



# РАДИОЛЮБИТЕЛЮ О МИКРОПРОЦЕССОРАХ И МИКРО-ЭВМ

Директива ввода «I» инициирует ввод данных с магнитной ленты в ОЗУ микро-ЭВМ. Данные на ленте должны быть записаны в указанном выше формате. Директива не содержит параметров, так как значения адресов зоны памяти, в которые будет произведена запись, считываются с ленты. Если вся запись считана верно, то по окончании ввода на экране высвечиваются в шестнадцатиричном виде значения начального и конечного адресов области памяти, в которую были записаны информация, считанная с ленты, и сообщение о готовности монитора. Признаком того, что при считывании были обнаружены ошибки, может служить то обстоятельство, что при окончании записи на ленте (о чем можно судить на слух) на экране дисплея не появилось сообщение о готовности монитора к выполнению следующей директивы. Константа для временной задержки при считывании также вынесена в ОЗУ. Перед началом операции чтения в ячейку F75CH необходимо поместить константу, значение которой рассчитывается так же, как и для записи. Эта константа должна быть в полтора раза больше константы записи.

По директиве «V» — сравнения содержимого области памяти и информации, записанной на ленту, можно проверить верность записи-считывания. Каждый байт, считанный с ленты, сравнивается с соответствующим байтом из области памяти. В случае ошибки на экран дисплея выводятся адрес ячейки памяти, содержимое этой ячейки и значение байта, считанного с ленты. После обнаружения первой же ошибки выполнение директивы заканчивается, и на экране дисплея появляется сообщение о готовности монитора к приему новых директив.

## ДИРЕКТИВЫ ЗАПУСКА И ОТЛАДКИ ПРОГРАММ

С помощью директивы «J» можно запустить в работу любую отлаженную программу, хранящуюся в памяти микро-ЭВМ. Для этого необходимо знать начальный адрес этой программы, т. е. адрес команды, которая исполняется в программе первой. Этот адрес использован в качестве параметра директивы.

Для отладки написанных вами программ МОНИТОР позволяет организовать отладочный режим. После того, как вы введете коды написанной вами программы в память микро-ЭВМ с клавиатуры дисплея, используя директиву «M», можно приступить к ее отладке. Конечно, предварительно за письменным столом вы должны отладить свою программу на листе бумаги, т. е. четко представить себе все действия и результаты при выполнении каждой ее команды.

МОНИТОР позволяет в отладочном режиме выполнить одну или несколько команд вашей программы (фрагмент программы). Отладочный режим характерен тем, что для каждого фрагмента программы можно задать некоторые начальные условия его выполнения — содержимое внутренних регистров и ячеек ОЗУ. Кроме того, можно задать адреса останова — ими могут быть адреса ячеек памяти, при достижении которых должно быть прекращено выполнение отлаживаемого фрагмента программы. При достижении адреса останова текущее содержимое регистров микропроцессора автоматически запоминается в специально отведенных ячейках памяти и управление передается МОНИТОРУ. При этом МОНИТОР предоставляет вам возможность просмотреть и при необходимости изменить содержимое регистров и содержимое любых ячеек памяти, назначить новый адрес останова, продолжить выполнение отлаживаемой программы, причем содержимое регист-

ров микропроцессора перед этим автоматически восстанавливается.

Для организации останова отлаживаемой программы по заданному адресу в нашем МОНИТОРе использован следующий способ. В ячейку памяти с адресом, равным адресу останова, при выполнении соответствующей директивы помещается код команды RST 7, при этом предварительно содержимое этой ячейки запоминается в рабочей области памяти МОНИТОРа. В ячейку памяти с адресом 0038H (адрес, по которому будет передано управление по команде RST 7) помещают команду перехода на соответствующую подпрограмму МОНИТОРа. Таким образом, при достижении адреса останова отлаживаемой программой выполняется команда RST 7, а управление передается МОНИТОРУ, который восстанавливает содержимое ячейки памяти, в которую был записан код команды RST 7. Все эти действия выполняются автоматически при выполнении соответствующих директив МОНИТОРа.

Рассмотрим теперь действия, выполняемые МОНИТОРОм по директивам отладки программ.

Директива «B» позволяет назначить один адрес останова в отлаживаемой программе.

Директива «G» предназначена для запуска отлаживаемой программы с адреса, используемого в ней в качестве параметра. При достижении адреса останова (заданного ранее с помощью директивы «B») управление передается МОНИТОРУ.

Директива «P» дает возможность организовать отладку циклически выполняемых программ с использованием двух адресов останова. В ней использованы три параметра: ADR1 — первый адрес останова, ADR2 — второй адрес останова и D8 — число проходов через второй адрес останова перед выполнением окончательного останова. После ввода директивы и нажатия на клавишу «BK» на экран будет выведено следующее сообщение:

START —

(Окончание. Начало см. «Радио», 1983, № 11, с. 31—34)

После этого оператор должен набрать начальный адрес запуска отлаживаемой программы и вновь нажать клавишу «ВК». На экране дисплея появится еще одно сообщение:

DIR.—

Теперь оператор может набрать одну из директив МОНИТОРА. Обычно это директива «D» с двумя параметрами — просмотр содержимого области памяти. Набор директивы заканчивается нажатием на клавишу «ВК». После этого управление передается по адресу: «START».

При достижении адреса останова на экран дисплея каждый раз будет выводиться содержимое внутренних регистров микропроцессора и выполняться директива МОНИТОРА, набранная в ответ на запрос DIR. Заметим, что оператор в ответ на этот запрос может просто нажать на клавишу «ВК», тогда при достижении адреса останова на экран будет выведено только содержимое регистров микропроцессора.

Рассмотренная директива не требует предварительного назначения адреса останова и является удобным средством отладки циклически работающих программ.

По директиве «X» на экране дисплея может быть выведено содержимое внутренних регистров микропроцессора в следующем формате:

A—00 B—00 C—00 D—00 E—00  
F—00 H—00 L—00 S—0000 O — 0000.

Здесь буквы означают название соответствующего регистра микропроцессора A, B, C, D, E, F (регистр признаков), H, L, SP (указатель стека) и O (адрес, по которому произошел останов программы, т. е. содержимое счетчика команд PC).

Директива «X» служит также и для изменения содержимого внутреннего регистра микропроцессора. Директива имеет параметр R, в качестве которого использовано однобуквенное наименование регистра: A, B, C, D, E, H, L, F или S. После нажатия на клавишу «ВК» на экран выводится текущее содержимое регистра, и курсор устанавливается справа от этого значения. Теперь оператор может набрать новое значение, и после нажатия на клавишу «пробел» оно будет записано в соответствующий регистр.

**Справочные директивы.** Кроме уже перечисленных, есть еще три директивы — справочные.

Директива «H» предназначена для подсчета суммы и разности двух шестнадцатиричных чисел. После набора директивы, двух четырехразрядных шестнадцатиричных чисел и нажатия на клавишу «ВК» на экран дисплея одновременно будут выведены их шестнадцатиричные сумма и разность.

Директива «A» позволяет вывести

на экран шестнадцатиричный код символа, заданного в качестве ее параметра. Например, если набрать A1(BK), то на экране высветится «31» — код символа «1».

Если задать МОНИТОРУ директиву «K», то после нажатия на клавишу «ВК» все, что набирает оператор на клавиатуре, будет отображаться на экране дисплея. В основном этот режим необходим для проверки работы дисплея и клавиатуры. Выйти из такого режима работы дисплея можно, нажав одновременно на клавиши «УС» и «А».

В состав МОНИТОРА входит ряд подпрограмм ввода-вывода, которые могут быть использованы программистами в своих программах. Перечислим эти подпрограммы и правила обращения к ним.

1. Подпрограмма ввода символа с клавиатуры.

Адрес вызова — F803H.

После возврата из подпрограммы код введенного символа находится в регистре «A» микропроцессора.

2. Подпрограмма ввода байта с магнитофона.

Адрес вызова — F806H.

После возврата из подпрограммы введенный байт находится в регистре «A» микропроцессора.

3. Подпрограмма вывода символа на экран дисплея.

Адрес вызова — F809H.

Перед вызовом этой подпрограммы необходимо поместить код выводимого символа в регистр «C» микропроцессора.

4. Подпрограмма записи байта на магнитофон.

Адрес вызова — F80CH.

Перед вызовом этой подпрограммы необходимо поместить выводимый байт в регистр «C» микропроцессора.

5. Подпрограмма проверки состояния клавиатуры.

Адрес вызова — F812H.

После возврата из данной подпрограммы в регистре «A» микропроцессора будет содержаться 00H — если клавиша не нажата, или FFH — если клавиша нажата.

6. Подпрограмма вывода на экран содержимого регистра «A» микропроцессора в шестнадцатиричном виде.

Адрес вызова — F815H.

Содержимое регистра «A» микропроцессора выводится на экран дисплея в виде двух шестнадцатиричных цифр.

7. Подпрограмма вывода сообщений на экран дисплея.

Адрес вызова — F818H.

Данная подпрограмма позволяет выводить на экран дисплея любые тексты, хранящиеся в памяти в виде последовательности кодов символов. Перед вызовом подпрограммы в регистровую пару HL записывают начальный адрес последовательности кодов. Признаком

конца текста служит код 00H, встретившийся в последовательности кодов символов.

Подпрограммы ввода-вывода МОНИТОРА позволяют программисту, разрабатывающему свои программы, не задумываться над тем, как устроены и какими подпрограммами обслуживаются дисплей, клавиатура и модуль сопряжения с магнитофоном в его микро-ЭВМ. Единые правила обращения к подпрограммам ввода-вывода позволяют радиолюбителям обмениваться программами. При этом их микро-ЭВМ, выполненные по различным схемам (но, конечно, на микропроцессоре КР580ИК80А), должны иметь управляющие программы, похожие на описываемый МОНИТОР, и содержать программы ввода-вывода «по-своему», обслуживающие отличные от описываемых дисплеи и клавиатуры. Но при этом правила обращения к этим подпрограммам должны быть едиными во всех микро-ЭВМ.

Принятые в описываемом мониторе правила обращения к подпрограммам, обслуживающим дисплей и клавиатуру, являются типичными для многих существующих микро-ЭВМ. Сложнее дело обстоит с подпрограммами обслуживания кассетного магнитофона. Здесь для достижения совместимости всех микро-ЭВМ необходимо также иметь одинаковые метод, скорость и формат записи информации на ленту.

Рассмотрим теперь пример, поясняющий использование некоторых возможностей монитора при написании и отладке программ. На рис. 1 представлена программа, реализующая следующие действия. При вводе с клавиатуры трех символов — A35, именуемых далее паролем, на экран выводится сообщение «правильно». Если при вводе любого из символов будет допущена ошибка, то на экран выводится вопросительный знак, и ввод пароля надо начать сначала. Пользуясь комментариями к программе, вы легко сможете восстановить ее алгоритм работы. При этом учтите, что для ввода символа с клавиатуры, вывода символа на экран, а также вывода сообщения «правильно», используются подпрограммы монитора, обращение к которым происходит по командам CALL F803H, CALL F809H и CALL F818H соответственно. Обратите также внимание на те действия, которые производятся в основной программе перед обращением к подпрограммам.

В ячейках памяти 0127H—0129H хранятся коды символов пароля, а в ячейках 012AH—0132H коды символов, из которых состоит сообщение «правильно». Не упустите из вида и то, что при написании текста программы (поля 3, 4, 5) в первом случае мы

непосредственно использовали коды символов пароля, которые затем при переводе программы в машинные коды и переписали в поле 2. Во втором случае при написании программы в поле 5 мы занесли слово «правильно», заключив его в апострофы (надстрочные запятые). И только на стадии перевода программы в машинные коды мы в поле 2 заменили коды букв этого слова. Два этих приема совершенно равноценны, но если текст программы предполагается переводить (транслировать) в машинные коды автоматически с помощью специальной программы — транслятора с ассемблера, то, конечно, используют второй способ. Ведь при этом вам не придется пользоваться таблицей перевода символов в коды, так как ассемблер автоматически поместит в поле 2 коды, соответствующие символам, заключенным в апострофы.

И еще одно пояснение к программе: внутренний регистр D микропроцессора используется в качестве счетчика правильно введенных символов пароля. Этот счетчик при вводе символов работает на вычитание и при его обнулении после трех введенных символов выполняется команда, загружающая в регистровую пару HL адрес кода первой буквы слова «правильно». Затем вызывается подпрограмма для вывода этого слова на экран. Вывод кодов символов происходит до момента, пока не встретится код 00H, записанный в ячейку 0133H.

В тексте программы допущена одна ошибка, которую вам предстоит найти и устранить в процессе отладки. Итак, с помощью директивы «M» введем коды программы по соответствующим адресам в память микро-ЭВМ. Правильность ввода можно проконтролировать по директиве «D», в ответ на которую на экран будет выведена таблица с содержимым области памяти в шестнадцатичном виде. Если вы воспользуетесь директивой «L», то сможете посмотреть содержимое памяти в символьном виде, проверив тем самым, верно ли введены коды пароля и слова «правильно».

Если при вводе кодов программы были допущены ошибки, то с помощью директивы «M» их необходимо устранить.

Теперь по директиве «J» запустим программу:

```
>J100 (BK).
```

Набирая на клавиатуре любые символы, не соответствующие паролю, убеждаемся, что они отображаются на экране, причем после каждого из них выводится также и вопросительный знак. Набираем теперь пароль. Однако после ввода пароля на экран слово «правильно» почему-то не выводится.

Чтобы вновь передать управление МОНИТОРУ, нажмем на кнопку «СБР» микро-ЭВМ.

Теперь попытаемся выяснить причину ошибочной работы программы. Для этого подготовим к запуску нашу программу в отладочном режиме, задав два адреса останова и число проходов через второй из них. Возможность одновременного задания сразу двух адресов останова позволяет при любых исходных данных приостанавливать выполнение отлаживаемой программы после команд условной передачи управления (т. е. в местах «разветвления» программы). Если бы при этом использовался только один адрес останова, то для гарантированной приостановки работы программы после команд условной передачи управления было бы необходимо заранее знать, будет или нет выполнено условие передачи управления. Другими словами, знать, по какому пути продолжится выполнение программы после «разветвления», что далеко не всегда возможно. В нашем случае мы будем контролировать содержимое внутренних регистров микропроцессора D и HL после выполнения команды JNZ ВВОД. При этом в качестве первого и второго адресов останова установим адреса следующих за ней команд при отсутствии условия передачи управления и при его наличии соответственно.

Число проходов через второй адрес останова установим равным 4 — на единицу больше числа символов в пароле. Адрес запуска программы зададим равным 100H, т. е. программе запускаем по ее начальному адресу. При прохождении программы через адрес останова будем следить только за содержимым внутренних регистров микропроцессора. Теперь введем следующие директивы и параметры:

```
P118,108,4(BK)
```

```
START—100(BK)
```

```
DIR—(BK)
```

Набираем символы пароля и после ввода каждого из них просматриваем содержимое внутренних регистров микропроцессора. Учтем, что содержимое регистров B и E может быть произвольным, так как они не использованы в программе.

Замечаем, что содержимое регистровой пары HL при вводе каждого нового символа увеличивается на 1, что полностью соответствует логике работы программы. Однако содержимое регистра D остается неизменным. Анализ программы позволяет выявить ошибку — вместо команды DCR D в программе ошибочно используется команда DCX D, уменьшающая каждый раз на 1 не содержимое регистра D, а содержимое регистровой пары DE, что противоречит нашему алгоритму. Выяс-

нив причину ошибки, нажимаем на кнопку «СБР» микро-ЭВМ и тем самым передаем управление МОНИТОРУ. Теперь исправим содержимое ячейки 0114H на код команды DCRD, равный 15H.

Отлаженную таким образом программу запускаем по директиве «J» с начального адреса и убеждаемся в том, что она работает правильно.

Описанная выше программа МОНИТОРА является одной из системных программ микро-ЭВМ. Рассмотрим теперь кратко другие компоненты системного программного обеспечения, которое может быть использовано в нашей микро-ЭВМ.

Программа «редактор текстовой информации» позволяет ввести в ОЗУ микро-ЭВМ с клавиатуры дисплея любой текст, например текст программы, просмотреть его на экране дисплея, удалить или вставить в текст любой символ или строку символов. Кроме того, обычно имеется возможность автоматического поиска и замены заданной последовательности символов в тексте. После того, как текст подготовлен и исправлен, он может быть переписан из ОЗУ на магнитную ленту, а в дальнейшем считан с магнитной ленты в ОЗУ микро-ЭВМ для внесения новых исправлений.

Записанный на ленту текст программы на языке ассемблера является входной информацией для работы программы-транслятора (называемой также ассемблером) при подготовке машинных кодов программы. Полученная в результате работы ассемблера программа в машинных кодах может быть записана на магнитную ленту и в дальнейшем загружена в ОЗУ для отладки или запуска в работу. Применение ассемблера значительно облегчает разработку программ. Но в то же время разработка программ на ассемблере значительно более трудоемка, чем на языках высокого уровня.

Наиболее распространенным языком программирования высокого уровня для микро-ЭВМ является язык БЭЙСИК. Это объясняется как легкостью изучения и использования самого языка, так и простотой его реализации для микро-ЭВМ. Для того чтобы писать или использовать программы на этом языке в составе микро-ЭВМ, необходимо иметь транслятор или интерпретатор с языка БЭЙСИК. Так же как и при использовании языка ассемблера, транслятор с языка БЭЙСИК формирует в результате своей работы программу в машинных кодах микропроцессора, которая после загрузки в память может быть непосредственно исполнена.

Однако чаще для языка БЭЙСИК вместо транслятора используют интер-

АДР.	КОД	МЕТКА	МНЕМ.	ОПЕРАНА	КОММЕНТАРИИ
1	2	3	4	5	6
0100	310001		LXI	SP, 0100H	НАСТРОЙКА СТЕКА
0103	1603	НАЧАЛО:	MVI	D, 3	ЧИСЛО СИМВОЛОВ В ПАРОЛЕ
0105	212701		LXI	H, ПАРОЛ	HL — УКАЗАТЕЛЬ НА НАЧАЛО
					ОБЛАСТИ ПАМЯТИ, ГДЕ ХРА-
					НИТСЯ ПАРОЛЬ
0108	CD03F8	ВВОД:	CALL	F803H	ВВОД СИМВОЛА
010B	4F		MOV	C, A	ПЕРЕСЛАТЬ ЕГО В РЕГ. С
010C	CD09F8		CALL	F809H	ОТОБРАЗИТЬ ЕГО НА ЭКРАНЕ
010F	BE		CMP	M	СРАВНИТЬ С ПАРОЛЕМ
0110	C21F01		JNZ	ОШИБК	НЕ СОВПАДАЕТ -> ОШИБКА
0113	23		INX	H	ПЕРЕИТИ К СЛЕД. СИМВОЛУ
0114	1B		DCX	D	УМЕНЬШИТЬ СЧЕТЧИК СИМВОЛОВ
0115	C20801		JNZ	ВВОД	ПОВТОРИТЬ ВВОД
0118	212A01		LXI	H, ВЕРНО	HL — УКАЗАТЕЛЬ НА СООБЩЕ-
					НИЕ 'ПРАВИЛЬНО'
011B	CD1BF8		CALL	F81BH	ВЫВОД СООБЩЕНИЯ НА ЭКРАН
011E	76		HLT		КОНЕЦ РАБОТЫ ПРОГРАММЫ
011F	0E3F	ОШИБК:	MVI	C, 3FH	3FH — КОД СИМВОЛА '?'
0121	CD09F8		CALL	F809H	ВЫВЕСТИ НА ЭКРАН '?'
0124	C30301		JMP	НАЧАЛО	ПОВТОРИТЬ ВВОД
0127	413335	ПАРОЛ:	DB	41H, 33H, 35H	КОД СИМВОЛОВ ПАРОЛЯ — АЗЗ
012A	707261	ВЕРНО:	DB	'ПРАВИЛЬНО'	СООБЩЕНИЕ
	77696C				
	786E6F				
0133	00		DB	0	; ПРИЗНАК КОНЦА СООБЩЕНИЯ

претатор. В этом случае в памяти ЭВМ должны одновременно находиться как текст (не машинные коды, а именно текст самой выполняемой программы), так и программа-интерпретатор. Интерпретатор при работе «просматривает» строки текста программы, распознает различные операторы языка и сразу же их выполняет (интерпретирует). Однако это значительно замедляет работу программы.

Интерпретатор БЭЙСИКА позволяет выполнить все действия, связанные с подготовкой, отладкой, записью на магнитную ленту и запуском в работу программ на языке БЭЙСИК. Для этой цели в состав интерпретатора входит упрощенный редактор текстов и специальные отладочные средства.

Объем различных версий интерпретаторов БЭЙСИКА для микропроцессора КР580ИК80А может быть от 2 Кбайт — «минимальный БЭЙСИК» до 20 Кбайт — «расширенный БЭЙСИК». Естественно, что и набор функций, ими реализуемый, также будет разным.

Особенностью интерпретатора БЭЙСИКА для нашей микро-ЭВМ является возможность работы с использованием псевдографических символов, отображаемых дисплейным модулем. Хотелось бы отметить, что игровые программы, как правило, пишут на языке БЭЙСИК. Подробнее с описанием языка Вы можете познакомиться в [Л].

В заключение мы хотели бы выразить надежду, что опубликованный цикл статей поможет радиолюбителям в освоении микропроцессорной техники, имеющей огромное народнохозяйственное значение. Радиолюбители могут приложить свои усилия в созда-

нии специализированных цифровых кассетных магнитофонов, цветных графических дисплеев, устройств управления бытовым радиокомплексом, синтезаторов музыки и речи, разнообразных системных и прикладных программ, а также устройств управления промышленными установками.

**Г. ЗЕЛЕНКО,  
В. ПАНОВ, С. ПОПОВ**

#### ЛИТЕРАТУРА

Уорт Т., Программирование на языке БЭЙСИК. «Машиностроение» — М., 1981.

★ ★ ★

Уважаемые читатели! Авторы считают своим долгом сообщить следующее.

Опытная эксплуатация нескольких экземпляров микро-ЭВМ показала, что номинал конденсатора С2 на схеме на рис. 2 («Радио», 1983, № 2, с. 41) должен быть уменьшен до 0,01 мкФ.

В схеме отладочного модуля («Радио», 1983, № 4) выявлены следующие ошибки: — вход 13 элемента D29.4 должен быть подключен не к проводу 70, а к проводу 30 (сигнал ОЖ);

#### КНИГА — ПОЧТОЙ

Магазин № 8 «Техника» Москниги имеет в продаже и высылает наложенным платежом (без задатка) следующие книги:

Бодилевский В. Г. Справочник молодого радиста. — М.: Высш. шк., 1983. — 320 с., ц. 1 р. 40 к.

Бенонсон З. М., Елистратов М. Р., Ильин Л. К. Моделирование и оптимизация на ЭВМ радиоэлектронных устройств. — М.: Радио и связь, 1981. — 272 с., ц. 1 р. 20 к.

— вместо элемента D11.4, выполняющего функцию логического «ИЛИ», в схему модуль должен быть установлен дополнительный элемент, выполняющий функцию «И» с инверсией.

— Входы D2 и D6 элементов D20 и D21 нужно подключить через резистор R3 к проводу «Земля».

В статье модуль программатора («Радио», 1983, № 6) при описании работы программы допущена ошибка. При верной записи информации в ППЗУ на светодиодах отладочного модуля появится комбинация 00001111, а не 10000001. В случае невозможности записать информацию в ППЗУ за 16 циклов появится комбинация 10000001.

На схеме дисплейного модуля («Радио», 1983, № 7, рис. 3) неправильно изображено подключение вывода 11 элемента D39. Этот вывод необходимо подключить к проводнику, соединяющему вход 14 элемента D41 и входы 9, 10 элемента D43.

На схеме клавиатуры («Радио», 1983, № 8) должны быть проставлены номера проводов в жгуте, соединяющем выводы 14, 15, 16 элемента D1 с выводами элементов D3 и D4: соответственно должно быть (слева направо) — 1, 2, 3.

В заключение приводим список литературы, которую мы рекомендуем прочитать радиолюбителям.

Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы. «Радио и связь». — М., 1981.

Каган Б. М., Сташин В. В. Микропроцессоры в цифровых системах. «Энергия». — М., 1978.

Клингман Э. Проектирование микропроцессорных систем. «Мир». — М., 1980.

Гибсон Г. Ю.—Ч. Лю. Аппаратные и программные средства микро-ЭВМ «Финансы и статистика». — М., 1983.

Коффон Дж. Технические средства микропроцессорных систем: Практический курс. — «Мир». М., 1983.

**Дорогие читатели! Вы прочитали последнюю статью цикла «Радиолюбитель о микропроцессорах и микро-ЭВМ». Судя по редакционной почте многие из Вас прочли его с большим интересом и почти в каждом письме пожелание или вопрос. Очевидно, у тех, кто взялся за повторение нашей микро-ЭВМ или какого-либо её модуля, вопросы будут возникать и в дальнейшем. Пишите нам. На конверте не забудьте сделать пометку «микро-ЭВМ»**

Бадулин С. С., Барнаулов Ю. М., Бердышев В. А. и др. Автоматизированное проектирование цифровых устройств. — М.: Радио и связь, 1981. — 240 с., ц. 1 р. 30 к.

Грешберг А. Е. Электронный луч и потенциальный рельеф в электроннолучевых приборах. — Л.: Энергоиздат. Ленингр. отд-ние, 1981. — 312 с., ц. 1 р. 80 к.

Адрес магазина: 103031, Москва, Петровка, 15, отдел «Книга — почтой».