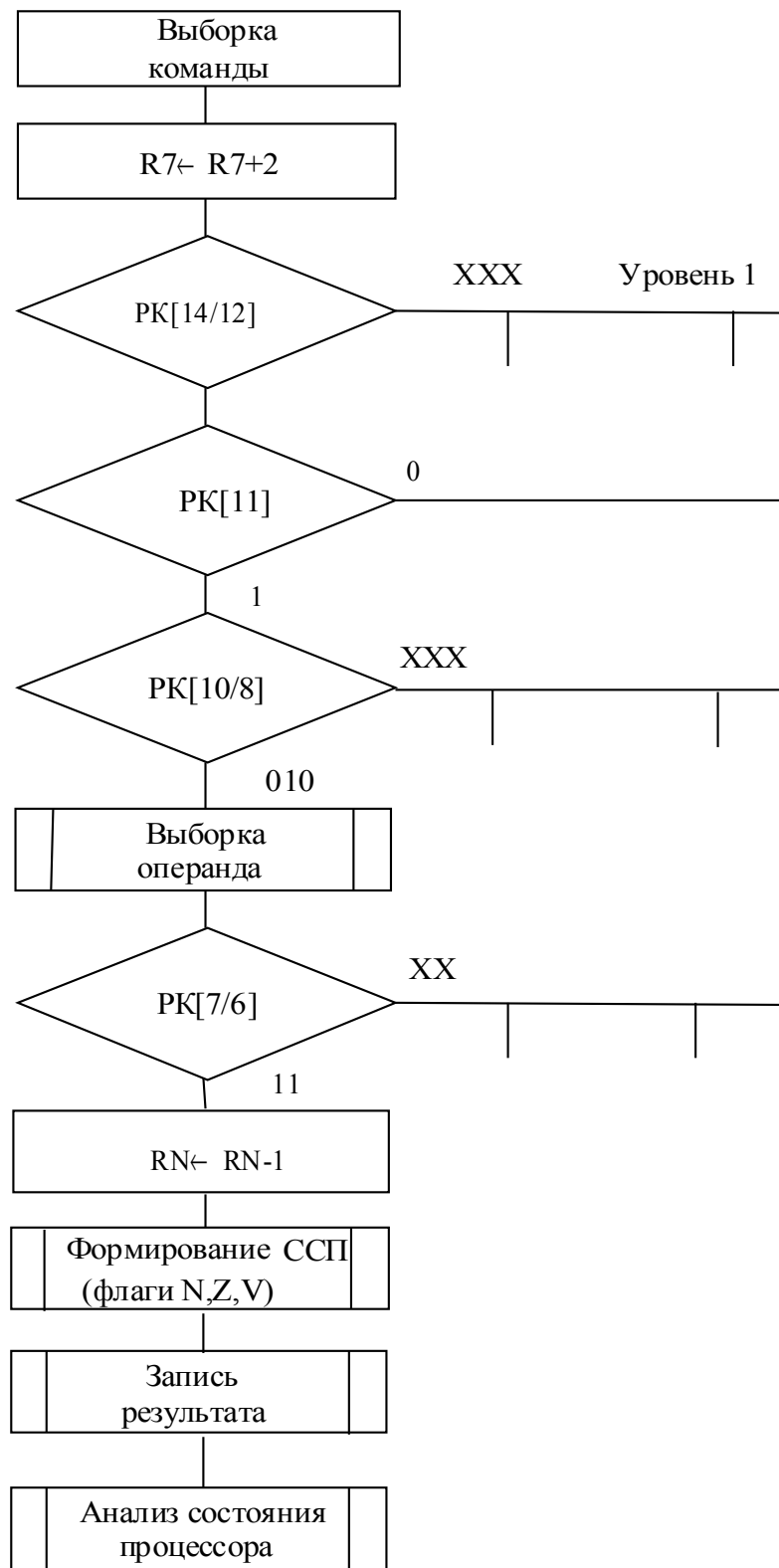


Рис.6. Алгоритм выполнения команды DEC

## Составление диаграмм микропрограммной логики

Составление кодированной микропрограммы работы процессора является весьма трудоемкой процедурой из-за значительной длины микрокоманды и сложности размещения микрокоманд в памяти микропрограмм. Облегчить кодирование микропрограмм (см. прил.2), а также их отладку позволяет составление диаграмм микропрограммной логики (ДМЛ).

ДМЛ представляет собой соединение блоков, каждый из которых является подробным описанием одной микрокоманды. Правило заполнения блока показано на рис.7.

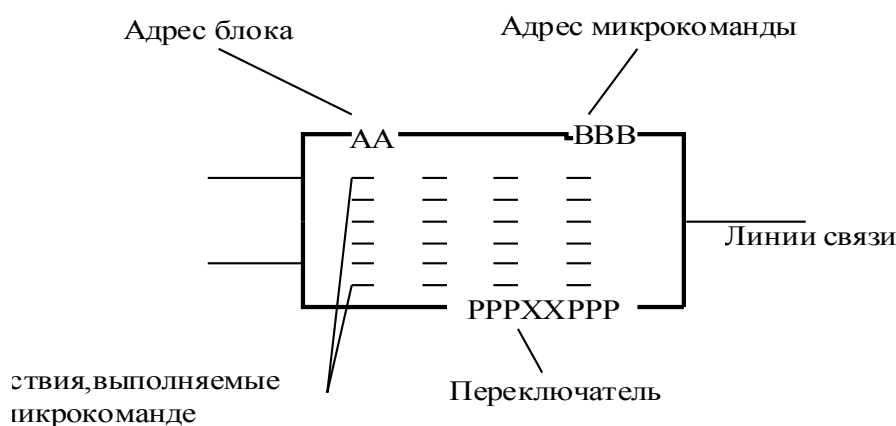


Рис. 7. Формат блока ДМЛ

Внутри блока заполняется 6 строк в соответствии с таблицей

Содержание строки	Пример Заполнения	Поля МК, задающие действия.
Описание действий над входной шиной данных в БОД	$D \leftarrow RDI; RDI \leftarrow BI$	M12
Описание операций над данными в БОД	$R0 \leftarrow (R0 + D)$	M5, M6, M7, M8, M9, M10, M11
Описание формирования PSW	$PSW \leftarrow C16, F15, Z, OVR$	M13
Описание действий над выходной шиной данных в БОД	$RDO \leftarrow Y$	M14
Описание перехода и условий перехода (формирования сигнала TST) в БМУ	$CJP; TST = PSW[0]$	M2, M3, M4
Описание полей констант и пользователя	7777;0	M1, M15

---

На периферии блочного символа в углах располагаются три адресных поля:

- адрес блока на листе;
- адрес микрокоманды, описываемой блоком;
- переключатель.

Поле адреса блока указывает координату на листе ДМЛ, в которой размещается блок. Адрес микрокоманды дается в восьмеричном представлении. Поле переключателя заполняется, если адрес следующей микрокоманды формируется по внешнему коду (переход типа JMAP или CJV) и представляет собой двоичный код адреса, в котором разряды, формируемые внешним кодом, отмечены знаком X.

Линии связи определяют последовательность исполнения микрокоманд (поток информации показывается слева направо и сверху вниз). Блок может иметь несколько входов и только один выход.

При ветвлениях по сигналу TST на линиях связи указывается значение сигнала TST (0 или 1), соответствующее разветвлению. При переходах по внешнему коду (переключателю) на линиях связи указываются двоичные значения разрядов, отмеченных знаком X в переключателе.

Над блоком можно давать краткий комментарий. Комментарий, относящийся ко всей диаграмме, записывается на свободном месте листа. При выполнении ДМЛ на нескольких листах все линии перехода с листа на лист заканчиваются или начинаются на правой и левой сторонах листа. На линиях указываются координаты блока, к которому и от которого идет линия.

Начало и конец диаграммы отмечаются "пустыми" блоками, адресные поля которых, кроме первого, не заполняются. Внутри блоков записываются слова "начало" и "конец". Подпрограмму ДМЛ обычно рекомендуется отмечать блоком, расшифровывающим ее работу. Содержание микропрограммы отображается на отдельной ДМЛ (рис.8).

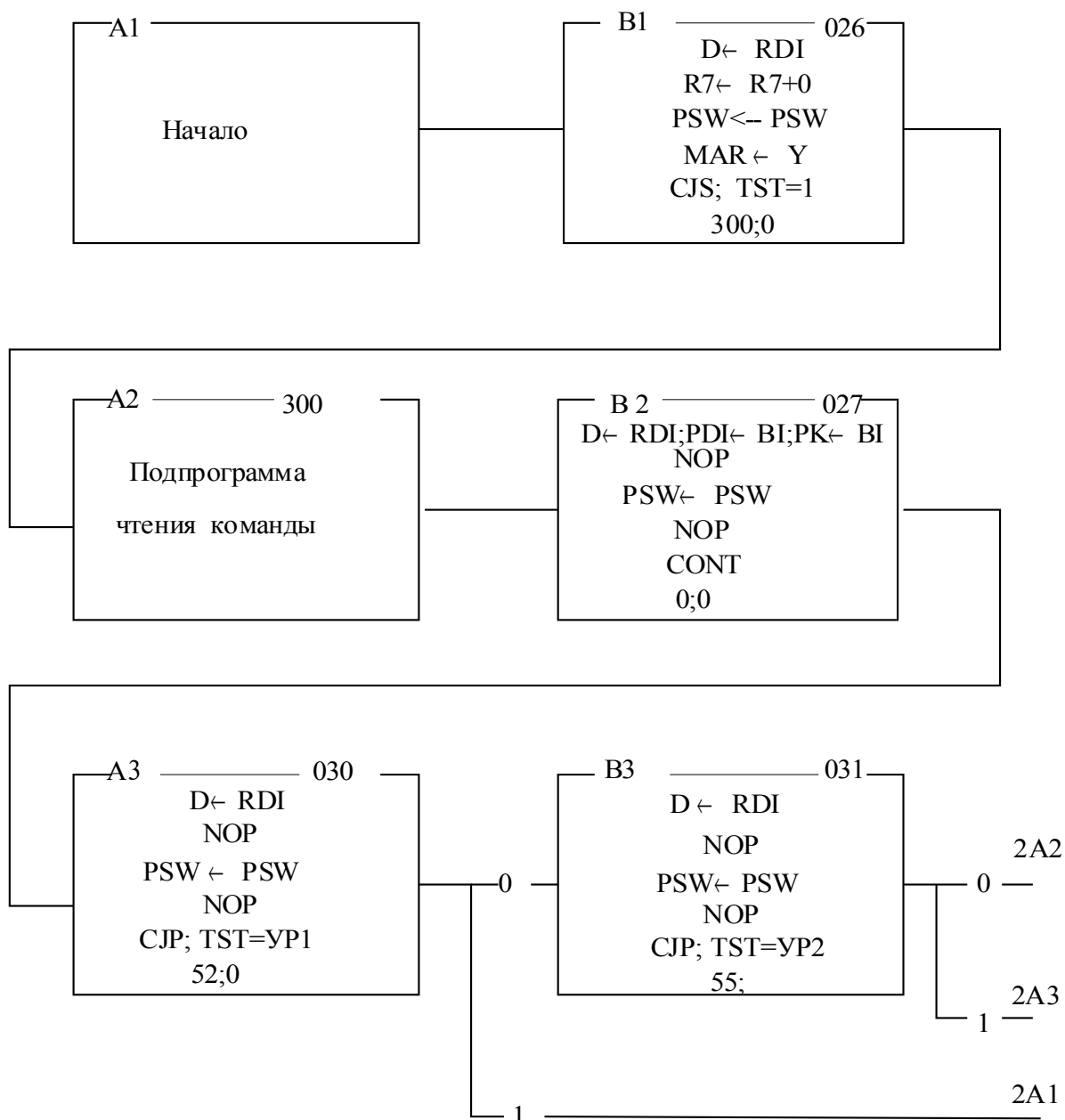


Рис. 8. Фрагмент ДМЛ

## Кодирование микропрограмм

На этом этапе важным моментом является определение начальных адресов участков микропрограмм для различных команд проектируемой ЭВМ. Для БМУ с аппаратным формированием начального адреса эти адреса определяются правилом работы соответствующего преобразователя ПНА. Для БМУ без аппаратного формирования начального адреса, приведенного

---

на рис.3, начальные адреса определяются по кодам команд проектируемой ЭВМ. Рассмотрим этот процесс более подробно.

При выполнении алгоритма в БМУ, приведенном на рис. 3, сначала определяется уровень команды, при этом ветвление производится с помощью команд условного перехода с использованием в качестве условий сигналов, поступающих с выхода схемы формирования уровней СФУР. После разделения команд на 6 уровней в соответствии с рис.5 производится переход по коду команды внутри данного уровня. К шине адреса ветвления (см.рис.3) могут подключаться либо старший, либо младший байты регистра команд РК, поэтому при выборе начальных адресов необходимо следить, чтобы не совпадали начальные адреса различных команд. Для хранения микрокоманд может использоваться ПЗУ на 512 слов, что соответствует девятиразрядному адресу.

Начальные адреса команд уровня 1 (кроме байтовых) определяются кодами 00XXX1111 (X=0 или 1) путем маскирования старшего байта регистра команд маской 000001111 и перехода по старшему байту РК. Например, команда MOV, имеющая в старшем байте код 00001AAAA (А - биты, определяющие адресную часть команды в старшем байте), получает начальный адрес 000011111; команды CMP имеют начальный адрес 000101111 и т.д. Необходимо отметить, что адрес 000001111 останется свободным, так как среди команд первого уровня нет команд, имеющих подобный код.

К уровню 2.1 относится только одна команда, и ее начальный адрес может быть помещен в любую ячейку ПЗУ. К уровню 2.2 относятся одноадресные команды. В этом уровне могут изменяться биты как в старшем, так и младшем байтах кода команды. Разделение по командам в этом уровне осуществляется в два этапа: сначала производится переход по старшему байту РК без маскирования (адреса 000001XXX), а затем в БОД определяются биты 7 и 6 в младшем байте кода команды и производится ветвление с использованием обычных микрокоманд условного перехода. Адреса ячеек, куда производится переход, выбираются из свободных ячеек ПЗУ.

---

Начальные адреса команд уровня 3 определяются кодами 0X0000XXX при переходе на адрес из старшего байта регистра команд.

Начальные адреса команд уровня 4 определяются кодами 1XXXXXXX (маска 100000000) при переходе на адрес по младшему байту регистра команд.

Начальные адреса команд уровня 5 определяются кодами 100000XXX (маска 100000000) при переходе на адрес по младшему байту регистра команд.

Аналогично определяются начальные адреса алгоритмов для различных способов адресации. Начальные адреса первого операнда определяются кодами 11111XXX1 путем маскирования маской 111110001, а начальные адреса второго операнда - кодами 011XXX111 (маска 011000111).

Однако полностью избежать дублирования начальных адресов не удастся. Так, например, при обработке команды 4 уровня SWAB возможно попадание, на адрес 11111111. На этот же адрес будет осуществляться переход и при способе адресации 7 через регистр 7.

В подобных случаях необходимо либо увеличить емкость ПЗУ, либо применять специальные приемы. Например, можно поступить следующим образом: переход на адрес 11111111 в случае обработки команды в свободный регистр БИС K1804BC1 занести 1, а в случае обработки операндов в этот же регистр занести 0, а затем разделить эти варианты с помощью команды условного перехода.

При использовании ПЗУ емкостью 1024 слова всю область памяти удобно разбить на следующие области:

- 0000-0377 - область для переходов по старшему байту регистра команд;
- 0400-0777 - область для переходов по младшему байту регистра команд;
- 1000-1377 - область для переходов по способу адресации первого операнда;
- 1400-1777 - область для переходов по способу адресации второго операнда.

При таком разбиении начальные адреса дублироваться не будут.

---

После определения начальных адресов необходимо выделить участки ПЗУ для общих частей всех команд: выборка команды, ее дешифрация, выборка операндов и т.д.

При составлении микропрограммы необходимо следить за тем, чтобы микропрограмма обработки какой-либо команды не наложилась на начальный адрес следующей команды. Если не хватит ячеек ПЗУ для кодирования какой-либо команды, можно перейти по микрокоманде безусловного перехода в свободную область памяти и там продолжить микропрограмму обработки данной команды.

### **Отладка микропрограмм на кросс-системе**

Учебная кросс-система проектирования (УКС) микропроцессорных устройств на БИС серии K1804 обеспечивает:

- подготовку и редактирование микропрограмм;
- отладку микропрограмм на имитационной модели базовой микроЭВМ;
- документирование результатов моделирования и сохранение микропрограмм на магнитном носителе.

Вызов УКС осуществляется при запуске программы **Gr1804.exe**, после чего на экран дисплея выводится заставка и система переходит к начальному диалогу, во время которого оператор задает ее конфигурацию, а именно: имя задачи, т.е. файла разрабатываемой микропрограммы (до 6 символов), номер варианта конфигурации базовым микроЭВМ, емкость ПЗУ-микрокоманд, необходимость создания файла ПЗУ на магнитном носителе (при первом обращении к системе). Затем система запрашивает "читать файл задачи или нет". По окончании начального диалога на экран выводится символ [0] >, указывающий, что кросс-система готова к работе под управлением команд, вводимых оператором.

Ввод и редактирование микропрограммы осуществляется с помощью команд: **R[Addr]**, **W**, **M[Fild]**, **<,>**, **Z**, **L[T]**, **L[F]**, **L[P]**, **F[L]**, **F[S]**.

Команда **R[Addr]** открывает ячейку ПЗУ с заданным адресом, при этом на экран выводится в двух последовательных строках адрес открытой ячейки и содержимое пятнадцати полей микрокоманды, (все данные здесь и далее представляются в восьмеричной системе счисления), т.е. с помощью этой команды можно прочитать микрокоманду.



---

Для чтения, записи и модификации микрокоманды используется команда **W**, по которой выводится в первой строке разметка адреса и полей микрокоманды, а во второй - выводится адрес открытой ячейки и последовательно вводятся коды всех полей микрокоманды. Для ввода кода поля набирается восьмеричный его код и нажимается клавиша **Enter**, затем маркер дисплея автоматически перемещается в соседнее поле.

Команды **<** и **>** являются комплексными и закрывают ранее открытую ячейку, открывают следующую (с большим или соответственно с меньшим адресом и читают ее содержимое).

Команда **M[Fild]** позволяет модифицировать одно из полей открытой ячейки. При этом вместе с командой **M** необходимо ввести номер ячейки, которую надо исправить.

Команда **Z** очищает ПЗУ, т.е. записывает во все ячейки коды 0. Пользоваться этой командой следует осторожно.

Команда **L** выводит оператору либо на принтер **L[P]**, либо на терминал **L[T]**, либо в файл **L[F]** листинг микропрограммы, записанной в ПЗУ. При этом содержимое ячеек с кодом 0 не выводится. Чтобы указать, на какое устройство выводить листинг, необходимо на запрос кросс-системы ввести буквы: **P** (принтер), **T** (экран) или **F** (файл). В последнем случае листинг выводится в файл с именем, указанным в начале программы без расширения.

Если оператору необходимо прочитать файл в ПЗУ по заданному адресу, то это необходимо сделать с помощью команды **F[L]**.

Для того, чтобы сохранить части ПЗУ в файле, надо ввести **F[S]**. Затем вводить начальный, конечный адреса и имя файла (до восьми знаков), куда запишется ПЗУ. Лучше это выполнять периодически, чтобы обеспечить автоматическую сохранность вводимой микропрограммы при случайном нарушении работы ЭВМ.

Моделирование подготовленной микропрограммы выполняется под управлением команд: **B, G, P, S, D, I** и **O**.

Установка начальных параметров модели выполняется по команде **B**, по которой последовательно вводятся: начальный и конечный адреса выполняемого участка микропрограммы, количество тактов моделирования (десятичное, число, равное ориентировочной длительности выполнения микропрограммы для предотвращения закливания).

---

Далее необходимо произвести установку начальных значений регистров базовой микроЭВМ, для чего последовательно вводятся имена регистров и код их содержимого. Для прекращения настройки модели необходимо нажать клавишу Enter , затем система предлагает оператору выводить на экран не все регистры. Для этого необходимо ввести имена тех регистров , которые оператор хочет увидеть на экране.

Запуск моделирования выполняется командой G, при этом уточняется, куда следует записывать результат моделирования как и в команде L, устройство для вывода протокола моделирования.

Только при выводе в файл кросс-система сохраняет результат с расширением имя\_задачи. doc.

При моделировании выполняется одна за другой микрокоманды, начиная с заданного адреса, до достижения конечного адреса или исполнения заданного при настройке количества тактов моделирования. Окончание моделирования сопровождается выводом на экран соответствующего сообщения.

При желании оператора можно по команде D вывести содержимое любого регистра, а по команде I - содержимое всех регистров" базовой микроЭВМ, командой S можно изменить содержимое любого регистра.

Для продолжения моделирования вводится команда P. Если необходимо задать несколько точек останова, то надо ввести команду O. Далее указывается количество остановов (от 0 до 20). При выполнении микропрограммы возможно отсутствие остановов, но при наличии остановов их адреса последовательно вводятся в систему, причем останов происходит после выполнения микрокоманды, адрес которой определяет этот останов. Команда O позволяет оперативно изменить адреса остановов, не прибегая к команде B.

Команда C позволяет вернуться к начальному диалогу, переопределив имя задачи, номер варианта, емкость ПЗУ.

Команда H вызывает справочник кросс-системы, приведенный в приложении. Работа с кросс-системой должны оканчиваться командой E, по которой микропрограмма сохраняет результаты в файле имя\_ задачи.pzu.

## **Расчёт быстродействия процессора**

---

Расчет быстродействия разрабатываемой схемы должен выполняться на микропрограммном уровне, когда учитывается время выполнения микроопераций.

Оценка быстродействия блока обработки данных (БОД) и блока микропрограммного управления (БМУ) производится отдельно.

Для расчета временных параметров БОД рекомендуется следующая методика.

1. Фиксировать структуру БОД. В качестве примера рассмотрим схему на рис.10, содержащую:

- четыре микропроцессорных секции (МПС) K1804BC1 и схемы ускоренного переноса (СУП) K1804BP1;
- входной регистр данных (RDI) и регистр микрокоманд (PMK);
- выходной регистр данных (RDO) и регистр состояния (PSW);
- мультиплексоры, замыкающие сдвиговые выходы МПС.
- Все регистры построены на микросхеме K1804IP1.

2. Фиксировать набор операций, выполнение которых предусматривается данной структурой. Для рассматриваемой схемы все операции при расчете временных параметров можно разбить следующим образом:

- логические и арифметические (в расчетах для арифметических операций учитывается задержка СУП);
- со сдвигом и без сдвига (в операциях со сдвигом учитываются, задержки мультиплексоров сдвига).

3. Для заданного набора операций определить источники и приемники информации, а также структурные элементы, участвующие в ее обработке. В рассматриваемой схеме источниками информации являются RDI, PMK, регистровое ЗУ (R0,....,R15); приемниками - RDO, PSW, R0,....,R15.

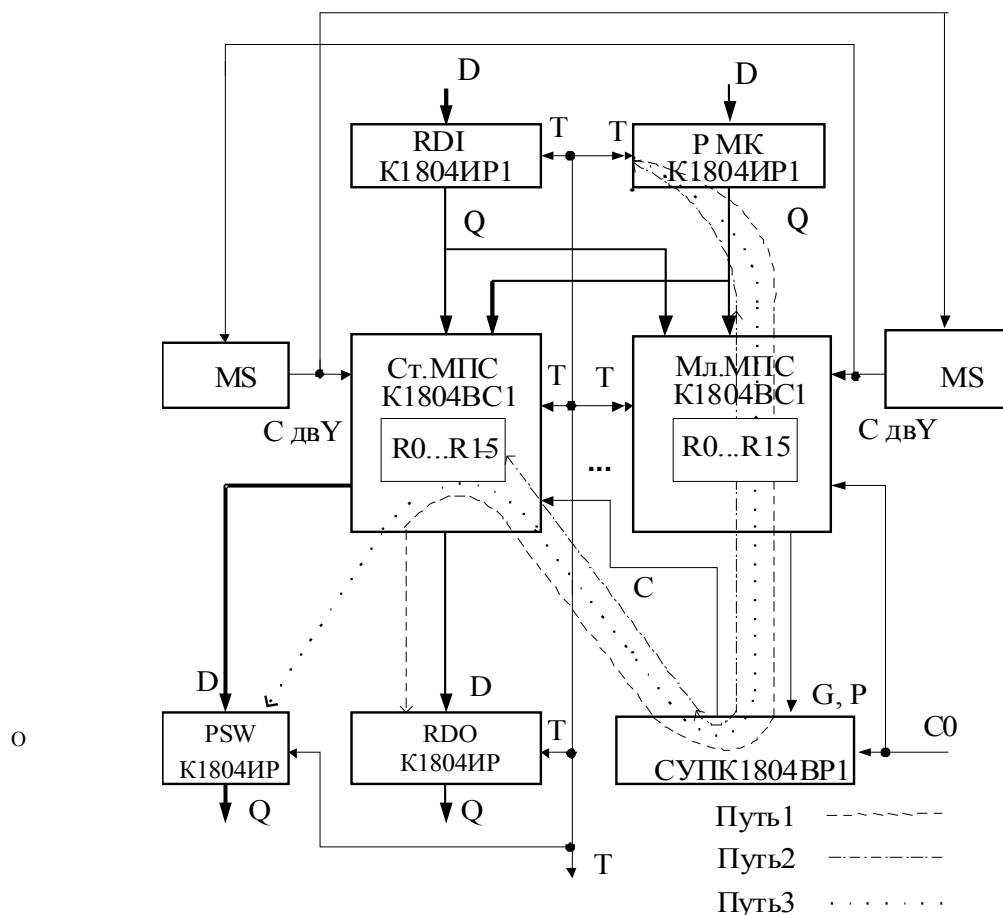


Рис. 10. Пути прохождения операндов в БОД

4. Для каждой группы операций определить пути прохождения информации от источников к приемникам. Для надежной фиксации информации необходимо учитывать время установки сигналов на входах приемников относительно положительного фронта тактового сигнала. Для схемы на рис.10 в общем случае каждая группа операций может использовать девять путей прохождения информации (три источника и три приемника). Однако практически все задержки, происходящие от РМК, больше задержек, происходящих от других источников, так как в этом случае происходит дополнительное обращение к РЗУ по адресным входам. Поэтому для каждой группы операций рассмотрим только три пути прохождения информации, отличающиеся местоположением приемников: RDO, PSW, РЗУ.

5. Рассчитать минимально допустимую длительность операций по каждому пути. Рассмотрим в качестве примера расчет длительности выполнения арифметических операций без сдвига, в которых участвует

---

СУП. Временные параметры микросхем серии K1804 приведены в [7]. Все арифметические операции включают в себя задержку чтения микрокоманды из РМК, определяемую от переднего фронта тактового импульса до формирования адреса РЗУ на входах А, В (11нс), выполнение операции в МПС и формирование сигналов Q, Р ускоренного переноса (59нс), задержку сигнала в СУП (7нс). На первом пути необходимо учитывать распространение переноса от входа С0 старшей МПС. Для надежной записи результата операции в РЗУ установка сигнала на этом входе должна произойти не раньше, чем за 55нс до переднего фронта следующего тактового сигнала. Таким образом, первый путь занимает не менее 132 нс.

На втором пути вместо времени установки сигнала на входе С0 старшей МПС учитывается задержка распространения информации от входа С0 старшей МПС до выхода Y (27нс) и время установки информации на входах RDO (5нс). Длительность второго пути составит 109 нс. На третьем пути при фиксации результата учитываются наибольшее время задержки от входа Е0 старшей МПС до выходов признаков и времени установки информации на входах PSW (5нс). Дольше всех формируется признак Z (46нс). Длительность третьего пути составит 128 нс. Таким образом, критическим является первый путь. Аналогично выполняется подсчет длительности для каждой группы операций.

6. Полученный набор длительностей операций использовать при расчете оптимальных длительностей такта работы БОД. Для рассмотренного примера частоту генератора можно задать примерно равной 7,2 МГц. Приведенная методика полностью применима и для оценки быстродействия БМУ. При этом операции БМУ можно классифицировать по типам выполняемых переходов, по способам формирования адреса следующей микрокоманды, выделить операции работы со стеком, подпрограммами и т.д. Определив длительность машинного такта, можно рассчитать время выполнения каждой из команд, подсчитав наибольшую длину последовательности микрокоманд, их реализующих и умножить ее на рассчитанную длительность машинного такта.

## **Оформление отчетной документации**

Отчетной документацией по курсовому проектированию являются пояснительная записка и схемная документация.

---

Пояснительная записка объемом 30-40 страниц выполняется на белой бумаге стандартного формата, рисунки можно выполнять на миллиметровой бумаге черным карандашом или чернилами. Все листы записки должны иметь сквозную нумерацию.

Пояснительная записка должна содержать:

- титульный лист;
- задание на курсовой проект, подписанное руководителем проекта и его исполнителем и утвержденное заведующим кафедрой;
- содержание с указанием страниц;
- раздел 1 с описанием форматов команд и данных проектируемого процессора;
- раздел 2 с описанием структуры базовой микроЭВМ и ее отдельных блоков и формата микрокоманд;
- раздел 3 с описанием алгоритма работы процессора при выполнении заданного набора команд;
- раздел 4 с листингом прошивки памяти микрокоманд и протоколом отладки микропрограмм;
- раздел 5 с описанием принципиальной схемы разработанного ТЭЗа;
- раздел 6 с расчетами временных соотношений сигналов процессора и времени выполнения отдельных операций;
- перечень использованной литературы.

Графическая часть проекта должна содержать:

- схему структурную электрическую процессора (1 лист формата А3);
- схемы функциональные электрические блоков (3-4 листа формата А3);
- схему принципиальную электрическую (1-2 листа формата А2);
- схемы алгоритмов (4-5 листов формата А3);
- диаграммы микропрограммной логики (4-5 листов формата А3).

Структурные электрические схемы должны раскрывать организацию процессора на уровне блоков и выполняются в соответствии с ГОСТ 2.708-81. Функциональные электрические схемы должны раскрывать функциональную организацию блоков на уровне функциональных групп (узлов) и элементов, условные графические обозначения которых должны

---

соответствовать ГОСТ 2.708-81. Рекомендуемыми форматами для структурных и функциональных схем являются А3 и А4.

Схемы алгоритмов должны оформляться в соответствии с ГОСТ 19.002-80. Рекомендуемым форматом для них является А3.

При оформлении принципиальных схем необходимо руководствоваться ГОСТ 2.708-81 "Правила выполнения электрических схем цифровой вычислительной техники" и ГОСТ 2.743-82 "Обозначения условные графические в схемах. Элементы цифровой техники" ЕСКД.

Схемы электрические принципиальные рекомендуется изображать на листах формата А2 с нанесенной на них координатной сеткой, шаг которой составляет 5 мм.

На каждый ТЭЗ оформляется своя принципиальная схема. Первый лист схемы должен иметь большой штамп (185 x 55 мм), остальные листы — малый (185 x 15 мм). Помимо условно-графического обозначения элементов и связей между ними схема электрическая принципиальная должна показывать схему подключения питающих напряжений к ТЭЗу с указанием соответствующих контактов разъема и отражать подводу питающих напряжений к отдельным микросхемам, а также схему подключения фильтрующих конденсаторов для подавления помех по питанию. Номиналы фильтрующих конденсаторов рекомендуется выбирать из расчета 0,07-0,1 мкФ на корпус микросхемы. Условные графические обозначения для БИС серии К1804 приведены в литературе [12].

На принципиальной схеме в соответствии с требованиями ГОСТ 2.702.-75 помещают перечень элементов, где однозначно должны быть определены все элементы, входящие в состав ТЭЗа и изображенные на схеме.

Рекомендуется перечень элементов изображать на листах формата А4 и записи в нем размещать в алфавитном порядке.

Примеры выполнения схемной документации приведены в [13].

СХЕМЫ БИС МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА К1804

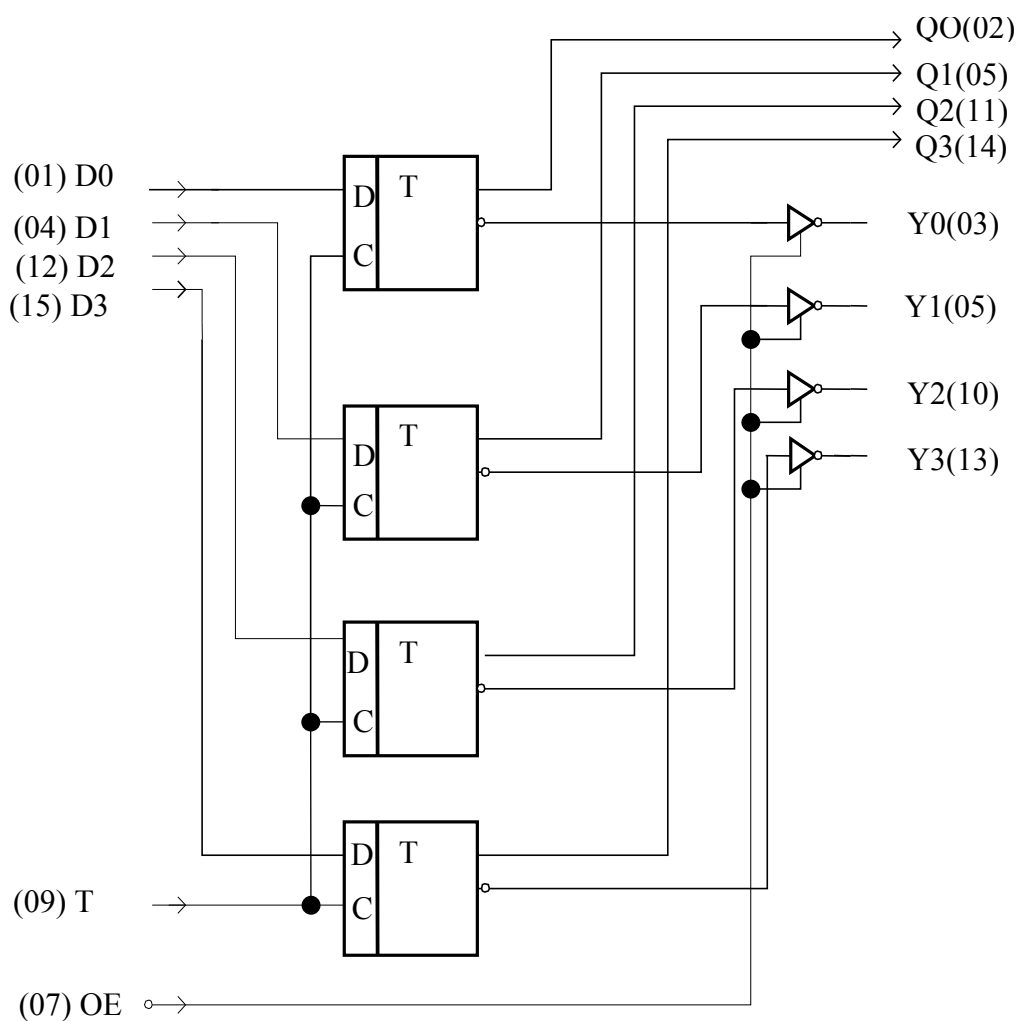


Рис. 1.П Универсальный регистр К1804ИР1



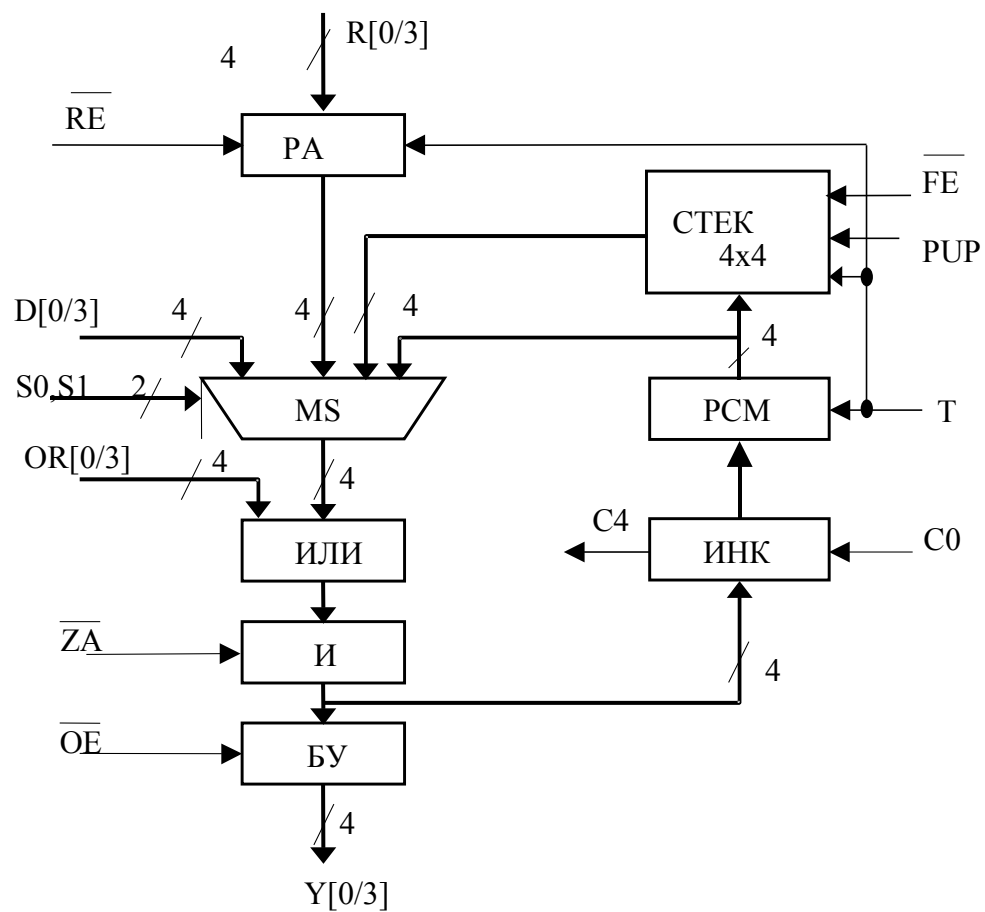


Рис. 2.П Схема управления адресом микрокоманды K1804BC1

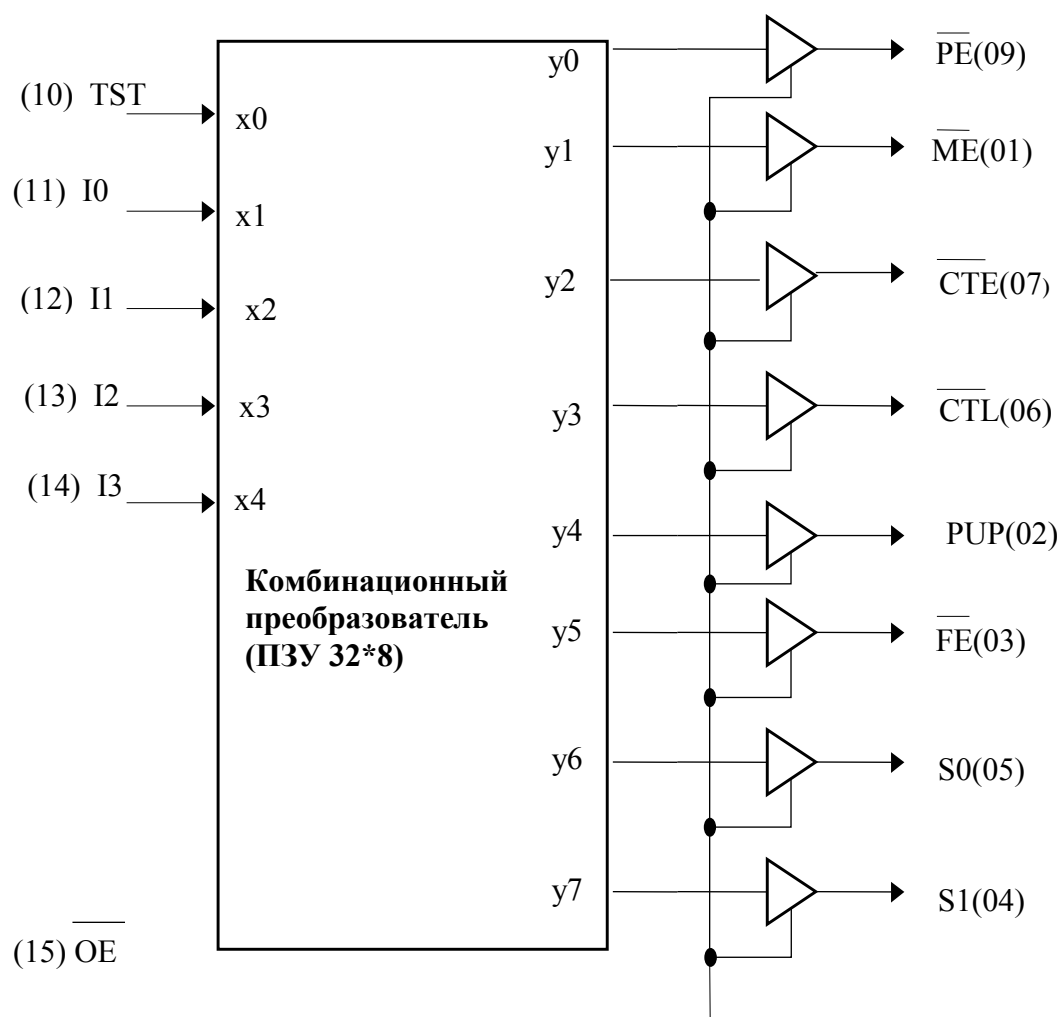


Рис. 3.П. Схема управления следующим адресом К1804ВУ3

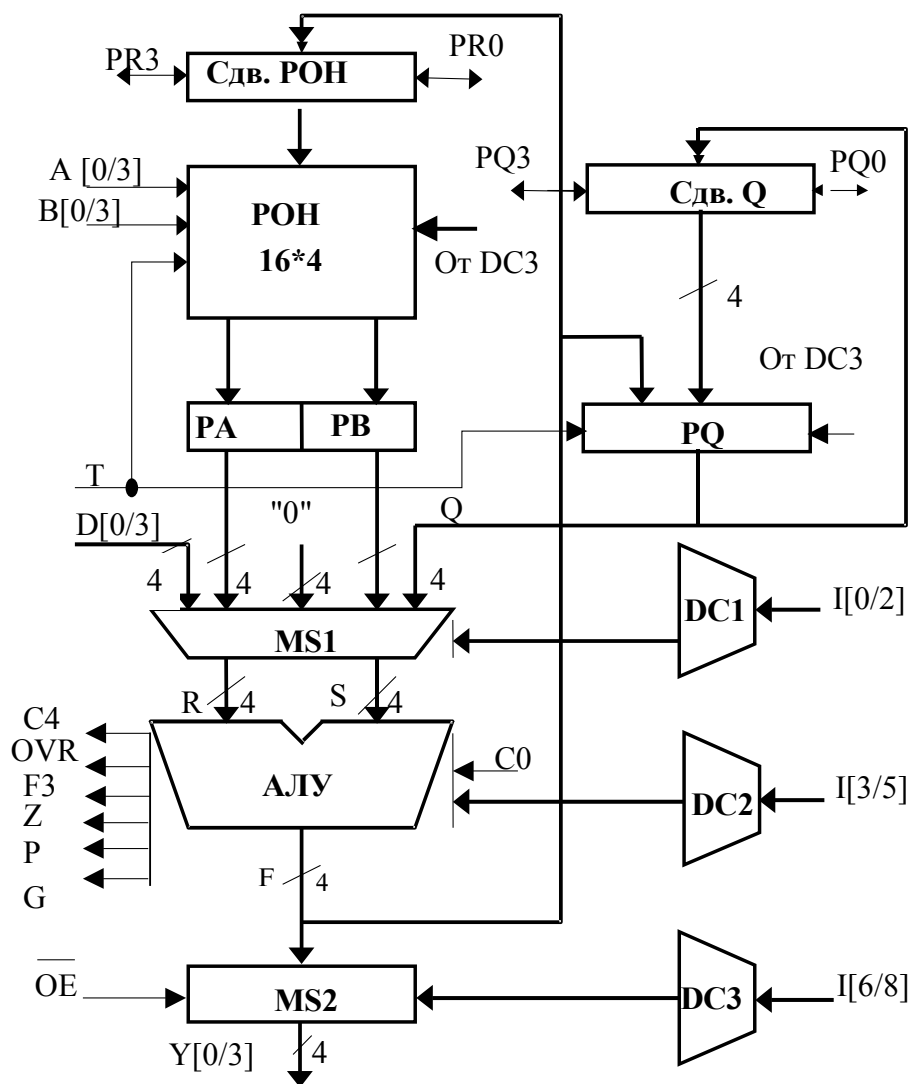


Рис. 4.П. Микропроцессорная секция К1804ВС1

## СПРАВОЧНЫЕ МАТЕРИАЛЫ УЧЕБНОЙ КРОСС-СИСТЕМЫ

Команды кросс-системы	
R[Addr]	- прочесть микрокоманду
W	- записать микрокоманду
<(>)	- открыть предыдущую (следующую) ячейку
Z	- очистить часть ПЗУ
M[Fild]	- модифицировать поле микрокоманды
L[T]	- вывести листинг микропрограммы на экран
L[P]	- вывести листинг микропрограммы на принтер
L[F]	- вывести листинг микропрограммы в файл
B	- установить начальные параметры модели
G	- начать моделирование
P	- продолжить моделирование
S	- установить начальные значения регистров
D	- вывести значения отдельных регистров модели
I	- вывести значения всех регистров модели
E	- конец работы
H	- обратиться к справочнику системы
F[L]	- чтение из файла
F[S]	- сохранение части ПЗУ в файле
C	- смена имени задачи

ФОРМАТ МИКРОКОМАНДЫ		
Поле	Разряды	Назначение
M1	63 - 48	Поле констант
M2	47 - 46	Управление маской и инверсией условия
M3	45 - 42	Управление мультиплексором условий
M4	41 - 38	Управление БМУ
M5	37 - 33	Управление мультиплексором адреса А
M6	32 - 28	Управление мультиплексором адреса В
M7	27 - 26	Управление сдвигами
M8	25 - 24	Управление входом переноса
M9	23 - 21	Управление источниками операндов АЛУ.
M10	20 - 18	Управление операцией АЛУ
M11	17 - 15	Управление приемником результата АЛУ
M12	14 - 12	Управление входной шиной D
M13	11 - 10	Управление регистром состояния
M14	9 - 8	Управление выходной шиной Y
M15	7 - 0	Управление вводом-выводом

M1 Поле констант
Константа или адрес

M2 Управление маской и инверсией условия
Вариант 1XX
M2 = 0,2 - нет инверсии условия M2 = 1,3 - инверсия условия Маскирование выполняется командами JMAP и CJV
Остальные варианты
0 - нет инверсии условия и нет маски 1 - инверсия условия, маски нет 2 - нет инверсии условия, маскирование адреса 3 - инверсия условия, маскирование адреса

М3 Управление мультиплексором условий.		
Вариант 1XX, M2=0,1		Вариант 1XX, M2=2,3
Остальные варианты		
0 - PSW[0]	10 - FLG[0]	10 - YP1
1 - PSW[1]	11 - FLG[1]	11 - YP2.1
2 - PSW[2]	12 - FLG[2]	12 - YP2.2
3 - PSW[3]	13 - FLG[3]	13 - YP3
4 - PSW[4]	14 - FLG[4]	14 - YP4
5 - PSW[5]	15 - FLG[5]	15 - YP5
6 - ПЕРЕНОС СТ	16 - FLG[6]	16 - RK[11/9]
7 - КОНСТАНТА 0	17 - FLG[7]	17 - RK[5/3]

М4 Управление БМУ
0 - перейти на нулевой адрес (JZ) 1 - условный переход к подпрограмме по адресу из PMK (CJS) 2 - переход по адресу из младшего байта PK (JMAP) 3 - условный переход по адресу из PMK (CJP) 4 - загрузка стека и условная загрузка счетчика СТ (PUSH) 5 - условный переход к подпрограмме по адресу из RA , либо из PMK (JSRP) 6 - условный переход по адресу из младшего байта PK (CJV) 7 - условный переход на адрес из RA , либо из PMK (JRP) 10 - повторение цикла, если счетчик не равен нулю (RFCT) 11 - повторение по адресу, выбираемому из PMK, если счетчик не равен нулю (RPCT) 12 - условный возврат из подпрограммы (CRTN) 13 - условный переход по адресу из PMK и прием из стека (CJPP) 14 - загрузка счетчика и последовательная выборка (LDCT) 15 - проверка условия окончания цикла (LOOP) 16 - последовательная выборка (CONT) 17 - переход по адресу выбираемому из PMK (JP)

M5 Управление мультиплексором адреса A.				
Разряды адреса A				
0	M5[3]	M5[2]	M5[1]	M5[0]
1	0	PK[2]	PK[1]	PK[0]
1	1	PK[8]	PK[7]	PK[8]
Для M5=(0-17) A=M5 Для M5=(20-27) A=PK[2/0] Для M5=(30-37) A=PK[8/6].				

M6 Управление мультиплексором адреса A.				
Разряды адреса B				
0	M6[3]	PK[2]	PK[1]	PK[0]
1	0	PK[2]	PK[1]	PK[0]
1	1	PK[8]	PK[7]	PK[6]
Для M6=(0-17) B=M6 Для M6=(20-27) B=RK[2/0]; Для M6=(30-37) B=RK[8/6]				

M7 Управление сдвигами.		
Код		Схема сдвига
M11[1]	M7	
0	0	"0"→[R]→[PSW(0)] "0"→[RQ]→
0	1	→[R]→[PSW(0)]→→[RQ]→ 
0	2	[PSW(0)]→[R]-----→[RQ]→[PSW(0)]
0	3	[(F15[+] OVR)]→[R]→[RQ]→[PSW(0)]
1	0	[PSW(0)]←[R]←"0"←[RQ]←"0"
1	1	←[PSW(0)]←[R]←←[RQ]← 
1	2	[PSW(0)]←[R]←-----[RQ]←"1"
1	3	[PSW(0)]←[R]←-----[RQ]←"0"

M8 Управление входом переноса.
--------------------------------

Код	Значение вх. переноса
0	"0"
1	"1"
2	PSW[0]
3	Инверсия PSW[0]

M9 Управление источниками операндов АЛУ.					
Код	Входы		Код	Входы	
	R	S		R	S
0	A	Q	4	0	A
1	A	B	5	D	A
2	0	Q	6	D	Q
3	0	B	7	D	0

M10 Управление операцией АЛУ.			
Код	Операция	Код	Операция
0	R+S+C0	0	R & S
1	S-R-1+C0	1	$\overline{R \& S}$
2	R-S-1+C0	2	R [+] S
3	R $\vee$ S	3	$\overline{R \& S}$ S)

M11 Управление приемником результата АЛУ					
Код	Сдвиг РЗУ	Загрузка РЗУ	Сдвиг RQ	Загрузка RQ	Выход "Y"
0	-	-	-	F → Q	F
1	-	-	-	-	F
2	-	F → B	-	-	A
3	-	F → B	-	-	F
4	→	F/2 → B	→	Q/2 → Q	F
5	→	F/2 → B	-	-	F
6	←	B ← 2*F	←	Q ← 2*Q	F
7	←	B ← 2*F	-	-	F

M12 Управление входной шиной D
--------------------------------



0 - RDI→ D	4 - RDI→ D; BI→ RDI
1 - ZI→ D	5 - ZI→ D; BI→ RDI
2 - M1→ D	6 - M1→ D; BI→ RDI
3 - PSW→ D	7 - RDI→ D; BI→ RDI ; BI→ RK

M13 Управление регистром состояния.			
0 - сохранение значений PSW			
1 - загрузка PSW по признакам МПС K1804BC1			
2 - загрузка PSW с учетом сдвигов			
3 - загрузка PSW с шины "Y"			
PSW	M13=1	M13=2	M13=3
0	C16-перенос из старшего	Выталкиваем разряд	Y[0]
1	разряда	OVR-сигнал	Y[1]
2	OVR-сигнал переполнения	переполнения	Y[2]
3	Z-сигнал нулевого результата	Z-сигнал 0 результата	Y[3]
4	F15-знак результата	F15 [+] OVR	Y[4]
5	-	-	Y[5]
6	-	-	Y[6]
7	-	-	Y[7]
	-		

M14 Управление выходной шиной Y.
0- -
1 - Y = RDO
2 - Y = MAR
3 - -

M15 Управление вводом-выводом.
Поле кодировки пользователя.

## Список литературы

- 
1. Микро- и мини-ЭВМ. / Б.П. Балашов, Г.А. Петров, В.Л. Григорьев. - Л.: Энергоатомиздат, 1984
  2. Малые ЭВМ и их применение. / Ю.А. Дедков, М.А. Островский, К.В. Песелев и др.; Под ред. Б.Н. Наумова. - М.: Статистика, 1980.
  3. Основы программирования на Ассемблере для СМ ЭВМ. / Г.В. Вигдорчик, А.Ю. Воробьев, В. Д. Праченко. - М.: Финансы и статистика, 1983.
  4. Липьят А. Архитектура малых вычислительных систем. - М.: Мир, 1981.
  5. Мик Дж., Брик Дж. Проектирование микропроцессорных устройств с разрядно-модульной организацией. - М.: Мир, 1984.
  6. Клингман Э. Проектирование специализированных микропроцессорных систем - М.: Мир, 1985.
  7. Проектирование цифровых систем на комплектах микропрограммируемых БИС. / С.С. Булгаков, В.М. Мещеряков и др.; Под ред. В.Г. Колесникова. - М.: Радио и связь, 1984.
  8. Калабеков Б. А. Микропроцессоры и их применение в системах передачи и обработки сигналов: - М.; Радио и связь, 1986.
  9. МикроЭВМ: В 8 кн. : Практическое пособие/Под, ред. Л. Н. Преснухина - М. : Высш. шк. , 1988. - Кн.1,2.
  10. Мини- и микроЭВМ семейства "Электроника"/Б. Л. Толстых И.Л. Топов, В.Г. Цывинский и др. - М.: Радио и связь, 1987.
  11. Применение интегральных микросхем в электронной вычислительной технике: Справочник / Под ред. Б.И. Файзулаева и Б.В. Тарабрина. - М.: Радио и связь, 1986.
  12. Микропроцессоры: Справочное пособие для разработчиков судовой РЭА/Г. Г. Гришин, А.А. Мошков и др. - Л. : Судостроение, 1987.
  13. Сапаров В.Е., Максимов Н.А. Системы стандартов в электросвязи и радиоэлектронике. - М.: Радио и связь, 1965.
  14. Коннов Н.Н., Пучков В. Г., Шашков Б.Д. Изучение архитектуры микроЭВМ / Методические указания. - Пенза: Пенз. политехн. ин-т, 1985.

---

15. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник / Под ред. В.А.Шахнова. - М.: Радио и связь, 1988, т 2.

16. Комплект БИС К1804 в процессорах и контроллерах. / В.М.Мещеряков, И.Е. Лобов, С.С. Глебов и др. Под ред. В.Б. Смолова - М. Радио связь, 1990.

---

## Содержание

ОПИСАНИЕ СТРУКТУРЫ БАЗОВОЙ МИКРОЭВМ .....	6
РАЗРАБОТКА АЛГОРИТМА ФУНКЦИОНИРОВАНИЯПРОЦЕССОРА.....	11
СОСТАВЛЕНИЕ ДИАГРАММ МИКРОПРОГРАММНОЙ ЛОГИКИ.....	17
КОДИРОВАНИЕ МИКРОПРОГРАММ.....	19
ОТЛАДКА МИКРОПРОГРАММ НА КРОСС-СИСТЕМЕ.....	22
РАСЧЁТ БЫСТРОДЕЙСТВИЯ ПРОЦЕССОРА.....	24
ОФОРМЛЕНИЕ ОТЧЕТНОЙ ДОКУМЕНТАЦИИ.....	27
ПРИЛОЖЕНИЕ 1.....	30
СХЕМЫ БИС МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА К1804 .....	30
ПРИЛОЖЕНИЕ 2.....	34
СПРАВОЧНЫЕ МАТЕРИАЛЫ УЧЕБНОЙ КРОСС-СИСТЕМЫ .....	34
СПИСОК ЛИТЕРАТУРЫ.....	39

---

Проектирование процессора ЭВМ на секционированных  
микропроцессорных ВИС

Методические указания  
к курсовому проектированию

Редактор *С.В. Сватковская*  
Технический редактор *И.А. Вьялкова*  
Корректор *И.И. Сафронова*  
Компьютерная верстка *Н.В. Ивановой*

ЛР № 020347 от 14.01.97

Сдано в производство 12.11.2001. Формат 60х 64 <sup>1/64</sup>.  
Бумага газетная. Печать офсетная. Усл. Печ. л. 2,56.  
Уч.-изд.л. 3, 05 Тираж 75. Заказ « 733. «С» 132.

---

Издательство Пензенского государственного университета  
440026, Пенза Красная, 40