

кую-либо ячейку микросхемы необходимо подать на входы питания «Е» и нагрузочные резисторы R18, R19, R25, R29 сигнал с формирователя, построенного на VT2 и VT3.

В каждый момент времени на ключи должна быть подана информация об одном бите, на всех остальных ключах низкий уровень должен шунтировать остальные биты данных ППЗУ. Чтобы сократить время прожига, программа записывает информацию лишь в те биты, в которые нужно записать уровень «Лог. 0». Программируют ППЗУ импульсы напряжения амплитудой 15 В и длительностью 300 мкс, следующие с паузами 2,7 мс. По каждому биту информация записывается 7 раз. Число циклов записи, а также длительность импульса программирования и паузы задаются программно. Практический опыт программирования БИС KP556PT5 показал, что заданные параметры прожига удовлетворительны для 80...90 % ППЗУ, а остальные кристаллы, как правило, не программируются и при повторной записи.

Программа чтения информации в ППЗУ KP556PT5 (табл. 6) считывает ППЗУ с начального адреса и записывает его содержимое в буферное ОЗУ, начиная с адреса 800011.

ТАБЛИЦА 6

программа чтения KP556PT5	
COPY15:	ORG 8000H
CALL ROT5	; РЕЖИМ ЧТЕНИЯ KP556PT5
CALL COPY	; КОПИРОВАТЬ ПЗУ
LBL	КОПИРОВАНИЕ ПЗУ
CALL READ	; ЧИТАТЬ ПЗУ
MOV M-P	; ЗАПИСЬ В ОЗУ
INX D	; УВЕЛИЧИТЬ НА 1 АДРЕСА ОЗУ И ПЗУ
DEC D	
INC D	
CPT OFFH	; ЕСЛИ АДРЕС НЕ ПОСЛЕДНИЙ,
JMP COOP	; КОПИРОВАТЬ ДАЛЬШЕ
	; ИНДИКАЦИЯ "GOOD"

Программы работы с ППЗУ серии K573 выполнены аналогично.

Телефон для справок о приобретении документации: 1-11-87 (Смоленск). Запросы и сведения о потребности в программаторах на 1986—1990 гг. направлять по адресу: 214000, Смоленск, ул. Шевченко, 97, СКТБ СПУ.

Статья поступила 15 января 1986 г.

УДК 681.3.06

И. К. Половинкин

КРОСС-СИСТЕМА ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА Д3-28» НА БАЗЕ ЭВМ «ИСКРА 226»

Системное программное обеспечение микроЭВМ «Электроника Д3-28» рассчитано на трудоемкое программирование в машинных кодах, требующее высокой квалификации программиста и значительного времени для отладки. Поставляемый с микроЭВМ Бейсик-интерпретатор несколько ускоряет разработку программы, но существенно снижает возможности системы. Он занимает значительный объем оперативной памяти, имеет невысокое быстродействие и не позволяет вести обработку символьных данных. Кроме того, для микроЭВМ «Электроника Д3-28» разработана кросс-система МИКРОИНФ, реализованная на ЕС ЭВМ [1]. Однако в этой системе отладку программы можно полностью выпол-

нить лишь на микроЭВМ. После выявления ошибок и коррекции программы требуется новая трансляция и передача на микроЭВМ.

Кросс-система подготовки программ на персональной ЭВМ «Искра 226» обеспечивает возможности ввода, полной отладки и компиляции программ в диалоговом режиме. Передача объектной программы на микроЭВМ «Электроника Д3-28» осуществляется через устройство обмена, работающее в range ИРПР и подключаемое к БИФ «Искра 015-82» [2]. В ПО системы входят:

интерпретатор Бейсик-2 (версия от 30.09.84), применимый на микроЭВМ «Искра 226»;

кросс-компилятор, разработанный для данной системы; драйверы устройства межмашинного обмена.

Технология программирования в кросс-системе включает следующие операции:

ввод программы с клавиатуры с визуальным контролем с помощью блока отображения символьной и графической информации (БОСГИ);

полная отладка программы в режиме интерпретации; компиляция программы с записью оттранслированной программы в архив;

передача объектной программы в микроЭВМ «Электроника Д3-28».

Для хранения промежуточных результатов на всех операциях используются гибкие магнитные диски. Значительную часть функций в системе выполняет интерпретатор. На этапе ввода программы осуществляется синтаксический контроль. Предусмотрены возможности ввода основных операторов и функций Бейсика с помощью одной клавиши, исправления ошибок и редактирования строк без повторного ввода. При выполнении отлаживаемой программы на БОСГИ отображаются возникающие ошибки. Обеспечены возможности по операторного выполнения программы, индикации промежуточных и конечных результатов счета. В качестве исходного модуля для кросс-компилятора используется Бейсик-программа. Более удобен для этой цели промежуточный код, являющийся результатом обработки Бейсик-программы интерпретатором и обладающий следующими особенностями:

содержит таблицу переменных;

элементы языка имеют идентификаторы, удобные для опознания;

каждый оператор и каждый номер строки снабжаются информацией о длине оператора (строки). Такая структура программы облегчает выполнение проходов кросс-компилятором.

Интерпретатор располагает развитыми средствами обработки информации байтового формата, что позволяет использовать Бейсик в качестве языка программирования самого кросс-компилятора. На Бейсике реализована также программа, обслуживающая межмашинный обмен со стороны ПЭВМ «Искра 226». Кросс-компилятор занимает 28К байт оперативной памяти микроЭВМ «Искра 226». Наряду с кросс-компилятором разработаны подпрограммы в машинных кодах «Электроника Д3-28» (объем 3,3К байта), реализующие операторы и отдельные формы операторов языка Бейсик. Эти подпрограммы включаются в объектную программу по мере необходимости.

Опыт эксплуатации кросс-компилятора показал, что быстродействие оттранслированных программ в 5...8 раз выше быстродействия программ на Бейсике микроЭВМ «Электроника Д3-28» (вариант ЗЛ). Объем оттранслированных программ в 1,1...1,5 раза больше, чем Бейсик-программ, однако отсутствие интерпретатора дает значительный выигрыш оперативной памяти. Экономия памяти может быть получена также за счет упаковки цифровых данных в символьные. Имеется возможность включения в программу кодовых вставок произвольной длины, которые позволяют с минимальными ограничениями использовать возможности системы команд микроЭВМ «Электроника Д3-28».

Адрес для запроса дополнительной информации:

ЛИТЕРАТУРА

- Б. И. Гехман, И. А. Равкин, В. Л. Темов. Кросс-система для микроЭВМ «Электроника ДЗ-28». — Электронная техника. Сер. 9. Экономика и системы управления. Вып. 3(52).— М.: ЦНИИ «Электроника», 1984.
- С. Н. Абрамович, В. В. Бойко, В. П. Бутрин и др. Профессиональные персональные ЭВМ «Искра 226». — Микропроцессорные средства и системы, 1985, № 2, с. 29—36.

Статья поступила 29 августа 1985 г.

УДК 681.322

В. И. Жихарев

ПРОГРАММАТОР НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА ДЗ-28»

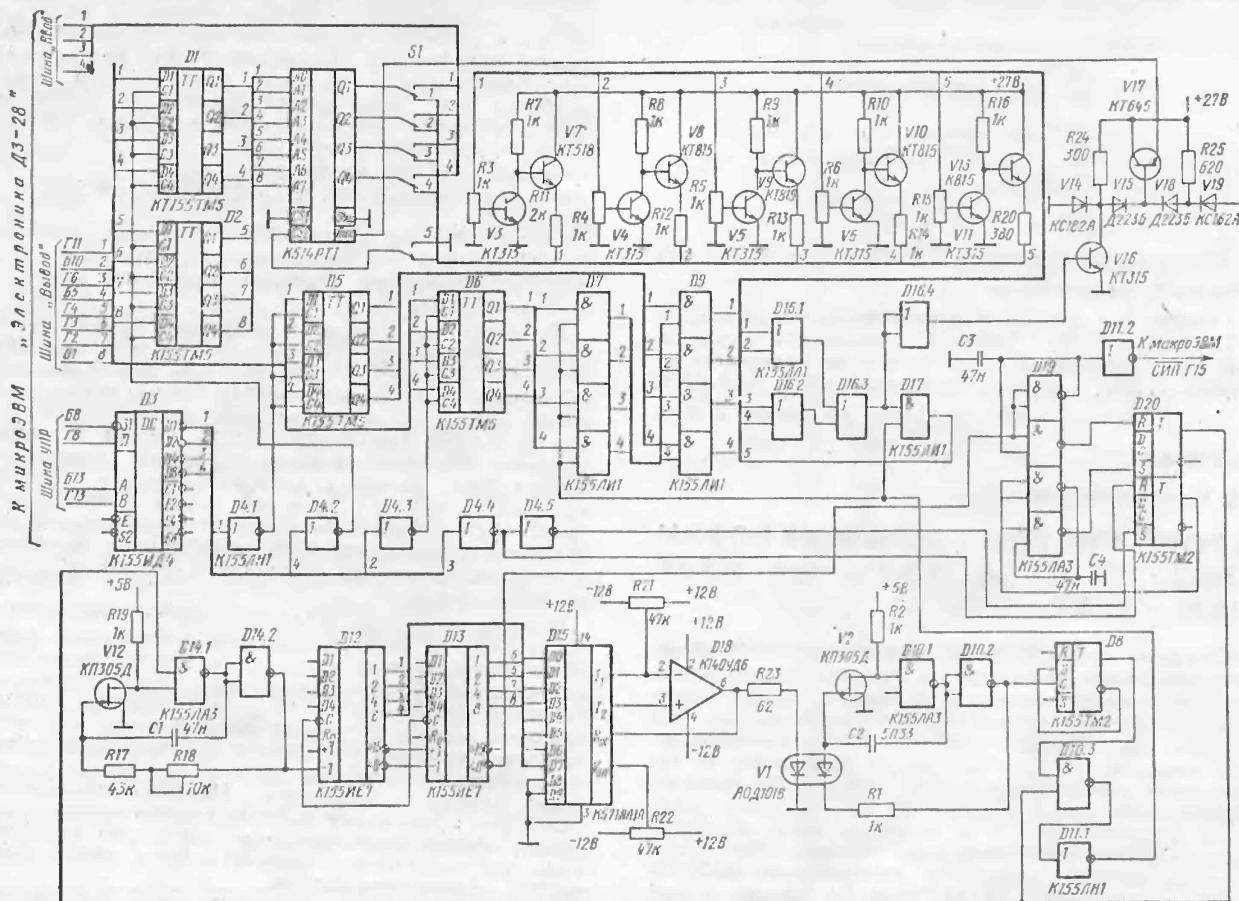
Программатор вместе с микроЭВМ «Электроника ДЗ-28» и алфавитно-цифровым дисплеем позволяет:
редактировать информацию для записи в ПЗУ;
отображать на экране дисплея записываемую информацию, содержимое ПЗУ в удобной для восприятия и понимания форме;

автоматизировать программирование и контроль правильности ввода информации в ПЗУ.

Особенность программатора: при небольших аппаратных затратах с его помощью решаются достаточно сложные задачи программирования ПЗУ. Для

этого в память микроЭВМ вводится программа управления программатором.

Работа программатора в режиме записи (рис. 1). С помощью дешифратора D3, состояния которого меняется шиной управления микроЭВМ, формируются четыре строба. Тремя из них информация с шины вывода микроЭВМ записывается в три регистра: адресный (микросхемы D1, D2), управляющий (D6), информационный (D5). Четвертый строб служит для запуска генератора программирующих импульсов. Генератор тайтвых импульсов D14, V12 заполняет счетчик D12, D13 за 0,4 с. За это время цифро-аналоговый преобразователь D15 и операционный усилитель D18 формируют линейно-возрастающее пилообразное напряжение, пропорциональное по величине числу тайтвых импульсов или выходному коду счетчика. Уровень и амплитуда напряжения устанавливаются потенциометрами R21, R22 соответственно. Оптрон V1, питаемый этим напряжением, осуществляет широтно-импульсную модуляцию генератора V2, D10. Длительность импульсов этого генератора, начиная от 1 мкс, линейно возрастает (в течение 0,4 с) до 8 мкс при постоянной скважности [2]. Эти импульсы подаются на управляемые ключи D7, D9, позволяющие в течение цикла (0,4 с) программировать только один бит. Пять усилителей мощности V3, V7, V4, V8, V5, V9, V6, V10, V11, V13 обеспечивают требуемые для K541PT1 величины напряжений и токов в режимах записи, подаваемые на соответствующие ее выводы. Одновременно с подачей импульсов на один из входов Q1, Q2, Q3, Q4 микросхемы K541PT1 такая же последовательность импульсов подается на ее 14-й вывод; если бит программируется не надо, то подавать ее на 14-й вывод не нужно. Эта процедура проводится



с помощью микросхем D16, D17. Они суммируют X1, X2, X3, X4 (входные сигналы микросхем, упомянутых выше, соответствующие определенным разрядам записываемого слова), затем умножают их сумму на X5 — программирующую последовательность импульсов. Созданный таким образом выходной сигнал у подается через усилитель V11, V13 на 14-й вывод микросхемы K541PT1. С помощью транзисторов V16, V17, стабилитронов V14, V19 и диодов V15, V18 напряжение на 16-м выводе микросхемы K541PT1 в момент программирования бита с +5 В увеличивается до +8 В.

Формирователи D19, триггеры D20 синхронизируют работу программатора.

По окончании цикла программирования формирователь D19 выдает импульс СИП, который через элемент D11.2 поступает на шину ввода микроЭВМ (гнездо Г15). Импульсом СИП программатор извещает микроЭВМ об окончании цикла программирования и готовности принять от нее следующую порцию информации: адрес записываемого слова, само слово и 4-разрядный код для управляющего регистра D6.

Программа обслуживания программатора состоит из трех подпрограмм.

Первая расположена с нулевого шага по 120-й. С ее помощью на экран дисплея из области ОЗУ микроЭВМ с начальным адресом, указанным в регистре микроЭВМ R0, выводятся 256 4-разрядных слов, представленных нулями и единицами.

Программа обслуживания программатора

0 MOVH X, R0	Установка начального адреса ОЗУ
2 CLR R4	Обнуление регистра R4
4 CLR R10	Обнуление регистра R10
6 MOV#0300, S	Запись кода единицы в регистр S0
8 MOV#0301, S1	Запись кода единицы в регистр S1
10 MOV#0200, S6	Запись кода пробела в регистр S6
12 MOV#0010, S7	Запись кода перевода строки в регистр S7
14 CLR R1	Обнуление регистра R1
16 DIG 1	
17 DIG 5	
18 MOvh X, R2	Запись числа 15 в регистр R2
20 DIG 1	
21 DIG 5	
22 MOvh X, R3	Запись числа 15 в регистр R3
24 MOV#1507, S2	
26 MOV#1515, S3	
28 OUTOWS	Обращение к дисплею
30 MARK 1	Установка метки 1
32 BBIC 4, @R0	Если четвертый бит в байте, расположенному по адресу R0, будет нулевым, то на экране дисплея выводится нуль. В противном случае — единица
34 BR 42	
36 MOV S0, @R10	
38 OUTO 1501	
40 BR 46	
42 MOV S1, @R10	
44 OUTO 1501	
46 BBIC 5, @R0	
48 BR 56	
50 MOV S0, @R10	
52 OUTO 1502	
54 BR 60	
56 MOV S1, @R10	
58 OUTO 1502	
60 BBIC 6, @R0	
62 BR 70	
64 MOV S0, @R10	
66 OUTO 1501	
68 BR 74	
70 MOV S1, @R10	

72 OUTO 1501	
74 BBIC 07, @E0	
76 BR 84	
78 MOV S0, @R10	
80 OUTO 1501	
82 BR 88	
84 MOV S1, @R10	
88 MOV S6, @R10	
90 OUTO 1501	
92 ADD#1, R0	
94 ADD#1, R1	
96 BSA R1, R2	
98 JMM 1	
100 MOV S7, @R10	
102 OUTO 1501	
104 CLR R1	
106 ADD#1, R4	
108 BSA R3, R4	
110 JMM 01	
112 OUTO 1501	
114 OUTO 1501	
116 OUTO 1501	
118 OUTO 1501	
120 STOP	Останов подпрограммы
121 MARK 3	Установка метки 3
123 CLR R2	Обнуление регистра R2
125 CLR X	Обнуление регистра X
126 STOP	Стоп
127 MOvh X, R0	Установка начального адреса R0
129 CLR X	Обнуление регистра X
130 STOP	Стоп
131 MOvh X, R1	Установка начального адреса R1
133 DIG2	
134 DIG5	
135 DIG6	
136 MOvh X, R3	Запись числа 256 в регистр R3
138 MARK 4	Установка метки 4
140 MOV @R0, S6	Пересылка байта на ОЗУ с начальным адресом R0 по адресу R1
142 MOV S6, @R1	
144 ADD#1, R0	Увеличение на единицу содержимого R0
146 ADD#1, R1	Увеличение на единицу содержимого R1
148 ADD#1, R2	Увеличение на единицу содержимого R2
150 BSA R28 R3	Если содержимое регистров R2, R3 одинаково, то останов программы. Если нет, то переход к метке 4
152 JMM 4	
154 STOP	

Если седьмой бит в байте, расположенным по адресу R0, будет нулевым, то на экране дисплея выводится нуль. В противном случае — единица
Запись в ОЗУ по адресу R10 кода единицы
Запись в ОЗУ по адресу R10 кода пробела
Вывод на экран изображения пробела
Увеличение на единицу содержимого регистра R0
Увеличение на единицу содержимого регистра R1
Если содержимое регистров R1, R2 не одинаково, то переход к метке 1, если равны, то перевод строки на экране дисплея
Обнуление регистра R1
Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Третья подпрограмма автоматически проверяет правильность записи 256 тетрад в ПЗУ. 256 тетрад, подлежащих записи, побитно сравниваются по логическому закону «Исключающее ИЛИ» с 256 тетрадами, считанными из ПЗУ по окончании записи. Результат проверки выводится на экран дисплея первой подпрограммой из области памяти микроЭВМ с начальным адресом, находящимся в регистре R4. При правильной записи на экране появятся 256 тетрад, все биты которых будут нулями. Единичные биты указут место ошибки программирования.

Подпрограмма контроля правильности ввода информации в ПЗУ

```

155 MARK 5
157 CLR R11
159 CLR R12
161 CLR R2
163 CLR X
164 STOP
165 MOvh X, R00
167 CLR X
168 STOP
169 MOvh X, R01
171 CLR X
172 STOP
173 MOvh X, R4
175 DIG2
176 DIG5
177 DIG6
178 MOvh X, R3
180 MARK 6
182 MOV @R0, S6
184 MOV @R1, S8
186 XOR S6, S8
188 MOV S8, @R4
190 ADD#1, R0
192 ADD#1, R1
194 ADD#1, R4

```

```

196 ADD#1, R2
198 BSA R2, R3
200 JMM 6

```

202 STOP

203 END

Программа считывания информации из ПЗУ позволяет размещать ее в любом месте оперативной памяти микроЭВМ, начальный адрес которой записан в регистре R2.

Программа считывания информации из ПЗУ и размещения ее в ОЗУ микроЭВМ «Электроника Д3-28» с начального адреса R2

```

0 DIG2
1 DIG5
2 DIG6
3 MOvh X, R4
5 CLR R11
7 CLR RI
9 MARK 7
11 MOV#0400, S2
13 MOV S7, S3
15 OUTOWS
17 INPOWC
19 MOV S3, @R2
21 ADD#1, R11

```

Пересылка байта из ОЗУ по адресу R0 в S6
Пересылка байта из ОЗУ по адресу R1 в S8
Сравнение S6, S8 по закону «Исключающее ИЛИ»
Пересылка результата сравнения по адресу R4

Увеличение содержимого регистров R0, R1, R4, R2 на единицу

Если содержимое регистров R2, R3 одинаково, то останов подпрограммы. Если нет, то переход к метке 6

Конец

Прием в регистр S3 слова ПЗУ
Пересылка слова ПЗУ по адресу R2 в ОЗУ
Увеличение на единицу содержимого регистра R11

23 ADD#1, R2

25 ADD#1, R1

27 BSA R4, R1

29 JMM 7

31 STOP

32 END

Увеличение на единицу содержимого регистра R2
Увеличение на единицу содержимого регистра R1
Если содержимое регистров R4, R1 одинаково, то конец программы. Если нет, то переход к метке 7

Конец

Приемма записи в любой области ОЗУ микроЭВМ с начальным адресом, находящимся в регистре R3, управляет и работой программатора в режиме записи (рис. 2).

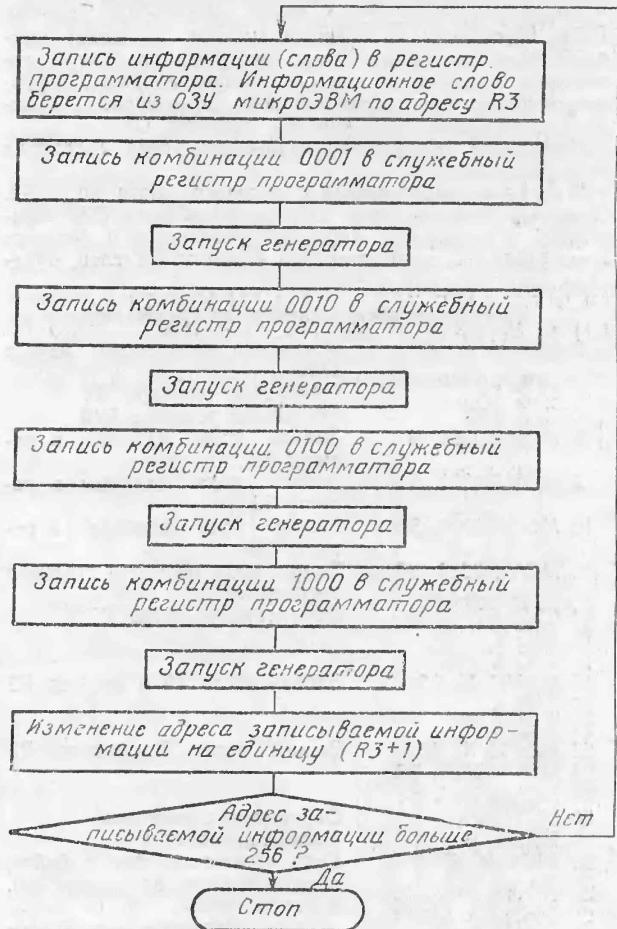


Рис. 2 Упрощенный алгоритм записи информации в ПЗУ

Программа записи информации в ПЗУ

0 DIG2	Запись числа 256 в регистр R4	Запись числа 256 в регистр R1
1 DIG5	Обнуление регистра R11	Обнуление регистра R11
2 DIG6	Обнуление регистра RI	Установка метки 8
3 MOvh X, R4	Установка метка 7	Запись 4-разрядного слова в информационный регистр программатора
5 CLR R11		
7 CLR RI		
9 MARK 7		
11 MOV#0400, S2		
13 MOV @R3, S2		
15 OUTOWS		
17 MOV#0700, S2		
19 MOV#0800, S3		
21 OUTOWS		
23 OUTO 0500		Запуск генератора

25 MOV #0400, S3

Запись комбинации 0010 в служебный регистр программатора

27 OUTOWS

29 OUTO 0500

31 MOV #0200, S3

33 OUTOWS

35 OUTO 0500

37 MOV #0100, S3

39 OUTOWS

41 OUTO 0500

43 ADD #1, R1

45 ADD #1, R3

47 ADD #1, R11

49 BSA R1, R4

УДК 681.3.06

А. Б. Глазов, С. А. Костарев, Е. В. Суханова

ЭФФЕКТИВНЫЕ ПРОГРАММЫ УМНОЖЕНИЯ ДЛЯ МИКРОПРОЦЕССОРА KP580ИК80А

При построении цифровых фильтров, реализации быстрого преобразования Фурье и других процессов обработки данных на микропроцессорных системах значительная часть времени обработки расходуется на выполнение умножения чисел. Ниже приводится несколько подпрограмм умножения чисел без знака различной разрядности для микропроцессора KP580ИК80А, имеющих малое время выполнения. «Время выполнения N тактов» означает, что программа выполняется не более чем за N тактов.

Основная идея приведенных алгоритмов заключается в одновременности сдвига результата произведения и одного из множителей. Подобный сдвиг в каждой подпрограмме оформлен в виде макрокоманды ROT и описан после соответствующей подпрограммы.

Ввиду краткости подпрограмм их регулярности комментарии к ним исчерпываются замечаниями, сделанными выше. Для начинающих программистов в дальнейшем комментарии сопровождают только макрокоманды. Из этих же соображений описание макрокоманд приводится после текста подпрограмм.

1. Подпрограмма умножения двух 8-разрядных чисел без знака.

Исходное положение: первый множитель находится в регистре И, второй в Е. Конечный результат: HL — произведение, D — 0, А, В, С, Е не изменяются.

MVI D,00	ROT	ROT
MOV L,D	ROT	ROT
DAD H	ROT	RNC
ROT	ROT	DAD D

Запуск генератора

Запуск комбинации 0100 в служебный регистр программатора

Запуск генератора

Запись комбинации 1000 в служебный регистр программатора

51 JMM 8

53 STOP

55 END

R4 одинаково, то конец программы. Если нет, то переход к метке 8.

Конец

Адрес для справок: 680028, г. Хабаровск, ул. Истомина, д. 88, кв. 53.

ЛИТЕРАТУРА

1. Коффрон Дж. Технические средства микропроцессорных систем. — М.: Мир, 1983.

2. Титие У., Шенк К. Полупроводниковая схемотехника. — М.: Мир, 1982.

3. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоаппаратуры на микропроцессорах. — М.: Радио и связь, 1984.

4. Григорьев В. Л. Программное обеспечение микропроцессорных систем. — М.: Энергоиздат, 1983.

5. Техническое описание микроЭВМ «Электроника Д3-28».

6. Справочник программиста для микроЭВМ «Электроника Д3-28».

Статья поступила 15 мая 1986 г.

Исходное состояние: А — 8-разрядный множитель, ВС — 16-разрядный множитель. Конечный результат: А, HL — произведение, Е — 00, В, С, D без изменений.

LXI H,0000	DAD B
MOV E, L	ADC E
ADD A	RET
ROT	ROT: MACRO
ROT	LOCAL Q
ROT	JNC Q
ROT	DAD B
ROT	ADC E
ROT	Q: DAD H
ROT	ADC A
RNC	END M

Время выполнения — 314 тактов. Данная подпрограмма незначительно отличается от описанной [3] (316 тактов) и в 1,5 раза быстрее — в [2] (475 тактов).

3. Подпрограмма умножения двух 12-разрядных чисел без знака.

Исходное состояние: С — старшие 8 разрядов первого множителя (в старшей тетраде D — младшие 4 разряда первого множителя, в младшей тетраде D — старшие 4 разряда второго множителя), Е — младшие 8 разрядов второго множителя. Конечный результат: А, HL — произведение, В — 0, старшая тетрада D — 0, содержимое С младшей тетрады D и Е не изменяется.

MVI B,00	ROT
MOV A,D	ROT
ANI F0	ROT
MOV H,A	ROT
MOV A,D	ROT
SUB H	RNC
MOV D,A	DAD D
MOV L,B	ADC B
MOV A,C	RET
DAD H	ROT: MACRO
ADC A	LOCAL Q
ROT	JNC Q
ROT	DAD D
ROT	ADC B
ROT	Q: DAD H
ROT	ADC A
ROT	END M

Замечание для ручного ассемблирования: в подпрограмму вместо каждого «ROT» вставляется только тело макрокоманды.

Время выполнения 257 тактов в 2 раза меньше времени выполнения одной из подпрограмм умножения, рассмотренной в [2] (520 тактов), и на 5 % меньше времени выполнения «быстрого» умножения с помощью таблиц произведений тетрад, описанного в [1, 2] (270 и 273 такта) и требующего громоздкой таблицы произведений, занимающей 256 байт в ПЗУ.

Наиболее удобной для системы команд микропроцессора KP580ИК80А является:

2. Подпрограмма умножения 8-разрядного числа без знака на 16-разрядное,