

**Автомобильная электроника:** БИС для управления бензиновым двигателем и построения систем контроля электроаппаратуры зажигания

**Двухкристальный микропроцессор** серии K1831 на основе КМОП БИС с микропрограммным управлением, совместимый по системе команд с микроЭВМ семейства «Электроника»

**ЭКСКОРТ-машина:** аппаратная поддержка системы программирования повышенного уровня

**Учебная микроЭВМ УМК-80М** — в лабораторном практикуме «Управляющие и микропроцессорные устройства роботов»

**СУБД для персональных ЭВМ,** совместимых с IBM PC, — обзор разработок зарубежных фирм

**Передача патентной информации** в составе ТВ-сигнала ускоряет внедрение изобретений в промышленность

## СВД-ВУЗ СВД-ВУЗ

**СИСТЕМА ВИРТУАЛЬНОГО ДОСТУПА  
ДЛЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ**



## СИСТЕМА ВИРТУАЛЬНОГО ДОСТУПА СВД-ВУЗ

Система предназначена для оборудования дисплейных классов, учебных кабинетов, залов курсового и дипломного проектирования, лабораторий высших учебных заведений. Вместе с тем может применяться и другими организациями для выполнения достаточно сложных вычислительных и научно-исследовательских работ, использующих принцип коллективного и равноправного пользования единым внешним оборудованием общего назначения.

Конструктивно система состоит из главной ЭВМ ДВК4 («Электроника МС 0507.04»), звездообразно соединенной с шестью учебными терминалами микроЭВМ («Электроника МС 0511») через шестиканальный интерфейс радиальной последовательной связи (ИРПС 6) с помощью витых пар проводов. Внешнее оборудование главной ЭВМ (общее для всех шести пользователей): накопитель на жестких магнитных дисках «Электроника МС 5401», накопитель на гибких магнитных дисках «Электроника МС 5305» и печатающее устройство «Роботрон СМ 6329.02М».

По сравнению с КУВТ-86 каждое рабочее место системы СВД-ВУЗ функционирует в режиме ПЭВМ, что позволяет использовать все ранее наработанное программное обеспечение для ЭВМ серии ДВК\* и дает широкие возможности при работе с компиляторами языков высокого уровня (Фортран, Паскаль, Си),

разделяемыми базами данных и полным набором утилит ОС RT-11.

Система обеспечивает одновременную (параллельную) загрузку с жестких дисков главной ЭВМ в микроЭВМ учебных терминалов полной ОС RT-11 с сохранением всех ее функций. Драйвер канала связи эмулирует устройство прямого доступа, обеспечивая прозрачность доступа к удаленным устройствам внешней памяти. При этом создается возможность интенсивной работы с файлами равноправно всеми пользователями, свободно обеспечивается запуск операционных программ, т. е. снимаются ограничения на размер оперативной памяти и расширяется круг решаемых задач.

Возможность использования ОС RT-11 и языков программирования Фортран IV, Паскаль, БЕЙСИК и др. без каких-либо ограничений функций позволяет создавать эффективные прикладные программы.

### Технические характеристики СВД-ВУЗ

Объем ОЗУ ЭВМ учебного терминала, Кбайт	192
Разрядность	16
Число рабочих (учебных) мест	6
Формат посылок в последовательном канале сети, бит	8
Скорость обмена в сети, бод	57600
Распределение объема (5 Мбайт) памяти жестких дисков главной ЭВМ, Кбайт:	
Операционная система и утилиты	150
Сетевое математическое обеспечение	100
Прикладное математическое обеспечение	250
Системы программирования:	
базы данных	250
Фортран IV	160
Паскаль	100
БЕЙСИК	30
Объем области каждого пользователя	660

Действующий образец СВД-ВУЗ используется в учебном процессе дисплейного класса Московского института тонкой химической технологии им. М. В. Ломоносова.

Госкомитет СССР по народному образованию и Министерство электронной промышленности утвердили техническое задание на разработку и производство СВД-ВУЗ.

Лимитная цена исполнения СВД-ВУЗ на шесть учебных мест составит не более 40 тыс. руб.

Объем серийного выпуска будет определяться числом заявок.

119831 ГСП, Москва, Г-435, М. Пироговская, д. 1



\* Корнюшко В. Ф., Авдеев В. Н., Фролов Г. М., Жедь А. Ю. МикроЭВМ в распределенной вычислительной системе // Микропроцессорные средства и системы. — 1988. — № 2. — С. 53—55.

ОРГАН  
ГОСУДАРСТВЕННОГО  
КОМИТЕТА СССР  
ПО ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКЕ  
И ИНФОРМАТИКЕ

Издается с 1984 года

# МП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 3/1989 МОСКВА

## СОДЕРЖАНИЕ

### МИКРОПРОЦЕССОРНАЯ ТЕХНИКА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Стоянов А. И., Хорошунов В. С., Ачкасов В. Н.— Двухкристальный микропроцессор серии K1831 . . . . .	2
Ненашев М. А., Петрыкин Ю. С.— Системы управления базами данных для персональных ЭВМ . . . . .	7
Христов П. В.— Архитектура ЭСКОРТ-машины и ее аппаратная поддержка . . . . .	13
Бельчинский И. Л.— Перенос ядра ИНМОС между машинами различных архитектур . . . . .	17
Антонов Ю. А., Бибило П. Н., Гольдберг Е. И., Каркоцкая И. П., Чигирь Н. П.— Система логического проектирования цифровых устройств на программируемых матричных БИС . . . . .	22
Андреев С. В., Ходулев А. Б.— Вывод графической информации на точечно-матричные печатающие устройства . . . . .	24
Мельник В. Е.— ДВК для отладки и контроля цифровых устройств . . . . .	27
Серебренников О. Ю., Шевченко Е. Г., Леонидов-Каневский Е. В.— Автоматизированная система диагностики технического и программного обеспечения микропроцессорных систем . . . . .	30
Гейман Л. М.— Этапы развития информатики как системы знаний . . . . .	31
Гиглавый А. В., Немец А. В.— Освоение технологии САПР БИС в КНР . . . . .	34
Микросхема РЗУ K573РФЗ . . . . .	37

### САПР

### Системы отладки

### Общие вопросы развития информатики Справочная информация ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Исаев А. Н., Блииников В. И., Кривошеев М. И., Сарьян В. К., Красносельский И. Н., Боловинцев Ю. М.— Передача патентной информации в составе телевизионного сигнала . . . . .	38
Горбачев В. С., Полубояринов О. Д.— Использование учебной микроЭВМ УМПК-80М в лабораторном практикуме по робототехнике . . . . .	42
Горбачев В. С., Полубояринов О. Д., Зуев М. В., Гительсон А. В.— MINI-VAL: язык автономного программирования учебного робота «РОБКО-01» . . . . .	47
Львов К. В., Гирба И. А.— Подключение к ЭВМ «Электроника ДЗ-28» устройств с последовательным интерфейсом ИРПС . . . . .	48
Джиган В. И.— Линия связи управляющей микроЭВМ с модулями антенной решетки . . . . .	50
Казанцев А. П., Майоров Л. Н., Данилов А. Б.— Указатель информации и интерфейс цветного телевизора для микроЭВМ «Электроника БК-0010» . . . . .	54
Казанцев А. П., Данилов А. Б.— Интерфейс пользователя ПЭВМ «Электроника БК-0010» . . . . .	57
Полянский П. В.— «Электроника БК-0010» в системах исследования объектов с распределенными параметрами . . . . .	58
Монахов В. Т.— Программное обеспечение ПЭВМ «Электроника БК-0010» . . . . .	60
Овчаренко А. И., Пушко П. В., Хилобок В. И., Зверев В. И.— Микропроцессорная система контроля датчиков зажигания . . . . .	63
Овчаренко А. И.— Применение таймера КР580ВВ153 для измерения временных интервалов при исследовании датчиков зажигания . . . . .	64
Овчаренко А. И., Хилобок В. Е.— Преобразователь параметров датчиков-распределителей зажигания с интерфейсом КТС ЛИУС-2 . . . . .	66
Лангуев В. В., Ольшак А. И., Гаврилюк С. Ю.— Процессор КМ1823ВУ1 . . . . .	69
Лангуев В. В., Болотов С. А., Трусов В. А.— Устройство ввода-вывода КМ1823ВВ1	72
Бобков В. А., Лангуев В. В., Болотов С. А., Ольшак А. И., Трусов В. А., Гаврилюк С. Ю.— Контроллер микропроцессорной системы зажигания автомобильного двигателя «Электроника МС2713» . . . . .	74
Дудихин В. В., Меньшикова Е. Г., Смирнов Н. В.— Программное обеспечение микропроцессорного контроллера комплекта группового вождения тракторов	76
Пройдаков Э. М.— Введение в мониторы микроЭВМ . . . . .	78
Дядькин А. И., Кушнир В. Е., Панфилов Д. И., Щаронин С. Г.— Способы построения и исследования интерфейсов устройств ввода и отображения информации микроконтроллеров на примере БИС КР580ВВ79 . . . . .	81

### Опыт применения ПЭВМ «Электроника БК-0010»

### Автомобильная электроника

### УЧЕБНЫЙ ЦЕНТР

УДК 621.3.049

А. И. Стоянов, В. С. Хорошунов, В. Н. Ачкасов

## ДВУХКРИСТАЛЬНЫЙ МИКРОПРОЦЕССОР СЕРИИ К1831

Микропроцессор (МП) серии К1831, состоящий из двух СБИС, выполненных по КМОП-технологии, представляет собой дальнейшее развитие многокристалльных МП серий К581 и К1811 с микропрограммным управлением. Сохраняет совместимость по системе команд с микроЭВМ семейства «Электроника». Ниже приведены основные технические характеристики МП.

Состав . . . . .	СБИС обработки данных и управления памятью КН1831ВМ1; СБИС микропрограммного управления КН1831ВУ1
Разрядность . . . . .	16
Объем адресуемой памяти, Мбайт . . . . .	4
Число команд . . . . .	140 (46 команд с плавающей запятой)
Число уровней прерывания . . . . .	4
Выстродействие, операция/с . . . . .	$3,5 \cdot 10^6$
Тактовая частота, МГц . . . . .	15
Время цикла, нс . . . . .	260
Потребляемая мощность, Вт, не более . . . . .	1
Напряжение источника питания, В . . . . .	5
Допустимые отклонения напряжения питания, В . . . . .	$\pm 0,25$

Шина адреса и данных — совмещенная. Микросхемы выпускаются в 84-контактных безвыводных металлокерамических корпусах и размещаются на многослойной 60-выводной керамической плате, обеспечивающей соединение выводных контактов СБИС между собой и подключение к внешним схемам. При этом МП имеет 59 внешних задействованных выводов, включая выводы питания и нулевого потенциала.

### Микросхема обработки данных и управления памятью КН1831ВМ1

Предназначена для выполнения логических и арифметических операций над данными и управления памятью. Условное графическое обозначение микросхемы приведено на рис. 1, назначение выводов показано в табл. 1. Структурная схема (рис. 2) включает следующие основные функциональные блоки и узлы: буферные схемы шины микрокоманд; логику управления и синхронизации; блок регистров общего назначения (РОН); арифметическое логическое устройство (АЛУ); буферные схемы шины адресов и данных; регистры системные плавающей запятой; регистры устройства управления памятью; устройство управления памятью (УПП).

Управление работой СБИС и взаимодействие с микросхемой КН1831ВУ1 осуществляется с помощью шин микрокоманд (ШМК), адреса и данных (ШАД), управления.

Информация, поступающая по ШАД, заносится в блок РОН, осуществляющий временное хранение данных и оперативную передачу АЛУ, внутренним регистрам и внешним схемам.

При этом буферные схемы ШАД обеспечивают предвыборку информации и конвейерный режим обработки. Блоки РОН и АЛУ имеют 32-разрядную структуру, что существенно повы-

шает эффективность обработки информации. Регистровый блок содержит восемь 32-разрядных регистров, доступных микропрограммно; две группы по шесть 16-разрядных РОН; регистры указателя стека; регистр слова состояния процессора; программный счетчик; регистр слова состояния УПП.

Программный счетчик содержит 16-разрядный адрес очередной выполняемой команды. Выбор одной из групп РОН определяется информацией регистра слова состояния процессора, отражающего текущий и предыдущий режимы работы процессора, используемую группу регистров, текущий уровень приоритета и коды условий последней выполняемой команды.

Регистровый блок имеет двухканальную (двухпортовую) структуру, один из которых (В-порт) предназначен только для чтения, а второй (А-порт) может использоваться для записи и чтения информации. Такая структура регистрового блока позволяет иметь минимум регистров, необходимых для эмуляции системных команд, и сократить время выполнения операций.

Порты А и В связаны с внутренней ШАД. Через эту шину осуществляется обмен информацией с внешними схемами, АЛУ и другими внутренними блоками.

АЛУ выполняет арифметические и логические операции, операции циклического и арифметического сдвигов вправо-влево, пересылки, проверки, установки и другие. Информация, под-

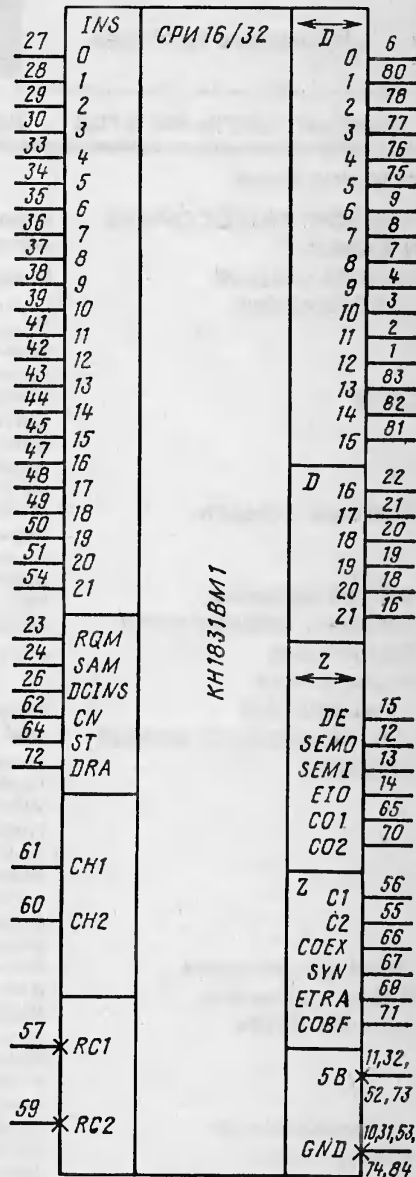


Рис. 1. Условное графическое обозначение микросхемы КН1831ВМ1

лежащая обработке, поступает в АЛУ по шинам А- и В-портов регистрового блока. Результат операции записывается в регистр по адресу А-порта.

Операция в АЛУ и размещение результата в регистровом блоке осуществляются за один тактовый цикл. Входной информацией для АЛУ может служить содержимое двух регистров микрокоманд или одного регистра и литерала (вид данных, содержащихся в микрокоманде) для литеральных микрокоманд.

Управление режимом работы АЛУ, а также всех других внутренних функциональных узлов и блоков осуществляет логика управления и синхронизации посредством внутренних шин микрокоманд и управления. Во время выполнения первой команды очередная

поступает из буфера в регистр команд и декодируется, а следующая загружается в буфер предвыборки, что значительно повышает эффективность обработки информации.

В выполнении операций с плавающей запятой участвуют 12 32-разрядных регистров плавающей запятой, в том числе регистр состояния плавающей запятой. Обмен информацией с регистрами-аккумуляторами плавающей запятой осуществляется по внутренней ШАД.

Устройство управления памятью, содержащее соответствующие регистры и логику управления, обеспечивает преобразование 16-разрядных вирту-

альных адресов в 22-разрядные физические.

В состав УУП входят регистры состояния УУП, описания страницы, адреса страницы, управления предвыборкой и др. Все регистры адресуются микропрограммно, а часть из них, кроме того, системными адресами (кодами).

Логика управления памятью включает виртуальный программный счетчик, мультиплексор виртуальных адресов, сумматор, схему сравнения, дешифратор системных адресов внутренних регистров. Непосредственное формирование физических адресов осуществляется сумматором устройства управления памятью.

Таблица 1

Назначение выводов микросхемы КН1831ВМ1

Вывод	Условное обозначение	Назначение
1...4	D12...D9	Входы-выходы данных
5	—	Не используется
6...9	D0...D6	Входы-выходы данных
10	GND	Общий
11	5B	Напряжение источника питания
12, 13	SEM0, SEM1	Выходы выбора памяти
14	EIO	Выход разрешения ввода-вывода
15	DE	Вход-выход запрета
16	D21	Выход данных
17	—	Не используется
18...22	D20...D16	Выходы данных
23	RQM	Вход запроса памяти
24	SAM	Вход состояния памяти
25	—	Не используется
26	DCI	Вход дешифрации инструкций
27...30	INS0...INS3	Входы микроинструкций
31	GND	Общий
32	5B	Напряжение источника питания
33...39	INS4...INS10	Входы микроинструкций
40	—	Не используется
41...45	INS11...INS15	Входы микроинструкций
46	—	Не используется
47...51	INS16...INS20	Входы микроинструкций
52	5 B	Напряжение источника питания
53	GND	Общий
54	INS21	Вход микроинструкций
55	C2	Выход такта
56	C1	Вход-выход такта
57	RCI	Вывод для подключения R и C
58	—	Не используется
59	RCO	Вывод для подключения R и C
60, 61	CH2, CH1	Входы контроля
62	CN	Вход продолжения
63	—	Не используется
64	ST	Вход «Старт»
65	CO1	Вход управления
66	COEX	Выход управления расширением
67	SYN	Выход синхронизации
68	—	Не используется
69	ETRA	Выход разрешения захвата адреса
70	CO2	Выход управления
71	COBF	Выход управления буфером
72	DRA	Вход готовности данных
73	5 B	Напряжение источника питания
74	GND	Общий
75...78	D5...D2	Входы-выходы данных
79	—	Не используется
80...83	D1, D15...D13	Входы-выходы данных
84	GND	Общий

Микросхема микропрограммного управления КН1831ВУ1

Предназначена для управления работой МП и генерации микропрограмм, эмулирующих систему команд. Условное графическое обозначение микросхемы приведено на рис. 3, назначение выводов — в табл. 2. Структурная схема микросхемы КН1831ВУ1 представлена на рис. 4. Микросхема состоит из следующих основных функциональных блоков и узлов: контроллеров адреса микрокоманды и прерываний, микропрограммной памяти, логики управления и синхронизации, логики управления ветвлением, буферных схем.

После включения питающего напряжения выполняется начальная программа подготовки, в процессе которой осуществляется установка всех блоков микросхемы в рабочее состояние. Все микрокоманды, кроме ввода-вывода, выполняются за четыре периода тактового сигнала, составляющих цикл микрокоманды. Микрокоманды ввода-вывода выполняются минимум за два цикла, кроме цикла «быстрого» чтения, занимающего один цикл. Время цикла микрокоманды определяется кодом микрокоманды, а также кодом внешних управляющих сигналов.

Взаимодействие всех узлов и блоков микросхемы осуществляется следующим образом: 32-разрядная микрокоманда считывается из микропрограммной памяти, поступает на внутреннюю ШМК, и через буферные схемы 22-разрядное операционное поле передается на внешнюю шину микрокоманд.

Микрокоманда, поступающая на логику управления, синхронизации и управления ветвлением, декодируется и вырабатывает соответствующие управляющие сигналы, определяющие режим работы микросхемы в следующем цикле, а также внешние управляющие сигналы, определяющие тип следующего цикла обработки информации.

Логикой управления ветвлением формируется и передается на внутреннюю 10-разрядную шину адреса микрокоманды адресное поле микрокоманды

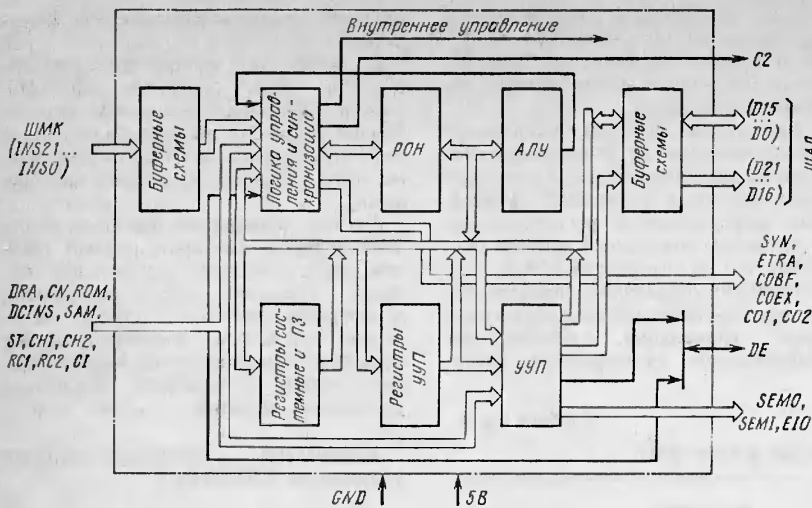


Рис. 2. Структурная схема микросхемы КН1831ВМ1

ветвления. Адрес выбранной микрокоманды поступает в контроллер адреса, который может передать его в шину адреса микрокоманды, сформировать модифицированный адрес или сохранить адрес, установленный логикой управления ветвлением. Вычисленный адрес микрокоманды передается в блок микропрограммной памяти, адресуя очередную микрокоманду.

Если в микропрограммном цикле происходит ввод информации по шине данных D15... D0, то она декодируется дешифратором кода ШАД и поступает в блок микропрограммной памяти, иницируя вместе с текущим адресом выборку очередной микрокоманды. Эта информация используется также логикой управления и синхронизации для формирования соответствующих управляющих сигналов.

Данные поступают, кроме того, в контроллер прерываний, обеспечивающий обработку запросов прерываний. После выполнения очередной команды данные о запросе прерывания разрешенного приоритетного уровня передаются для организации выборки первой микрокоманды подпрограммы обслуживания прерывания.

Логика управления и синхронизации при выполнении последней микрокоманды микропрограммы эмуляции текущей команды формирует активный уровень сигнала DCINS — начало выполнения следующей команды.

Все операции, связанные с формированием адреса микропрограммной памяти, в том числе и в случае операций ветвления, а также вывода и дешифрации микрокоманд, осуществляются в конвейерном режиме и не требуют удлинения микрокомандного цикла. В конвейерном режиме обработки происходит также ввод команд и данных. Запросы прерываний анализируются в каждом микрокомандном цикле, а предоставляется прерывание только по

Рис. 3. Условное графическое обозначение микросхемы КН1831ВУ1

окончании выполнения текущей команды.

Архитектура микросхемы позволяет использовать в составе МП до 32-х микросхем микропрограммного управления, что обеспечивает возможность наращивания системы команд. Управление между микросхемами передается посредством специальных микрокоманд передачи управления, имеющих соответствующее адресное поле «выбора кристалла».

#### Работа микропроцессора

Взаимодействие микросхемы обработки данных и управления памятью с микросхемами микропрограммного управления осуществляется с помощью двух 22-разрядных шин: ШАД и ШМК. ШАД — совмещенная шина; 16 ее разрядов используется для передачи

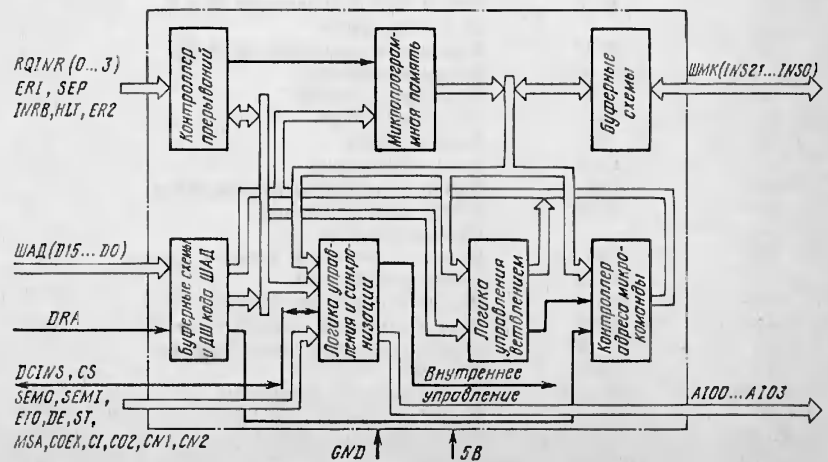
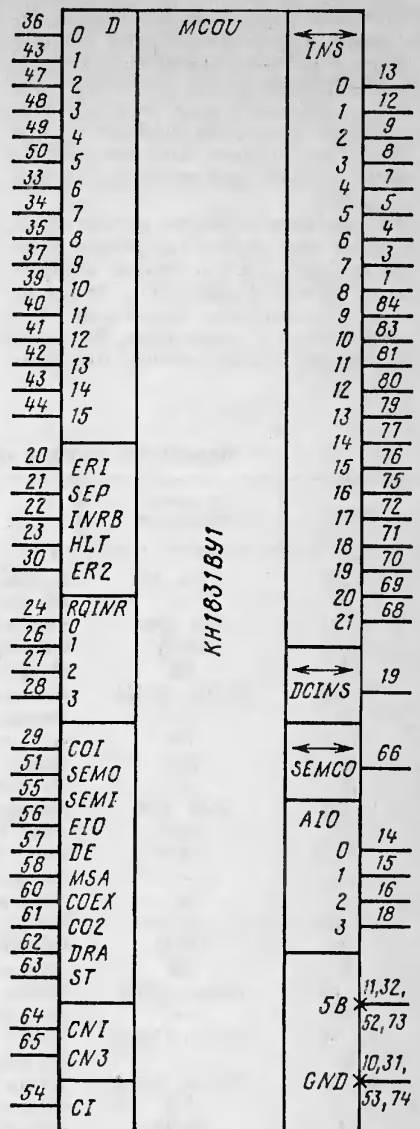


Рис. 4. Структурная схема микросхемы КН1831ВУ1

Назначение выводов микросхемы КН1831ВУ1

Вывод	Условное обозначение	Назначение
1	INS8	Вход-выход инструкции
2	—	Не используется
3...5	INS7...INS5	Входы-выходы инструкции
6	—	Не используется
7...9	INS4...INS2	Входы-выходы инструкции
10	GND	Общий
11	5 В	Напряжение источника питания
12, 13	INS1, INS0	Входы-выходы инструкции
14...16	A100...A102	Выходы, входы-выводы адреса
17	—	Не используется
18	A103	Выход, ввод-вывод адреса
19	DCI	Вход-выход дешифрация инструкции
20	ER1	Вход «Ошибка 1»
21	SEP	Вход «Выбор процессора»
22	INRB	Вход «Прерывания шины»
23	HLT	Вход «Останов»
24	RQINP0	Вход «Запрос прерывания 0»
25	—	Не используется
26...28	RQINR1...RQINR3	Входы «Запрос прерывания 1, 2, 3»
29	CO1	Вход «Управление 1»
30	ER2	Вход «Ошибка 2»
31	GND	Общий
32	5 В	Напряжение источника питания
33...35	D6...D8	Входы данных
36, 37	D0, D9	Входы данных, разряды 0 и 9
38	—	Не используется
39...44	D10...D15	Входы данных
45	D1	Вход данных, разряд 1
46	—	Не используется
47...50	D2...D5	Входы данных
51	SEM0	Вход «Выбор памяти 0»
52	5 В	Напряжение источника питания
53	GND	Общий
54	C1	Вход «Такт 1»
55	SEM1	Вход «Выбор памяти 1»
56	EIO	Вход «Разрешение ввода-вывода»
57	DE	Вход «Запрет»
58	MSA	Вход «Состояние памяти»
59	—	Не используется
60	COEX	Вход «Управление расширением»
61	CO2	Вход «Управление 2»
62	DRA	Вход «Готовность данных»
63	ST	Вход «Пуск»
64	CH1	Вход «Контроль 1» (технологический)
65	CH3	Вход «Контроль 3» (технологический)
66	SEMCO	Вход-выход, выбор управляющей памяти
67	—	Не используется
68...72	INS21...INS17	Входы-выходы инструкции
73	5 В	Напряжение источника питания
74	GND	Общий
75...77	INS16...INS14	Входы-выходы инструкции
78	—	Не используется
79...81	INS13...INS11	Входы-выходы инструкции
82	—	Не используется
83, 84	INS10, INS9	Входы-выходы инструкции

адресов и данных (D15... D0), старшие 6 разрядов (D21...D16) — только для вывода адреса и имеет однонаправленную структуру.

ШМК — внутренняя шина МП. Для связи с внешними схемами кроме ШАД используется шина управления, включающая широкий набор управляющих сигналов.

Выполнение команды начинается с

операции выборки, для чего в ШАД выводится адрес данной команды. После поступления и дешифрации команды выполняются операции выборки операндов с использованием соответствующих методов адресации. Затем реализуется операция собственно использования данной команды.

Физический адрес, устанавливаемый во второй половине цикла, формирует-

ся по всем 22 разрядам ШАД. По младшим 16-ти разрядам данной шины осуществляется обмен информацией между МП и внешними схемами. По старшим разрядам в это время формируется служебная информация.

По ШМК поступают микрокоманды, обеспечивающие выполнение текущей команды, а на шине управления формируются сигналы, управляющие работой МП и обменом информацией с внешними схемами. Команды МП делятся на безадресные, одноадресные, двухадресные.

В безадресных командах содержится только код операции. В кодах одно- и двухадресных команд обычно содержится информация, которая определяет: выполняемую функцию (код операции), адрес РОИ, используемых при выборе операндов, метод адресации (способ использования выбранного РОИ).

Регистры общего назначения могут быть использованы в следующем качестве:

- накопителей (обрабатываемые данные хранятся в регистрах);
- указателей адреса (РОИ содержит адрес операнда, а не сам операнд);
- указателей адреса, содержимое которых изменяется автоматически с шагом 1 или 2, что позволяет обратиться к последовательно расположенным ячейкам памяти. При этом автоматическое увеличение содержимого указателя при обращении к последовательно расположенным ячейкам памяти называется автоинкрементной, а уменьшение — автодекрементной адресацией;

индексных регистров, содержимое которых прибавляется к индексному слову для вычисления адреса операнда.

Применяются следующие методы прямой адресации:

- регистровой — операнд находится в выбранном РОИ, который используется как накопитель;
- автоинкрементной, обеспечивающей обращение по очередному адресу;
- автодекрементной — для последовательной обратной адресации;
- индексной — для формирования произвольного адреса.

В режиме косвенной адресации содержимое выбранных регистров является не операндом, а адресом. Методы косвенной адресации используются совместно с основными методами адресации.

В режиме адресации с использованием программного счетчика (счетчика команд) информация данного регистра влияет на выборку очередного слова из памяти, после чего содержимое регистра автоматически увеличивается на два и является уже адресом следующей команды (или очередным адресом нового слова, используемого при выполнении текущей команды).

Автоинкрементный и автодекрементный методы адресации позволяют организовать стековую память. В ка-

честве указателя стека может быть выбран любой программно доступный РОН, но используется, как правило, специальный регистр, указывающий адрес части памяти, отводимой под стек. С помощью такой адресации информация записывается в стек или происходит последовательная выборка. Возможна также и произвольная выборка информации посредством индексной адресации. Стек используется для оперативного доступа к наиболее часто требуемым данным.

32-разрядная внутренняя структура АЛУ и информационного тракта, наличие аппаратных средств умножения и сдвига обеспечивают эффективную

обработку информации: большинство операций выполняется за один микрокомандный цикл, равный 260 нс.

Высокую производительность МП определяет конвейерный режим обработки информации на всех уровнях: последняя микрокоманда текущей команды декодируется и немедленно передается следующая команда из буфера предварительной выборки в регистр команд микросхемы обработки данных и управления памятью. Одновременно микросхема микропрограммного управления начинает дешифрацию новой команды и формирование соответствующей микропрограммы, управ-

ляющей работой микросхемы обработки данных и управления памятью.

Во время выполнения микросхемой обработки данных и управления памятью текущей микрокоманды микросхема микропрограммного управления формирует следующую микрокоманду и выдает ее в ШМК, исключая ожидание обработки данных. Параллельно с обработкой данных и дешифрацией команд реализуется предварительная выборка команд, очередная команда загружается в буфер предварительной выборки, что также исключает ожидание считывания очередной команды.

Статья поступила 7.07.88

## КРАТКОЕ СООБЩЕНИЕ

УДК 681.5:681.32—181.48

Е. А. Коноплев

### ПРОГРАММАТОР РПЗУ С УЛЬТРАФИОЛЕТОВЫМ СТИРАНИЕМ

Для записи информации в репрограммируемые ПЗУ (РПЗУ) с ультрафиолетовым стиранием применяются серийно выпускаемые программаторы. Пользователь, имеющий в своем распоряжении микроЭВМ, может воспользоваться ее ресурсами для создания несложного программатора, максимально использующего все возможности ПЭВМ.

Наиболее простая схема программатора — для пересылки информации между микроЭВМ и БИС РПЗУ используется параллельный интерфейс ИРПР (стандартное средство связи с печатающим устройством почти всех отечественных микроЭВМ).

В этом случае программатор представляет собой счетчики адреса, одновибратор и контактирующие устройства для различных типов БИС РПЗУ (рис. 1).

При включении питания программатора (достаточно подавать +5 В, напряжение +25 В генерируется на плате программатора) счетчики сбрасываются в состояние, соответствующее нулевому адресу БИС РПЗУ. Инкрементирование адреса счетчиков обеспечивается логической схемой задержки при окончании цикла записи по текущему адресу. При этом микроЭВМ освобождается от работы по пересылке адреса (для 8-битового канала ИРПР затруднительно).

Работа с программатором, подключенным к микроЭВМ вместо печатающего устройства, происходит в режиме диалога при запуске программы PROM, написанной на языке Паскаль (осуществляется переносимость программы на уровне языка).

Программа занимает около 10 Кбайт динамической памяти и обеспечивает:

выбор БИС РПЗУ с УФ-стиранием из следующего ряда: К573РФ2, К573РФ4, К573РФ5, К573РФ6, К573РФ7, К1816ВЕ48;

выбор режима работы с БИС РПЗУ: чтение/запись/верификация;

подготовку и запись файла данных на магнитный носитель для последующей записи на БИС РПЗУ;

работу по одиночным адресам с любой из выбранных БИС РПЗУ. Алгоритм программы представлен на рис. 2.

Телефон 484-95-17, Киев

Сообщение поступило 26.02.88

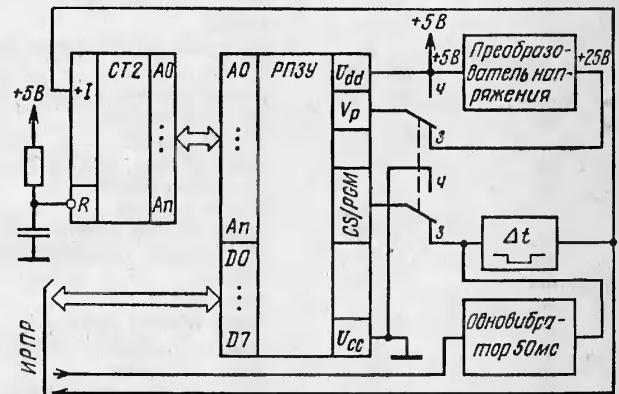


Рис. 1. Функциональная схема программатора РПЗУ

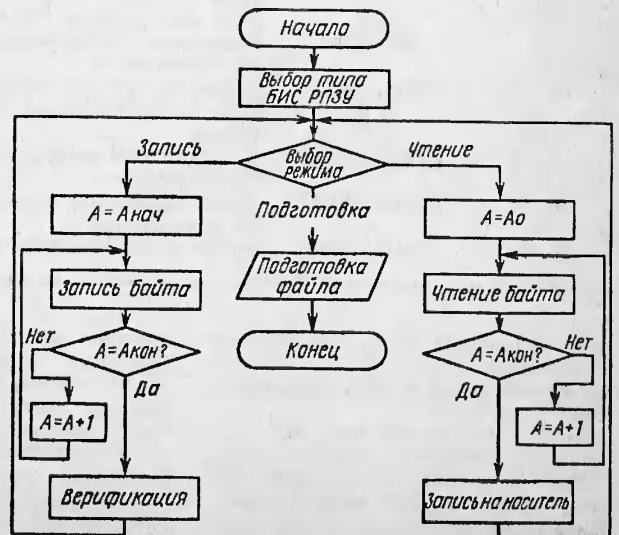


Рис. 2. Алгоритм работы программатора



УДК 681.3.06

М. А. Ненашев, Ю. С. Петрыкин

## СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ДЛЯ ПЕРСОНАЛЬНЫХ ЭВМ

Одними из наиболее развитых и широко используемых программных продуктов стали системы управления базами данных (СУБД). К настоящему времени, например, число высокоразвитых СУБД для IBM PC превышает двести наименований [1].

Видимо, в ближайшие годы отечественные ПЭВМ типа ЕС1840, «Искра 1030», «Нейрон И9.66», программно совместимые с IBM PC, станут основными машинами этого класса в нашей стране. Поэтому необходимо оценить нарабатанное зарубежными фирмами ПО для подобных ЭВМ.

**Структуры СУБД.** Основные модели организации баз данных — это иерархическая, сетевая и реляционная. Возрастающие требования к различным аспектам представления данных (неизбыточность, независимость, взаимосвязанность, защищенность и пр.) привели к тому, что в последнее время наиболее интенсивно развиваются исследования по сетевой и реляционной моделям [4]. Подавляющее число СУБД (для ПЭВМ) используют реляционный подход к построению данных в силу описанных ниже причин.

Сетевая модель обеспечивает высокую эффективность создаваемых на ее основе информационных систем, но требует от пользователя спецификации структур хранения, способов доступа, логических структур данных. Это снижает гибкость системы и усложняет базу данных.

При реляционном подходе к построению баз данных упрощается структура логических и физических баз данных, отпадает необходимость предсказания способов доступа, появляется возможность обработки непредсказуемых запросов. Достоинства этого подхода — а) простота представления данных (в виде двумерных таблиц, теоретически можно использовать таблицы и большей размерности); б) гибкость (операции процеивования и соединения позволяют получать таблицы данных в нужной форме); в) точность (отношения по своей природе обладают более точным смыслом, чем направленные связи, и поддаются математически точным методам манипулирования); г) простота контроля секретности; д) независимость данных от прикладных программ; е) простота ведения и модификации данных. Основным недостатком реляционных баз данных — низкая эффективность в ряде случаев использования ресурсов вычислительной системы. Однако по мере снижения стоимости обработки информации на ЭВМ и одновременного увеличения затрат на программирование привлекательность реляционного подхода к построению баз данных возрастает.

**Краткая характеристика лучших СУБД.** Набор пакетов для организации баз данных условно разделим на три большие группы:

I. Карточечные системы, позволяющие в одном сеансе пользоваться только одним файлом данных и обеспечивающие только последовательный доступ к этим данным. Они предназначены для автоматизации неспециалистами простых функций, например хранения списка оборудования на складе небольшого предприятия.

II. СУБД, реализующие одну из моделей организации баз данных. Как мы уже указывали, в подавляющем большинстве случаев это системы, основанные на реля-

ционной модели. Пользователь такой СУБД может построить существенно более гибкую и удобную автоматизированную систему, чем при помощи картотечного пакета. Распределение информации на несколько взаимосвязанных файлов позволяет избежать дублирования данных, индексирование дает возможность быстрой выборки и т. д.

III. Программируемые СУБД, предоставляющие пользователю не только стандартные средства манипулирования данными, но и проблемно-ориентированный язык для создания приложений с нестандартными функциями в соответствии с конкретной задачей. Хотя многие из имеющихся пакетов этой группы можно использовать и без дополнительного программирования, полное их освоение под силу только профессионалу.

Разнообразие задач по разработке автоматизированных информационных систем определяет повышенный интерес к СУБД третьей группы. Ниже кратко описаны наиболее развитые из них.

1. **dBase III Plus.** Самая оптимальная по сочетанию мощности и удобства использования, а потому и самая распространенная — реляционная СУБД dBase III компании Ashton-Tate. Последняя (лето 1986 г.) версия этой системы под названием dBase III Plus обладает еще большими возможностями, включая обеспечение доступа в локальной сети (т. е. имеются команды защиты файлов и записей от одновременного обращения нескольких пользователей), более быстрые процедуры сортировки и индексации, около 50 новых команд и функций и мощный пользовательский интерфейс, позволяющий выполнить практически любые действия над данными, не прибегая к собственному языку системы. Например, построить критерий фильтрации просматриваемых записей, установить взаимосвязь между файлами, сгруппировать файлы одного банка, задать формат выдачи данных на экран или устройство печати и многое другое. Компания сделала свой продукт доступным для любого пользователя: исключительно подробно (около 1800 страниц!) документированы имеющиеся возможности и методы, организованы оперативный справочник в самой СУБД и отдельная интерактивная обучающая программа со многими примерами.

Помимо достаточно развитого языка программирования запросов и (ограниченного) выхода на ассемблер, имеются удобные средства отладки программ — пошаговое исполнение, сохранение так называемой «истории» выполнения команд, возможность приостановить программы для проверки значений переменных и т. д. Можно кодировать текст программ для защиты их от чтения или модификации. Таким способом самой фирмой защищен имеющийся в пакете генератор прикладных программ (APPSGEN).

В целом, dBase III Plus проще в использовании и значительно шире по возможностям по сравнению с предшествующими версиями. Им могут пользоваться новички в области обработки данных и разработки СУБД. Большой набор вспомогательных программ для пользователей этого пакета, включая компиляторы и готовые прикладные системы, выпущен разными фирмами. dBase определяет часть символов из верхней половины расширенной таблицы ASCII как управляющие и требует доработки при необходимости хранения данных на русском языке.

Помимо приведенных в табл. 1, 2 показателей, отметим наличие в dBase III Plus следующих удобных средств: стандартные поля — символьное, числовое, логическое, дата, время, текст (до 512 Кбайт);

проверка-обработка данных при вводе — по числовому диапазону, преобразование регистра (первые 128 символов

Характеристика	KMap/2	dBase III Plus	DDQuery	filePRO 16 Plus	Goldata Base	Informix-SQL	Mainstay
Минимальный объем требуемой оперативной памяти	320	256	384	256	192	512	512
Минимальный объем внешнего ЗУ	750	150	323	2100	650	905	1300
Число полей в записи	255	128	255	999	300	H/O	H/O
Максимальное число символов в записи	64 000	4000	4096	16 000	16 000	32 000	H/O
Максимальное число записей в файле	H/O	1 млрд.	2 млрд.	16 млн.	H/O	H/O	H/O
Максимальное число одновременно открытых файлов	H/O	10	9	20	20	13	H/O
Максимальное число записей в базе	H/O	10 млрд.	2 млрд.	16 млн.	H/O	H/O	H/O
Максимальное число символов в поле	64 000	254	4096	999	80	32 000	H/O
Максимальное число одновременно обрабатываемых таблиц	H/O	2	5	10	10	13	H/O
Максимальное число индексных файлов	H/O	7	7	18	H/O	H/O	H/O
Максимальное число действующих полей при сортировке	H/O	7	32	7	20	8	H/O
Число стандартных типов полей	8	6	5	8	5	8	7
Число форматов импорта данных	3	8	3	4	—	1	3

Характеристики времени обработки

Характеристика времени обработки	KMap/2	dBase III Plus	DDQuery	filePRO 16 Plus	Goldata Base	Informix-SQL	Mainstay
Сортировка тестового файла (500 записей), с	57	14,2	—	210	241	40	14
Индексирование тестового файла, с	33	10,7	—	450	305	2,5	16
Выборка данных по индексу, с	1	0,1	—	1,5	0,5	0,1	18
Последовательная выборка, с	2	3,7	—	8	4,5	2,5	13,5
Распечатка всего файла в виде таблицы, мин	4,3	8,1	3,3	4,0	3,0	2,0	6,0

ASCII), запрет повторов, перенос данных из предыдущей записи, пользовательская обработка ошибок; преобразование формата представления данных — ASCII, DIF, DBF, PFS, SDF, SYLK, WKS, пользовательский формат;

оперативное изменение структуры файла данных (длина/тип поля, добавление/удаление полей);

пакетная обработка групп записей; наличие математических функций (ABS, MIN, MAX, LOG, SQRT, MOD, STR, CHR и т. д.) и функций обработки строк.

На сегодняшний день dBase III — это фактически стандарт реляционных СУБД, что привело к появлению большого числа вспомогательных продуктов (трансляторы, генераторы программ, оконечные интерфейсы и пр.) и ряда пакетов-имитаторов. Наиболее известны следующие пакеты:

— Clipper, Autumn'86 фирмы Nantucket Corp. — полная реализация dBase III в виде транслятора, дающая возможность генерировать программы в загружаемом формате. Это не только закрывает доступ к исходным текстам (что само по себе важно для разработчиков ПО), но и в 2—3 раза повышает скорость обращения к базе данных, исправляя, таким образом, основной недостаток dBase III. Хотя Clipper можно использовать как независимый продукт, гораздо удобнее составлять и отлаживать программу в режиме интерпретации при помощи dBase III, а затем

транслировать текст полученной программы, получая обычный объектный модуль. После обработки редактором связей (стандартным или поставляемым вместе с Clipper) этот модуль можно запускать из ОС, как любой другой EXE файл.

— dBase III / Compiler фирмы WordTech Systems Inc. (основной конкурент Clipper'a) — псевдокомпилятор, дающий промежуточный «d-код», для работы которого требуется интерпретатор низкого уровня. Конечная программа получается не такой быстродействующей, как EXE файл, но позволяет более гибко распределять оперативную память. Кроме того, dBase III / Compiler включает в себя удобный отладчик, а также позволяет использовать полностью совместимые с dBase III индексные файлы, тогда как Clipper создает индексные файлы в своем формате.

— СУБД FoxBASE Plus фирмы Fox Software Inc. является, пожалуй, самой близкой имитацией dBase III Plus на уровне программ и файлов данных, выгодно отличающаяся по своему быстродействию, более удобным средствам программирования и работы с взаимосвязанными файлами, алгоритмам сетевого доступа и компактности при существенно меньшей цене. Сам пакет реализует режим интерпретации, как и dBase, но в базовом наборе имеется и компилятор, программы, написанные при помощи FoxBASE, можно без всяких изменений обрабатывать Clipper'ом. К сожалению, не все из широкого ассорти-

Таблица 1

Meta-file	Oracle	Paradox	PC/Focus	R:Base 5000	Revelation	The Sensible Solution	Team-Up	Zim	10 Base	Enable	BOSS	Data-Ease
128	512	512	640	256	320	128	256	386	192	256	384	384
272	1140	410	3500	546	400	400	709	300	750	1065	500	435
250	254	255	256	400	H/O	H/O	1000	8000	H/O	254	255	255
64 000	90 000	4000	4096	1532	64 000	26 000	8187	8000	32 000	64 517	65 000	4000
2000	H/O	65 000	H/O	H/O	H/O	16 млн.	H/O	H/O	H/O	H/O	2 млрд	64 000
5	H/O	H/O	16	40	H/O	16	10	20	16	H/O	40	28
10 000	H/O	H/O	H/O	H/O	H/O	16 млн.	H/O	H/O	H/O	65 000	H/O	16 млрд.
235	240	255	250	1532	32 000	255	8187	8000	9999	254	255	255
2	H/O	H/O	16	2	H/O	16	255	11	16	2	40	10
3	H/O	H/O	H/O	H/O	H/O	10	100	H/O	H/O	10	7	255
250	H/O	255	32	10	H/O	1	100	H/O	5	8	—	H/O
3	8	7	7	7	8	6	8	7	4	7	6	8
2	2	5	5	7	3	—	2	3	2	3	3	5

Таблица 2

Meta-file	Oracle	Paradox	PC/Focus	R:Base 5000	Revelation	The Sensible Solution	Team-Up	Zim	10 Base	Enable	BOSS	Data-Ease
85	—	7	7	349	61	—	65	59	240	24	—	—
31	87	14	—	31,2	120	9	120	59	78	25	44	—
4	1	4	128	3,52	0,5	2,2	0,5	6,3	5,5	14,5	1	2
4	7	5	128	5,69	6	1,5	4	—	16	11,5	41	1,7
13,7	7,1	1,6	11,7	11,4	2,0	0,25	2,1	5,8	5,8	1,2	8,0	7,6

мента вспомогательных программ пригодны и для работы с FoxBASE (например, оконный интерфейс Flash-Code).

FoxBASE использует эффективные алгоритмы обращения к внешним посетителям и памяти экрана, а также весьма компактные индексные файлы на основе двоичных деревьев. Это дает в 2—4 раз большее быстродействие по сравнению с dBase III. Интересно отметить, что внутренний редактор FoxBASE допускает использование всех символов верхней половины таблицы символов ASCII (т. е. русского алфавита), но при поиске могут применяться не все эти символы.

Сам пакет FoxBASE написан на языке Си.

2. **KNOWLEDGEMAN.** Другой широко известный пакет — Knowledge Manager версии 2.0 или сокращенно KMan/2 (продукт фирмы Micro Data Base Systems Inc.). Его возможности те же, что и у dBase III Plus, но этот пакет радикально отличается по структуре, дизайну пользовательского интерфейса и по расстановке акцентов на различных аспектах обработки данных.

Отметим, что KMan/2 был перенесен с больших ЭВМ и сохраняет совместимость (а вместе с тем и недостатки) СУБД MDBS/III.

По дизайну KMan/2 скорее напоминает интегрированный пакет типа "Symphony" или "Ability" наличием встроенного электронного расчетного листа (Spreadsheet) с полным набором стандартных функций, подсистемы деловой графики, текстового редактора и возможностью

выхода на коммуникационный порт. Отличие в том, что в других интегрированных пакетах подсистема СУБД — самое слабое место и представляет собой скорее аналог картотеки, ограниченный при этом размером имеющейся оперативной памяти. В то же время в KMan/2 реализована полная реляционная СУБД с минимальным числом ограничений, а все остальные подсистемы предназначены для обслуживания этого ядра. Данные, как и в любой другой реляционной СУБД, хранятся в виде записей в двумерных таблицах, состоящих из строк и столбцов. Таблицы связаны посредством одинаковых полей в них. Но в отличие от многих реляционных СУБД (например, R:Base 5000, Mainstay или PC/Focus) KMan/2 не организует постоянных связей между таблицами. Необходимые связи задаются пользователем при запросе на вывод данных. Такой подход позволяет не накладывать ограничения на количество одновременно открываемых файлов данных и индексов и на общее количество файлов в базе. Однако пользователю необходимо помнить, что и в каком виде у него хранятся, поэтому одновременное использование более 4—5 файлов затруднительно.

Каждый из нескольких десятков оверлейных модулей базового пакета реализует определенную функцию.

Система развивается простым добавлением новых модулей. Дополнительные пакеты позволяют использовать KMan/2 из программ, написанных на языке Си, графи-

чески представлять данные, интерактивно создавать форматы вывода на экран и на печать, использовать базу в локальной сети и даже вести диалог на естественном языке. Отметим имеющийся в базовом наборе механизм макроопределений, позволяющий создавать свой собственный язык работы с системой. Кроме того, KMap/2 обеспечивает гибкие возможности защиты данных от случайного просмотра. Помимо того, что каждая вводимая запись немедленно шифруется (а при выводе автоматически дешифруется), имеется механизм пароллирования практически всех уровней работы системы, начиная от входа в нее и кончая доступом к отдельному полю избирательно на чтение-модификацию. Написанные программы обработки данных также можно зашифровать при помощи специальной утилиты.

К достоинствам пакета относятся возможность использования цвета и графики, а особенно его прозрачность для всего расширенного набора кодов ASCII. Это, в частности, позволяет без всяких доработок обрабатывать русский шрифт, встраиваемый на зарубежных ПЭВМ в верхнюю половину таблицы символов.

В целом, KMap/2 предназначен для разработки весьма квалифицированными программистами прикладных информационно-справочных систем. Пакет предоставляет достаточно развитый язык программирования с широким набором функций арифметических, статистических и строковой обработки, двумерными массивами, четырьмя типами переменных, похожий по своей структуре на Паскаль (локальные и глобальные переменные, передача аргументов и т. п.). Кроме того, в самой базе можно задать виртуальные поля, значение которых не хранится, а может вычисляться по достаточно сложному алгоритму. Однако программы из-за отсутствия транслятора чрезвычайно медленны, да и сама система значительно уступает в быстродействии тому же dBase III Plus. Качество сопровождающей пакет документации неудовлетворительно, так как разобраны в ней под силу только профессионалу.

3. **R:Base Series 5000.** Судя по зарубежным публикациям, едва ли не основную конкуренцию dBase III составляет СУБД R:Base Series 5000, Version 1.1 фирмы Microrim Inc. (2).

Хотя и dBase и R:Base предназначены для решения одних и тех же задач, отличия в принципах построения этих СУБД заставляют пользователя совершенно иначе проектировать свою систему.

Несмотря на наличие мощного языка обработки запросов и возможности использовать данные из программ, написанных на Си, Паскале и Фортране, R:Base скорее ориентирована на пользователя, не склонного к изучению программирования, но желающего организовать достаточно сложную информационную систему. В отличие от dBase III Plus здесь имеется мощный генератор приложений (Express), преобразующий интерактивно формируемую пользователем структуру в соответствующую программу, тут же компилируемую в двоичный код. Создается всего один файл, хранящий все данные и связи между ними. Одновременно организуется интерфейс с пользователем (набор вертикальных и горизонтальных меню). Созданная структура модифицируется далеко не так просто, как в dBase III Plus. Грамотное предварительное описание структур хранения и выборки данных дает существенно более компактную и целостную систему.

По сравнению с dBase III в СУБД R:Base 5000 существенно больше стандартных возможностей организации проверок при вводе-изменении данных. Например, можно задавать сравнение вводимого значения поля с набором допустимых величин, используя при этом логические операторы OR и AND. Что касается «импорта-экспорта» данных (т. е. ввода-вывода данных, хранящихся на внешнем ЗУ в формате, отличающемся от принятого в данной СУБД), то меньшее разнообразие форматов компенсируется существенно большими удобствами самого процесса.

Еще одна интересная особенность R:Base 5000 — можно использовать близкий к естественному язык запросов

Claut (похожий на аналогичный модуль KMap/2). При этом работа с базой для непрограммиста еще более упрощается, хотя практика показывает, что такой способ выбора данных, как правило, не самый эффективный.

4. **Revelation.** С момента выпуска СУБД Revelation, Version G. 2B фирмы Cosmos Inc в начале 1983 г. продано более 30 000 копий, в 1986 г. ежемесячная продажа составляла около 1500 копий [3]. Успех пакета определяется его практически неограниченными возможностями — переменные длины полей, записей и файлов с автоматической компрессией данных (чего нет ни в одной другой из рассматриваемых СУБД), одновременное использование любого количества файлов данных и индексов, произвольная глубина сортировки и индексации и многое другое. Но все это доступно для весьма квалифицированного программиста, имеющего достаточно хорошую конфигурацию аппаратуры (640 К оперативной памяти, сопроцессор i8087 и жесткий диск), хотя фирма и заверяет, что подсистема меню позволяет любому пользователю немедленно организовать нужную ему систему.

Основное достоинство пакета, включающего в себя практически все стандартные средства (генератор приложений, генератор отчетов и пр.) — использование в качестве языка запросов сильно расширенной версии языка БЕЙСИК (RBASIC) и компилятора к нему. При этом пользователь получает свободный выход на все средства ОС, контроль над клавиатурой и экраном, возможность организации файлов произвольной длины, отладку программы в режиме интерпретатора и быстро исполняемый двоичный код после компиляции.

Интересно отметить, что предлагаемая за значительную дополнительную плату (около 500 долл. на каждые четырех пользователей) доработка пакета для работы в вычислительной сети дает возможность выбора более чем из 20 вариантов (в случае с dBase III Plus всего два варианта).

В целом Revelation версии G.2B мощнее любого из перечисленных пакетов, но эта мощь доступна только при условии больших затрат труда программистов высокой квалификации.

5. **Zim.** Одна из самых интересных и мощных СУБД — пакет ZIM канадской фирмы Zante Information Inc. Пакет основан на последних теоретических исследованиях в области реляционных баз данных и использует уникальную модель хранения данных в виде «объектов» (а не записей). Система обеспечивает создание рекурсивных и многозначных взаимосвязей между объектами, сохраняемыми в специальных файлах.

Работа с ZIM предполагает знание не только теории структурного программирования, но и концепций теории реляционных объектов. Это определяет необходимость высокого профессионализма пользователя (точнее — разработчика прикладного пакета). По существу, ZIM — это скорее высокоструктурированный язык программирования, имеющий помимо богатого набора функций (как арифметических, так и строковой обработки), библиотек и команд управления необычные средства автоматизации хранения и защиты данных и организации гибких пользовательских интерфейсов. В пакет входит и компилятор, дающий весьма эффективный и компактный код и удобные возможности отладки программ. Модификация созданных структур достаточно проста. Набор утилит позволяет интерактивно строить конечную систему, а затем компилировать и использовать полученный программный текст (тип генератора приложений).

В целом, ZIM — одно из лучших (для специалистов) в настоящее время средств разработки очень сложных баз данных.

6. **Paradox, Version 1.1.** Интересна СУБД Paradox фирмы Ansa Software. Этот пакет выгодно отличается от ранее описанных гибкостью и удобством интерфейса с пользователем, позволяющим интуитивно использовать самые тонкие из имеющихся возможностей (благодаря реализации некоторых идей из области искусственного интеллекта). Система пробует обработать любое, даже ошибочное дейст-

вие пользователя, сама предлагает варианты и запрашивает дополнительную информацию. Практически любое преобразование, добавление и выборку данных, генерацию отчетов и форматов вывода можно выполнить посредством меню и подсказок. Особенно удобно реализуется выборка определенной части и пакетная модификация данных (находящихся даже в разных таблицах) заполнением граф в автоматически формируемой таблице запросов. При этом неправильное синтаксически или по смыслу запроса заполнение граф не приводит просто к выводу сообщения об ошибке и необходимости начинать весь процесс сначала (таким способом борьбы с пользователем особенно отличается KMap/2). Таблица запросов и получаемые данные доступны до тех пор, пока пользователь явно не укажет, что он получил необходимую информацию и желает перейти к другой работе.

Отметим широкие возможности проверки правильности вводимых данных, а также генерации отчетов с 16 уровнями группировки (dBase III Plus допускает до 7 уровней) и автоматическим подсчетом средних, суммарных и граничных значений.

Недостаток такого подхода — крайняя перегруженность экрана разнообразными подсказками, горизонтальными и вертикальными разделителями, сообщениями Paradox'a вперемежку с данными, а следовательно, быстрая утомляемость при работе с системой. Программа Assist из dBase III Plus, хотя и не предоставляет полностью аналогичных возможностей, но выглядит более привлекательно.

Необычный подход фирмы Ansl Software к организации своего продукта выразился и в предлагаемом языке запросов — PAL. Помимо обычного набора команд и функций процедурного языка, здесь имеется привычная для современного программиста, но отсутствующая практически во всех имеющихся пакетах возможность пользоваться системными средствами. В частности, программист может полностью обрабатывать клавиатурные прерывания (включая ALT-, ESC- и CTRL-последовательности) и выходить из командный процессор ОС. Встроенный отладчик позволяет при запланированной или случайной остановке программы просматривать значения переменных, таблиц и функций. Вместо обычного генератора приложений, не использующего, как правило, все возможности СУБД, можно встраивать в программу отработанные в интерактивном режиме запросы. Это существенно снижает затраты времени и труда.

К сожалению, исходный вариант пакета Paradox, хотя и пропускает символы верхней половины таблицы ASCII, но не реализует запросы, связанные с сортировкой и поиском по этим символам. Таким образом, при использовании его для хранения данных на русском языке требуется модификация.

7. Team-Up. Пакет Team-Up, Version 1.2 фирмы Unlimited Processing Inc.— это еще один пример довольно успешной попытки предоставить пользователю все средства системы посредством интерфейса.

Несмотря на широкие возможности, эта СУБД настолько проста в эксплуатации, что ею можно начать пользоваться, даже не заглядывая в документацию. Система меню и подсказок позволяет в считанные минуты создать необходимые файлы, задать правила и модифицировать структуры. Как и в Paradox'e, запросы организуются заполнением экранной формы, и скорость выполнения запросов примерно такая же. Кроме того, в пакете есть система парольной и многопользовательской защиты и автоматическое кодирование данных. Это обеспечивает возможность ее использования в сети ЭВМ.

Недостаток системы — совершенно неудовлетворительное качество внутреннего языка и отсутствие интерфейса с ОС. Это снижает ее привлекательность для программистов.

Пакет, несомненно, может быть рекомендован для начинающего пользователя СУБД, имеющего достаточно высокие требования к компактности, быстрдействию и простоте

использования получаемой системы.

8. BOSS. СУБД BOSS фирмы American Planning Corp.— это еще одна система, в основе которой лежит расширенный вариант языка БЕЙСИК. BOSS предназначен для разработчиков прикладных пакетов и имеет средства создания проблемно-ориентированных СУБД, включая средства сетевого доступа. Программист может модифицировать имеющуюся подсистему меню и подсказок.

Отличительные черты системы — можно обрабатывать большие таблицы данных и модифицировать их во время вывода информации, при большом наборе математических функций быстрое действие — достаточно высокое.

Разработанный фирмой язык Megabasic позволяет адресовать до 1 Мбайт оперативной памяти, использовать принципы структурного программирования и оверлейного построения программ.

9. Mainstay. Один из самых дорогостоящих из имеющихся на сегодняшний день пакетов для ПЭВМ — СУБД Mainstay Version 2.53 фирмы Mainstay Software Co. (около 1500 долл.). Под стать цене и требованиям, предъявляемые к аппаратуре — минимум 512 Кбайт ОЗУ и 1,3 Мбайт свободного места на жестком диске. Но зато в разделе «Основные характеристики» значения практически всех граф характеризуются фразой «не ограничено» (правда, всегда стоит помнить о том, что конфигурация вашей аппаратуры так или иначе наложит свои ограничения).

Отметим, что принципы работы пользователя с этой СУБД настолько отличаются от dBase III Plus и ей подобных, что без подробного изучения прилагаемой документации (а это четыре увесистых тома) вряд ли удастся построить требуемую в данном конкретном случае систему. Здесь нет даже привычной таблицы, состоящей из строк и столбцов, вместо них надо оперировать «переменными» и «измерениями». Пакет в основном предназначен для работы с такими данными, которые можно хранить в матрицах, дающих ряды значений переменной, например временные.

Благодаря простоте модификации заставок, подсказок и меню, а также включению в язык запросов широкого набора встроенных функций пакет в первую очередь предназначен прикладным программистам для разработки проблемно-ориентированных систем. Но при этом словарь и синтаксис языка серьезно отличаются от широко распространенных правил и требуют значительных усилий для освоения. По мощности он вряд ли превосходит язык dBase III Plus.

Недостатки пакета — «жесткое» сохранение созданной структуры хранения данных, практически не допускающее ее изменения. Средства «импорта-экспорта» данных другого формата также невелики.

10. Oracle. Более традиционной, но также ориентированной на профессиональных программистов, требующей больших аппаратных ресурсов и дорогостоящей СУБД, является Oracle, Version 4.1.4 — продукт фирмы Oracle Corp. Пакет широко распространен на больших и мини-ЭВМ и полностью совместим с собственной разработкой фирмы IBM SQL/DS и DB2. Построен он на базе языка запросов SQL (Structured Query Language) с удобным пользовательским интерфейсом, позволяющим интуитивно формировать структуры данных и запросы.

Для программистов, которым требуется реализация функций, невозможная в SQL (хотя для обработки данных это один из самых удобных аппаратов), предлагается C Precompiler (интерфейс с популярным языком Си). Удобные встроенные средства организации форматов ввода-вывода данных и формирования отчетов позволяют экономить затраты труда при разработке прикладных систем.

11. Informix-SQL. Одна из самых быстродействующих СУБД (сравнимая по скорости выполнения запросов только с Revelation) — пакет Informix-SQL известной своими разработками для ОС Unix/Xenix фирмы Relational

Database System. К сожалению, скорость, — пожалуй, единственное преимущество этого пакета по сравнению с dBase III Plus. Подсистема меню и подсказок крайне запутана, сам пакет не использует цветовые возможности дисплея, а для работы необходимо запоминать нестандартные сочетания CTRL и ALT последовательностей.

Имеющийся процедурный язык не отличается своими возможностями от аналогичных продуктов, а компилятор не даст практически никакой диагностики, которая могла бы помочь при отладке программы.

Среди стандартного набора встроенных функций выделяется генератор отчетов (ACE), позволяющий формировать весьма сложные таблицы вывода.

12. **Metafile.** Несмотря на более высокую цену, пакет Metafile, Version 8.2 фирмы Metafile Information Systems Inc. несколько проще в использовании и значительно уступает dBase III Plus по своим возможностям. Интересен метод хранения данных: структура файла хранится независимо от самих данных, представляющих обычный текст, в котором записи разделяются запятыми. Экономия дискового пространства при этом нет, скорее наоборот — даже при «импорте» структурированных данных Metafile отводит в среднем в 2,5 раза больше места по сравнению с исходным текстом.

Язык запросов достаточен для начинающего программиста, но также уступает dBase III Plus, хотя и обладает удобными отладочными и интерактивными средствами (оперативная проверка значений функций, пошаговое и процедурное исполнение, трассировка и немедленное редактирование). Методы доступа к данным довольно ограничены, особенно при использовании индексных файлов.

В целом, пакет предназначен для создания не слишком сложных приложений при сравнительно небольших затратах труда.

13. **DataEase.** Один из пакетов, удачно сочетающих наличие развитых средств и простоту использования, — DataEase, Version 2.5 компании Software Solution Inc.

Начальный запуск пакета позволяет задать тип используемого устройства печати, вид меню и назначение функциональных клавиш, использование цветов экрана и уровень защиты данных. Очень удобно организован ввод данных с переносом-просеркой-изменением значений и также хорошо продуманы средства выборки и отображения данных.

Недостаток пакета — отсутствие в языке стандартных средств сортировки и индексации данных. Эти операции можно производить только временно в ОЗУ.

В целом, Data Ease представляет собой не самую мощную, но одну из самых удобных из современных реляционных СУБД.

14. **СУБД The Sensible Solution, Version 2.0** фирмы O'Nolan Computer Systems предполагает почти противоположный способ организации базы. От пользователя требуется создание процедурного файла на предлагаемом языке с использованием набора вспомогательных программ и специального редактора. Высокие скоростные характеристики обеспечиваются компиляцией полученного кода.

Ядро СУБД — редактор, ориентированный на создание необходимых процедур с помощью специальной системы меню. В зависимости от контекста пользователю предлагается выбор только допустимых команд и проводится ряд проверок допустимости указываемых значений. Компиляция осуществляется, не выходя из редактора. Для еще большего сокращения времени на написание кода, программы подготовки форматов ввода и вывода данных автоматически генерируют соответствующие процедуры.

В целом, такой подход приводит к созданию достаточно удобной системы за счет больших затрат времени и труда на ее разработку и отладку. Существенный недостаток всего пакета — отсутствие средств «импорта» данных.

15. **СУБД PC/Focus** фирмы Information Builders Inc. нельзя, строго говоря, назвать просто реляционной, поскольку она позволяет использовать и иерархические, и некоторые виды сетевых структур. Пакет — это почти полный повтор СУБД Focus (больших машин).

По своим возможностям PC/Focus далеко превосходит большинство из имеющихся на ПЭВМ пакетов, но сложность работы с ним такова, что фирма рекомендует предварительно пройти обучение на специальных курсах. Из-за большой сложности организуемых структур данных требуется предварительная проработка задачи. Негромкое описание проектируемой системы усложняет последующие работы, а всякая модификация крайне затруднительна.

**Достоинство системы** — полная ее совместимость с исходной СУБД Focus, позволяющая с минимальными затратами использовать накопленные данные, а также наличие необычайно широкого набора средств манипулирования данными. Но несмотря на наличие подсистемы меню и подсказок, достаточно полное освоение пакета требует высокого профессионализма.

16. **СУБД Goldatabase, Version 6.0** компании Goldata Computer Services Inc. — одна из самых простых в освоении. Удобная система меню и подсказок и использование экранного редактора для одновременного создания структуры и формата ввода-изменения данных позволяют в считанные минуты организовать требуемую систему. Язык запросов — обычный БЕЙСИК. Сам пакет также написан на БЕЙСИК-компиляторе и предназначен для решения несложных задач начинающим пользователем. Тем не менее, можно создать многофайловые реляционные системы и даже работать в локальной сети.

17. **В СУБД 10 Base, Version 2.0** фирмы Fox Research Inc. организован необычный пользовательский интерфейс. Здесь нет ставшей стандартной в других современных пакетах системы меню. Пользователю предлагается работа в специальном редакторе, а точнее, интерпретаторе языка SQL. Этот язык позволяет создавать и изменять структуру данных, проводить сортировку и выборку данных. После выполнения команды остаются доступными для модификации или запоминания.

Для случаев, когда команда языка SQL недостаточно для организации сложного запроса, используется процедурный язык, дополняющий SQL.

Недостатки пакета — ограниченные возможности «импорта» данных и необходимость предварительного знакомства с языком SQL.

18. Пакет **Enable, Version 1.10** компании Software Group — это пример создания интегрированного продукта, все подсистемы которого обладают хорошими характеристиками. В отличие от аналогичных пакетов Symphony или Framework II в Enable реализована развитая реляционная СУБД со своим языком запросов. Встроенный редактор, электронная таблица и подсистема графики позволяют более гибко организовать хранение и выборку данных. По допустимым объемам хранения и одновременной обработки данных и времени выполнения запросов Enable лишь немного уступает похожему пакету KMan/2.

Особенно Enable отличается качеством оперативных подсказок, зачастую предоставляющих несколько путей решения текущей задачи.

Основной недостаток пакета — отсутствие в документации полного описания языка запросов, необходимого для выполнения нестандартных задач.

19. **СУБД filePRO 16 Plus, Version 02.00.00** компании The Small Computer Company предназначен для профессиональных разработчиков прикладных пакетов, знакомых с терминологией, принятой в системе Unix / Xenix. С помощью этой СУБД можно создавать достаточно сложные информационные системы, хотя и не очень современными методами. В частности, в системе практически отсутствуют оперативные справочники, не используется общепринятое на ПЭВМ разделение экрана на окна.

Язык запросов — некоторая разновидность БЕЙСИКа.

уступающая по удобству языку того же dBase III Plus.  
20. **DDQuery.** Завершим обзор описанием самой дорогостоящей из имеющихся на сегодняшний день СУБД для персональных ЭВМ.

С помощью пакета DDQuery, Version 3.1 фирмы Venet-Uiams Inc. (цена пакета более чем в 5 раз превышает цену dBase III Plus) профессионал может создать прикладную систему такой сложности, для которой всего несколько лет назад требовалась большая ЭВМ. Правда, затраты труда при этом будут тоже соответствовать работе с большой машиной. Причина этого, видимо, в том, что как ряд аналогичных пакетов, DDQuery была разработана более 20 лет назад, а затем перенесена на микроЭВМ.

Система состоит из набора программ, каждая из которых позволяет реализовать определенную функцию обработки данных, строчного редактора и языка запросов TRANSACT. Структура базы описывается на специальном языке и обрабатывается отдельным процессором. При этом указываются взаимосвязи между полями разных файлов («наборов данных»), составляющих базу, способы доступа и степень защиты данных. DDQuery не предназначена для решения простых задач. Об этом говорят такие возможности, как, например, одновременная сортировка девяти файлов по семи параметрам или виртуальное объединение нескольких файлов. Кроме того, система ориентирована на работу в локальной вычислительной сети.

Рассмотренные 20 пакетов СУБД, предлагаемые зарубежными фирмами для использования на ПЭВМ, — на сегодняшний день наиболее серьезные разработки в этой области. Существует большое число гораздо более простых систем, предназначенных, как правило, для решения конкретных задач учета и управления. Отличительная черта всех рассмотренных пакетов — их универсальность. Наличие в них того или иного языка запросов позволяет создать систему, наиболее полно соответствующую заданным условиям хранения, изменения и выборки данных, организовать автоматическое отслеживание взаимосвязей и избежать ненужного дублирования.

Большинство имеющихся задач можно решить при помощи пакетов dBase III Plus, R:Base 5000 и DataEase. Для повышения быстродействия лучше применить Paradox или Revelation. Для достаточно сложных приложений можно рекомендовать Zim, Revelation и Enable. Для построения особо сложных систем можно воспользоваться пакетами MainStay или DDQuery, Revelation, Zim или Enable.

Телефон 330-55-56, Москва, Ненашеву М. А.

#### ЛИТЕРАТУРА

1. Мартин Дж. Организация баз данных в вычислительных системах. — М.: Мир, 1980.
2. What Micro? The Business Buyers Guide. — 1986. — May. — P. 52—56.
3. PC Magazine, June, 24, 1986.
4. Byte, April 1986 — November 1986.
5. The Software Catalog, Systems Software produced from «Menu» International Software Data Base, Elsevier, 1986.

Статья поступила 12.08.88

УДК 681.3.06

П. В. Христов

## АРХИТЕКТУРА ЭСКОРТ-МАШИНЫ И ЕЕ АППАРАТНАЯ ПОДДЕРЖКА

### Введение

Инструментальная система ЭСКОРТ [1—3] разрабатывается в Научно-производственном центре АН СССР и ПО ЗИЛ по проблемам САПР в машинностроении как средство повышения производительности труда профес-

сиональных программистов, создающих и сопровождающих большие программные комплексы. Реализация ЭСКОРТА опирается на следующие основные положения: используется нетекстовое представление программы. Выбранная единая семантическая модель приближает уровень понятий, которыми оперирует инструментальная система, к уровню понятий программиста;

программист работает в рамках единой среды редактирования — трансляции — выполнения. Это резко уменьшает длительность цикла отладки, дает возможность в произвольной последовательности чередовать выполнение и редактирование фрагментов программ;

каждое редактирующее действие сопровождается анализом изменений. Поэтому программисту всегда доступна исчерпывающая информация о разрабатываемой программе;

программист может прервать сеанс и при редактировании, и при выполнении программ с последующим возобновлением сеанса с прерванного места;

программисту предоставлены средства восстановления предыдущих состояний программы во время редактирования и выполнения.

Для подсистемы выполнения ЭСКОРТ-программ важны два решения, вытекающие из перечисленных положений:

1. Для хранения разрабатываемого программного комплекса используется специализированная сетевая база данных (БД), с которой взаимодействуют все компоненты интегрированной системы: редактор, визуализатор, интерпретатор, компилятор, подсистемы анализа, тестирования, конфигурационного управления. При этом программа рассматривается как структурированная совокупность программных объектов: процедур, управляющих конструкций, типов данных, переменных, констант, выражений и т. п. Всем программным объектам взаимно-однозначно соответствуют объекты БД. Кроме основной (семантической) информации о программном объекте БД содержит дополнительные атрибуты, заданные пользователем либо вычисленные соответствующим компонентом интегрированной системы.

2. Программы, разрабатываемые в рамках системы ЭСКОРТ, интерпретируются. Достоинство такого подхода — возможность выполнения программ в разных режимах, обратного выполнения, диагностики программных ситуаций в исходных терминах и т. д. Кроме того, интерпретация позволяет реализовать языковые средства работы со сложными структурами и более гибкий механизм параметризации по сравнению с традиционными языками, такими, например, как Ада. Основным недостатком интерпретации — низкая эффективность. Производительность вычислительной системы понижается из-за интенсивного взаимодействия интерпретатора с БД.

Вместо непосредственной интерпретации объектов БД было решено сначала компилировать во внутреннее представление так называемый сшитый код [4,5], а затем интерпретировать уже этот код, что позволит уменьшить число обращений к БД и воспользоваться информацией, собранной компонентом анализа. Редактирование программ сопровождается трансляцией измененных фрагментов в сшитый код ЭСКОРТА. Транслятор каждому программному объекту ставит в соответствие сегмент команд; выполнение программного объекта заключается в интерпретации данного сегмента. Аналогично тому, что программный объект имеет подобъекты, сегмент кода объекта содержит команды перехода к сегментам кода подобъектов.

### Общая характеристика ЭСКОРТ-машины и сшитого кода

**Особенности сшитого кода.** Сшитый код системы ЭСКОРТ разработан таким образом, чтобы найти компромисс между двумя противоречивыми требованиями — максимальным использованием информации, предоставляемой компонентом анализа программ, и минимизацией затрат на перетрансляцию при локальных изменениях программы.

При переходе от непосредственной интерпретации объектов БД к интерпретации сшитого кода вся информация о статической семантике программного объекта представляется в формате, наиболее удобном для обработки управляющей программой интерпретатора. Адрес сегмента команд можно получить за одно обращение к БД. Применение сшитого кода сокращает число запросов к БД в процессе интерпретации в три-пять раз.

При формировании сшитого кода транслятор использует результаты проверки корректности синтаксиса и статической семантики программы, которая осуществляется в процессе редактирования. Команды сшитого кода должны контролировать соответствие только динамических атрибутов операндов.

ЭСКОРТ-машина стековая. Такое решение вызвано требованиями минимальности перетрансляции, так как именно для стековой машины наиболее просто описываются ее состояния, что предельно упрощает «соглашения о связях» интерпретируемых программных объектов.

Самоопределяемость данных в ЭСКОРТ-машине дает возможность обойтись без перетрансляции при локальных изменениях описаний переменных и типов данных, а также уменьшить число команд обработки данных.

В сшитый код включены такие понятия, как конструкторы типов «массив», «запись», «множество», «строка», «перечислимый», высокоуровневые операции обработки данных и управляющие конструкции. ЭСКОРТ-машина непосредственно поддерживает параметризацию (причем в качестве параметров допускаются не только данные, но и типы), средства обработки исключительных ситуаций, а также механизм пакетов, в том числе и настраиваемых (аналогичный механизм родовых пакетов предоставляет язык Ада [6].)

Отличительная особенность ЭСКОРТ-машины — введение в нее понятия времени. Каждой переменной в программе сопоставляется вся совокупность значений, которые когда-либо приняла переменная. Вместе со значениями сохраняется информация о моментах времени их формирования. Временной счетчик увеличивается на единицу при выполнении операций *присвоить* и *шаг\_итератора*. ЭСКОРТ-машина с помощью операции *откатить* восстанавливает значение переменной на любой заданный момент времени. На рассмотренном механизме основывается реализация возможности «обратного выполнения».

**Структура ЭСКОРТ-машины.** ЭСКОРТ-машина работает с сегментами команд сшитого кода и динамически формируемыми описателями типов, данных и других объектов. На текущую интерпретируемую команду указывает счетчик команд (СК). Команды, не являющиеся командами перехода, увеличивают значение счетчика на единицу. Операции ввода-вывода взаимодействуют с буфером, который содержит вводимые или выводимые данные, представленные в символической форме.

Для перехода к сегменту кода подобъекта используются команды *войти<Об>* и *выйти* (здесь и в дальнейшем <Операнд> обозначает аргумент команды, например <Об> — объект, над которым надо выполнить соответствующую операцию, <Сдвиг> — значение, на которое изменяется СК при выполнении команд перехода). По команде *войти в Стек возвратов* заносится значение СК+1, а в СК — начальный адрес сегмента кода, соответствующий подобъекту. Каждый сегмент заканчивается командой *выйти*, обозначающей возврат к предыдущему сегменту.

Для организации вычислений используется *Стек данных*. Большинство операций берут из стека адреса описателей операндов и (или) помещают в него адрес описателя результата. По команде *получить\_описатель <Об>* в *Стек данных* загружается адрес соответствующего объекту <Об> описателя.

*Стек контекстов* организует рекурсивные вычисления и переход от одного экземпляра родового пакета к другому. При повторном обращении к программному объекту (процедуре, пакету) в этом стеке сохраняются преды-

дущие экземпляры переменных, параметров и типов, определенных в данном объекте.

Кроме того, ЭСКОРТ-машина использует счетчик времени (СВ) и признак исключительной ситуации (ИС). Спектр возможных исключительных ситуаций традиционен.

### Организация данных в ЭСКОРТ-машине

В процессе работы ЭСКОРТ-машины создаются, модифицируются и удаляются структуры, называемые описателями типов, данных, контекстов и итераций. Описатели могут размещаться в нескольких связанных фрагментах памяти переменной длины. Различают временные и сохраняемые описатели. Временные описатели используются, например, для представления значений выражений; они формируются в результате выполнения операций обработки данных и удаляются при выполнении операций, операндами которых являются. Сохраняемые описатели соответствуют константам, переменным, параметрам, типам и пакетам; создаются и удаляются они только явно.

**Описатели типов** содержат динамические атрибуты типов, определенных в интерпретируемой программе. Они могут быть простыми («логический», «вещественный», «диапазон целых чисел» и «перечислимый») или составными («массив», «строка», «множество» и «запись»). Простой описатель типа размещается в одном или нескольких последовательных словах памяти. Особенность составных описателей — они содержат ссылки на описатели других типов (типов компонентов или элементов).

**Описатели данных** содержат заголовок, ссылку на описатель типа данных и ссылку на поле значений. Для простых типов данных поле значений имеет структуру однонаправленного списка. Одно слово резервируется под «голову» списка, указывающую на текущее значение. Элементы списка содержат собственно значение, момент времени, в который это значение сформировано, а также ссылку на элемент, соответствующий предыдущему значению.

У массивов поле значений представлено вектором ссылок, каждая из которых является головой списка простых значений. Поле значений множество и строк отличается от поля значений простого типа тем, что вместо простых значений список содержит ссылку на таблицу элементов структуры. В поле значений записи находится таблица ссылок на списки значений (по одной ссылке на каждый из компонентов).

Во всех составных структурах данных на месте простых значений может содержаться ссылка на поле значений подструктуры. Так представляются объекты данных с вложенной структурой, например множество строк, массив записей и т. п.

**Описатели контекстов.** Каждому пакету сопоставляется так называемый описатель контекста, содержащий таблицу ссылок на описатели данных, типов и контекстов, определенных в пакете (в том числе и описатели параметров настройки родового пакета).

**Описатели итераций.** При выполнении программы может создаваться еще один вид динамических структур — описатели итераций, используемые для организации циклических вычислений. Описатель итераций содержит атрибуты цикла (например, начальное и конечное значения управляющей переменной цикла и шаг итерации) и ссылку на описатель управляющей переменной.

### Система команд ЭСКОРТ-машины

Работа ЭСКОРТ-машины инициируется по команде *начать\_выполнение <Об>* и завершается по команде *закончить\_выполнение*.

Ссылки на динамические описатели объектов программы хранятся в БД. Команда *получить\_описатель <Об>* запрашивает у БД описатель. Команда *привязать\_описатель*



тель <Об> посылает БД запрос на смену описателя объекта, ее предыдущий описатель помещается в стек контекстов. Команда *удалить\_описатель* <Об> восстанавливает предыдущий описатель.

В ЭСКОРТ-машину включены арифметические, логические, текстовые операции, а также операции преобразования типов и обработки структур: *преобразовать\_тип*, *сделать\_структурой*, *конкатенация*, *индекс*, *длина строки*, *пересечение*, *объединение*, *разность*, *принадлежит*, *мощность*. При выполнении команд обработки данных проверяется соответствие атрибутов типов операндов, а также присвоение операндам начального значения.

Операция *присвоить* допустима только для операндов эквивалентных типов. При выполнении этой операции временной счетчик увеличивается на единицу и формируется новое значение операнда-приемника, которое замещает предыдущее значение.

Для изменения последовательного порядка интерпретации сегмента команд используются команды условного и безусловного перехода, а также итерации. Команда *создать\_итератор* по операндам (управляющей переменной, началу, концу и шагу цикла) формирует итератор. Команда *шаг\_итератора* <Сдвиг> в случае, если выполнение цикла завершено, удаляет итератор и увеличивает СК на Сдвиг, иначе модифицирует значение управляющей переменной цикла и увеличивает СК на единицу.

Команды, формирующие описатели типов *целый*, *логический*, *вещественный*, возвращают описатели определенных простых типов; *диапазон* вырабатывает описатель отрезка множества целых чисел; *множество* и *строка* по описателю типа элемента создают описатель результирующего типа; *массив* <N> получает N+1 операнд — описатель типа элемента и N описателей типов индексов, а команда *запись* <N> — N описателей компонентов; *тип* по описателю данных находит описатель его типа, команды *тип\_элемента* и *тип\_индекса* — описатель типа элемента и, соответственно, типа индекса; *перечислимый* <N> (где N — мощность типа) формирует описатель типа и описатели констант данного типа.

Операция *создать\_переменную* создает описатель данных типа, который определяется операндом. Операция *создать\_параметр* вырабатывает описатель формального параметра. Ссылки на описатель типа и поле значений копируются из описателя фактического параметра.

Для выделения подструктуры предназначены команды *подмассив* (ее результатом может быть массив меньшей размерности), *подстрока* и *компонент\_записи*.

Команды для реализации пакетов: *создать\_контекст* <N> формирует описатель контекста по N описателям; *установить\_контекст* выполняет обратные действия — выделяет из описателя контекста описатели данных и типов; *взять\_из\_контекста* <N> выделяет из контекста N-й описатель.

Для организации ввода и вывода используется текстовый буфер, доступный другим компонентам ЭСКОРТа. Команда *прочитать* получает из буфера текст и формирует по нему описатель данных. Команда *записать*, наоборот, по описателю данных формирует в буфере соответствующий текст.

Команды *исключение* и *обработать\_исключение* используются для обработки ИС. Первая команда устанавливает признак ИС, вторая его сбрасывает. Если признак ИС установлен, то большинство команд виртуального кода пропускается. Так же, как и в обычном режиме, выполняются команды *выйти* и *удалить\_описатель*, что позволяет последовательно завершать выполнение объемлющих объектов, пока не будет достигнут объект, в котором предусмотрена обработка ИС.

Команда *откатить* восстанавливает значение операнда на заданный момент времени.

## Расширение сшитого кода

Для организации взаимодействия ЭСКОРТа с существующими средствами программного обеспечения необходимо применить в ЭСКОРТе программные единицы, разработанные вне его, и разработать препроцессор с входного языка системы ЭСКОРТ на один из производственных языков (ПЯ).

ЭСКОРТ-машина позволяет использовать наряду с предопределенными командами сшитого кода ПЯ-программы. Фактические параметры вызываемой программы загружаются в стек данных, специальный препроцессор данных преобразует параметры в соответствии с форматом, принятым в ПЯ. Затем ЭСКОРТ-машина передает управления ПЯ-программе. После выполнения ПЯ-программы постпроцессор вновь переводит ее параметры в формат ЭСКОРТа. Так как при передаче управления СВ увеличивается на единицу, для выходных параметров ПЯ-программы постпроцессор сохраняет предыдущее значение и формирует новое. Следовательно, возможна «откатка» выполнения ПЯ-программы как единого действия.

Решение обратной задачи позволяет построить компилятор ЭСКОРТ-программ. Он получается объединением ЭСКОРТ-ПЯ-препроцессора и ПЯ-компилятора. Откомпилированные ЭСКОРТ-программы с помощью рассмотренного механизма также применяются для расширения предопределенного множества команд сшитого кода.

Для ускорения выполнения ЭСКОРТ-программ используется частичная компиляция. Модули, для которых достигнут высокий уровень надежности, компилируются и в дальнейшем выполняются как единые действия. Если возникнет необходимость, программист всегда сможет вернуться к редактированию и интерпретации этих модулей.

## Анализ производительности ЭСКОРТ-машины

При создании первой версии ЭСКОРТа, которая находится сейчас в опытной эксплуатации, мы стремились разработать прототип, а не высокоэффективную вычислительную систему. Реализация данной версии выполнена полностью программно на языке Фортран-77. Не самые оптимальные, но простые технические решения были приняты исключительно из желания быстрее получить работающую версию системы. Тем не менее, эксперименты с данной версией ЭСКОРТа позволяют выделить некоторые характеристики производительности ЭСКОРТ-машины.

Оценка производительности ЭСКОРТ-машины и относительных затрат времени характерных этапов интерпретации была выполнена на наборе тестов. Три наиболее простых теста, имеющих прямые аналоги на Фортране:

тест 1:  $i:=1; j:=1$ ,  
пока  $i < 10\ 000$ , выполнять  $i:=i+j$ ;  
тест 2:  $i:=1; j:=1$ ,  
пока  $i < 10\ 000$ , выполнять  $i:=i+j; j:=j*j$ ;  
тест 3:  $i:=1; j:=1$ ,  
пока  $i < 10\ 000$ , выполнять  $a(i):=i; i:=i+j$ .

В таблице сравнивается время выполнения тестов

Временные характеристики ЭСКОРТ-машины

Номер теста	Время выполнения тестов, с		Относительные затраты, %, на			
	ЭСКОРТ	Фортран	операции со стеком	запрос к БД	динамический захват и освобождение памяти (ДП)	управляющую программу (УП) интерпретатора
1	31	07	23	25	15	12
2	51	14	24	30	12	11
3	59	13	28	25	13	12

и соответствующих Фортран-программ, а также показаны «вклады» в общие временные затраты блоков (СТ, БД, ДП, УП).

Измерения показывают, что простейшие программы в текущей версии ЭСКОРТа выполняются в 300...500 раз медленнее, чем соответствующие Фортран-программы, причем, как и следовало ожидать, основное время затрачивается не на сами вычисления, а на их организацию. Но ЭСКОРТ-машина выполняет операции над сложными структурами при значительно меньших накладных расходах; например, арифметические операции над векторами и матрицами большого размера — в 7...10 раз медленнее, чем программы на Фортране, в которых действия с массивами моделируются с помощью циклов.

### Предложения по аппаратной поддержке

Интерпретатор сшитого кода ЭСКОРТ-машины в настоящее время реализован полностью программно на языке высокого уровня. Самый перспективный способ повышения эффективности выполнения ЭСКОРТ-программ — аппаратная поддержка интерпретатора сшитого кода.

Аппаратная (либо микропрограммная) реализация языка высокого уровня достаточно широко освещена (см., например, [7]). Примерами машин с подобной архитектурой могут служить система SYMBOL [8], FORTH-машина [9], система ДССП [5]. Главным аргументом за непосредственную аппаратную реализацию языка является высокая эффективность выполнения программ, написанных на этом языке. Эта эффективность достигается за счет максимальной специализации архитектуры. Однако учет особенностей языка высокого уровня оборачивается утратой универсальности вычислительной системы.

Сшитый код ЭСКОРТ-машины более сложен и специализирован, чем код любой из указанных систем. Его реализация в аппаратуре затруднит расширение ЭСКОРТ-машины. Альтернативным решением может быть выделение и аппаратная поддержка наиболее тяжелых и часто используемых функций в интерпретаторе.

Анализ архитектуры ЭСКОРТ-машины и сшитого кода позволяет сделать вывод, что наибольшее время при интерпретации занимают следующие действия: выборка и декодирование команд, взаимодействие с БД, работа со стеком, динамическое управление памятью. Кроме того, значительные затраты приходится на операции со списочными структурами и на обращения к памяти с большим уровнем косвенности.

Результаты измерения производительности ЭСКОРТ-машины подтверждают вывод: для тестов, работающих с простыми типами данных, суммарное время работы управляющей программы интерпретатора, запросов к БД, операций над стеком, динамического захвата и освобождения памяти составляет 75...80% от всего времени интерпретации.

Один из возможных подходов к аппаратной поддержке интерпретации заключается в расширении системы команд процессора операциями, которые обеспечат эффективное выполнение действий, связанных с наибольшими затратами. Это могут быть команды с повышенным уровнем косвенности, операции динамического управления памятью и над стеками, очередями, списками и таблицами, что позволит оптимизировать обработку запросов к БД.

Для реализации операций над стеками и таблицами, а также повышения косвенности достаточно возможности, предоставляемые микропроцессором 68020 фирмы Motorola [10]. Например, при работе со стеками и очередями используется косвенная регистровая адресация с априорным отрицательным или апостериорным положительным приращением содержимого регистра и команда BOUNDSCHECK, проверяющая, лежит ли в указанных пределах содержимое регистра.

Для организации односвязных списков сегментов были бы полезны следующие команды:

взять из начала m, n, D  
вставить в начало m, n, d

Первая команда помещает содержимое регистра Rn в регистр Rm, а содержимое ячейки памяти с адресом Rn+D в регистр Rn. Операция выполняется только, если Rn не содержит 0; в противном случае устанавливается соответствующий бит признака:

если  $Rn \neq 0$ ,  $Rn - > Rm$ ,  $(Rn + D) - > Rn$ .

Вторая команда помещает содержимое регистра Rn в ячейку с адресом Rm+D, а содержимое регистра Rm в регистр Rn:

$Rn - > (Rm + D)$ ,  $Rm - > Rn$ .

Архитектура, на которой реализуется ЭСКОРТ-машина, может выглядеть следующим образом: ЭСКОРТ-программы выполняются специализированным сопроцессором, который взаимодействует с процессором БД программ (быть может, также специализированным), посылая ему запросы вида

адрес\_кода <Об>  
получить\_описатель <Об>  
установить\_описатель <Об>

Преимущество такого подхода — возможность раздельной специализации процессоров; на начальном этапе вместо любого из них может выступать универсальный процессор.

Телефон 277-87-31, Москва

### ЛИТЕРАТУРА

1. Бетелин В. Б. Системы автоматизации труда программиста. — Препринт. — М., 1986. — 40 с. — (НСК АН СССР).
2. Прилипко Г. А. Инструментальная среда программирования «ЭСКОРТ» // Вопросы кибернетики. Проблемы обработки графической информации в САПР машиностроения. — М.: НСК АН СССР, 1987. — С. 120—137.
3. Галатенко В. А., Митропольский А. Ю., Мурашкин А. И., Прилипко Г. А., Христов П. В. Основные компоненты системы ЭСКОРТ // Вопросы кибернетики. Проблемы автоматизации инженерного труда в машиностроении. — М.: НСК АН СССР, 1988. — С. 22—35.
4. Уокерли Д. Ж. Архитектура и программирование микроЭВМ. — М.: Мир, 1984.
5. Брусенцов Н. П. Структурированное программирование и сшитый код // Архитектура и программное обеспечение цифровых систем. — М.: МГУ, 1984. — С. 3—9.
6. Вегнер П. Программирование на языке АДА. — М.: Мир, 1983.
7. Майерс Г. Архитектура современных ЭВМ — М.: Мир, 1985.
8. Laliotis T. A. Architecture of the SYMBOL Computer System / in Y. Chu, Ed. // High-Level Language Computer Architecture. — New York: Academic, 1975. — P. 110—185.
9. James J. S. What is FORTH? A tutorial introduction // BYTE. — 1980. — Vol. 5. N8. — P. 100—126.
10. MC68020 32-Bit Microprocessor User's Manual. — Prentice-Hall, Inc., Englewood Cliffs, 1984.

Статья поступила 25.02.88

*Читатели часто задают редакции одни и те же вопросы в своих многочисленных письмах и телефонных разговорах: как в редакции работают над рукописью и отбираются материалы в номер? Мы решили познакомить читателей МП с «редакционной кухней» на примере прохождения одной конкретной статьи, которая типична в том отношении, что мнения о ней рецензентов, каждый из которых — крупный специалист в своей области, расходятся... Следуя духу гласности, редакция выносит предмет этого рядового конфликта из непрерывной, по выражению А. П. Ершова, «священной войны редакторов и авторов» на суд самих читателей. Разумеется, здесь мы приводим уже доработанный материал.*

УДК 681.3.06

И. Л. Бельчинский

## ПЕРЕНОС ЯДРА ИНМОС МЕЖДУ МАШИНАМИ РАЗЛИЧНЫХ АРХИТЕКТУР

Перенос ядра инструментальной мобильной операционной системы (ИНМОС) осуществляется на совместимых машинах (IBM PC XT, Искра 1030), далее называемых персональными компьютерами (ПК). Для переноса было взято ядро ИНМОС, работающее на машинах типа SM (SM4 и SM1420).

**Подходы к переносу.** Для каждого нового компьютера необходима ОС, разработка которой — процесс длительный и трудоемкий. Первым шагом, обещавшим упростить этот процесс, было создание языков системного программирования высокого уровня и реализация ОС на них, что на ассемблере требуется писать только около 10% исходного текста ОС.

Второй шаг — попытка переноса существовавших ОС и конструирования мобильных ОС, не зависящих от архитектуры машин. Исследовались следующие основные подходы к переносу [1]:

установка (перенос) эмулятора старой машины на новую. При этом резко понижается быстродействие; создание виртуальной машины высокого уровня, разработка для нее программного обеспечения и эмуляция ее на существующих компьютерах. Быстродействие понижается меньше благодаря специальному конструированию набора инструкций. Так устроена UCSD П-система;

разработка ОС и компиляторов на языке высокого уровня и максимальная изоляция машинно-независимых частей, а затем разработка кодогенератора и машинно-зависимых частей ОС для новой машины и перекompиляция ОС полученным компилятором. Этот подход оказался наиболее удачным по быстродействию и эффективности ОС.

Третий шаг — конструирование архитектур машин, обеспечивающих максимальную эффективность унифицированных мобильных ОС.

### Цель работы:

1. Создать реализацию ОС для ПК. ОС должна быть совместима на уровне команд пользователя, системных вызовов, языка программирования с ИНМОС на SM.

2. В процессе переноса получить исходный текст ядра унифицированной многозадачной ОС для ПК.

3. Исследовать трудности переноса ОС без предварительных детальных знаний устройства данной ОС.

**Особенности ОС ИНМОС с точки зрения переноса.** ИНМОС — многозадачная ОС с разделением времени, предназначенная для разработки и отладки программ. Она стала популярной благодаря своей простоте и сравнительной легкости, с которой переносится на машины других архитектур. Эти ее характеристики сравнимы

с аналогичными ОС Unix [2—4]. Простота и переносимость обусловлены тем, что, во-первых, ОС опирается на очень простую виртуальную машину, в ней не используются особенности архитектуры, во-вторых, более 90% кода ядра написано на языке Си.

Система ИНМОС ориентирована на процессы, управление которыми в ней довольно просто. Информация о каждом готовом к выполнению процессе размещается в списке. Процессы расположены в нем по приоритетам. Каждый процесс получает процессор на единственный отрезок времени по алгоритму, распределяющему процессор и динамически пересчитываемому приоритету. Время разделяется за счет прерываний от системного таймера и (или) при обращении процессора к ядру.

ИНМОС предполагает минимум два режима: пользовательский и системный (ядра). Вся работа драйверов, управление процессами, управление вводом-выводом осуществляются в системном режиме. Код процессов выполняется в пользовательском режиме. Всякий раз, когда процесс выдает системный вызов, режим меняется на системный, а при возвращении к тексту процесса вновь становится пользовательским. Тем самым обеспечивается разделение системной и пользовательской фаз процессов.

Управление памятью в прикладной программе, выполняемой в ИНМОС, не зависит от аппаратных средств управления памятью. Поэтому можно обойтись даже без защиты памяти, если система будет в основном использоваться для работы с уже отлаженными программами, например с прикладными пакетами.

В ИНМОС обеспечивается ввод-вывод двух видов: блочный — для высокоскоростных устройств большой емкости, таких как диски, и символьный — для устройств типа терминалов. Интерфейс к драйверам в ядре прост и строго определен, что существенно упрощает разработку драйверов — камень преткновения во многих ОС.

В ИНМОС процесс — пользовательская задача, которой выделяются некоторые ресурсы, такие как память, время процессора, место в системных таблицах. Многозадачность осуществляется благодаря разделению времени. Каждый процесс состоит из системной и пользовательской фаз, и каждая фаза обладает своим стеком, данными и текстом. Текст системной фазы процесса — это текст ядра, т. е. код ядра исполняется в системной фазе процессов, а код пользователя — в пользовательской. Единственным исключением является процесс 0 — начальный процесс, специально порождаемый, не имеющий пользовательской фазы.

Контекст процесса — область данных, где содержится описание текущего состояния процесса. Переключение контекстов заключается в приостановлении текущего выполняемого процесса и возобновлении выполнения другого процесса. Так как ИНМОС — ОС разделения времени, ориентированная на процессы, то переключение контекстов проводится часто (по окончании обработки системного вызова или при некоторых прерываниях, например от таймера).

Процессор K1810BM86 [5] с точки зрения постановки ИНМОС имеет положительные и отрицательные стороны. Сегментная организация памяти упрощает разделение памяти между задачами, их взаимное обособление. Отсутствие приоритетных режимов в процессоре усложняет реализацию системного и пользовательского режимов ИНМОС. Процессор не обеспечивает защиты памяти.

В ПК создается восемь независимых уровней прерывания. За счет выборочного маскирования прерываний мы можем обеспечить достаточную гибкость любой схемы прерываний при работе с внешними устройствами, отделяя быстрые от медленных. Независимые уровни прерываний удобны для организации эффективной работы драйверов с высокоскоростными устройствами внешней памяти и низкоскоростными дисками, клавиатурой.

**Подготовка переноса.** Для переноса было взято ядро

ИНМОС СМ. В качестве инструментальной среды использовался ПК с ОС типа Unix. Осуществлять перенос ИНМОС, применяя ОС, не совместимую с Unix, значительно сложнее, так как для генерации ядра, командных процессоров, всех команд и библиотек требуются средства, доступные под ИНМОС или Unix. В первую очередь это Си-компилятор, Си-препроцессор (удобный и как препроцессор более общего назначения, например для ассемблера), инструментальная программа поддержки компьютерных программ make.

Сначала были проведены детальный разбор ядра, определение машинно-зависимых блоков и функций и выявление мест в исходном тексте ядра для СМ, требовавших переделки. При этом использовались специально разработанные методы переноса. В связи с отличиями архитектуры процессора K1810VM86 от процессоров в СМ и разными архитектурами самих машин (ПК и СМ) необходимо определить:

• способ реализации системных вызовов и передачи параметров;

• минимальный размер распределяемой памяти;

• расположение информации о процессах в памяти и метод переключения с процесса на процесс;

• расположение процессов в памяти;

• режимы системы.

Выделение, решение и реализация этих проблем (в том числе модификация соответствующих алгоритмов ядра) составляют перенос ядра ИНМОС между СМ и ПК. Очевидно, при переносе приходится модифицировать и некоторые алгоритмы ядра (в данном случае распределения памяти, переключения между процессами).

#### Перенос ИНМОС: принятые решения

**Системные вызовы и передача параметров.** Пользовательская фаза процесса обращается к ядру, выдав прерывание (с вектором, общим для всех системных вызовов). Требуемый системный вызов указывается в регистре ВХ, а остальные РОН и (или) верхушка стека пользовательской фазы содержат параметры вызова. Передача параметров в регистрах быстрее, чем через стек, хотя в РС/IX (совместимой с Unix по системным вызовам ОС) реализована быстрая передача через стек. Там библиотечные функции для системных вызовов не используют стек, поэтому параметры, переданные этим функциям, доступны ядру по адресу на слово выше текущего указателя стека пользовательской фазы процесса. А по текущему указателю хранится адрес возврата из системного вызова. При такой организации не требуется еще одно копирование в регистры или стек, но каждый системный вызов занимает свой вектор прерываний, и накладываются жесткие ограничения на формат ассемблерных программ, реализующих пользовательскую фазу системных вызовов, — в них запрещается использовать стек.

**Управление памятью.** Память рассматривается как блоки по 512 байт с абсолютными адресами, кратными 512. При загрузке системы именно так устанавливается карта свободного пространства памяти. Хотя из соображений сегментации минимальный размер блока можно было сделать 16 байт, но совпадение размеров блока памяти и физического блока ввода-вывода упростило управление вводом-выводом и повысило его эффективность. Размер логического блока ввода-вывода (единицы распределения памяти в файловой системе) выбран 1 Кбайт. Тем самым обеспечивается повышение скорости ввода-вывода и совместимость с файловой системой ИНМОС СМ.

**Размещение контекста.** На СМ детальная информация о процессе (ДИП) и стек ядра для данного процесса (СЯП), составляющие контекст процесса, хранятся в одной области физической памяти на процесс (ОПП). ДИП находится в начале ОПП, СЯП — в конце. ОПП отображается в виртуальное адресное пространство ядра, когда данный процесс становится текущим. Так как

в ПК нет страничной организации памяти, то эта схема не годится.

Вместо отображения ОПП можно использовать копирование всего контекста, сохраняя тот же формат ОПП (так реализован Venix/86). В этом случае при каждом переключении с процесса на процесс требуется копирование всей ОПП в адресное пространство данных ядра. Так как размер ОПП не менее 1 Кбайт, то при нескольких запущенных процессах и частых переключениях быстродействие системы падает.

Можно хранить ДИП в адресном пространстве ядра, в описателе процесса, а СЯП отображать в адресное пространство стека ядра с помощью установки сегментного регистра SS. Тогда мы избегаем необходимости всякого копирования. Но так как при обращении к статическим данным абсолютный адрес вычисляется с использованием DS, а при обращении к локальным — через SS, то статические и локальные данные программы (в данном случае ядра) оказываются в различных сегментах. Тем самым нарушается подразумеваемое компилятором Си единство адресного пространства данных. Для правильной работы программы с такой сегментной организацией требуется крайне аккуратное программирование при работе с указателями на локальные данные — дело в том, что память под локальные данные отводится в сегменте стека, определяемом по SS, в то время как каждое адресное выражение интерпретируется относительно сегмента данных, который определяется по DS. Следовательно, запрещаются операции записи по указателю на локальные данные, недопустимы локальные массивы. Такие ограничения должны бы соблюдаться не только при репрограммировании собственно ядра, но и в разрабатываемых пользователями драйверах. Поэтому разделение адресного пространства данных на два сегмента было отвергнуто. Отметим все же, что это не бесплодный подход: он реализован в РС/IX.

В ИНМОС ПК ДИП хранится в адресном пространстве ядра, а в ОПП располагается только СЯП. При переключении процессов копируется только занятая верхушка стека ядра — обычно несколько десятков байт. Тем самым обеспечивается быстрое переключение процессов без искусственных ограничений на программирование ядра и драйверов.

**Размещение процессов в памяти.** В ИНМОС ПК процесс в памяти представлен четырьмя областями: текстом (собственно программой), данными, стеком процесса и ОПП. Из предыдущего рассмотрения относительного расположения сегментов стека и данных для программы на Си в этой архитектуре следует, что величины сегментных регистров DS и стека SS должны совпадать. Для обработки переполнения стека и запросов дополнительной памяти требуется возможность изменения размеров сегментов данных и стека процесса при выполнении программы. Поэтому желательно, чтобы данные процесса начинали, а стек заканчивал физический сегмент памяти максимально допустимого размера (64 Кбайт), т. е. чтобы данные начинались с адреса 0 адресного пространства данных процесса, а стек завершался адресом 0xFFFF того же пространства.

Можно при запуске процесса сразу выделять под его данные и стек сегмент максимально допустимого размера (64 Кбайт), что реализовано в РС/IX. Тогда при попытке наложения данных и стека возникает неустраиваемая ошибка, вызывающая останова программы. Тем самым повышается надежность ОС. Так как никакой процесс, написанный на Си, физически не может обратиться вне максимально допустимого сегмента 64 Кбайт, то и не сможет испортить информацию, ему не принадлежащую.

Но учитывая, что для запуска большинства программ (системные команды) требуется несколько килобайт, захватывать по 64 Кбайт оперативной памяти на данные каждого работающего в системе процесса слишком накладно. Поэтому в ИНМОС ПК память под

данные и стек процесса выделяется в таких размерах, чтобы обеспечивалось условие взаимного расположения данных и стека в начале и конце 64 Кбайт-отрезка физической памяти, а центральная часть отрезка свободна и может быть занята текстом и (или) другим процессом.

При расширении стека или данных процесса взаимное их расположение сохраняется. Иногда приходится захватывать свободную область размером не в величину расширения, а для новых стека и данных, затем копировать их и освобождают память, занятую старыми.

Для повышения надежности в ИНМОС ПК добавлены форматы загрузочного файла, в которых стек процесса размещается перед данными, с адреса 0. При таком формате процесс не может изменить размер стека. Компилятор в этом случае генерирует код с проверкой размера стека перед каждым входом в подпрограмму, и полученная программа позволяет обнаружить ошибки перехода за границу стека. В то же время все отлаженные программы занимают минимально необходимый размер оперативной памяти.

**Режимы ИНМОС ПК.** В ИНМОС ПК существуют три режима: пользовательский, системный, прерывания (внешнего). Переход из пользовательского режима в системный происходит, когда процесс обращается к ядру. При этом сохраняются все регистры, кроме указателя стека SP, в стеке пользователя. Затем ядро переключает процессор на текущий системный стек (СЯП), в котором запоминается указатель стека пользователя. Через этот указатель ядро получает доступ к содержимому пользовательских регистров. Далее устанавливается адресное пространство ядра, ядро переходит в системный режим, и вызывается эмулятор сопроцессора с плавающей запятой или подпрограмма ядра, выполняющая требуемый системный вызов. Аналогично обрабатываются прерывания из пользовательского режима. Если прерывание пришло в системном режиме или во время другого прерывания, то переключения стека не требуется — прерывания вкладываются и по окончании восстанавливают указатель стека SP.

**Загрузка ядра.** В ИНМОС GM начальный загрузчик запускает мощный вторичный загрузчик, который, в свою очередь, позволяет загрузить нужное ядро в указанную область памяти в требуемом режиме. В диалоге с загрузчиком можно задать имя ядра, режим, установки имен системных устройств и т. д. В ИНМОС ПК начальный загрузчик, находящийся в первом секторе дискеты, сразу грузит ядро с заданным именем из головной директории, а ядро потом само копирует себя в рабочую область памяти. Аналогично происходит загрузка с жесткого диска.

**Числа с плавающей запятой.** Ядро обеспечивает эмуляцию подмножества команд сопроцессора с плавающей запятой 18087. Компилятор при обработке команд над числами с плавающей запятой генерирует инструкции из подмножества команд сопроцессора, которые должны либо выполняться им, либо эмулироваться. Перед каждой последовательностью инструкций сопроцессора генерируется инструкция внутреннего прерывания по эмуляции сопроцессора с плавающей запятой. Эмулятор распознает инструкцию сопроцессора, определяет расположение, типы операндов, требуемое действие и выполняет его.

#### Отладка

ПК не имеет ни встроенного монитора, ни пульта. Недоступен был и аппаратный отладчик. При загрузке ИНМОС захватывает для собственного использования векторы прерываний, устанавливает в нуль всю память машины.

Клавиатура и терминал тоже обслуживаются ИНМОСом, поэтому понимание причин очередного зависания превращалось в гадание на кофейной гуще. В итоге выработалась следующая технология отладки: собранное ядро ИНМОС с неподключенными векторами прерываний пере-

носилось в ОС MS-DOS, там загружалось самым простым доступным отладчиком, чтобы возникало как можно меньше коллизий, регистры вручную устанавливались для запуска ядра, и оно отлаживалось практически в пошаговом режиме. Когда все системы начальной инициализации, выделения ресурсов, переключения процессов, распределения процессора заработали, был написан и отлажен начальный загрузчик с дискеты, подключены прерывания и таймер в ядре, и отладка вновь пошла методом проб и ошибок, но существенно быстрее: основные ошибки все-таки были исправлены.

#### Заключение

Перенос ядра на ПК занял четыре человеко-месяца до получения первой примитивной работающей версии с драйверами строчного терминала и гибких дисков (работы начались в мае 1986). В дальнейшем были разработаны: драйвер дисплея с буферизацией вывода и эмулятором управляемого символическими последовательностями экранного дисплея; четырехрегистровый (строчные и прописные буквы, кириллица и латиница) драйвер клавиатуры; драйвер жесткого диска емкостью 5...64 Мбайт; драйверы для нескольких принтеров; загрузчики с гибкого и жесткого дисков, программы инсталляции. Создан комплект поставки (девять гибких дисков). В декабре 1987 ИНМОС-И (модификация ИНМОС ПК для Искры 1030) прошел государственные испытания.

Телефон 132-19-56, Москва

#### ЛИТЕРАТУРА

1. Jalics P. G., Heines S. H. Transporting a portable operating system: Unix to an IBM minicomputer // Communications of the ACM. — 1983. — Vol. 26, N. 12. — P. 1066—1072.
2. Tilson M. Moving Unix to New Machines // Byte. — October 1983. — P. 266—276.
3. Беляков М. И. Принципиальные основы и перенос Инмос. — В кн.: Готье Р. Руководство по операционной системе — М.: Финансы и статистика, 1985. — С. 205—228.
4. Беляков М. И., Ливеровский А. Ю., Семик В. П., Шяудкулис В. С. Инструментальная мобильная операционная система ИНМОС. — М.: Финансы и статистика, 1985.
5. Микропроцессоры. В 3-х книгах. Кн. 1. Архитектура и проектирование микроЭВМ. Организация вычислительных процессов. — М.: Высшая школа, 1986. — С. 219—244.
6. Rood A. L., Cline R. C., Brewster J. A. Unix and MC68000 // Byte. — 1986. — Vol. 11, N 9. — P. 179—200.

Статья поступила 20.03.87

#### РЕЦЕНЗИЯ

на статью И. Л. Бельчинского «Перенос ядра ИНМОС на машины различных архитектур»

Важность и актуальность работы, посвященной мобильной операционной системе (ОС) типа Unix, не вызывает сомнения. Интригующий заголовок статьи сразу же вызывает повышенный интерес. Однако синтаксическая ошибка в первом же предложении<sup>1</sup> противоречие второго абзаца заголовку и, наконец, перл из третьего абзаца, что «всякий компьютер обладает своей собственной ОС»<sup>2</sup>, создают у читателя гнетущее впечатление, которое не рассеивается до самого конца статьи.

При самом доброжелательном отношении и повышенном интересе к работе я не нашел в рецензируемой статье ни одного положительного момента.

Название статьи «Перенос ядра ИНМОС на машины различных архитектур» не соответствует объявленному в «вступлении» переносу «на ряд совместимых машин...», что в свою очередь не соответствует содержанию работы. Фактически, кроме претенциозных заголовков типа «Подходы к переносу», «Перенос ИНМОС: принятые решения» и поверхностного изложения технологии отладки, в статье о проблемах переноса ядра ничего не говорится. Автор даже не упомянул о трудностях переноса ядра на новые машины, о возникающих при этом несовместимостях и их причинах, ни слова не сказано о соотношении трудностей переноса ядра и системы в целом (заметим, что крупным фирмам на полномасштабную и качественную адаптацию Unix для новой машины требуется 2...3 года работы).

Вместо этого автор в первой половине статьи приводит свое понимание функционирования отдельных элементов ИНМОС, а вторая половина посвящена описанию опять-таки отдельных элементов ИНМОС ПК.

Из статьи не ясна цель проделанной автором работы. Спорными являются также сформулированные автором три цели, которые, кстати говоря, не имеют отношения к излагаемому далее материалу:

1. «Исследовать практически (?) требования к архитектуре процессора для многозадачной ОС, которые выявятся в процессе переноса».<sup>3</sup>

Исследовать требования — задача стоящая, но при чем здесь перенос? Или существуют какие-то особые требования, которые важны именно для процесса переноса, а не для функционирования «многозадачной ОС»?

2. Исследовать трудности переноса ОС людьми без предварительных детальных знаний ее устройства».<sup>2</sup>

Создается впечатление, что в стране (в ИПИ АН СССР) так много машин новых архитектур, что для переноса ОС уже не хватает подготовленных и квалифицированных людей и приходится прибегать к услугам «людей без... детальных знаний». Или это проблема из области «опытов над людьми»? Тогда ее уместно исследовать в работах по психологии, философии или педагогике.

3. Обеспечить унифицированную многозадачную ОС для ПК, совместимую... с популярной ОС Unix.

Но ведь у автора в распоряжении уже имеется ОС Venix/86, о чем он сообщает читателю. Так что эта цель достигнута еще до начала работы.

Статья написана крайне сумбурно, со множеством неточностей и фактических ошибок. Так, характеризуя архитектуру процессора K1810VM86, автор растворяет действительно важные элементы (стековую организацию, структуру виртуальной памяти, разрядность данных) в потоке несущественных для статьи деталей. Статья сообщает, что в процессоре «предусмотрено множество команд, расширяющих общие возможности»<sup>4</sup>, хотя об «общих возможностях» умалчивается. «Отсутствие приоритетного режима» никак не может усложнить реализацию ни системной, ни тем более пользовательской фазы, разве что увеличить максимальное время отклика на прерывание. Доступ к регистрам устройств, расположенным в адресном пространстве памяти, возможен не только без страничной организации, но и без виртуальной памяти вообще.

Сваливаются в кучу линейное виртуальное пространство, страничная и сегментная организации памяти — совершенно различные вещи; сегментированное виртуальное пространство ошибочно называется линейным (неоднократно). Утверждение «процессы выполняются в пользовательском режиме» противоречит существованию системной фазы процесса. Список нелепостей можно продолжить, иногда дело доходит до абсурда: «желатель-

но при смене режима...» иметь возможность указать «менять ли режим».

Некоторые рассматриваемые вопросы (например, о примитивах fork/exec) относятся к теме не более, чем любые другие из книг о Unix; в то же время машинно-зависимые функции явно так и не локализованы.

Большая часть «выводов» — тривиальна, остальные — нелепы. ИНМОС требует для эффективной и надежной работы 16-или 32-битовый процессор? Разве автор видел многопользовательскую систему на 8-битовом? Или может быть 64-битовый CRAY работает под Unix ненадежно? Разве страничная и сегментная организации памяти нужны в Unix больше, чем в любой другой системе разделения времени? Команда переключения контекста, конечно, очень полезна, но в Unix меньше, чем в ОС PB или MOS ВП ИНМОС нужна страничная организация памяти? Но в K1810VM86 ее нет (в CRAY-1, впрочем, тоже!). Т. е. те «необходимые требования», которые стали «очевидны» автору в «процессе переноса»<sup>3</sup>, другим специалистам очевидны и без того.

Сумбурность статьи дополняется полным отсутствием обозначенной иерархии разделов, которая по замыслу автора все же имеется.

Что касается стиля, то полная непродуманность и несвязанность изложения оставляет еще более тягостное впечатление, чем множество грамматических ошибок, терминологических неточностей и использование жаргона («плавучка», «голая машина»<sup>1</sup>). Разделы «управление памятью» и «размещение процессов в памяти» невозможно понять даже при знании ядра Unix и архитектуры целевого процессора (1—2 картинки были бы здесь полезны и при корректном описании).

Если упоминаемый перенос ИНМОС выполнялся так же аккуратно и квалифицированно, как написана рецензируемая статья, и полученная версия сносно работает, то это действительно эффективная демонстрация мобильности ИНМОС.

В статье рассматриваются очень частные и специфические вопросы, да и те вне их связи с существующим опытом. Излагаемые теоретические результаты самоочевидны.

Таким образом, статья не содержит сколько-либо значительного общественно-полезного опыта или анализа проблем переноса ИНМОС. Научная ценность статьи сомнительна и публиковать ее в журнале не целесообразно.

Доктор физ.-мат. наук С. В. Клименко

#### УВАЖАЕМАЯ РЕДАКЦИЯ!

Я прочел рецензию на статью нашего сотрудника Бельчинского И. Л. «Перенос ядра ИНМОС на машины различных архитектур». Как один из старейших рецензентов и авторов Вашего журнала считаю себя вправе высказать свои замечания по поводу этой рецензии, поскольку не могу согласиться с выводом рецензента о целесообразности публикации статьи.

Основная причина такого вывода содержится, видимо, в том месте рецензии, где рецензент заявляет, что сама постановка задачи — перенос ИНМОС на машины типа ЕС1841 бессмысленна, так как у автора уже имеется система типа Юникс — Venix. ОС типа Юникс мобильна, т. е. допускает постановку на ЭВМ различной архитектуры, обеспечивая тем самым единую операционную среду для всего остального программного обеспечения. Именно в этом одно из основных достоинств Юникс. Однако за рубежом имеется ряд не вполне совместимых версий Юникс, и на машинах разных фирм стоят разные версии. Поэтому системы типа Юникс, поставляемые, например, для машин семейств PDP-11, VAX и IBM-PC, не полностью совместимы. При прямой «адаптации» этих

<sup>1</sup> Исправлено. <sup>2</sup> Формулировка изменена на эквивалентную по смыслу. <sup>3</sup> Часть первой редакции статьи, относящаяся к требованиям к архитектуре процессора и ЭВМ, снята. <sup>4</sup> Исключено из статьи. (Прим. авт. статьи).

систем к аналогичным отечественным ЭВМ совместимости также не будет. Именно такой путь принят при разработке ОС ДЕМОС (также типа Юникс), в результате чего уже защищаются диссертации по разработке методики обеспечения программной совместимости версий мобильной ОС ДЕМОС для разных машин. При таком подходе теряется смысл создания мобильной ОС (не говоря о том, что «адаптация» ОС к полностью совместимой машине квалификации, естественно, не требует и интереса с точки зрения публикаций не представляет).

Автор статьи ставит задачу именно переноса существующей ОС на машину с другой архитектурой. Это принципиально иной подход, гарантирующий получение полностью совместимой операционной среды. Именно в этом смысл и ценность работы.

Неправильно трактует рецензент также выводы автора. Автор вовсе не пытается научно обосновать действительно очевидные и тривиальные утверждения о желательности аппаратной поддержки для сложной многопользовательской ОС. Он говорит о том, что реализация Юникс на процессоре K1810 возможна, но система будет работать значительно эффективней при наличии дополнительных аппаратных средств. Единственное, в чем здесь рецензент прав, — это в том, что заключение практически не связано с основной темой статьи.

Необоснованы также сомнения рецензента в качестве работы, послужившей темой статьи, и в сроках ее выполнения. В статье речь идет только о переносе ядра ИНМОС, а не о создании новой версии системы, которая является собственностью разработчика; последнее, естественно, гораздо более трудоемкая работа. А перенос ядра (что, кстати, стало только частью работы по постановке ИНМОС на ПЭВМ типа ЕС1841) действительно выполнен автором в указанные сроки.

Я не хочу останавливаться на замечаниях рецензента о «сумбурности» или использовании неверной терминологии. С требованием исправить и сократить статью я полностью согласен, однако возражаю против вывода рецензента: «В статье рассматриваются очень частные и специфические вопросы, да и те вне их связи с существующим опытом. Излагаемые теоретические результаты самоочевидны. Таким образом, статья не содержит сколько-либо значительного общественно полезного опыта и анализа проблем переноса Юникс. Научная ценность статьи сомнительна и публиковать ее в журнале нецелесообразно».

Я считаю статью очень существенной по тематике (хотя теоретических результатов она вообще не содержит), весьма актуальной и полагаю, что она должна быть (после сокращения и некоторой корректировки) опубликована в Вашем журнале.

К т. н. Ландау И. Я.

## УВАЖАЕМАЯ РЕДАКЦИЯ!

Давайте для начала определим тему разговора. Вопреки замечанию И. Я. Ландау, предшествующая рецензия отнюдь не касалась «качества работы, послужившей темой статьи»: оценить его на основании статьи невозможно. А вот сомнение в качестве самой статьи действительно выразилось и очень сильное.

Некоторая нечеткость изложения нередко встречается в первых вариантах и, но это же вопиющий случай! Даже специалисту в данной области, обладающему опытом работы в Unix, знающему ядро, т. е. фактически тему статьи, стоило немалых усилий и времени понять, о чем идет речь в большей ее части; некоторые места «расшифровать» так и не удалось. Что же ожидать большинству читателей? Написанное претендует на манжетные записки, но никак не на

печатную работу и не может быть представлено широкой аудитории. Даже если бы работа была очень важной и актуальной (см. далее), ее следовало бы не подправлять, а писать заново.

«Автор статьи ставит задачу именно переноса существующей ОС на машину с другой архитектурой...»

И. Я. Ландау

Но Xenix и Vcpix — тоже результаты переноса уже существовавшей ОС Unix V7 для PDP-11 на машину архитектуры IBM-PC, т. е. отправная точка, конечная точка и, по крайней мере, общее начертание пути совпадают, причем автор не сопоставил полученную им версию с существующими. В чем же «принципиально иной подход». «Смысл и ценность» работы? Наконец, связь с существующим опытом, очевидно, отсутствует.

Резюме: какая-либо необходимость (актуальность) обсуждаемой работы не была обоснована не только в статье, но и после, когда вопрос был поднят специально рецензентом.

Мы можем представить следующие побудительные причины для переноса Unix V7 с PDP-11 на ЕС1840:

- 1) обеспечить «прозрачность» работы с русским символьным набором;
- 2) получить систему в исходных текстах;
- 3) приобрести технологический опыт для дальнейшего развития системы.

Ни одна из этих причин не была упомянута И. Л. Бельчинским или И. Я. Ландау. Единственная попытка мотивации предпринята в письме И. Я. Ландау, увы, несостоятельная. Действительно, сам по себе перенос не гарантирует получения полностью совместимой операционной среды. Более того, *подход* вообще ничего *гарантировать* не может. Таким образом, основная идея письма И. Я. Ландау — аккуратный перенос одной версии на все машины как папачей — технологически ошибочна. Можно также показать, что она бесплодна и в стратегическом плане, отметив неизбежность различий и фундаментальную причину их возникновения — развитие системы.

Окончательная точка в развитии живой системы только одна — смерть. Поэтому, пока система нужна, абсолютной совместимости между версиями не будет никогда: это непреложная диалектика развития. Но смысл создания мобильной ОС из-за этого никоим образом не теряется. Понять, что, в конце концов, хочет И. Л. Бельчинский, нельзя, но т. Ландау стремится привести результат обсуждаемых трудов в эту «замороженную» точку. Между тем, «...переносимость» средств, разрабатываемых в системе типа Unix, основывается не на жестко аксиоматизированной модели виртуальной машины, правильная работа которой гарантируется лишь в случае точной ее реализации ..., а, совсем напротив, на свободном сочетании и взаимодействии машинно-зависимых и машинно-независимых средств, при котором заранее не известно, что при переходе в новое окружение сохранится, а что заменится. Переносимость обеспечивается не тем, что система отгораживается от изменений окружения, а тем, что она хорошо расчленена на элементы, и изменения в окружении затрагивают лишь немногие из них» (Г. С. Цейтин, частная переписка).

Таким образом, с нашей точки зрения, более правильным путем, в противоположность предлагаемой канонизации определенной (причем уже устаревшей) версии Unix и отгораживанию от развивающегося мира, является выработка и программная поддержка ориентированной на развитие и изменение окружения методологии мобильного программирования, куда мы также относим текущие работы подкомитетов IEEE, ANSI

и ряда специальных объединенных групп ведущих производителей вычислительной техники или/и программного обеспечения Европы, США и Японии по определению лучшего (на данный момент и обозримое будущее) варианта спецификаций интерфейсов операционной среды и куда, по крайней мере в принципе, могут быть отнесены отечественные диссертации о методике поддержки мобильности программного обеспечения в Unix.

Машинная независимость и переносимость — различные понятия. ОС Unix является переносимой, но машинно-зависимой. В свете настоящего соображения мы рискуем посоветовать т. Ландау, по возможности, избегать гладких переходов от «не вполне совместимы» к «совместимости не будет».

Кстати, адаптация даже к совместимой машине, как и любая работа с операционной системой (да и разработка программного продукта вообще), должна требовать квалификации, и весьма высокой, в зависимости от целей.

Безграмотность, низкий профессиональный уровень и оторванность статьи от существующего опыта несомненны. Причем И. Я. Ландау не отрицает ее основных недостатков, а также не опровергает соответствующих утверждений и многочисленных (но далеко не исчерпывающих) примеров предыдущей рецензии. В силу сказанного мы опять приходим к вы-

воду: статья не содержит сколь-либо значительного общественно-полезного опыта или анализа проблем переноса ИНМОС; научная ценность статьи сомнительна, и публиковать ее в журнале нецелесообразно.

Вообще же печатная работа на данную тему может быть интересной, что никак не связано с актуальностью или полезностью. Но не такая.

Нам хочется верить, что сделанный вывод относится только к статье и не приложим к работе в целом. Мы надеемся, что в проведенной работе на самом деле были поставлены разумные цели, приняты любопытные решения, получены примечательные результаты, и только низкое качество самой статьи растворило их. В таком случае вполне возможно вновь представить статью к публикации «после сокращения и некоторой корректировки» согласно замечаниям рецензента (правда, если следовать замечаниям рецензента, то будет другая статья). Естественно, что после существенной переработки статью следует опять представить рецензенту (хотя и не обязательно первому, впрочем, этот вопрос уже находится в компетенции редакции).

Наконец, имеет смысл повторить: недопустимо отрывать перевод ядра от переноса системы в целом. Конечно, статья посвящена ядру, но невозможно полностью умолчать об остальном.

С. В. Клименко

## САПР

УДК 681.5.04—519.714

Ю. А. Антонов, П. Н. Бибило, Е. И. Гольдберг, И. П. Каркоцкая, Н. П. Чигирь

## СИСТЕМА ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ НА ПРОГРАММИРУЕМЫХ МАТРИЧНЫХ БИС

Система позволяет получать настроечные матрицы ПЛМ и ПЗУ автоматизированным способом из различных функциональных описаний проектируемого цифрового устройства, таких как микропрограммы, скобочные и матричные формы систем булевых функций и системы секвенций. САПР реализована на мини-ЭВМ СМ4, состыкована с программатором микросхем и позволяет синтезировать логические сети на ПЛМ и ПЗУ серии К556. Синтезируемые схемы могут быть одноуровневыми и многоуровневыми.

Структура системы и ее использование. Система состоит из подсистем ввода-вывода и контроля данных; диалогового синтеза цифрового устройства, представляющего собой комбинационный либо последовательностный автомат; программирования микросхем ПЛМ и ПЗУ (рис. 1). Входные данные вводятся в символьном или табличном виде и представляют собой текстовые файлы, которые могут редактироваться стандартными редакторами операционной системы. Программы ввода данных преобразуют внешние формы входных дан-

ных в компактные внутренние формы, с которыми работают программы синтеза. Внутренние формы входных и выходных данных могут быть неизвестны проектировщику, работающему с системой. Синтезированные схемы выдаются на АЦПУ или дисплей. Для каждой микросхемы ПЗУ или ПЛМ формируется своя таблица настройки. Система позволяет сохранять в архиве входные данные и результаты синтеза. Имеется возможность сохранения в архиве промежуточных результатов синтеза, что позволяет прерывать процесс синтеза и проводить синтез схемы

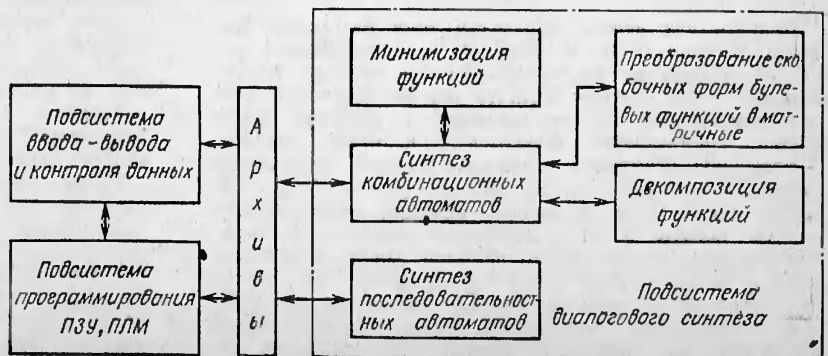
за несколько сеансов работы с системой.

Представление входной и выходной информации. Входными данными являются системы секвенций, булевых функций и микропрограммы. Системы секвенций и булевых функций представляются в матричной (табличной) форме. Функции могут быть определены полностью и частично и задаваться наборами значений множеств аргументов и в интервальной форме системами дизъюнктивных нормальных форм (ДНФ) булевых функций [1].

Система булевых функций  $F_1, F_2, F_3$  может быть задана в алгебраической форме или Парой матриц.

Исходная система скобочных форм булевых функций представлена ниже.

рис. 1. Структура системы проектирования





$A1 \ B1 \ A2 \ B2 \ A3 \ B3 \ A4 \ B4;$   
 $C1 \ C2 \ C3 \ C4 \ S1 \ S2 \ S3 \ S4;$   
 $S1 \ S2 \ S3 \ S4 \ C4;$   
 $S1 = A1 * \neg B1 + \neg A1 * B1;$   
 $C1 = A1 * B1;$   
 $S2 = C1 * (\neg A2 * \neg B2 + A2 * B2) + \neg C1 * (\neg A2 * B2 + A2 * \neg B2);$   
 $C2 = A2 * B2 + (A2 + B2) * C1;$   
 $S3 = C2 * (\neg A3 * \neg B3 + A3 * B3) + \neg C2 * (\neg A3 * B3 + A3 * \neg B3);$   
 $C3 = A3 * B3 + (A3 + B3) * C2;$   
 $S4 = C3 * (\neg A4 * \neg B4 + A4 * B4) + \neg C3 * (\neg A4 * B4 + A4 * \neg B4);$   
 $C4 = A4 * B4 + (A4 + B4) * C3;$

Система функций  $S1... S4, C4$  описывает четырехразрядный сумматор для сложения чисел  $A=(A4, A3, A2, A1), B=(B4, B3, B2, B1), A1, B1$  — младшие разряды,  $C1, C2, C3, C4$  — переносы,  $S=(S4, S3, S2, S1)$  — сумма  $A+B$  чисел  $A, B$ . Сумматор реализован в виде одноуровневой схемы на двух ПЛМ К556РТ1.

Микропрограммный автомат, граф-схема которого изображена на рис. 2 (пример взят из работы [5]), задается в виде двух текстовых файлов [6].

**GSA1**  
 1H:2.  
 20:MK1,3.  
 3Y:X1,4,10.  
 4Y:X2,6,5.  
 5Y:X3,7,4.  
 60:MK2,8.  
 70:MK3,9.  
 80:MK4,9.  
 9Y:X2,14,15.  
 100:MK5,11.  
 11Y:X4,12,8.  
 12Y:X1,12,13.  
 130:MK7,15.  
 140:MK6,13.  
 15K:

**MKGSA1**  
 MK1:F1,F3.  
 MK2:F2,F3.  
 MK3:F4.  
 MK4:F1,F4.  
 MK5:F2.  
 MK6:F5.  
 MK7:F6.  
 END

Секвенциальный автомат задается парой трюичных матриц [1].

Выходными данными являются логические сети (схемы) в базисе ПЛМ и ПЗУ. Информация о структуре отдельной ПЛМ, задающей какой-то элемент логической сети, представляется в виде текстового файла, разделенного на записи. Каждая запись задает конъюнкцию (логическое произведение входных переменных) и вхождение этого произведения в ДНФ тех булевых функций, которые будут реализованы на данной ПЛМ. По желанию проектировщик может запрограммировать ПЛМ или ПЗУ самостоятельно, подготовив исходные данные для программатора, не пользуясь программами синтеза.

Подсистема диалогового синтеза обеспечивает решение следующих задач.

1. Реализация системы секвенций на «абстрактной» ПЛМ с памятью в виде регистра RS-триггеров. Абстрактной ПЛМ назовем ПЛМ, не удовлетворяющую хотя бы одному из ограничений на параметры ПЛМ.
2. Реализация микропрограммного автомата на абстрактной ПЛМ с памятью в виде регистра D-триггеров.
3. Преобразование скобочных форм системы булевых функций в матричные (табличные) формы.
4. Реализация матричных форм булевых функций сетями ПЛМ и ПЗУ.

Эти программы позволяют представить абстрактные ПЛМ в виде сети базисных ПЛМ и (или) ПЗУ. Базисные ПЛМ задаются при синтезе тройкой чисел  $t, q, s$ , где  $t$  — число входных полюсов ПЛМ,  $q$  — число промежуточных шин ПЛМ,  $s$  — число выходных полюсов. Для задания базисных ПЗУ достаточно чисел  $t$  и  $s$ , так как  $q=2$ . Например, для ПЛМ К556РТ1  $t=16, q=48, s=8$ , для ПЗУ К556РТ5  $t=9, s=8$ .

Программы дают возможность проектировщику минимизировать системы булевых функций и число аргументов функций за счет удаления фиктивных аргументов, провести одноуровневый и многоуровневый синтез. Реализованы методы многоуровневого синтеза: ортогонализации, группирования ортогональных конъюнкций, множественных отображений в пространстве промежуточных переменных [1,4].

Синтез схем можно вести и комбинированными методами. Выбор метода

или их комбинации осуществляет проектировщик в зависимости от вида и размерности исходного задания и требований к схеме (быстродействие, сложность). Синтезируемые схемы состоят не более чем из 15 ПЛМ К556РТ1.

5. Извлечение элемента синтезированной логической сети и представление его в формате для программирования ПЛМ или ПЗУ. В последнем случае система ДНФ булевых функций преобразуется в таблицу истинности.

Программы подсистемы реализованы на ЭВМ СМ4 в языке ЛЯПАС-М [5], эффективным для решения задач логического синтеза. Функционируют в среде операционной системы РАФОС-2. Для работы с подсистемой синтеза требуются два пакета магнитных дисков для архива пользователей.

Проблема повышения быстродействия и минимизации объема оперативной памяти, отводимой непосредственно под программы, решена путем оптимального размещения подпрограмм по оверлейным сегментам. Это позволяет обрабатывать системы булевых функций с числом аргументов  $p \leq 128$ , функций  $m \leq 128$ , конъюнкций  $k \leq 400$ . («разреженные» системы функций такой размерности эффективно реализуются на ПЛМ). Системы функций с большим числом конъюнкций дизъюнктивно разлагаются на ряд подсистем, реализуемых независимо.

Подсистема программирования микросхем ПЛМ и ПЗУ представляет собой программно-аппаратурный комплекс, включающий программатор и систему программ, обеспечивающую программирование микросхем ПЛМ и ПЗУ серии К556 и позволяющую реализовать следующие функции:

- «ручная» подготовка и редактирование таблицы данных для программирования;
- входной и выходной контроль;
- программирование;
- допрограммирование (корректировка).

Программатор (рис. 3) представляет собой микропрограммный контроллер, выполненный на плате типа Е2 конструктива УК СМ, и подключается к «общей шине» управляющей ЭВМ непосредственно или через блок расширения системы.

Программируемая микросхема устанавливается в розетку адаптера типа микросхемы, который через внешний разъем подключается к программатору.

Экспериментальные данные. 1. Автомат, представляющий порт активного устройства для осуществления обмена данными через канал связи типа «общая шина», был представлен ГСА, содержащей 92 вершины. За 56 с работы ЭВМ СМ4 описание преобразуется в систему ДНФ 47 функций, зависящих от 17 аргументов и заданных на 97 конъюнкциях. Полученная система ДНФ реализована одноуровневой схемой из семи ПЛМ К556РТ1. Время

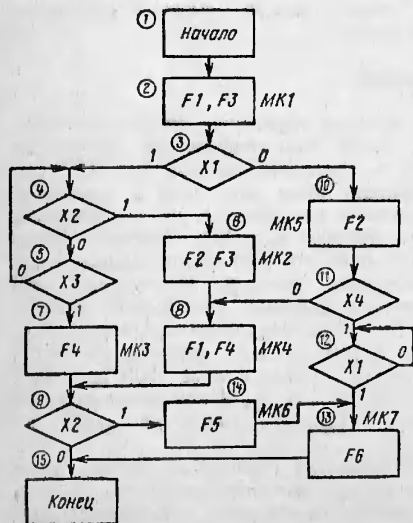


Рис. 2. Граф-схема алгоритма (ГСА)

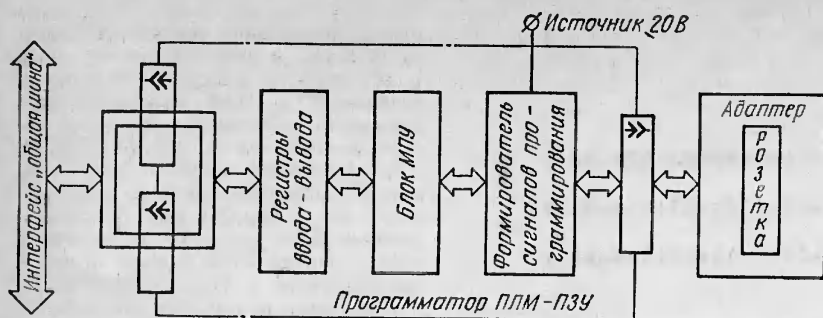


Рис. 3. Структура программатора

синтеза составило 40 с. Синтезированное автоматизированным способом устройство обладает в три раза большим быстродействием, чем синтезированное «вручную», при том же объеме аппаратных средств.

2. Автоматизированный синтез умножителя двух 4-разрядных чисел, заданного алгебраическими скобочными выражениями, потребовал 2 мин времени для получения ДНФ восьми функций, зависящих от восьми аргументов. Состоящая из 161 конъюнкции система ДНФ может быть реализована одноуровневой схемой из четырех ПЛМ К556РТ1 либо на одном ПЗУ К556РТ5. При задании того же умножителя парой булевых матриц синтез схемы в базисе ПЛМ сводится к сов-

местной минимизации системы совершенных ДНФ булевых функций. Время минимизации данной системы, дизъюнктивно разбитой на четыре подсистемы по 64 конъюнкции, составляло 36 мин процессорного времени.

Минимизированная система содержала 157 конъюнкций. Таким образом, выбор исходного функционального описания сильно влияет на время синтеза. Этот эксперимент говорит о том, что использование средств автоматизации логического проектирования переносит центр тяжести на этап алгоритмического проектирования.

С помощью системы проектирования на ПЛМ и ПЗУ выполнены типовые устройства комбинационной логики: сумматоры, умножители, кодовые пре-

образователи, схемы сравнения и устройства нерегулярной управляющей логики. В настоящее время программы диалогового синтеза адаптируются для ПЭВМ серии ЕС.

Телефон 39-50-84, Минск

## ЛИТЕРАТУРА

1. Закревский А. Д. Логический синтез каскадных схем.— М.: Наука.— 416 с.
2. Баранов С. И., Скляр В. А. Цифровые устройства на программируемых БИС с матричной структурой.— М.: Радио и связь, 1986, 272 с.
3. Баранов С. И., Журавина Л. Н., Скляр В. А. Автоматизация проектирования ЦВМ.— Минск: Высшая школа, 1981.— 206 с.
4. Бибило П. Н., Гольдберг Е. И., Каркоцкая И. П., Чигирь Н. П. Система логического проектирования дискретных устройств на программируемых матричных БИС. Минск: Ин-т техн. кибернетики АН БССР, 1987, 82 с.
5. Закревский А. Д., Торопов Н. Р. Система программирования ЛЯ-ПАС-М.— Минск: Наука и техника, 1987.— 238 с.

Статья поступила 25.04.88

УДК.681.3.

С. В. Андреев, А. Б. Ходулев

## ВЫВОД ГРАФИЧЕСКОЙ ИНФОРМАЦИИ НА ТОЧЕЧНО-МАТРИЧНЫЕ ПЕЧАТАЮЩИЕ УСТРОЙСТВА

Для получения твердой копии обычно используют две группы устройств: символьные печатающие устройства и графопостроители. Традиционные символьные печатающие устройства печатали только буквы и цифры, графопостроители чертили ломаные и кривые, из которых складывался чертеж, рисунок или график. Сейчас же обе группы по многим функциям перекрывают друг друга.

Большинство точечно-матричных печатающих устройств (принтеров) кроме литер позволяют также печатать и индивидуальные точки и при этом обеспечивают достаточно высокое разрешение. Причем, на одном и том же устройстве можно получить изображения с различным разрешением. Например, устройство EPSON FX может в различных режимах печатать точки с интервалами 1/60 дюйма (0,423 мм), 1/72 дюйма (0,353 мм), 1/120 дюйма (0,212 мм) и 1/144 дюйма (0,177 мм) по горизонтали. Расстояние между точками по вертикали определяется расстоянием между девятью иглами печатающей головки (для EPSON FX это 1/72 дюйма). Для устройства EPSON LQ-2500 допустимы режимы печати точек с интервалами 1/60 дюйма, 1/80 дюйма (0,318 мм), 1/90 дюйма (0,282 мм) и 1/180 дюйма (0,141 мм) по горизонтали. (Существуют также и другие режимы печати.) При этом печатающая головка LQ-2500 состоит из 24 игловок, что позволяет варьировать расстоянием между

точками по вертикали: это 1/60 дюйма, если одновременно печатается один байт (т. е. работает каждая третья игла головки), или 1/180 дюйма (работают все 24 иглы). Последующее изложение будет вестись применительно к устройству EPSON FX.

На этом устройстве значение перемещения бумаги по вертикали задается с точностью до 1/216 дюйма (1/3 расстояния между иглками), поэтому, печатая каждую строку за три прохода со сдвигом бумаги между проходами на 1/216 дюйма, можно получить плотность раstra 216 точек на дюйм.

### Построение изображения

Изображение на принтер выводится последовательно, строка за строкой, тогда как графические программы строят изображение в произвольном порядке. Поэтому все изображение должно быть накоплено в некотором буфере и затем целиком распечатано. В описываемых ниже драйверах в буфере хранится битовый образ изображения. Каждый байт отвечает вертикальному столбику из восьми соседних позиций. Естественно, при использовании различных режимов потребуются буферы различной длины. Так, при заполнении прямоугольного поля размерами 8×10 дюймов потребуются 43200 байт в режиме печати с плотностью 60×72 точек на дюйм (43200=8×10×60×72/8), а при использовании режима с плотностью 144×216 точек на дюйм потребуются 311040 байт.

Первоначально изображение строится в буфере, который можно рассматривать как двумерный массив битов. Реально используется одномерный байтовый массив, из которого можно скомпоновать прямоугольную матрицу с произвольными границами по горизонтали и

вертикали. Таким образом, границы поля вывода не фиксированы и могут быть динамически изменены. Поле вывода ограничено лишь шириной каретки точечно-матричного печатающего устройства (ТМПУ) и размерами буфера.

Основным примитивом структурных схем, гистограмм и графов, архитектурных и машиностроительных чертежей являются отрезки прямых линий. Кроме того, наборы коротких отрезков могут служить достаточно хорошей аппроксимацией кривых линий. Следовательно, для построения изображения необходимо уметь строить отрезки на EPSON FX.

В данном случае рисование отрезков — это установка единиц в соответствующие разряды буфера. Для построения отрезков можно воспользоваться хорошо известным алгоритмом Брезенхема [1]. Произвольный доступ к элементам матрицы позволяет легко справиться с проблемой несовпадения отрезков [A, B] и [B, A], построенных с помощью алгоритма Брезенхема [2]. Драйвер просто рисует все отрезки в направлении слева направо. При этом не возникает лишних точек при рисовании сразу обоих отрезков [A, B] и [B, A].

### Печать с высоким разрешением

Как уже было сказано, существует возможность увеличивать плотность точек по горизонтали и тем самым улучшать качество изображения. Но при этом возникает проблема, состоящая в том, что горизонтальные линии будут выглядеть темнее вертикальных при использовании режимов с высокой плотностью горизонтальных точек, так как расстояние между точками по вертикали зависит от расстояния между иглами печатающей головки. Но, несмотря на то, что расстояние между иглами фиксировано, можно увеличить плотность точек по вертикали, выполняя сдвиг бумаги на расстояние, меньшее шага иглока [3]. При печати изображения можно воспользоваться одним из следующих алгоритмов:

первая байтовая строка составляется из 1,4,7,10,13,16,19,22 битовых строк матрицы изображения. После печати этой строки бумага сдвигается на 1/216 дюйма и печатается вторая строка, составленная из 2,5,8,11,14,17,20,23 битовых строк. Затем бумага сдвигается на 1/216 дюйма и печатается третья строка, состоящая из 3,6,9,12,15,18,21,24 битовых строк. Именно такой подход используется в описываемом ниже драйвере;

матрица изображения разбивается на три. При подготовке изображения для каждой точки с целочисленными координатами (X, Y), если остаток от  $Y/3$  равен 1, то точка заносится в первую матрицу, если остаток равен 2, то точка заносится во вторую матрицу, если кратен трем, то точка заносится в третью матрицу. При этом координаты точки в каждой матрице будут (X,  $[Y/3]$ ), где  $[Y/3]$  — целая часть от  $Y/3$ . При печати сначала печатается строка из первой матрицы, после сдвига бумаги на 1/216 дюйма печатается строка из второй матрицы, происходит сдвиг бумаги на 1/216 дюйма и печатается строка из третьей матрицы [3]. Результат работы этого алгоритма точно такой же, как у предыдущего, т. е. плотность точек по вертикали увеличивается втрое.

Для принтера EPSON LQ-2500 данный алгоритм не нужен, так как можно воспользоваться режимом печати, при котором работают все 24 игла печатающей головки. При этом достигается плотность по вертикали 180 точек на дюйм.

### Методы утолщения линий

В машинной графике более известна обратная задача: из имеющегося изображения необходимо выделить каркас за счет утончения толстых линий. Утолщение же линий необходимо для полиграфических целей, когда результаты графического вывода на ТМПУ непосредственно используются в качестве иллюстраций к статьям, пре-

принтам и т. д. Тонкие линии в этом случае не воспроизведутся фотокопировальной аппаратурой, в особенности при износе красящей ленты на принтере.

При этом необходимо утолщать все линии на изображении в равной степени. Иногда возникает потребность утолщения части линий с тем, чтобы выделить их на рисунке. Для решения этих двух задач применяются разные методы.

1. **Утолщение всех линий.** В этом случае можно, имея изображение с низкой плотностью (например,  $72 \times 72$  точек/дюйм), использовать режим печати с повышенной плотностью (например,  $144 \times 216$  точек/дюйм), печатая каждую точку исходного изображения в виде прямоугольника  $2 \times 1$ ,  $2 \times 2$  или  $2 \times 3$ . Лучше использовать прямоугольник, близкий к квадрату (на бумаге), с тем, чтобы линии разных направлений утолщались в равной степени.

Заметим, что размеры буфера изображения отвечают фактической низкой плотности изображения ( $72 \times 72$ ), несмотря на использование принтера в режиме высокой плотности. Необходимое размножение каждой точки достигается (в случае прямоугольника  $2 \times 2$ ) путем двукратной посылки каждого байта на принтер и двукратной печати каждой строки со сдвигом по вертикали.

### 2. Выборочное утолщение:

а. Один из способов утолщения отрезков прямых линий — некоторое изменение работы алгоритма Брезенхема. Для удобства рассмотрения пусть доминирующей будет координата X, т. е. на каждом шаге алгоритма координата X получит приращение +1. В этом новом алгоритме, если делается шаг по горизонтали, то в матрицу изображения заносится также и диагональная точка, а если был выбран диагональный шаг, то в матрицу заносится и горизонтальная точка. Недостаток этого алгоритма — довольно большая «ступенчатость» полученной таким образом линии.

б. Утолщение линий можно сопоставить с классическим алгоритмом прореживания [4]. Аналогично этому алгоритму алгоритм утолщения просматривает содержимое соседних элементов матрицы изображения, т. е. если  $A(I, J)$  — элемент матрицы изображения, то рассматриваются также восемь соседних элементов:

$$A(I-1, J-1), A(I-1, J), A(I-1, J+1), \dots, A(I+1, J+1).$$

Алгоритм работает следующим образом: если у элемента матрицы изображения не менее k его соседей содержат единицы, то в данный элемент также заносится единица. Коэффициент k задается в зависимости от требуемой толщины линий. Можно также применить этот алгоритм несколько раз для большего утолщения линий. Недостаток приведенного алгоритма — большая затрата времени для обработки больших матриц изображения.

в. При работе алгоритма Брезенхема в матрицу изображения заносится единицы как в очередной элемент матрицы  $A(I, J)$ , выбранный алгоритмом, так и в элементы

$$\begin{matrix} A(I-1, J-1), & A(I-1, J), & A(I-1, J+1), \\ A(I, J-1), & & A(I, J+1), \\ A(I+1, J-1), & A(I+1, J), & A(I+1, J+1). \end{matrix}$$

Таким образом, вместо обычной точки на бумаге печатается как бы более крупная точка, состоящая из девяти обычных. Этот алгоритм дает тот же результат, что и предыдущий при  $k=1$ , а затраты времени на обработку (для реальных изображений) существенно меньше.

г. Аналогично предыдущему алгоритму, в матрицу изображения заносится единицы как в  $A(I, J)$ , так и в

$$\begin{matrix} & A(I-1, J) & \\ A(I, J-1), & & A(I, J+1), \\ & A(I+1, J), & \end{matrix}$$

т. е. вместо обычной точки печатается более крупная, состоящая из пяти обычных.

Результат работы этого алгоритма близок к результату работы рассмотренного выше алгоритма (б.) при  $k=2$ , хотя в точности с ним не совпадает. Горизонтальные и вертикальные линии, полученные с помощью этого алгоритма, будут иметь ту же толщину, что и в предыдущем алгоритме, а диагональные линии несколько тоньше.

д. Подобно алгоритмам (в.) и (г.) можно, заносив в элемент матрицы  $A(I, J)$  единицу, ставить единицы и в

$$A(I+1, J), \quad A(I, J+1), \\ A(I+1, J+1).$$

т. е. ставить точку, состоящую из четырех обычных.

е. Можно также использовать следующий алгоритм: если угол наклона прямой содержится в промежутке от  $-45^\circ$  до  $+45^\circ$  (т. е. прямая больше наклонена к оси абсцисс), то заносить единицы в

$$A(I, J), \quad A(I+1, J),$$

а в противном случае, заносить единицы в

$$A(I, J), \quad A(I, J+1).$$

Горизонтальные и вертикальные прямые будут иметь ту же толщину, что и при работе алгоритма (д.), а диагональные прямые будут тоньше.

## Драйверы

На основе рассмотренных алгоритмов построены драйверы, связывающие графические системы Графор [5] и ГКС [6] с ТМПУ EPSON FX-800, EPSON FX-1000 и EPSON LQ-2500+. Программы Графора и ГКС написаны на Фортране, а программы драйверов — на языке Макроасемблера IBM PC.

Каждый драйвер состоит из двух частей. Одна часть служит для построения изображения в памяти и для вывода его на печать. Вторая — выполняет функции управления: задает режим работы ТМПУ и устанавливает текущую толщину линии.

В Графоре изменены некоторые программы, в частности, исключена программа PLOT, введены новые программы SETMOD и SETWID [7].

Программа SETMOD используется для установки одного из четырех режимов печати. В первом режиме изображение строится на растровой сетке с шагом 1/72 дюйма по горизонтали и вертикали (1 дюйм=25,4 мм). Второй режим устанавливает печать с шагом 1/120 дюйма по горизонтали и 1/72 дюйма по вертикали. В третьем режиме (режим повышенной жирности) используется алгоритм двойной печати, когда каждая точка печатается дважды на растровой сетке с горизонтальным шагом 1/144 дюйма и каждая строка печатается дважды со смещением 1/216 дюйма. В четвертом режиме изображение строится на растровой сетке с шагом 1/144 дюйма по горизонтали и 1/216 дюйма по вертикали. Для печати с высоким разрешением используется алгоритм, описанный в разделе «Печать с высоким разрешением». При обращении к программе SETMOD с параметром 0 режим вводится с клавиатуры.

Программа SETWID устанавливает толщину линии в текущем режиме. Обеспечивается рисование линий двух толщин — тонких (стандартный алгоритм Брезенхема) и толстых (последний алгоритм). Задание режима возможно только перед открытием страницы, а задание толщины линии — и внутри страницы. Программы, изменяющие толщину линий, необходимы и в ГКС, где есть программа установки толщины линий.

Пример использования функций Графора, иллюстрирующий работу драйвера (рис. 1):

```
REAL X(100),Y(100),Z(100)
X(1)=-1.9
Y(1)=-SIN(-1.9)
Z(1)=-COS(-1.9)*1.5
DO 5 I=2,90
X(I)=X(I-1)+1
Y(I)=SIN(X(I))
5 Z(I)=-COS(X(I))*1.5
CALL GRINIT
C Интерактивная установка режима печати: обращение
C к программе SETMOD с параметрами (0,0)
CALL SETMOD(0,0)
CALL PAGE(32,0,20,0,0,1)
CALL LIMITS(-2.7,-1.5,1.5)
CALL REGION(3,3,25,15,0,0,1)
CALL SYMBOL(10,18,2,1,'Пример',6,0)
CALL AXES('X',1,1,5,'Y',1,5,2,0)
CALL SYMBOL(25,17,1,'Пример',6,0)
C Установка толщины линии 2:
CALL SETWID(2)
CALL LINEO(X,Y,85)
C Установка толщины линии 1:
CALL SETWID(1)
CALL LINEO(X,Y,85)
CALL ENDPG(0)
END
```

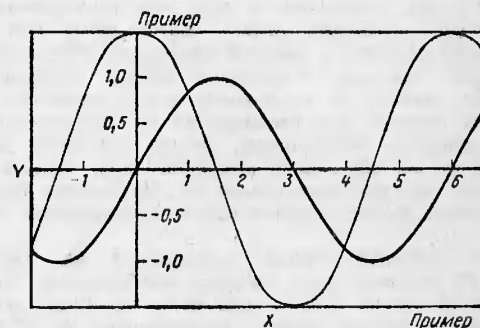


Рис. 1

Далее приведен фрагмент программы построения чертежа с использованием функций Графора (рис. 2):

```
C Вычислительная часть:
-----
C Построение:
C Инициализация Графора (обращение к подпрограмме GRINIT)
CALL GRINIT
C Обращение к подпрограмме SETMOD для интерактивной
C установки режима:
CALL SETMOD(0,0)
CALL MMS
C Открытие страницы (подпрограмма PAGE)
CALL PAGE(255,159,0,0,0)
C Установка толщины линии 2 (подпрограмма SETWID)
CALL SETWID(2)
C Рисование чертежа детали толстыми линиями:
-----
C Установка толщины линии 1 (подпрограмма SETWID)
CALL SETWID(1)
C Рисование разметок тонкими линиями:
-----
C Закрытие страницы:
CALL ENDPG(0)
END
```

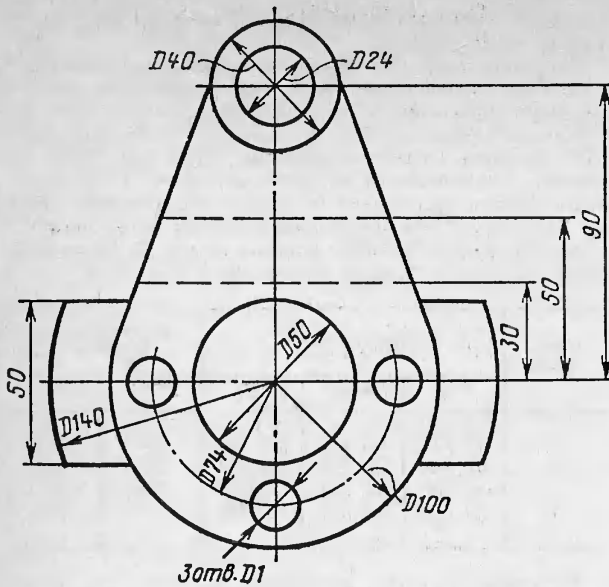


Рис. 2

Работу драйвера в четвертом режиме иллюстрирует рис. 3.

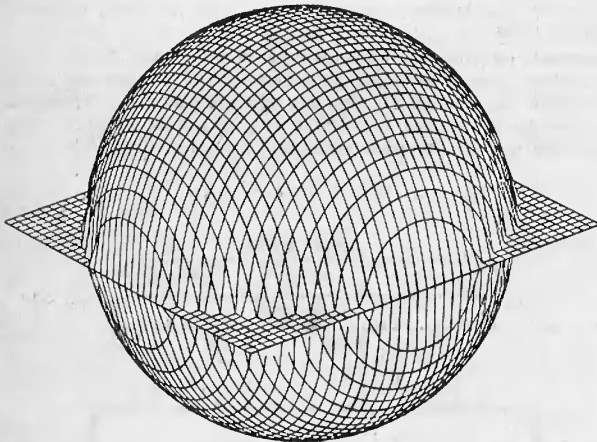


Рис. 3

### Заключение

Существуют также печатающие устройства, обладающие более широкими возможностями для вывода графической информации, чем рассмотренные здесь. Например, ТМПУ, оборудованные цветной красящей лентой, струйные и лазерные принтеры. Приведенные методы пригодны и для таких устройств, но возникает своя специфика. Например, при работе с цветным изображением необходимо отводить в памяти несколько битов на каждую точку. Соответственно изменяются алгоритмы занесения в матрицу изображения и распечатки. Лазерный принтер предоставляет широкий набор команд, в том числе и команду изменения размеров точки, делающую ненужными программные методы утолщения.

Рассмотрению вопросов вывода графической информации на эти устройства предполагается посвятить отдельную статью.

Телефон 333-71-89, Москва

1. Bresenham J. E. Algorithm for Computer Control of Digital Plotter // IBM Systems Journal.— 1965.— Vol. 4, N 1.
2. Bresenham J. E. Ambiguities in Incremental Line Rastering // Computer Graphics and Application.— 1987.— Vol. 7, N 5.
3. Bridger M., Goresky M. High Resolution Printer Graphics // BYTE.— 1985.— November.
4. Павлидис Т. Алгоритмы машинной графики и обработки изображений.— М.: Радио и связь.— 1986.
5. Баяковский Ю. М., Галактионов В. А., Михайлова Т. Н. Графор. Графическое расширение Фортрана.— М.: Наука.— 1985.
6. Computer Graphics. Special GKS issue.— 1984.— February.
7. Андреев С. В., Михайлова Т. Н., Мишакова С. Т., Рыжова И. Г., Ходулев А. Б. Комплекс Графор для персональных ЭВМ.— Препринт № 92 ИПМ АН СССР.— 1988.

Статья поступила 29.09.88

## СИСТЕМЫ ОТЛАДКИ

УДК 681.322

В. Е. Мельник

### ДВК ДЛЯ ОТЛАДКИ И КОНТРОЛЯ ЦИФРОВЫХ УСТРОЙСТВ

ДВК позволяют создавать достаточно универсальные программируемые стенды для отладки и контроля разнообразных по функциям цифровых устройств (ЦУ) как в производстве, так и, что важно, на этапе разработки. ТЭЗы современных ЦУ являются функционально насыщенными и для подключения требуют до 100...200 контактов, по которым осуществляется передача управляющих и информационных слов с рабочими частотами до 10 МГц. Выбор устройства сопряжения (УС) ДВК с ЦУ весьма неоднозначен и определяется целями, достижение которых должно быть обеспечено процессом отладки. Естественной является отладка и контроль ЦУ на рабочих частотах. Однако в этом случае УС должно быть весьма сложным программируемым блоком, способным после программной настройки в автономном режиме обрабатывать генерацию тестовых воздействий, принимать и хранить реакцию. В работе [1] рассмотрен блок логических устройств (БЛУ) «Электроника НЦ-803», который позволяет организовать режимы отладки и контроля ЦУ по 64 двунаправленным каналам с частотой выдачи тестовых наборов и фиксации реакции до 10 МГц.

Разработка программ режимов отладки и контроля ЦУ с помощью БЛУ является весьма трудоемким процессом. Созданное для БЛУ программное обеспечение в виде пакета DISLA [1], ориентированное на отладку устройств с системой команд ЭВМ «Электроника 60», имеет следующие недостатки:

- поддержка одновременной работы только 32 каналов; весьма заметная инерционность диалога с оператором; сложность системы «меню»;
- большой объем резидентной части программы, что делает ее несовместимой с версией РТ11-5.0 для дискет Ø 133 мм;
- большое число оверлейных сегментов, что снижает надежность работы программы;
- отсутствие поддержки программирования генератора сигналов и др.

Указанные недостатки в значительной степени затрудняют использование пакета DISLA для отладки произвольных ЦУ и ставят разработчика аппаратуры перед необходимостью создания своих программ отладки и контроля.

Учитывая сложность БЛУ даже при хорошем знании архитектуры последнего для разработки программы со средними функциональными возможностями (объем порядка 10—20 тыс. команд) программисту средней квалификации потребуется не менее 1—1,5 лет, что во многих практических случаях является неприемлемым.

Для разработчиков, имеющих в своем распоряжении ДВК и не являющихся профессиональными программистами, компромиссный выход заключается в создании более простых УС и соответствующей программной поддержки. В работе [2] описана система отладки, в которой УС подключено к ИРПП ДВК. Хотя в этом случае существенно упрощается подключение УС, но значительно уменьшается (до сотен герц) частота выдачи тестовых воздействий и прием реакции ЦУ.

С точки зрения упрощения программного обмена с УС, а также повышения частоты выдачи тестовых воздействий на ЦУ (до десятков килогерц) целесообразно размещать регистры связи УС с ЦУ в адресном пространстве 160 000... 177 600 (странице регистров внешних устройств). К наиболее простым вариантам относится использование стандартного устройства И2, входящего в состав микроЭВМ «Электроника 60». Такой же эффект даст и использование контроллера параллельного обмена (КПО), в котором два 16-разрядных канала сопряжения с ВУ в отличие от И2 выполнены не на плате, а на полуплате конструктива «Электроника 60» (рис. 1).

Основу канала КПО составляют одна БИС К1801ВП1-033 (АС, PгC, БУ) и две БИС К1801ВП1-034 (PгВых, MпVx). Адреса регистров и векторов прерывания (ВП) канала задаются распойкой логических уровней на управляющие входы БИС К1801ВП1-033 согласно [3, стр. 156]. Указанная БИС позволяет выбрать четыре различных набора адресов. Таким образом, без введения дополнительных схем селек-

ции адреса можно организовать УС на двух КПО, содержащих четыре канала.

Для сопряжения с ВУ каждый из каналов имеет выходную и входную 16-разрядные шины, а также шину управления. По шине управления к ВУ передаются сигналы Выв, Вв, Сбр, соответствующие магистральным сигналам ДЗП, ДЧТ, СБР. Внешнее устройство сигналом ТПРА или ТПРВ при наличии разрешения в регистре состояния (РС) может инициировать прерывание от канала по соответствующему ВП. Управление процедурами канала осуществляется через РС, формат которого показан на рис. 2. Назначение битов РС стандартное, за исключением РС1, РС0.

Номер канала	Адреса регистров			Адреса векторов прерывания	
	PгC	PгВых	PгVx	ТПРА	ТПРВ
1	167 770	167 772	167 774	300	304
2	167 760	167 762	167 764	310	314
3	167 750	167 752	167 754	320	324
4	167 740	167 742	167 744	330	334

Указанные разряды, доступные по записи и чтению, могут использоваться как дополнительные информационные или управляющие сигналы для ВУ. УС на двух КПО представляет собой четыре идентичных канала, занимающих на МПИ ДВК адреса, указанные в таблице. Оно может выдавать сигналы на отлаживаемое устройство по 64 (без учета РС0, РС1) и принимать от него реакцию также по 64 линиям. Такого числа линий сопряжения во многих случаях оказывается недостаточно для полного «охвата» отлаживаемого устройства необходимыми входными и выходными сигналами. Но упомянутый недостаток не является слишком серьезным для инженера-разработчика на этапе отладки. На основе УС создано три отладочных комплекса, с помощью которых в течение года была проведена автоматная и комплексная отладка разнообразных ЦУ: ком-

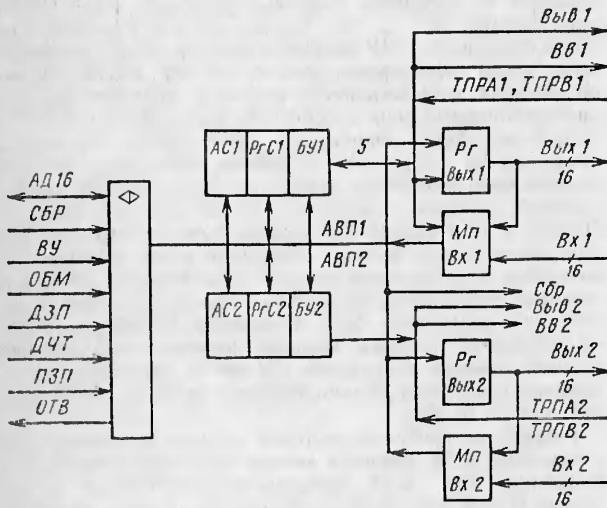


Рис. 1. Структурная схема КПО:

АС — адресный селектор, PгC — регистр состояния, БУ — блок управления, PгВых — регистр выходов, MпVx — мультиплексор входов, АВП1 — адрес вектора прерывания

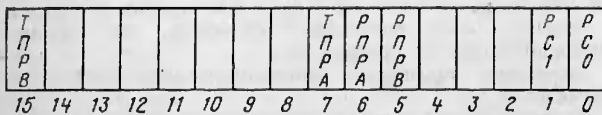


Рис. 2. Формат регистра состояния канала КПО:

РС0, РС1 — вспомогательные разряды РС; РПРА, РПРВ — разрешения прерываний А, В (доступ по записи и чтению); ТПРА, ТПРВ — разрешения прерываний А, В (доступ по чтению)

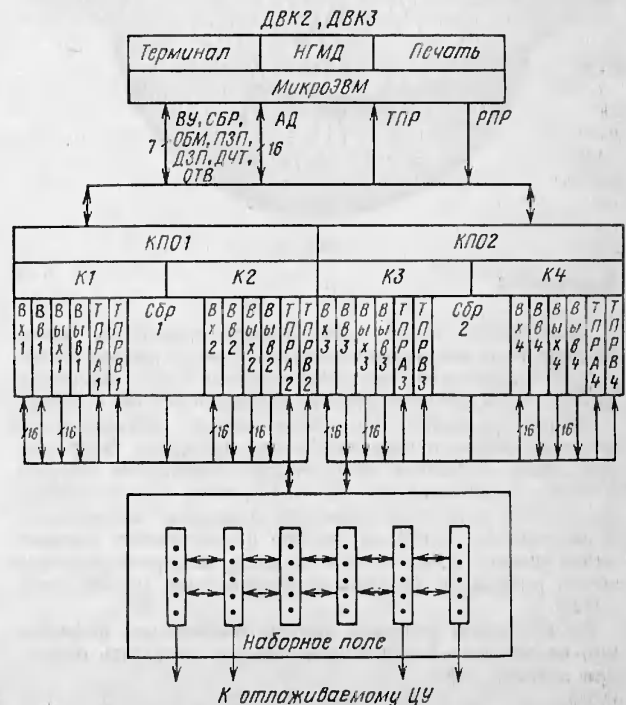


Рис. 3. Структурная схема стенда отладки

плексных и специализированных АУ с микропрограммным управлением, устройств двухпортовой двухблочной памяти, устройств управления аппаратного и микропрограммного типов, интерфейсных устройств. Каждое из ЦУ выполнено в основном на ИС большой и средней интеграции общим числом свыше сотни корпусов и содержит четыре 60-контактных разъема.

Практика отладки показала, что при подключении отлаживаемого устройства к УС через наборное поле разработчик практически всегда может вести отладку устройства по схемным фрагментам или режимам, обеспечивая передачу через УС необходимых динамических сигналов, подавая, если нужно, на контакты наборника статические комбинации необходимых сигналов. Упрощенная структурная схема программируемого стенда с использованием рассмотренного УС показана на рис. 3.

Принцип программируемости универсального стенда можно использовать, реализуя два основных методически различных подхода. Первый подход заключается в составлении для каждого ЦУ своей индивидуальной программы. Создание такой программы, а точнее, набора программ по числу отлаживаемых фрагментов ЦУ приемлемо, если разработчик свободно владеет одним из языков программирования или имеется достаточное число программистов. В большинстве случаев более целесообразно иметь унифицированную программу, которая с помощью сравнительно простого интерфейса с разработчиком настраивается на отладку конкретного ЦУ. Интерфейс может быть организован по типу подсказки или «меню» (рис. 4).

Программная логика модулей 3...7.9 инвариантна к структуре отлаживаемых ЦУ и работает с двумя входными (массивы входных данных и управляющих сигналов) и одним выходным (массив выходных данных или реакции) внутренними буферами.

Информация, необходимая для поддержки процесса отладки конкретного ЦУ, вводится в буферы с файлов на диске модулями 1 и 8, а выводится — модулями 3 и 10. Привязка буферов к адресам каналов и, следовательно, к структуре ЦУ производится модулем 2. Указанные модули в диалоговом режиме запрашивают имена входных и выходных файлов, адреса выдачи данных, управляющих сигналов, приема реакции.

Стоит уточнить функции модуля 3. Первоначально на диске в общем случае нет необходимых входных массивов для конкретного ЦУ. В режиме диалога они формируются полностью или по частям модулем 3 и под заданным именем записываются на диск. Ввод массива ведется в виде, задаваемом оператором. Ясно, что входной массив данных удобно задавать в виде двоичных или восьмеричных чисел, а массив управляющих сигналов целесообразно задавать в виде временной диаграммы сигнала на определенном временном отрезке. При наличии откорректированных массивов диалог с программой можно упростить и исключить лишние обращения к диску. Для этого достаточно программе с введенными буферами, заданными именами и адресами каналов сохранить с помощью мониторинговой команды SAVE и использовать на этапе контроля в производстве. Рабочий вариант программы с буферами по 512 слов занимает в памяти около 8К слов и после загрузки, если не введется коррекция входного или управляющего массива, не требует обращения к диску.

Следует отметить, что программа поддерживает процесс отладки, если используется любое другое УС, в котором не менее четырех регистров: два выходных, один входной и регистр состояния.

Таким образом, применение ДВК в сочетании с простыми регистровыми УС позволяет оперативно и эффективно производить отладку ЦУ.

Использование унифицированных программ значительно снижает трудозатраты на разработку программной поддержки процесса отладки большого количества типов ЦУ. Ориентированные на неподготовленного разработчика такие программы позволяют поставить процесс отладки ЦУ в течение одного рабочего дня.

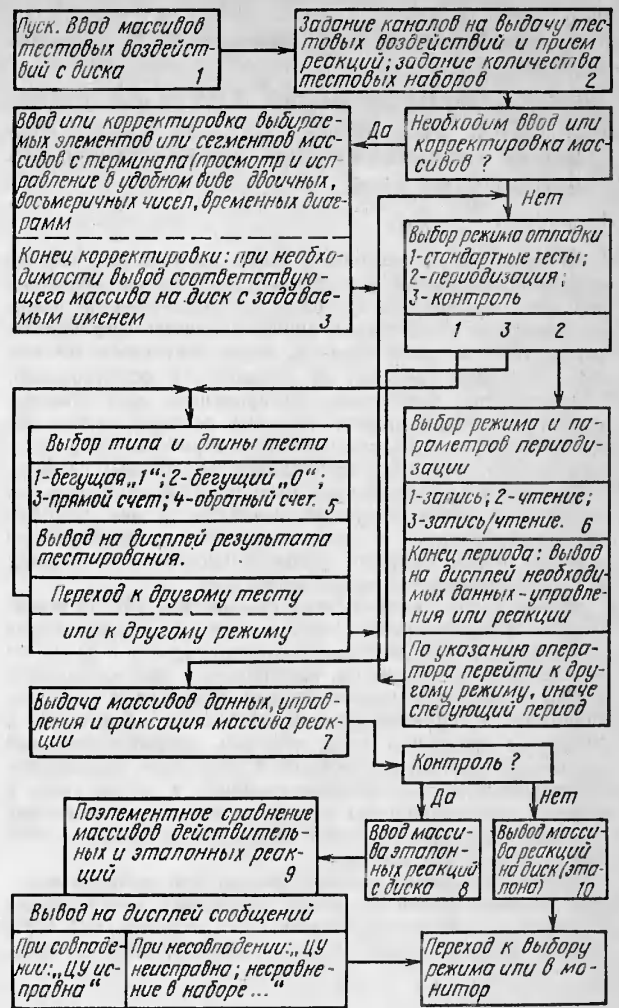


Рис. 4. Структура унифицированной программы отладки ЦУ

Недостатком УС на основе программно доступных регистров является сравнительно низкая частота выдачи тестовых воздействий и записи реакции ЦУ — 20...40 кГц в зависимости от типа ДВК.

Телефон 274-36-97, Ленинград

#### ЛИТЕРАТУРА

1. Средства отладки. лабораторный практикум и задачник. / Под ред. чл.-корр. АН СССР Л. Н. Преснухина. М.: Высшая школа, 1986.
2. Безобразов В. С., Ларин Б. И., Сохранин В. Ю. Отладочный комплекс для 32-разрядной микроЭВМ на основе ДВК2 // Микропроцессорные средства и системы. — 1986. — № 3. — С. 49—50.
3. Микропроцессоры. Справочное пособие для разработчиков судовой РЭА // Под ред. Ю. А. Овечкина. Ленинград: Судостроение, 1987.

Статья поступила 16.03.88

О. Ю. Серебренников, Е. Г. Шевченко, Е. В. Леонидов-Каневский

### АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДИАГНОСТИКИ ТЕХНИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Из известных в настоящее время методов диагностики выбран программный способ самотестирования, предполагающий использование программных средств-тестов, функционирующих в собственно диагностируемых устройствах. Тесты строятся таким образом, чтобы результаты тех или иных операций зависели от исправности оборудования.

Программное управление тестированием дает возможность генерировать пакеты тестовых последовательностей достаточно большой длины, логически обрабатывать результаты работы тестов без специальных аппаратных средств.

Определенная программная доступность функциональных частей МП-систем позволяет выдавать на них тестовые последовательности и анализировать соответствующую реакцию. Метод обладает универсальностью и не зависит от сферы применения микропроцессоров.

Автоматизированная система диагностики (АСД) технического и программного обеспечения микропроцессоров КТС ЛИУС-2 позволяет контролировать запуск и функционирование 32-х МП-систем, тестировать и диагностировать отказы в режимах предстартового контроля и рабочем, принимать и протоколировать сообщения об отказах с указанием времени и даты, подавать предупредительный сигнал при поступлении сообщений об отказах, наращивать программные модули самотестирования в соответствии с набором функциональных элементов, получать справочные данные о конфигурации любой локальной системы, включенной в систему диагностики.

Основой созданной системы диагностики является комплект базовых тестов (комплекс программ), рассчитанных на проверку основных вычислительных ресурсов (ЦП, ОЗУ, ПЗУ) и средств отсчета времени (таймер). Тестовые про-

граммы вместе с программами обмена образуют локальный программный модуль самотестирования (ЛПМС).

В процессе самотестирования локальной системы проверяются выполнение микропроцессором всех команд и сравнение результатов с заранее известными: работа ОЗУ методами «бегущий ноль», «бегущая единица», «падающий дождь», запись (чтение) случайной информации; контрольная сумма кодов, записанных в ПЗУ.

Система диагностики структурно состоит из трех частей: ЛПМС, концентратора, центральной части (рис. 1). Концентратор и центральная часть связаны между собой двухпортовым ОЗУ, равнодоступным обеим частям системы.

Сообщение, подготовленное в ЛПМС, посылается через элементы КС 52.25, частично дешифрируется в концентраторе, после чего попадает в доступное центральной части двухпортовое ОЗУ, где организован буфер сообщений. В центральной части производится полная идентификация и диагностика сообщений с помощью базы данных, готовится буфер на печать и видеополосы на дисплей. После запуска любой локальной микропроцессорной системы первым всегда начинает работать ЛПМС, который после предстартового контроля осуществляет передачу результатов тестирования в концентратор, причем инициатива в этом случае принадлежит ЛПМС. Концентратор, приняв посылку о предстартовом контроле, заносит данную систему в аппаратную таблицу оборудования, по которой в дальнейшем опрашивает данную систему в рабочем режиме. Основные характеристики АСД приведены ниже.

Разрядность процессоров АСД	8
Разрядность процессоров локальных систем	8
Тип интерфейса связи локальных систем с АСД	ИРПС токовая петля
Протокол обмена	Для сетей X.25
Максимальное число локальных систем	32
Объем памяти АСД, Кбайт,	
ОЗУ	8
ПЗУ	24
двухпортового ОЗУ	4
базы данных	8
Максимальная интенсивность поступления, сообщений / с.	32

Посылка ответа в систему диагностики после запроса концентратора подпрограммы НОТДОГ (рис. 2) свидетельствует об исправности линии связи в рабочем режиме

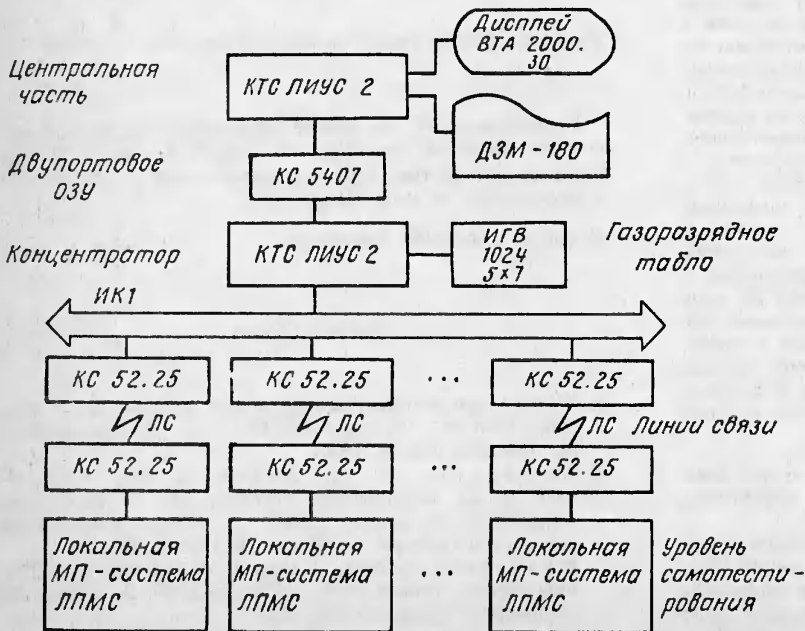


Рис. 1. Структурная схема АСД

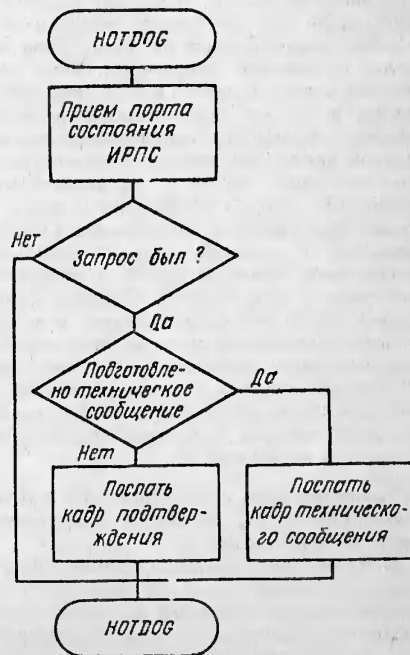


Рис. 2. Структура подпрограммы анализа на зависание



локальной системы. Отсутствие ответа трактуется как зависание или останов локальной системы. При отсутствии ответа формируется соответствующее сообщение на дисплей и печатающее устройство. Подпрограмма запускается не реже одного раза в секунду или в цикл.

Все сообщения в системе передаются в виде 32-х байтовых сообщений-кадров. Концентратор и ЛПМС в штатном режиме обмениваются кадрами, каждый из которых может содержать до 10 кодов отказов одновременно или символическое сообщение до 20 символов, несущих дополнительную информационную нагрузку по усмотрению разработчиков локальных систем. Это могут быть технические сообщения, сообщения об отклонении параметра технологического процесса, рекомендации эксплуатационному персоналу. Элемент связи ИРПС КС 52.25 полностью буферизирует 32 байта, составляющих кадр сообщения. Поэтому потеря времени при приеме-передаче не происходит.

Основу АСД составляет база данных, описывающая оборудование локальных микропроцессорных систем. Если принимается кадр сообщения с признаками отказа или неисправности, то производится диагностирование и локализация отказа. На экране дисплея высвечиваются после-

довательно план машинного зала с указанием шкафа, изображение шкафа с указанием этажа и каркаса, на котором установлена данная МП-система, изображение каркаса с элементами, неисправный элемент. Таким образом эксплуатационный персонал получает оперативную информацию о возникающих отказах, достаточную для принятия конкретных решений по их устранению.

Для функционирования системы в период профилактики на внешних устройствах (печать, дисплей) предусмотрено цифро-буквенное газоразрядное табло, на которое выводится вся информация, приходящая от локальных систем. Сообщения полностью не расшифровываются, однако по ним можно судить о работоспособности и отказах диагностируемых систем.

Описанная автоматизированная система внедрена в промышленную эксплуатацию на стане 2000 горячей прокатки Новолипецкого металлургического комбината.

Телефон 77-04-90, 77-69-51 н. Липецк

Статья поступила 12.05.88

## ОБЩИЕ ВОПРОСЫ РАЗВИТИЯ ИНФОРМАТИКИ

УДК 681.3

Л. М. Гейман

### ЭТАПЫ РАЗВИТИЯ ИНФОРМАТИКИ КАК СИСТЕМЫ ЗНАНИЙ

Потребность записи информации в виде символов и их дешифровки возникла на самых ранних этапах становления человеческого общества, благодаря чему был впервые достигнут эффект обезличенности процесса передачи информации и реализована функция надежности ее хранения. Ранняя история информатики — это история информационной символики, появления новых средств, развития коммуникаций на общем фоне непрерывного накопления в обществе массива информации событийного и научного характера. В историческом плане информатика развивалась изначально по двум тесно связанным направлениям: язык системы представления знаний и система информационных коммуникаций.

Символика для представления знаний на лингвистической основе проделала путь от общедоступных форм через многообразие национальных фонетически разнородных письменностей до современных формализованных языков высокого уровня для общения с ЭВМ (макрокоманды); символика на математической и химической основах всегда носила абстрактный, общедоступный (межнациональный) характер.

Вначале носителем человеческих знаний была речь, выражающая собой кодированные сигналы для координации выполняемых трудовых процессов. Первая в истории информационная символика была представлена в каменном веке образами, которые на камне отражали в виде рисунка отдельное представление — так называемое пиктографическое письмо. В бронзовом веке появляются изображения повторяющихся систем понятий — идеограммы (идеографическое письмо). Эти примитивные информационные единицы превращаются с конца 4-го тыс. до н. э. в рисуночное иероглифическое письмо (рис. 1, 1). Усложнение информационной символики в виде скорописи сделало ее доступной исключительно узкому кругу социально обособленных (в условиях складывающейся государственности) людей.

В это же время благодаря развитию средств производства и, позднее, торговли совершенствуется числовая символика, которая возникла в виде счета из двух чисел — один и два (количество предметов более двух обозначалось как «много») одновременно с человеческой речью. Дальнейшее развитие система счета получила благодаря пальцам на руках: появился счет до пяти и соединение двух «пятерок» в десятку. Клинописная запись счета приме-

нялась в Вавилоне в 3-м тыс. до н. э. (рис. 2, а). Так была изобретена позиционная система счисления, когда от положения (позиции) знака меняется его смысл. Эта система шестидесятиричная, и следы ее сохранились до

The diagram illustrates the evolution of writing systems through five stages:

- 1:** Pictographs: A fish (рыба), a hand (рука), and a star (звезда).
- 2:** Pictographs with phonetic values: fish (рыба) and hand (рука).
- 3:** Cuneiform script: A grid of symbols representing the fish, hand, and star.
- 4:** Cuneiform script with phonetic values: A grid of symbols representing the fish, hand, and star with their respective phonetic values.
- 5:** A grid of symbols representing the fish, hand, and star with their respective phonetic values in various scripts.

1	2	3	4	5

Рис. 1. Развитие систем письменности:

1 — иероглифический текст на печати «Тархумвас — царь страны Мера»; 2 — превращение рисуночных знаков в клинописные; 3, 4, 5 — родственные линейные письменности Средиземноморья; 6 — финикийское письмо, X...IX вв. до н. э.; 7 — ранне-греческое письмо, VII в. до н. э.; 8 — классическое греческое письмо, V...IV вв. до н. э.; 9 — западно-греческое письмо, до V в. до н. э.; 10 — латинское письмо, IV в. до н. э.; 11 — латинское письмо, IV в. н. э.

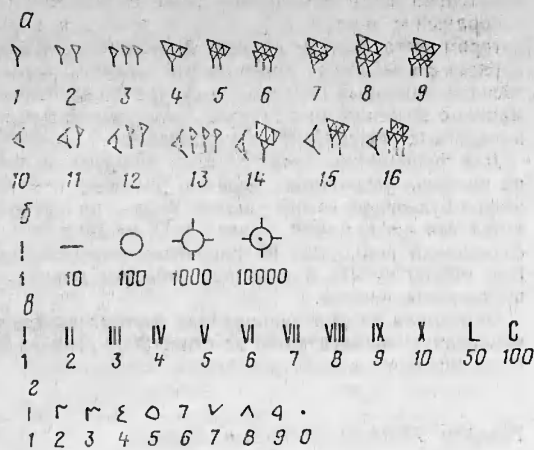


Рис. 2. Развитие системы записи счета:

а — вавилонский; б — критский; в — латинский; г — арабский

сих пор (час делится на 60 мин, минута — на 60 с). Вавилонская система счета позволяла вести запись чисел до миллиона и больше и выполнять действия с простейшими дробными числами. В V...IV вв. до н. э. в Древней Греции появились значки для обозначения произвольных величин. В этот период на Крите применяется удобная для записи десятичная символика счета (рис. 2,б).

Древние римляне положили (предположительно) в основу знаков счисления иероглифы, обозначающие пальцы рук. Ко времени расцвета римской культуры эти значки были заменены похожими на них латинскими буквами (рис. 2,в). Восприняв от индусов искусство счета, арабы заметили у них и значки для записи чисел — цифры (рис. 2,г), которые в VII...VIII вв. распространились на европейском континенте. Эти значки использовались для записи чисел в том же порядке, как принято сейчас.

В Древнем мире для обозначения веществ, химических операций и приборов применяли символические изображения, буквенные сокращения, а также их сочетания. Совершенствование этих символов началось в XV...XVIII вв. (рис. 3) и продолжалось вплоть до начала XIX в., когда шведский химик Й. Берцелиус (1814 г.) предложил химические знаки современного вида.

Иероглифическое письмо сохранилось как усовершенствованный реликт в ряде регионов (например, в Китае с 2-го тыс. до н. э.). Этому явлению, наряду с малой необходимостью в передаче грамматических показателей, а также удобством для общения между носителями фонетически различающихся диалектов, способствовали этническая однородность и многовековой абсолютизм государственного строя, исповедовавший консервативные формы не только в общественных отношениях, но и в письменности как привилегии избранных. Качественно иной характер носило развитие системы письма в средиземноморском регионе, где имелись предпосылки для его совершенствования: многообразие языковых форм, широкие межрегиональные торговые контакты, относительная нестабильность политической обстановки в государствах, возникновение прогрессивных государственных образований, смещение различных по национальному признаку культур, миграция населения. Это позволило в короткое историческое время завершить информационный переход от образной рисуночной иероглифической системы письма к абстрактной и более удобной для чтения системе клинописи (рис. 1,2) на сырых глиняных табличках (3...2-го тыс. до н. э.).

Следующим заметным этапом стало создание линейного словесного письма на глиняных табличках (рис. 1,3...5). В этот период аккадский (вавилонский) язык впервые в истории начинает выполнять международные функции в

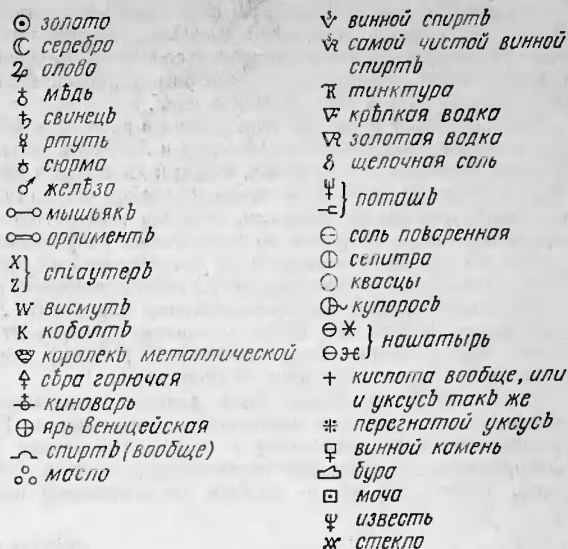


Рис. 3. Некоторые химические знаки XVIII в. (из кн.: Еркслейб И. Х. П. Начальные основания химии / С немецкого на русский язык перевел Н. Соколов: СПб, 1788. — С. 16—17). Сурьма — сурьмяный блеск  $Sb_2S_3$ . Орпимент — аурипигмент  $As_2S_3$ . Мышьяк — трехокись мышьяка  $As_2O_3$ . Спиаутер — цинк. Ярь Венецѣйская — основной ацетат меди. Знак меди надо перевернуть (опечатка в подлиннике)

дипломатии и торговле и, выводя информацию из узконациональных рамок, придавать ей широкие коммуникационные и терминоподобные формы.

Новым этапом явилось создание в X...IX вв. до н. э. финикийского алфавита. Революционный по сути и многоэтапный по времени переход к алфавитным системам завершается в VIII в. до н. э. созданием на основе финикийского письма греческого алфавита — основы всех западных письменных систем (рис. 1,6...11). Удобство этой информационной символики способствовало распространению письменности в древнегреческом мире. Усовершенствованной алфавитной письменности послужило введение во II...I вв. до н. э. в Александрии начал пунктуации. Развитие письменной символики завершается в Европе в XV в. созданием пунктуации современного вида. Совершенствованию языка способствовало появление древнегреческих терминов научного характера на основе разговорной речи, благодаря чему началось устранение информационной избыточности. В период Возрождения древнегреческий и латинский языки послужили корневой основой для создания национальных по форме и специальных по содержанию терминологических систем в различных областях знаний. В период технической революции (конец XIX в.) терминологические системы значительно расширяются в объеме, упорядочиваются за счет фундаментального переосмысления законов природы и общества, происходит диффузия терминов между областями знаний.

Математическая символика продолжает качественно развиваться благодаря углубленным исследованиям и фундаментальным открытиям в математике: создается совершенная алгебраическая символика (XIV...XVII вв.), вводятся знаки сложения, вычитания и умножения (вначале буквенные, конец XV в.), знаки равенства, бесконечности, скобки, дроби, корни, логарифмы (XVI...сер. XVII вв.); в конце XVII в. появляются знаки степени, дифференциала, интеграла, производных, в первой половине XVIII в. — знак переменных операций — функция (1718 г.), знаки синуса, косинуса, тангенса (1748...1753 гг.).

Особой формой представления знаний являются карты, которые отображают пространственное размещение объек-

## НЕСТАНДАРТНЫЕ ВНЕШНИЕ УСТРОЙСТВА ДЛЯ ПЭВМ «ЭЛЕКТРОНИКА БК-0010»

**Устройство формирования цветной палитры (УФЦП) для ПЭВМ «Электроника БК-0010» (далее БК-0010) предназначено для:**

получения различных цветов и их оттенков на экране цветного ТВ при работе с БК-0010 без специального ПО;

работы БК-0010 с цветным ТВ в монохромном режиме. Это позволяет программно (без каких-либо механических переключений, перестыковок кабелей и т. д.) выводить на экран цветные монохромные изображения с градациями яркости и символы (64 символа в строке);

установки каждой строке экрана дисплея своей палитры цветов;

выполнения функций таймера с частотой  $\sim 1,6$  кГц.

УФЦП преобразует стандартный RGB сигнал с ТВ-выхода ПЭВМ. При этом в ручном и программном режимах управления задаются номера палитры.

Пульт-манипулятор для ПЭВМ «Электроника БК-0010» позволяет взаимодействовать с программой на языках ФОКАЛ, БЕЙСИК, в машинных кодах и т. д. без клавиатуры (либо вместе с клавиатурой) через выносное устройство и не требует дополнительного системного ПО.

Пульт-манипулятор состоит из выносного пульта управления (с двумя каналами плавной регулировки) и блока адаптера, подключенно-



Пульт манипулятора

ного к регистру 177714 порта управления внешними устройствами БК-0010.

Каждый из двух каналов плавной регулировки ручками управления допускает установку 32 состояний (5 бит на каждый канал).

Тел. 155-09-19, Москва



Устройство формирования цветной палитры

(Окончание. Начало см. на 1 с. вкладки)

на материале «Polaroid» 691» аппаратом «Vivitar». Получение одной копии на бумаге требует 1,5—2 мин.

Кооператив «Информ» продает указатель графической информации и ПО для микроЭВМ «Электроника БК-0010».



Тел. 497-86-52 (Майоров Леонид Николаевич с 19 до 21 ч.), Москва

# УЧЕБНЫЙ МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКТ — УМК

Разработан Рижским производственным объединением ВЭФ для обучения основам программирования и конструирования микро-ЭВМ. УМК необходим преподавателям и учащимся профессионально-технических училищ, техникумов, вузов, кружков технического творчества,

станций юных техников

УМК имеет системную программу «Монитор», которая обеспечивает:

- чтение и модификацию содержимого памяти и регистров микропроцессора;
- выполнение программ пользователя;
- вычисление контрольной суммы массива памяти;

заполнение массива памяти константой;

перемещение заданного массива памяти в адресном пространстве.

За дополнительную плату УМК комплектуется набором плат, существенно расширяющих функциональные возможности:

**М2** — макетирование схем сопряжения с устройствами, имеющими параллельный интерфейс (предусмотрена розетка для БИС ППЗУ К573РФ2);

**ППИ** — освоение сопряжения с устройствами, имеющими последовательный синхронный или асинхронный интерфейсы;

**ПГМ** — программирование БИС ППЗУ типа К573РФ2. Источник питания, обеспечивающий напряжение +24 В;

**АЦА** — имеет два АЦП и два ЦАП с длиной слова 10 бит;



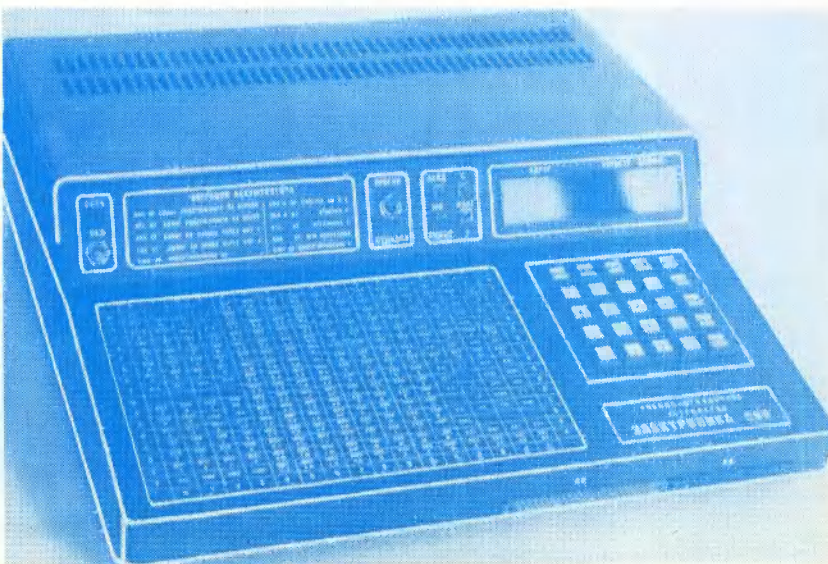
**ПС** — изучение основ вычислительной техники от элементарных логических операций до модели простейших ЭВМ;

**КОП** — сопряжение УМК с КОП и позволяет создать локальную сеть на базе УМК.

Набор сменных плат позволит значительно расширить область применения УМК, создав на его базе программатор, музыкальную приставку, электронный вольтметр и другие устройства.

**УЧЕБНЫЙ МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКТ** — это современный метод подготовки к завтрашнему дню информатики и программирования.

Позаботьтесь о своевременности обучения!  
**ЦЕНА УЧЕБНОГО МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА — 700 руб.**



## УЧЕБНО- ОТЛАДЧНОЕ УСТРОЙСТВО

Во втором номере журнала за 1985 г. была опубликована статья «Учебно-отладочное устройство «Электроника 580». В настоящее время завод-изготовитель имеет возможность поставлять это устройство без фондов по прямым связям.

Если Вас интересуют самые простые, но эффективные средства отладки программ для МПК К580, направляйте ваши запросы по адресу:

424003, Йошкар-Ола, ЗПП, отдел сбыта.

## ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР ПК-85

Программируемый контроллер ПК-85 предназначен для управления дискретными и дискретно-непрерывными технологическими процессами, а также для сбора, обработки, отображения (в текстовом и графическом виде) аналоговой и дискретной информации.

Контроллер имеет свободно компоновемую структуру: собранный на одной панели размером  $480 \times 280 \times 400$  мм — содержит 192 дискретных входа-выхода; в многопроцессорной конфигурации — до восьми микроЭВМ и 4096 дискретных входов-выходов.

Функциональные возможности ПК-85 обеспечиваются за счет развитого ПО при минимизации аппаратных средств. Минимальная конфигурация ПК состоит из одноплатной микроЭВМ и малогабаритного пульта ввода и индикации, с помощью которых осуществляется диагностика остальной части контроллера.

Одноплатная микроЭВМ разработана с применением БИС К1801ВМ1А, КР1801ВР1, К573РФ4А, КР573РУ10. Объем РПЗУ — 16 Кслов, объем ОЗУ — 8 Кслов. Все модули ввода-вывода — с гальванической развязкой.

Для программирования ПК-85 предлагается несколько языков:

упрощенный автоматически-ориентированный язык ПАТ (Параллельный Ассемблер Технологий) интерпретирующего типа. Ввод и отладка



Модули

программы на этом языке ведется с малогабаритного пульта ввода и индикации. ПАТ содержит инструкции управления дискретными процессами, арифметические и логические, позиционирования и ряд других. Программное обеспечение для поддержки программирования на этом языке подготовлено на языке Си и «зашиито» в 16 Кслов РПЗУ;

язык Си (имеется пакет поддержки программирования на этом языке);

язык РАЯ (Расширяемый Адаптируемый Язык) разработан лабораторией ЭВМ МГУ.

Серийное производство контроллеров начато с 1989 г.

278000, Молдавская ССР, Тирасполь, ул. Сакриера, 2-В, СКБТЛ, тел. 9-35-62, 9-34-88

## УКАЗАТЕЛЬ ИНФОРМАЦИИ И ИНТЕРФЕЙС ЦВЕТНОГО ТЕЛЕВИЗОРА ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА БК-0010»

(К ст. Казанцева)

Бытовой телевизор, применяемый в качестве видеомонитора с данным интерфейсом, позволяет генерировать компьютером цветные изображения хорошего качества. Авторами апробирован экспресс-метод, использующий моментальные диапозитивные 35-мм фотопленки «Polachrome» (фирма Polaroid), экспонируемые любым отечественным фотоаппаратом. С помощью аппарата «Autoprocessor» можно получить цветной диапозитив через 1,5—2 мин. При необходимости он увеличивается до размеров  $8,3 \times 10,8$  см и тиражируется в виде цветных фотографий на материале «Polacolor 669» или слайда того же размера

(Продолжение см. на 3 с. вкладки)



Одноплатная микроЭВМ

## МУЛЬТИМЕТР С АВТОМАТИЧЕСКИМ ВЫБОРОМ ПРЕДЕЛОВ МРЦ-4

Компактные, с автономным питанием цифровые приборы для измерения напряжения, тока и сопротивления — мультиметры, пользуются особой популярностью у радиолюбителей и специалистов по ремонту аппаратуры и поэтому являются перспективным товаром народного потребления.

В последнее время отечественная промышленность наладила выпуск мультиметров типа В7-41, В7—47, а мультиметр «Электроника ММЦ-01» уже продается в фирменных салонах «Электроника» (96 руб.). Имея хорошие точностные параметры, эти приборы все же отстают от технического уровня, достигнутого ведущими западными фирмами, так как переключение пределов измерения выполняется вручную (механическим переключателем). Это уменьшает удобство в работе и снижает надежность прибора.

Мультиметр МРЦ-4 по сервисным показателям значительно превосходит отечественные аналоги и приближается к лучшим мировым образцам. Он имеет автоматический выбор пределов измерения и указатель полярности. Это делает процесс измерения более удобным, быстрым и точным.

Прибор МРЦ-4 выполнен на базе мультиметра МРЦ-3 с электронным переключением пределов (см. «МП».—1988.—№ 6.—С. 3 обл.) и отличается от него элементной базой — использован аналого-цифровой преобразователь типа КР572ПВ5 и микросхемы широкого применения серий К561 и КР140УД14.

Мультиметр МРЦ-4 значительно легче и меньше зарубежных аналогов. Резисторов в нем — 37 (в «Электронике ММЦ-01» — 48); конденсаторов — 17 (23); транзисторов — 3 (7); микросхем — 10 (7). В МРЦ-4 всего два переключателя на два положения, отсутствие тумблера включения питания (оно включается при вставлении измерительных проводников в гнезда) исключает разрядку батареи питания после измерений.

Мультиметр имеет индикатор разряда батареи питания и электронную защиту по всем входам. Жидкокристаллический индикатор мультиметра обеспечивает 3 1/2-разрядную десятичную индикацию результата измерения с отображением десятичной запятой.

Максимальное показание на индикаторе 1999. При превышении измеряемой величиной этого значения переход на более старший предел измерения происходит автоматически.



### Технические характеристики МРЦ-4

Пределы измерения напряжения, В; сопротивления, кОм . . . . .	2, 20, 200, 2 000
тока, мА . . . . .	2, 20, 200
Основная погрешность при измерении постоянного напряжения, зн.	
постоянного . . . . .	± (1+2)
переменного . . . . .	± (2+3)
сопротивлений, % . . . . .	± (2+2)
Диапазон частот измерений:	
переменного напряжения, кГц	
в пределах 0...2 В . . . . .	0 ... 20
в остальных пределах . . . . .	0 ... 40
Входное сопротивление, МОм . . . . .	1
Потребляемый ток, мА . . . . .	5
Габаритные размеры, мм . . . . .	133×75×22
Масса, г . . . . .	185

Энергии одной батареи «Корунд» хватает на 100 ч непрерывной работы. Минимальное напряжение для работы МРЦ-4 — 6,5 В.

Мультиметр собран на микросхемах КР572ПВ5Б (1 шт.); КР140УД14 (3 шт.); К561ЛП2 (2 шт.); К561ЛА7 (1 шт.); К561ИЕ11 (1 шт.); Индикатор — типа ИЖЦ-5—4/8, переключатели — ПД2—2П4Н (2 шт.).

На мультиметр МРЦ-4 подготовлена документация: схема принципиальная (1 лист); схема печатной платы (М1:1 и М2:1)—(5 листов); техническое описание и указание по настройке (14 листов).

тов и явлений природы и общества в виде образно-знаковых моделей. Первые карты, дошедшие до наших дней, были составлены в Вавилоне и Древнем Египте в 3-м...1-м тыс. до н. э. Карта мира впервые была составлена К. Птолемеем (II в.). Создание новых картографических проекций и коренное совершенствование карт происходит в конце XVI в. под влиянием Великих географических открытий, развития мировой торговли и мореплавания. С конца XIX в. разрабатывается большое число тематических карт со специальной знаковой и цветовой нагрузкой, среди которых особое место по сложности и высокой информативности занимают геологические карты. Во второй половине XX в. широкое распространение получают синтетические карты, комплексующие разнородную информацию.

Единый информационный язык присущ произведениям технической графики, в которой с помощью линейных форм воспроизводятся орудия труда, технологические процессы, строительная тематика, взаимосвязь технологических процессов во времени и пространстве, функциональные математические зависимости, организация работ, управление производством и т. п.

Возникновение технической графики относится ко времени появления ранней письменности. Техническая графическая документация развивается в связи с сооружением сложных объектов (пирамиды, дворцы, тоннели, рудные шахты) в 3-м...2-м тыс. до н. э. (Древний Египет, Вавилон и др.) и созданием измерительных инструментов и первых механизмов с сер. 1-го тыс. до н. э. (Древняя Греция, Древний Рим).

Совершенствование технической графики происходит в эпоху Возрождения в связи с конструированием кинематически сложных машин и механизмов, гидросиловых установок, коренным усовершенствованием подземной разработки рудных тел, созданием крупных городов.

В период энергетической революции в конце XVIII в. техническая графика является основным инструментом изобретательской деятельности для создания паровых двигателей и разнообразных автономных рабочих машин на этой основе, широко применяются графические формы для установления математических зависимостей. С конца XIX в. техническая графика становится основой проектирования типовых машин, технологий, строительства. С середины XX в. широко внедряются объемная техническая графика с функциональным введением цветовой гаммы, производство графической информации в автоматизированном режиме по заданным программам.

Подводя итог рассмотрению эволюции системы представления информации, можно отметить общую тенденцию к созданию наиболее рациональных форм человеческой (наднациональной) информационной символики. В наиболее трудно доступной лингвистической области это реализуется введением машинных языков.

**Информационные коммуникации** в обществе возникли при появлении человеческой речи, которая стала нематериализованной несущей информацией. Начало этой фазы несомненно можно считать первым информационным взрывом в истории технологоической цивилизации. В течение следующей фазы — до бума жной — информационные взрывы характеризовали переход ко все более совершенным носителям: запись на камне позволила впервые добиться эффекта обезличенности процесса передачи информации, закрепленной навечно в определенном месте; переход к записям на сырых глиняных табличках и деревянных дощечках с 4-го тыс. до н. э. придало информационным коммуникациям динамический характер (камень сохраняет значение носителя символической монументальной информации); изобретение папируса (с 3-го тыс. до н. э.) значительно повышает единичную емкость носителя и его расширяющую способность благодаря большой площади свитка и возможности применить краски; появление пергамента (III в. до н. э.) завершает добужающую фазу новым информационным взрывом: появляется оптимальный носитель информации — книга (IV в. н. э.).

На развитие механизма информационного взаимодействия людей в добужающую фазу оказывали влияние социальные, политические и региональные факторы. В каменном веке никтограмма на камне представляла собой общедоступную информационную коммуникацию, что соответствовало низкому уровню дифференциации физического и умственного труда и отсутствию социальной иерархии. На этапе создания первых государств динамичные информационные носители (глиняные и деревянные таблички, папирусы) превращаются в объекты централизованного закрытого хранения, а пользование ими приобретает прерогативные формы для высшей аристократии. Наличие информационных источников в ведении правящих слоев общества вызывает потребность в обучении наследников этой власти (царской, военной, жреческой), т. е. избирательную по форме передачу знаний. Информационные источники приобретают значение как фактор внутрисоциальных, так и межгосударственных отношений. Упорядочение системы хранения информационных носителей и накопление их в одном месте (например, в столице Хеттского государства — г. Хаттусае во дворце хранилось 20 000 глиняных клинописных табличек) позволяет говорить о создании первых «библиотек», доступных только отдельным лицам и символизирующих накопление знаний и исторических фактов для последующих поколений. В условиях политической нестабильности «библиотеки» были легко уязвимыми, что привело в конечном итоге к почти полному их уничтожению.

Качественно новый характер приобрели информационные коммуникации, когда в крупных государствах Древнего мира (Греция, Персия, Египет, Китай, Рим) возникла хорошо налаженная государственная почтовая связь: письменная информация передавалась гонимами по принципу эстафеты. С образованием древнегреческих городов-полисов создаются библиотеки, доступные для свободных граждан. Библиотека в этот период становится первым в истории центром сосредоточения информационных носителей на папирусных свитках (позднее на пергаменте) для передачи широкому кругу пользователей. своего рода первым институтом массовой информационной коммуникации. Это обстоятельство в значительной мере способствовало началу продуцирования информации в новой форме — авторских сочинениях. Книга приобретает функции товара, чем порождается новая ремесленная профессия переписчиков книг. Книжные лавки превращаются в своего рода интеллектуальные микроцентры, где происходит интенсивный обмен знаниями. Наряду с общественными библиотеками формируется новая форма массовой коммуникации — большое число личных библиотек у наиболее обеспеченных граждан. Принципиально важным моментом является выделение пока узкого социального слоя людей — производителей знаний, закрепляемых ими в книгах и передаваемых наиболее способным продолжателям. Несомненно, что такое широкое информационное взаимодействие людей привело к накоплению не только духовных, но и материальных (в том числе культурных) ценностей, способствовало появлению рациональных для того времени законодательных форм, регулировавших социальные отношения в обществе.

Великое переселение народов, сопровождавшее падение Римской империи, принесло сложившейся культуре (в том числе информационной) практически полное уничтожение. В этой связи нельзя не отметить важное историческое факта миграции информационных источников — рукописных тиражей первых сочинений. Вначале с первыми христианами, гонимыми Римом, они попадают в Византию и Среднюю Азию (главным образом в Бухару). В VII...IX вв. в Бухаре эти сочинения переводятся на арабский язык. После падения Константинополя с сер. XV в. рукописные древнегреческие и латинские книги с беженцами перемещаются в Западную Европу, где становятся основным информационным массивом накопленных знаний.

Бумажная фаза развития информатики начинается с X в., когда бумага (изобретенная еще во II в. в Китае) становится объектом промышленного производства в стра-

нах Европы. Эпоха Возрождения и последующий за ней период сыграли исключительную роль в развитии информатики. С расширением торговли и ремесел появились городские почты, с XV в. — частная почта (Зап. Европа), в XVI..XVII вв. — централизованная королевская почта (Франция, Швеция, Англия и др. страны). Благодаря этим стабильным коммуникациям в информационную деятельность вовлекается большое число людей и она охватывает крупные регионы. Центрами хранения и передачи информации становятся первые университеты Италии, затем Франции, Германии, Англии.

Революцией в информатике стало изобретение в Германии книгопечатания (сер. XV в.), придавшее ей форму массовой деятельности, особенно с конца XVII в., т. е. со времени возникновения науки и появления парового двигателя — основы машинного производства. По существу это стало началом нового научно-технического этапа в естествознании. Главным качественным содержанием информатики стало рождение систем научно-технической терминологии в основных отраслях знаний, а количественным — выпуск многотиражных книг, журналов, газет, географических карт, технических чертежей, а также первых энциклопедий — своего рода стационарных информационно-поисковых систем на алфавитной основе.

Новый этап в развитии информатики, связанный с технической революцией конца XIX в., характеризуется созданием почтовой связи как формы стабильных международных коммуникаций (Всеобщий почтовый союз с 1874 г. и Всемирная почтовая конвенция с 1878 г.), фотографии (с 1839 г.), изобретением телеграфа (1832 г.), телефона (1876 г.), радио (1895 г.), кинематографа (с 1895 г.), а позднее беспроводной передачи изображения (1911 г.) и промышленного телевидения (с конца 20-х гг.). В развитии информационных коммуникаций наступил период создания общемировой системы сосредоточения, хранения и быстродействующей передачи информации в наиболее удобной для пользователей форме. Это превратило информацию в движущую силу технического, социального и экономического прогресса, определило ей ведущую роль на этапе современной технологической революции, которая придает информатике форму интеллектуальной индустрии. Благодаря этому было разрешено давнее историческое противоречие между накоплением гигантского объема информации в обществе и невозможностью эффективного его использования с помощью традиционных немашинных методов. Обширные исследования в области создания и модернизации информационных систем и технических средств привели к созданию новой крупной области знания — технологической информатике.

Информация превращается в один из наиболее ценных по содержанию и массовых по форме продуктов цивилизации, потребителем которой становится все человечество. Этап информационной революции второй половины XX в. знаменует начало безбумажной фазы развития информатики, когда на качественно новом уровне завершается крупнейший исторический виток перехода к неосознаваемой несущей информации, причем скорость ее передачи (посредством электромагнитных волн) возрастает в миллион раз (по сравнению с человеческой речью). Машинная интуиция (экспертные системы) превращается в производительную силу, а искусственный интеллект позволяет решать качественно новые задачи технического прогресса. Исключительное значение машинных динамических информационных систем в жизни современного общества выдвинуло на первый план проблемы создания все более совершенных ЭВМ и связанных с ними технологий. Это привело к узко практическому толкованию термина «информатика» со стороны ряда специалистов. История развития механизма информационного взаимодействия между людьми (а теперь и между человеком и машиной) дает основание для понимания информации как единой интеграционной системы развития всех областей знаний, этапы которой в основном совпадают с периодами становления естествознания

и с более ранними по времени периодами накопления знаний в обществе. Поэтому средства информатики в их историческом развитии представляют собой инструмент познания закономерностей эволюции технологической цивилизации.

Телефон 297-77-62, Москва

Статья поступила 23.08.88

УДК 681.322.1

А. В. Гиглавы, А. В. Немец

## ОСВОЕНИЕ ТЕХНОЛОГИИ САПР БИС В КНР

Общая ситуация в разработке и производстве БИС в КНР

Стратегия развития электронной промышленности в КНР на рубеже 1970...1980 гг. претерпела существенные изменения, что нашло отражение в формировании долговременной целевой программы развития микроэлектроники с одновременной децентрализацией управления предприятиями этой отрасли.

В ходе шестой пятилетки (1981...1985 гг.) впервые в истории КНР была принята и в основном выполнена комплексная программа научно-технического прогресса из 38 пунктов, значительная часть которых была связана с электроникой. За период пятилетки было импортировано более 1000 комплектов передового оборудования для электронной промышленности; 1/3 предприятий этой отрасли прошла коренную техническую реконструкцию.

Динамика производства ИС демонстрируется в таблице.

Выпуск ИС, млн. шт. (за 1987 г. — предварительные данные)

1978	1980	1981	1982	1983	1984	1985	1986	1987
30,0	16,8	12,8	13,5	23,6	39,2	53,0	46,0	75,0

Уменьшение выпуска ИС в начале 80-х гг. связано с прекращением производства устаревших малых и средних ИС. После того как КНР освоила выпуск ИС для телевизоров, магнитофонов и ПЭВМ, производство ИС вновь стало расти. В 1985 г. было импортировано порядка 200 млн. ИС и к концу года на складах скопилось 135 млн. ИС, что привело к падению их выпуска в 1986 г. В 1987 г. в результате ограничения импорта ИС и повышения качества китайских изделий, кризис сбыта был преодолен.

В середине 80-х гг. выпуск ИС в КНР в основном был сосредоточен на 10 базовых заводах, подчиненных Министерству электронной промышленности (МЭП) КНР и действующих в Пекине, Шанхае и граничащих с Шанхаем районах провинций Цзянсу и Чжэцзян. В производстве ИС было занято примерно 40 тыс. чел. включая 5 тыс. ИТР. Определенными мощностями по выпуску ИС располагало и Министерство космической промышленности.

До 1985 г. БИС в КНР практически не выпускались, а в 1985 г. их выпуск составил 2,1 млн. шт., главным образом динамических ОЗУ (16 Кбит). В 1986 г. началось мелкосерийное производство динамических ОЗУ (64 Кбит), статических ОЗУ (16 Кбит), а также 16-разрядных микропроцессоров. Маштабы выпуска этих изделий (и доля БИС в производстве ИС) значительно выросли в 1987 г.

В 1986 г. заводами МЭП и Министерства космической промышленности был освоен серийный выпуск ИС с шириной линии 5 мкм (динамических ОЗУ (16 Кбит), 8-разрядных микропроцессоров типов 6502 и 6800); в опыт-



ном порядке выпускались ИС с шириной линии 2,5...4 мкм (ДОЗУ (64 Кбит) и 16-разрядные микропроцессоры типа Зайлог 8000); в НИИ и вузах разрабатывалась технология обработки изделий с шириной линии 1 мкм. К 1988 г. электронной промышленностью освоена технология серийного производства БИС с шириной линии 3 мкм.

Главные задачи КНР в седьмой пятилетке (1986...1990 гг.) в области разработки и производства ИС, утвержденные в 1986 г.:

1. Создание технологий СБИС, КAD/КАМ и автоматизированного контроля основных этапов производства.
2. Стабильное производство БИС и разработка СБИС.
3. Организация производства 16-разрядных микропроцессоров в промышленных масштабах.
4. Разработка 32-разрядных микропроцессоров.
5. Разработка вентиляемых матриц.
6. Выпуск в 1990 г. 400 млн. ИС.

Можно сделать вывод, что микроэлектроника КНР к 1990 г. должна по качественным показателям выйти на мировой уровень середины 80-х гг. [1].

В 1984...1986 гг. КНР импортировала большое количество современного оборудования для изготовления ИС, включая производственные линии большой мощности; кроме этого, электронная промышленность самостоятельно разрабатывала и начала выпускать ряд важных элементов этого оборудования. Например, в 1986 г. были созданы фотоповторитель, обеспечивающий стабильный выпуск изделий с шириной линии 2...3 мкм и пригодный для серийного изготовления ДОЗУ (64 и 256 Кбит), и устройство ионного внедрения мощностью 200 кэВ. Из-за отсутствия необходимой технологии мощность оборудования полностью не использовалась и выпуск изделий высокого качества не обеспечивался, что стало одной из причин массового импорта ИС в 1985 г. и сокращения их выпуска в 1986 г. На состоявшемся в конце 1986 г. в Пекине первом ежегодном заседании Китайского научного совета по полупроводникам указывалось, что импортированная техника осваивается неудовлетворительно [2].

Очевидно, что для достижения целей, поставленных перед электронной промышленностью КНР, необходимо было создать обладающую высоким уровнем технологию разработки БИС и прежде всего САПР БИС.

### Начальный этап разработки САПР БИС

Работы в этой области в КНР начались еще в 1984 г. Осенью 1985 г. центральная пресса КНР опубликовала список премий за наиболее выдающиеся достижения науки и техники в 1983...1984 гг. В частности, первую премию получила система автоматизированного проектирования и анализа БИС и СИС, разработанная Шанхайским транспортным университетом (ТУ), а одну из третьих премий — САПР шаблонов БИС, созданный Шанхайским НИИ металлургии АН КНР (именно этот НИИ в 1985 г. впервые в КНР разработал ИС с шириной линии 2 мкм, а в 1986 г. — БИС 16-разрядного микропроцессора-аналога Зайлог 8000) [3].

В конце 1985 г. Пекинский авиационный институт (ВУЗ) создал САПР БИС, реализованную на сепер-мини-ЭВМ VAX-11/750.

В 1985 г. Шанхайский ТУ продолжал совершенствовать свою САПР БИС. На передний план была выдвинута задача создать базу данных САПР для интеграции функций системы моделирования с учетом проектных норм.

Поскольку самостоятельная разработка специализированной базы заняла бы много времени, специалисты ТУ решили использовать в качестве основы универсальную базу данных DBMS TOTAL. В результате к концу 1985 г. был создан прототип базы данных САПР БИС, позволивший существенно улучшить исходную САПР [4].

Продолжая эту работу, лаборатории САПР и компьютерной графики Шанхайского ТУ в мае 1986 г. создали подсистему размещения и анализа фрагментов СИС и БИС, по своим характеристикам достигающую мирового уров-

ня [5]. В результате летом 1986 г. фирма «Аполло компьютер» (США) приобрела САПР печатных плат, разработанную на факультете электронной техники, — это важный показатель уровня, достигнутого университетом в области САПР для микроэлектроники [6].

В НИИ электронно-вычислительной техники АН КНР (Пекин) в конце 1985 г. был создан прототип кремнивого компилятора, предназначенного для проектирования БИС и СБИС и обеспечивающего выполнение ряда функций иерархического проектирования. Это программное обеспечение было представлено на 3-м всекитайском семинаре по САПР ИС, состоявшемся в Пекине в 1985 г. На втором этапе разработок предполагалось создать кремниевый компилятор, соответствующий передовому мировому уровню. САПР БИС была одной из нескольких специальностей, по которым осенью 1986 г. НИИ ЭВТ проводил набор в докторантуру [7].

Важным центром разработки САПР БИС является Пекинский политехнический университет (ПУ) Цинхуа. НИИ микроэлектроники ПУ Цинхуа уже в 1985 г. имел в своем составе отдел САПР БИС. Именно этот НИИ был ведущей организацией в разработке и широком внедрении в КНР технологии БИС с шириной линии 3 мкм.

В середине 1986 г. факультет ЭВМ ПУ Цинхуа совместно с 25 НИИ и заводами завершил разработку САПР ИС. Создание данной САПР было расценено китайскими специалистами в ходе аттестации как успешный шаг на пути к производству в КНР самостоятельно разработанных ИС. Вскоре после этого радиофакультет ПУ Цинхуа создал САПР ИС для цветных телевизоров [8].

В марте 1986 г. в Пекине прошла аттестация системы анализа и проектирования ИС, совместно созданная НИИ автоматизации АН КНР в Пекине и Шанхайским НИИ металлургии АН КНР. Система базируется на модели представления знаний, являющейся соединением методов искусственного интеллекта и распознавания моделей. В этой системе использована база данных высокого уровня, хранящая информацию о процессе проектирования ИС и разновидности технологий.

В сообщении о создании этой системы указывалось, что ранее анализ ИС, используемых в качестве прототипов для выпуска китайских изделий, выполнялся в основном вручную и очень медленно. Внедрение новой системы, которая еще до аттестации успешно применялась в ходе освоения производства высокоскоростных КМОП ИС, позволило повысить производительность труда при разработке ИС в 6...9 раз, обеспечить автоматическое регулирование скорости и точности ведения технологических процессов. Члены государственной комиссии по приемке системы пришли к выводу, что она находится на ведущем месте в КНР, по качественным показателям соответствует передовому мировому уровню и ее внедрение позволит существенно ускорить развитие микроэлектроники в КНР [9, 10].

В июле 1986 г. в Пекине состоялось всекитайское совещание по применению ЭВМ, на котором была выработана стратегия развития данной области до 1990 г. В ходе совещания, в частности, принято решение о внедрении в электронной промышленности КНР САПР БИС. Возможности для реализации этого решения были созданы в результате успешных разработок в ведущих вузах и академических НИИ.

По мнению американских специалистов, технология и программное обеспечение для освоения производства ДОЗУ (64 Кбит) и 16-разрядных микропроцессоров, выпуск которых начался в 1986 г., были разработаны в КНР самостоятельно [11].

### Современное состояние разработок САПР БИС в КНР

В конце мая — начале июня 1987 г. несколько организаций, успешно разрабатывавших САПР БИС в 1985—1986 гг., завершили создание новых систем, что качественно изменило ситуацию в данной области.

Шанхайский ТУ создал автоматизированную систему проектирования, контроля, измерения характеристик БИС и СБИС. С использованием этой системы была спроектирована и запущена в опытное производство на одном из заводов Шанхая СБИС, содержащая 128 тыс. транзисторов. По оценке китайских специалистов эта система, без которой невозможно было обойтись при проектировании БИС и СБИС, была создана впервые в КНР, что позволило начать подготовку к производству различных БИС специального назначения, в частности для супер-ЭВМ.

Разрыв между КНР и развитыми странами резко сократился. Специалисты двух американских фирм, тщательно изучив эту систему, пришли к выводу, что в области разработки БИС она достигла мирового уровня 80-х годов. Эти фирмы подписали с Шанхайским ТУ контракт о закупке разработанного в университете программного обеспечения [12].

Одновременно в Пекине прошла аттестацию созданная НИИ ЭВТ АН КНР комплексная система контроля и измерения характеристик БИС/СБИС, в результате был выполнен один из пунктов осуществляемой в КНР комплексной научно-технической программы. Ранее системы такого уровня в КНР отсутствовали [13].

Наконец, НИИ полупроводников АН КНР в Пекине также создал САПР БИС. В сообщении об этом указывалось, что прорывная техническая блокада, осуществляемая в отношении КНР западными странами. Одна из фирм в Гонконге, получившая эту САПР для опытного использования весной 1986 г., в течение месяца успешно спроектировала первую БИС. После этого ряд фирм США, Англии, Японии, Италии начали переговоры с НИИ полупроводников о закупке программного обеспечения и сотрудничестве в дальнейших разработках.

В сообщении также указывалось, что в данной области КНР уже вышла на передовый мировой уровень; китайские разработчики могут теперь стабильно поставлять эффективные средства проектирования для производственных линий большой мощности по выпуску ИС, импортированных КНР за последние несколько лет. Новая САПР обеспечивает существенное повышение производительности труда разработчиков и повышение точности на 50 % по сравнению с использовавшимися до этого методами проектирования; ее сфера применения охватывает цифровые и частично линейные схемы [14].

К середине 1987 г. электронная промышленность КНР освоила массовый выпуск большинства ИС и всех печатных плат, необходимых для сборки ПЭВМ класса РС/XT. К концу года был освоен серийный выпуск супер-мини-ЭВМ типа migco-VAX II с долей китайских деталей на уровне 70 % или выше. Осваивался и выпуск различной электронной техники высокого уровня. Перед разработчиками БИС встали новые задачи.

В начале марта 1988 г. центральная печать КНР поместила сообщение о том, что успешно реализован важный пункт комплексной научно-технической программы седьмой пятилетки (1986...1990 гг.) — создана и прошла аттестацию САПР БИС, по своим характеристикам превосходящая все прежние системы.

Система была разработана самостоятельно китайскими специалистами, реализована на базе супер-микроЭВМ с функциями рабочей станции и включает разнообразное программное обеспечение для проектирования ИС. Отмеча-

ется, что по своим функциям новая система превосходит образцы САПР БИС, импортированные различными китайскими организациями за последние несколько лет; часть программного обеспечения достигает передового мирового уровня.

До аттестации эта система использовалась 3-м Пекинским заводом полупроводниковых деталей, 871-м заводом МЭП в Шаосине, пров. Чжэцзян, 13-м НИИ МЭП и другими организациями. В результате было спроектировано более 40 ИС, включая ИС для полностью автоматической стиральной машины, для «электронной арфы» и др.; часть ИС уже запущена в серийное производство. В ходе разработки данной системы накоплен опыт и подготовлены кадры для развития в КНР САПР СБИС.

При создании этой системы по инициативе МЭП КНР создано объединение из 10 вузов, НИИ и заводов, включая факультет ЭВМ университета Цинхуа, факультет электронной техники университета Фудань (в 1985...1986 гг. он вел успешные работы по внедрению наиболее передовых импортных САПР БИС), НИИ микроэлектроники АН КНР в Пекине, 3-й Пекинский завод полупроводниковых деталей. Около 100 ИТР вели работу в течение трех лет; в результате был создан комплект программного обеспечения из 190 программ, содержащих около 600 тыс. команд. Новая САПР БИС является конкурентоспособной на мировом рынке, незадолго до аттестации она была отправлена на Международную выставку изобретений в Брюсселе [15].

Можно сделать вывод, что создание новой САПР БИС практически решает проблемы разработки и производства в КНР БИС, содержащих до 100 тыс. транзисторов, обеспечивает возможности для быстрого роста выпуска ИС и полного использования наличных производственных мощностей. Следующим шагом должна стать разработка САПР СБИС, содержащих несколько сотен тысяч транзисторов.

Во второй половине 1988 г. впервые в КНР была создана новая, усовершенствованная САПР БИС, предназначенная и для выпуска СБИС. К концу года различные организации страны внедрили более 20 единиц этой САПР.

Телефон 133-24-35, Москва

#### ЛИТЕРАТУРА

1. Никкэй электроникс / На японс. яз.— 1987.— 12.01.— С. 265.
2. Гуанмин жибао.— 1986.— 18.12.
3. Гуанмин жибао.— 1985.— 8—11.10.
4. Цзисуаньцзи яньцзю куй фачжань (Исследования и развитие ЭВМ).— 1986.— № 1.— С. 27
5. Вэньхой бао.— 1986.— 9.06.
6. Жэньминь жибао.— 1986.— 23.07.
7. Цзисуаньцзи яньцзю куй фачжань.— 1986.— № 3.— С. 7.
8. China daily.— 1986.— 25.07.
9. Жэньминь жибао.— 1986.— 15.03.
10. Гуанмин жибао.— 1986.— 15.03.
11. Electronics weekly.— 1987.— 10.06.
12. Жэньминь жибао.— 1987.— 6.06.
13. Гуанмин жибао.— 1987.— 2.06.
14. Жэньминь жибао.— 1987.— 2.06.
15. Жэньминь жибао.— 1988.— 1.03.

Статья поступила 15.06.88.

#### УВАЖАЕМАЯ РЕДАКЦИЯ!

Полностью присоединяюсь к опубликованным в № 2 за 1988 г. предложениям Р. А. Бронштейна и М. И. Потемкина о создании вневедомственного фонда алгоритмов и программ и об организации Союза программистов.

Принципы, по которым функционирует ГОСФАП, аномальны. В него попада-

ют сырые, неотлаженные программы. Вместе с тем, хотелось бы отметить, что процедура оформления программ очень сложная.

Новому фонду необходимо: резко упростить процедуру оформления, принимать программы по отзывам независимых пользователей;

создать при фонде мощную службу оперативного тестирования для особо важных программ.

Круг задач, которые помог бы решить творческий Союз программистов, огромен. Создание правового государства невозможно без адекватных юридических норм по вопросам программирования, автоматизированной обработки и хранению информации.

О. А. Миленький, ведущий инженер  
РОВИ МЛН БССР

О. Ф. Ускова, М. Н. Карпачева

### ПОЛУЧЕНИЕ СПРАВОЧНОЙ ИНФОРМАЦИИ С ПОМОЩЬЮ ПРОСТЕЙШЕГО ЭКРАННОГО РЕДАКТОРА ТЕКСТОВ

Для получения оперативной справочной информации в любой предметной области разработана программа, работающая в режиме диалога «пользователь-ЭВМ». С помощью этой программы можно получать в памяти ЭВМ (на диске) информационный файл, изменять и дополнять его новыми порциями информации, выводить текст на экран дисплея. Она позволяет работать с текстовой информацией четырехуровневой структуры. Информация может быть представлена либо сплошным куском текста, либо параграфами, состоящими из пунктов или непрерывного текста. Пункты внутри параграфа (уровень 2) также могут быть непрерывным текстом или состоять из подпунктов. Подпункты (уровень 3) внутри пунктов — непрерывный текст.

Разработанная программа не требует обширных знаний по организации операционной среды, проста в эксплуатации. Вся работа с текстовыми файлами (поиск, открытие и закрытие, распределение памяти как внешней, так и оперативной, поиск нужной информации внутри файла, запись в файл текста, ведение файла каталога ранее созданных файлов и т. д.) осуществляется автоматически. Она скрыта от пользователя и лишь соответствующие сообщения выдаются на экран терминала.

Для удобства пользователя в программе в качестве сервисного средства включен простой редактор текста со следующими функциями: удаление строки, вставка строки, удаление символа, вставка символа, движение маркера по символам одной строки, движение маркера по строкам, вывод на экран дисплея информации о функциях редактора, восстановление экрана, выход из режима редактирования с записью текста на диск, освобождение динамической памяти, отражение на экране производимых действий с текстом.

Система состоит из управляющей части, простого редактора текста, информационного файла, содержащего краткие сведения о работе программы. Управляющая часть обеспечивает диалоговый режим работы для создания новой информации, имеющей четырехуровневую структуру, чтения или корректировки любой ее составной части.

Разработанная программа может использоваться автономно или в составе автоматизированных систем обучения и контроля знаний. Она внедрена в лаборатории вычислительной техники факультета прикладной математики и механики Воронежского Государственного университета, где с ее помощью созданы справочные файлы по языку программирования Модуль-2 и алгоритмическому языку, изучаемому в средних школах.

Телефон 56-63-08, Воронеж

Сообщение поступило 3.12.87

### ОПЕРАЦИОННАЯ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ И ПРОГРАММА-МОНИТОР ДЛЯ 16-РАЗРЯДНОЙ МИКРОЭВМ НА БАЗЕ МИКРОПРОЦЕССОРА КМ1810ВМ86

Операционная система реального времени реализует многозадачный режим работы, синхронизацию задач, обработку событий в режиме реального времени. Расширение ядра операционной системы: программа работы с пультом, дисковая файловая система, драйвер постстрочного печатающего устройства, начальный загрузчик, интерактивный отладчик.

Программа-монитор обеспечивает доступ оператора к регистрам памяти, портам ввода-вывода, пуск программ с заданного адреса с остановами в указанных местах и ряд других сервисных функций.

Пользователям предоставляется полный комплект документации, включая исходные тексты программ, а также консультации по установке и первоначальному запуску программ.

Телефон 924-67-06, Москва, Невзнер С. Я.

### МИКРОСХЕМА РПЗУ К573РФ3

РПЗУ К573РФ3 — память с ультрафиолетовым стиранием и электрической

Таблица 1

Вывод	Назначение
1	Вход сигнала чтения, RE
2	Выход сигнала ответа, RPLY
3	Вход сигнала синхронизации, CE
4...11	Входы-выходы адреса-данных, AD4...AD11
12	Общий, GND
13...16,	Входы-выходы адреса-данных, AD3...AD0,
17...20	AD12...AD15
21	Напряжение источника питания, U <sub>CC2</sub>
22	Запись-считывание, WE/RE
23	Выбор микросхемы, CS
24	Напряжение источника питания, U <sub>CC1</sub>

записью данных. Информационная емкость — 65536 бит (4096×16 разрядов).

Микросхема выполняется по n-канальной МОП-технологии; выпускается в 24-выводном металлокерамическом корпусе типа 210Б.24-5. Число элементов в микросхеме — 140.000. Условное графическое обозначение РПЗУ К573РФ3 приведено на рис. 1, назначение выводов описано в табл. 1, электрическая структурная схема дана на рис. 2.

Микросхема предназначена для построения блоков памяти микроЭВМ. Информационная емкость и адресация ячеек показаны в табл. 2. Основные

Таблица 2

Микросхема	Емкость, Кбит	Адресация
К573РФ3	64	0...4096
К573РФ31	32	0...2047
К573РФ32	32	2048...4096

### СПРАВОЧНАЯ ИНФОРМАЦИЯ

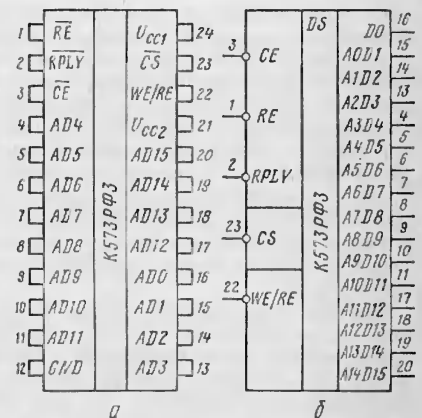


Рис. 1. Условное графическое обозначение микросхемы К573РФ3 по порядку расположения (а) и функциональному назначению (б) выводов

(Начало. Окончание см. на с. 77)

В статье рассматривается актуальная задача циркулярной передачи патентной информации заинтересованным организациям и предприятиям.

Учитывая широкий охват первой программой центрального телевидения всей территории СССР, авторы справедливо считают наиболее перспективным использование разветвленной ТВ-сети для передачи информации.

Технические решения, предлагаемые в статье, основаны на уже выполненных проработках и экспериментальных исследованиях и могут быть реализованы. Срок практической реализации предложений, обсуждаемых в статье, будет зависеть прежде всего от решения вопросов создания банков данных, каналов по сбору информации и предоставления потребителям технических средств для приема циркулярной информации.

С выходом нового закона об изобретательстве актуальность рассматриваемого в статье вопроса повысится, так как без развитого информационного обеспечения Государственный комитет по делам изобретений и открытий не сможет эффективно проводить политику резкого ускорения внедрения изобретений в промышленность и народное хозяйство.

Заместитель начальника ГНТУ  
Минсвязи СССР  
к. т. н. В. В. Плеханов

УДК 681.3.06—181.4

А. Н. Исаев, В. И. Ближников, М. И. Кривошеев,  
В. К. Сарьян, И. Н. Красносельский, Ю. М. Боловинцев

## ПЕРЕДАЧА ПАТЕНТНОЙ ИНФОРМАЦИИ В СОСТАВЕ ТЕЛЕВИЗИОННОГО СИГНАЛА

С возрастающим потоком патентной информации можно справиться, если широко внедрить ее автоматизированную обработку, хранение и распространение на машинночитаемых носителях. Но преимущества автоматизированной системы можно реализовать в полной мере только в том случае, если машинночитаемые носители используются на всех уровнях системы. Сегодня, когда заявки на изобретения поступают во ВНИИГПЭ на бумажных носителях, перед внедрением новой технологии возникают трудности, связанные с нехваткой во ВНИИГПЭ, ВНИИПИ и ВИНТИ людских ресурсов и аппаратных средств, необходимых для обработки поступающих материалов заявок и перевода их на машинночитаемые носители.

Поэтому пока создана только база данных (БД) по реквизитам заявок (ВНИИГПЭ и ВНИИПИ) и создаются БД по их рефератам (ВНИИПИ и ВИНТИ). Эти БД не могут удовлетворять всем патентно-информационным потребностям пользователей, так как заметного сдвига в информационном обслуживании не происходит: после автоматизированного поиска в этих БД (включая системы доступа по каналам связи с удаленных терминалов) пользователь вынужден заказывать во ВНИИПИ полные описания изобретений, получать их традиционными «долгими» способами (например, путем ксерокопирования и рассылки по почте) и обрабатывать вручную.

Эти принципиальные трудности внедрения новой информационной технологии можно преодолеть, если снабдить низовых пользователей, изобретателей и патентоведов удобным инструментальным средством — автоматизированной системой по оформлению первичных документов по заявкам [1, 2]. Основные принципы, заложенные при разработке и модификации этой системы, таковы:

- система должна быть привлекательной для автора изобретения (этого можно достичь, например, обеспечивая пользователя в виде подсказки нужным объемом знаний нормативных документов, по которым должна быть составлена его заявка); выявляя формализацию и избыточность, заложенную в тексте описания и других первичных документах заявки, обеспечивать сокращение объема вводимой пользователем в ЭВМ информации, ее структуризацию, возможность накопления и многократного использования отдельных единиц структуризации;

- система должна быть реализуема на массовых отечественных и импортных ЭВМ любой производительности;

- алгоритм обработки информации в системе должен учитывать особенности делопроизводства по заявкам во ВНИИГПЭ;

- должна быть возможность получения документов в привычном нормативном виде (в виде распечаток с АЦПУ).

В различных версиях системы, в зависимости от выбранной ЭВМ и развитости сопутствующей этой системе нормативных и предметных баз знаний, может быть достигнуто соответствующее сокращение избыточности по вводимой пользователем в ЭВМ информации, необходимой и достаточной для формирования всех первичных документов по заявкам (включая и рисунки)\*:

- 80 Кбайт — при выделении только реализующих структур [2];

- 20 Кбайт — при выделении реализующих структур и информационно-независимых частей [2];

- 5 Кбайт — при хорошо развитой предметной базе знаний, сопутствующей данной системе.

Сокращение избыточности по входу не только важно для автора заявки (уменьшается время оформления заявки на изобретение), но и может иметь решающее значение для формирования БД по заявкам и изобретениям.

Автоматизированная система по оформлению заявок, построенная на этих принципах, прошла в 1985—1987 гг. опытную эксплуатацию и стала частью создаваемой в стране системы безбумажной технологии делопроизводства по заявкам [3].

Широкое использование автоматизированной системы на предприятиях страны и в консультационных пунктах ВОИР, а также внедрение в практику других элементов безбумажной технологии делопроизводства по заявкам приведут к тому, что во ВНИИГПЭ будет стремительно накапливаться база данных полных документов по заявкам и описаниям к выданным изобретениям.

Наличие таких структурированных БД позволит получить во ВНИИГПЭ в автоматизированном режиме вторичные (такие, как реферативная и реферативная) базы данных, а также извлекать из них в автоматизированном режиме различную обзорно-аналитическую информацию (например, такую как матрицы [4], используемые во ВНИИГПЭ для обеспечения достоверности и полноты проводимого патентного поиска в процессе экспертизы).

\* Цифры приведены для заявки среднего объема (~100 Кбайт).

Наиболее перспективна для быстрого распространения этой информации среди потребителей передача патентной информации в составе ТВ-сигнала. Основанием к такому утверждению может служить следующее: патентная информация — циркулярная и ТВ-вещание — циркулярное, ТВ-вещанием охвачено 95 % населения.

Возможные объемы передаваемой информации будут определяться числом заявок, поступающих во ВНИИГПЭ, и количеством признанных из них изобретениями; числом выделенных категорий пользователей, для которых необходим свой специфический патентно-информационный продукт; числом разделов или классов МКИ, по которым целесообразно группировать передаваемую информацию; установленными сроками накопления информации и временными интервалами между пакетными передачами накопленной информации.

Ежегодно во ВНИИГПЭ поступает около 150 тыс. заявок, открытая публикация которых возможна; из этого числа примерно половина признается изобретениями. Таким образом, в стране ежемесячно создается около 6—7 тыс. изобретений. Если установить временной интервал между передачами равным трем дням и группировать изобретение только по девяти разделам МКИ, то объем одного пакета информации составит 700 изобретений. Умножив это значение на приведенные выше значения объема первичных документов по одной заявке на изобретение, можно получить максимальные значения одного пакета информации.

Выделим категории потребителей патентной информации: публичные патентные библиотеки (региональные, городские), библиотеки предприятий, патентные отделы предприятий, консультационные пункты ВОИР, индивидуальные пользователи, отраслевые и территориальные патентные фонды.

Каждая из этих категорий пользователей отличается не только полнотой собираемой информации, но и требуемой степенью ее безыбыточности (т. е. степенью структуризации), а также потребностью во вторичной информации и результатах обзорно-аналитической переработки. В этом отношении особое положение в категории пользователей занимают патентные отделы предприятий и консультационные пункты ВОИР, т. е. учреждения, где установлены автоматизированные системы по оформлению заявок на изобретение. Поскольку такие системы становятся одним из главных инструментальных средств новой патентно-информационной технологии, необходимо, чтобы они были постоянно в рабочем состоянии. Это означает, что нормативные знания в каждой отдельной пользовательской системе должны оперативно изменяться при изменении нормативных актов Госкомизобретений (что происходит не реже одного раза в два месяца). Для правильного оформления заявки пользователем очень большое значение имеет также адекватная предметная база знаний на стороне пользователя и ВНИИГПЭ, а также такие результаты обзорно-аналитической переработки, как матрицы. Эти нужные для патентных отделов и консультационных пунктов ВОИР информационные продукты и изменения в ПО ЭВМ, на которых реализована система, также целесообразно и очень эффективно передавать в составе ТВ-сигнала.

Для индивидуального пользователя (и/или предприятия) следует (на приемной стороне) предусмотреть возможность режима автоматизированной выборки и накопления из передаваемого пакета патентной информации определенного узкого подкласса или рубрики МКИ. Это сделать легко, так как каждая заявка изначально содержит все необходимые реквизиты.

Как показывают расчеты, переход к такой технологии экономически оправдан, когда не менее 70 % заявок будут поступать во ВНИИГПЭ на машиночитаемых носителях, выполненных с помощью автоматизированной системы. Остальная часть информации (поступающая на бумажных носителях) будет перерабатываться с помощью таких же автоматизированных систем непосредственно во ВНИИГПЭ. Таким образом, будет создаваться советский фонд изобретений на машиночитаемых носителях. Для внедрения этой технологии в настоящее время решаются правовые

вопросы, связанные с приемом заявок во ВНИИГПЭ на безбумажном носителе. Эта задача приобретает интернациональный, межгосударственный характер, так как тенденция подачи заявок по безбумажной технологии развивается и в других странах (США, Япония и т. д.). Создание национальных баз на магнитных носителях и обмен такими базами приведут к тому, что по данной системе можно будет распространять и зарубежную патентную информацию.

Как показывают предварительные технико-экономические расчеты, абонентская плата за пользование такой системой будет меньше средств, уплачиваемых предприятиями за информационные изделия и услуги. Существенное сокращение сроков поступления непосредственно к потребителям различной информации в подготовленном для обработки виде значительно повысит эффективность накапливаемого в стране и в мире научно-технического потенциала, который содержится в изобретениях и патентах, и, тем самым, ускорит развитие производительных сил страны.

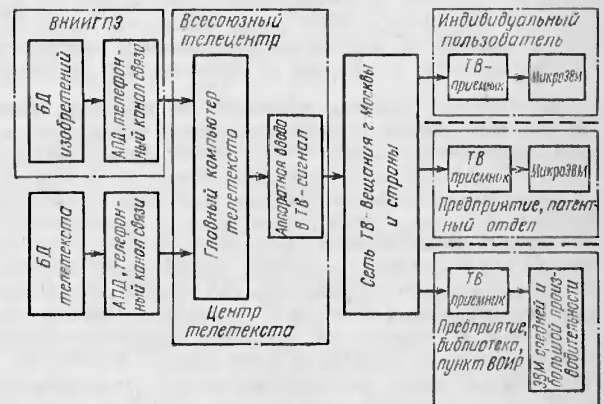
Система распределения патентной информации может основываться на методах передачи дополнительной информации в составе ТВ-сигнала.

Наиболее известными и развитыми техническими средствами подобного типа являются системы телетекста. Они обеспечивают передачу зрителю справочной информации одновременно с сигналом ТВ-программы, причем на экран ТВ-изображения или страницы телетекста выводятся по желанию зрителя.

Основные элементы системы телетекста — БД телетекста, аппаратура ввода этих данных в свободные строки ТВ-сигнала в интервале гашения по полям, сеть распределения и трансляции ТВ-программ на абонентские телевизоры, снабженные декодерами телетекста. В нашей стране разносторонне испытаны различные системы телетекста и развешиваются работы по их производству и внедрению. Поэтому уже сейчас необходимо учитывать при разработке оборудования те специфические требования, которые налагает передача патентной информации в составе ТВ-сигнала.

Система циркулярной передачи ведомственной информации по принципам действия схожа с системой телетекста, но отличается повышенной помехозащищенностью. Эта система разрабатывается и должна использоваться одну из строк ТВ-сигнала первой общесоюзной программы [5].

Сеть циркулярной связи более приспособлена для распределения информации, предназначенной выделенным абонентам (ТАСС, АПН, Госметцентру и т. п.), имеющим соответствующие средства документальной регистрации (ЭВМ, печатающие устройства, факсимильные аппараты и т. п.). Параметры системы циркулярной связи (и, в первую очередь, тактовая частота замещения дополнительной информации в состав ТВ-сигналов) подбираются так, чтобы обеспечить оптимальное согласование и необходимую поме-



Обобщенная структурная схема передающей части системы телетекста и циркулярной передачи для распределения изобретений из центральной базы данных

хоустойчивость с возможностью выделения сигналов в любой точке приема ТВ-программы без дополнительных регенерирующих устройств, устанавливаемых обычно перед ТВ-передатчиками.

Использование всего возможного интервала введения дополнительной информации в пределах импульсов гашения полей (в процессе ТВ-вещания) и в активной части изображения (во время перерывов) при условии применения соответствующей адресной системы позволяет параллельно передавать по ТВ-сети информацию различных источников.

При указанном подходе массив данных по изобретательству может передаваться в любом режиме — после накопления или по мере подготовки. Он не влияет на передачу других видов информации и, в первую очередь, на информацию телетекста для населения.

Рассмотрим возможный вариант использования систем телетекста и циркулярной передачи для распределения изобретений из центральной БД. Передающая часть системы (см. рисунок) включает в себя базы данных изобретений и страниц информации телетекста, систему управления БД (СУБД), главный компьютер центра телетекста, аппаратуру передачи данных (АПД) по телефонным каналам, посредством которой связаны удаленные БД изобретений (БДИ) и центр телетекста.

При такой организации системы БДИ — это, по сути дела, один из поставщиков информации системы телетекста, имеющий, однако, некоторые особенности. Во-первых, это значительная временная дискретность выдачи массивов информации (например, 700 заявок один раз в два дня). Во-вторых, ограниченное время вещания в составе ТВ-сигнала, обеспечивающее уверенный прием и запись информации региональными получателями. В-третьих, повышенная помехоустойчивость и исключение несанкционированного доступа со стороны широких масс пользователей. Рассмотрим, как эти особенности влияют на параметры и конфигурацию всей системы.

**Объем передаваемых данных и время передачи.** Если принять объем одной заявки за 20 Кбайт, то при 700 заявках объем данных одного массива примерно равен  $1,4 \cdot 10^4$  Кбайт. Однако для повышенной помехоустойчивости необходимо применить коды, исправляющие ошибки. Вероятность ошибки в приеме патентной информации не должна превосходить значений  $10^{-8} \dots 10^{-7}/1$  бит. В то же время для системы телетекста допустима вероятность ошибки  $10^{-4}$ , а при передаче данных по телефонной сети в неблагоприятных ситуациях она может достигать  $10^{-3}$ . Поэтому необходимо применить мощные корректирующие коды, гарантированно снижающие вероятность ошибки с  $10^{-3}$  до  $10^{-7}$  и устойчивые к воздействию пакетов ошибок. Таким требованиям удовлетворяют сверточные самоортогональные коды в сочетании с оптимизированным пороговым алгоритмом их декодирования [6]. Исследования показывают, что приемлемые кодовые скорости таких кодов —  $2/3 \dots 4/5$  [6]; таким образом, полный объем передаваемой информации может возрасти в 1,5 раза и составить  $2,1 \cdot 10^4$  Кбайт.

**Возможности системы телетекста по передаче такого объема информации.** В системе телетекста мгновенная скорость передачи в пределах строки данных установлена равной 6,9375 Мбит/с, а пакет данных состоит из 40 байт [7]. Пропускная способность системы телетекста зависит от числа используемых строк данных. Исходя из особенностей технологии отечественного ТВ-вещания выделение для передачи телетекста более двух строк в каждом поле ТВ-сигнала не представляется возможным (по крайней мере, в течение ближайших нескольких лет). Ориентируясь на две строки данных в каждом поле, получаем пропускную способность, равную 4 Кбайт/с (массив данных изобретений в  $2,1 \cdot 10^4$  Кбайт может быть передан за 1,5 ч при условии, что никакая другая информация по системе телетекста не передается).

Практически одного сеанса телезагрузки удаленных БДИ пользователей мало, поскольку это в ряде случаев сопровождается потерями информации. Причины потерь —

пересторопность персонала приемных БДИ, сбой в работе компьютеров БДИ и таймерных устройств, внезапные интенсивные помехи в канале передачи и т. п. Поэтому целесообразно передавать информацию в течение нескольких последовательных сеансов, например четырех. Но и время занятия канала при этом возрастает пропорционально (до 6 ч), что физически не реализуемо. Из этого противоречия может быть два выхода: загружать БДИ ежедневно, отводя для этого существенную часть общего объема журнала, или (более перспективно) вводить данные во все доступные строки кадра ТВ-сигнала (при отсутствии передач сигнала изображения).

В последнем случае число строк в кадре, в которые можно вводить данные, составляет примерно 600 (исключается ряд строк импульса синхронизации полей и строк с контрольными измерительными сигналами). При этом пропускная способность системы возрастает до 600 Кбайт/с, а время передачи массива сокращается до 140 с (четыре сеанса).

Соответственно буфер данных на передающей стороне системы должен быть рассчитан на хранение  $2,1 \cdot 10^4$  Кбайт информации с возможностью быстрого чтения из него — четырежды за 140 с. Для загрузки этого буфера в течение всего десятидневного срока между сеансами требуется скорость передачи по телефонному каналу, равная 0,19 Кбит/с. С учетом необходимых технологических перерывов эта скорость должна быть повышена, например до стандартных значений 1200...4800 бит/с.

Таким образом, информация по изобретательству может быть передана в составе ТВ-сигнала в рамках либо системы телетекста, либо системы циркулярной передачи дополнительной информации с возможностью вывода данных на устройство документальной регистрации.

Предлагаемые варианты системы передачи патентной информации в составе ТВ-сигнала позволяют быстро решить важную народнохозяйственную задачу и при этом существенно не увеличат загрузку систем «Телетекста» и циркулярной связи.

Телефон 261-11-09, Москва, Сарьян В. К.

## ЛИТЕРАТУРА

1. Сарьян В. К., Александров В. А., Смолич Г. Г., Ефимов А. Л. Г. Использование реализующих структур для оформления первичных документов по заявкам и формализованная проверка описаний к изобретениям // Вопросы изобретательства. — 1987. — № 3. — С. 30—33.
2. Сарьян В. К., Корытов В. В. Увеличение времени жизни диалоговой системы по оформлению первичных документов по заявкам // Микропроцессорные средства и системы. — 1987. — № 4. — С. 40—41.
3. Блинные В. И., Минашин В. П., Бедин Б. Д., Сарьян В. К. Система безбумажной технологии делопроизводства по заявкам на изобретения // Там же. — С. 42—43.
4. Блинные В. И., Александров Л. В., Ерофьева С. Б. и др. Методика построения структурных матриц. — М.: ВНИИГПЭ, 1983.
5. «Известия». — 1988. — 15 января.
6. Красносельский И. И. Повышение помехоустойчивости передачи неподвижных изображений посредством сверточного кодирования // Тез. докл. XXXVIII Всесоюз. науч. сессии, посвященной Дню Радио // НТОРЭС им. А. С. Попова. — М.: Радио и связь, 1983, ч. 1.
7. МККР. XVI Пленарная Ассамблея. Дубровник, 1986. Рекомендация 653: Системы телетекста // Зеленая книга. — Женева. — 1987. — Т. XI.

Статья поступила 12.07.88.

Л. А. Григорьев, А. В. Прокопчик, Б. А. Разживин

## ТЕХНОЛОГИЧЕСКАЯ ПОДГОТОВКА ПРОИЗВОДСТВА С ПОМОЩЬЮ ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ «ДЕТАЛЬ»

В условиях динамичного развития машиностроительного производства, сокращения сроков технологической подготовки производства (ТПП) нередко целесообразно использовать готовые технологические решения: техпроцессы-аналоги и унифицированные (типовые и групповые) технологические процессы (УТП). Кроме того, в начальной стадии ТПП новых изделий необходимо иметь возможность проводить с достаточной степенью точности экспресс-анализ различных параметров готовящегося производства, например структурного состава и числа единиц оборудования, размеров производственных площадей, материально-технических ресурсов, затрат на технологическое проектирование и т. п.

Для такого анализа требуются сведения о геометрических, точностных, весовых и других конструкторско-технологических параметрах каждой детали, чтобы иметь возможность группировать их для разработки УТП, адресовать новые детали к определенным группам, на которые уже имеются УТП, или определять деталь-аналог и соответственно техпроцесс-аналог.

Информация о детали должна быть представлена в обезличенном закодированном виде, чтобы не работать с чертежом и ускорить процесс обработки информации. Для полного описания данных о детали требуется 20...25 показателей, поэтому ручная обработка информации очень трудоемка, а часто и невозможна. Использование информационно-поисковой системы (ИПС) «Деталь» позволило значительно сократить время, затрачиваемое на ТПП, увеличить производительность труда технолога, повысить качество технологических разработок, организовать работу по унификации деталей и их элементов, получить оценку таких параметров производства, которые оценивались раньше исходя только из опыта и интуиции, повысить привлекательность труда технолога.

Для кодирования данных о детали используется «Иллюстрированный определитель деталей общемашиностроительного применения», на основании которого формируется шестизначный конструкторский код детали, и «Технологический классификатор деталей машиностроения и приборостроения» для формирования четырнадцатизначного технологического кода детали.

В конструкторском коде используется пять классификационных группировок (класс, подкласс, группа, подгруппа, вид). В качестве основных классификационных признаков для деталей выбраны геометрическая форма, конструктивные характеристики отдельных элементов, взаимное расположение элементов, параметрический признак.

В структуру технологического кода детали включены параметры, описывающие размеры, группу материала, вид детали по технологическому методу, вид исходной заготовки, классы точности и чистоты поверхностей, характеристики элементов зубчатого зацепления и термической обработки, масса детали и заготовки.

Кроме того, в базу данных сведены дополнительные реквизиты для обеих частей кода, что позволяет в существующих рамках системы расширять ее возможности. Таким образом, общий конструкторско-технологический код детали состоит из 24 реквизитов, содержащих достаточно полное описание различных параметров детали. Система реализована на ПЭВМ «Искра 226». Основным пользователем системы является инженер-технолог.

Начиная работать с ИПС, пользователь определяет те параметры детали, по которым будет организован поиск. Затем для каждого выбранного параметра задает режим

поиска и устройство ввода-вывода (экран дисплея или АЦПУ). В ИПС «Деталь» реализованы следующие режимы поиска, связанные с определением параметра: равен значению; равен выделяемой части; больше или равен минимальному значению, но меньше или равен максимальному значению; не равен значению; не равен выделяемой части; больше минимального, но меньше максимального значения; больше значения; равен контексту.

Для вывода информации предусмотрены различные выходные формы, вид которых зависит от характера составленного запроса. В форме типа «паспорт» выдаются все параметры детали. В форме типа «таблица» выводится ряд основных параметров, а в сокращенной форме — цифровое обозначение и наименование детали. По требованию пользователя можно проектировать любые виды выходных форм, достаточно указать номер нужной формы. В дальнейшем можно изменить набор параметров и режимы поиска и выполнить новый запрос.

С помощью ИПС «Деталь» можно формировать описание параметров, вводить и корректировать классификаторы, разрабатывать формы вывода и их описания, вводить данные в информационные массивы; корректировать информацию в массиве, организовывать поиск информации с выводом на печать, формировать инвертированные списки, обеспечивать удобство в работе с блоками пользователя, вводить и выводить информацию с магнитного носителя. Все работы выполняются технологом в диалоговом режиме.

При разработке программного обеспечения использовались язык программирования БЕИСИК-02 редакции от 10.10.84 г. и система ведения локальных баз данных (ВЕЛОБАД) редакции 2.

Максимальный объем базы данных ИПС «Деталь» составляет 25000 записей, максимальная длина записи — 1024 байт, максимальное число реквизитов — 255. Тип данных: символьные и десятичные. Время поиска одновременно по 3, 4 атрибутам при объеме БД, равном 500 записям, составляет 1...2 мин.

Телефон: 29-11-23, 29-11-98, г. Тула

Статья поступила 02.08.88

УДК 681.3.071

А. В. Покрепо

## СИСТЕМА «ЛОТЕРЕЯ»: ИСПОЛЬЗОВАНИЕ МИКРОЭВМ В УЧЕБНОМ ПРОЦЕССЕ ВУЗА

Система «Лотерея» предназначена для проверки знаний студентов по ряду предметов и применяется, в частности, для автоматизированной защиты лабораторных работ в курсе «Программирование на Фортранс» кафедры математики, в курсах «Механика», «Электричество», «Оптика» кафедры физики, в курсе «Горное оборудование» кафедры горных машин.

Система «Лотерея» представляет собой программу, составленную на символьном языке микроЭВМ «Искра». Необходимые технические средства для работы с программой: микроЭВМ «Искра 1256» с памятью не менее 64 Кбайт и встроенным кассетным накопителем на магнитной ленте, принтер «Искра 001-41» или Роботрон.

Программа обеспечивает опрос студентов по выбранной теме или лабораторной работе и автоматически выставляет оценку каждому студенту по пятибалльной системе. Предусмотрена возможность включения режима одновременного сбора статистической информации и записи на кассету.

Работа с программой складывается из трех этапов в диалоговом режиме: ввод и начальная настройка, защита лабораторных работ и завершение работы. Этапы 1 и 3

выполняет преподаватель или лаборант, этап 2 — студенты. При работе с программой выдается сводный протокол-расчетка в виде оформленной ведомости (см. таблицу).

### ПРИМЕР ВЕДОМОСТИ — РЕЗУЛЬТАТА КОНТРОЛЯ СКГМИ

Кафедра физики

Дата контроля — 15.03.88

### ВЕДОМОСТЬ ОЦЕНОК ПО ЗАЩИТЕ ЛАБОРАТОРНЫХ РАБОТ ГРУППЫ МО-88 5

Л/Р	Фамилия И. О.	Оценка
3.4	Иванов И. И.	4.0
3.6	Петров П. П.	1.5
3.9	Кузнецов А. С.	3.0

Студент в ответ на запрос вводит свою фамилию с инициалами и выбирает номер лабораторной работы, которую он будет защищать. На экране появляются вопросы, относящиеся к данной лабораторной работе и возможные варианты ответов, из которых только один является правильным.

Число вариантов ответа может быть от двух до пяти. Число вопросов в конкретной лабораторной работе может быть любым, но из методических соображений оно выбирается в пределах 5...10. В конце опроса студенту на экран выдается его оценка. Одновременно та же оценка печатается в сводной ведомости, предназначенной для преподавателя.

Оценка выставляется следующим образом. Подсчитывается сумма баллов, набранных студентом при выборе правильных вариантов ответа. Эта сумма складывается из всех вопросов, причем вопросы могут иметь одинаковый (например, единица) или различный вес в зависимости от сложности. Оценка вычисляется по следующей формуле:  $O = (2+3 \cdot V) / V$ , где  $O$  — оценка, представляющая собой целую часть выражения в правой части;  $V$  (наб) — набранная сумма баллов;  $V$  (общ) — общая сумма баллов всех вопросов.

Для повышения надежности системы порядок следования вариантов ответа к каждому вопросу меняется случайным образом для каждого отвечающего. Пропускная способность системы составляет примерно 10...12 человек в час.

Программа на символьном языке состоит из основного ядра и переменной части. В переменную часть программы входят наборы вопросов с вариантами ответов, которые даются соответствующими кафедрами. Число возможных вариантов системы, таким образом, не ограничено. Систему можно применять в любом учебном курсе, допускающем с точки зрения методики возможность такой формы контроля.

Система адаптируется к нужному курсу добавлением необходимых вопросов и вариантов ответов в фиксированные части программы на символьном языке. Эту работу может выполнить любой программист, знакомый с основами работы на микроЭВМ «Искра 1256». К преимуществам системы следует отнести и то, что для работы с ней необходимо иметь только одну или две кассеты МК-60. Программу можно загрузить в память нескольких микроЭВМ, используя только одну копию на кассете.

УДК 681.326—181.4

В. С. Горбачев, О. Д. Полубояринов

## ИСПОЛЬЗОВАНИЕ УЧЕБНОЙ МИКРОЭВМ УМПК-80М В ЛАБОРАТОРНОМ ПРАКТИКУМЕ ПО РОБОТОТЕХНИКЕ

### 1. Учебный манипулятор УМ-2

Манипулятор (рис. 1) имеет два последовательно сочлененных звена. На конце второго установлен пишущий узел, регистрирующий траекторию движения конечной (рабочей) точки при изучении алгоритмов управления роботом.

Базовая, декартова система координат (СК) совмещена с осью вращения первого звена манипулятора (рис. 2).

Из начального положения манипулятор приводится в движение с помощью шаговых двигателей, имеющих по четыре обмотки. При подаче напряжения (в определенной последовательности) создается вращающееся магнитное поле в требуемом направлении. Более подробно работа шагового двигателя изучается в других курсах [3].

Привод шагового двигателя (рис. 3, 4) включает в себя:

блок питания, преобразующий напряжение +27 В, поступающее с внешнего источника питания, в напряжение  $+5 \text{ В} \pm 10\%$  для питания ТТЛ-элементов привода;

входное устройство, служащее для гальванической развязки управляющих сигналов микроЭВМ и их согласования с логическими элементами привода. Один вход обеспечивает вращение по, а другой — против часовой стрелки;

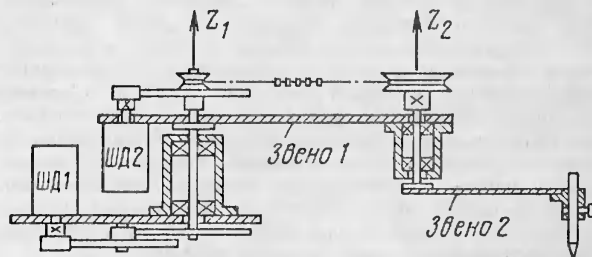


Рис. 1. Схема манипулятора

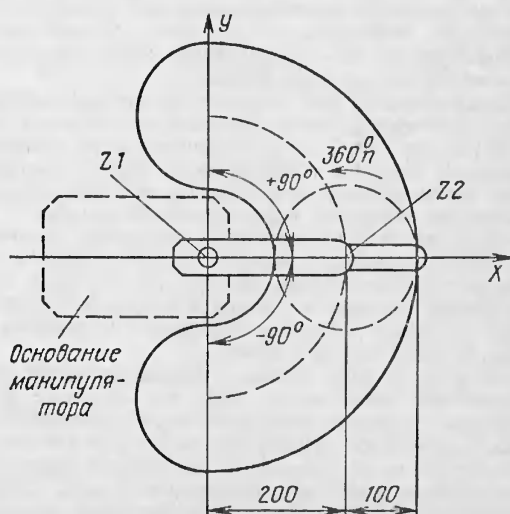


Рис. 2. Система координат и зона обслуживания

Телефон 47-4-49, г. Орджоникидзе

Статья поступила 02.08.88



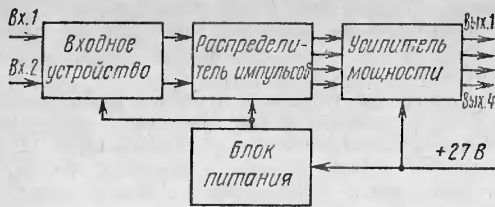


Рис. 3. Блок-схема шагового привода

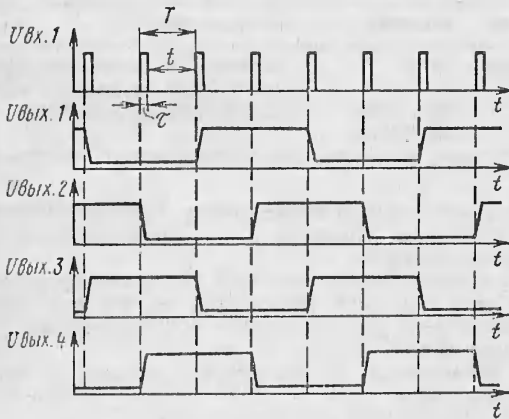


Рис. 4. Временная диаграмма привода

распределитель импульсов, преобразующий одноканальную последовательность управляющих импульсов в четырехфазную, четырехтактную, реверсивную систему сигналов; усилитель мощности, осуществляющий необходимое усиление сигналов для создания требуемого магнитного поля.

Временная диаграмма, иллюстрирующая работу привода, изображена на рис. 4.

## 2. Управление учебным манипулятором

**2.1. Общие вопросы управления учебным манипулятором**  
Укрупненная структура лабораторной установки изображена на рис. 5. Учебная микроЭВМ УМПК-80М используется в качестве управляющего устройства и через выходной регистр связана с манипулятором.

В табл. 1 приведены управляющие слова, которые требуется загружать в выходной регистр для получения перемещения в необходимом направлении.

При разработке программы, управляющей перемещением манипулятора, целесообразно пользоваться подпрограммами, имеющимися в ПЗУ микроЭВМ.

Ниже приведены описания некоторых встроженных подпрограмм. KIND — подпрограмма ввода и декодирования

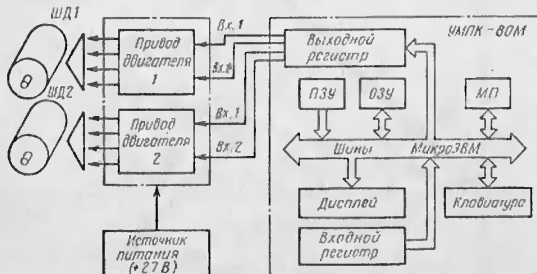


Рис. 5. Блок-схема лабораторной установки

Таблица 1

Направление вращения	Управляющее слово										
	Z1	Z2	D7	D6	D5	D4	D3	D2	D1	D0	Код (HEX)
→	0	0	0	0	0	0	0	0	0	1	01
←	0	0	0	0	0	0	0	1	0	0	04
↻	0	0	0	0	1	0	0	0	0	0	10
↺	0	0	1	0	0	0	0	0	0	0	40
↻	0	0	0	0	1	0	0	0	0	1	11
↺	0	0	1	0	0	0	0	1	0	0	44
↻	0	0	0	0	1	0	1	0	0	0	14
↺	0	0	1	0	0	0	0	0	0	1	41

числа с клавиатуры микроЭВМ. Начальный адрес 014BH. Подпрограмма ожидает нажатия клавиши.

В результате выполнения подпрограммы KIND в аккумулятор МП заносится код нажатой клавиши. Код соответствует обозначению, нанесенному на клавишу.

MPL — подпрограмма умножения двоичных однобайтовых чисел. Начальный адрес 04EH. Входные параметры: множимое — в регистре D МП; множитель — в регистре E МП; результат — в регистрах B и C МП.

BEER — подпрограмма выдачи звукового сигнала БИП. Начальный адрес 0010H.

DELB — подпрограмма регулируемой временной задержки. Начальный адрес 0430H. Входной параметр — длительность задержки (мс) — в регистрах B, C МП.

RST 1 — подпрограмма прерывания выполнения программы. Начальный адрес 0008H. При обращении к подпрограмме RST 1 прерывается выполнение текущей программы с сохранением содержимого всех внутренних регистров МП, а в PC МП заносится адрес команды, следующий за командой обращения к подпрограмме RST 1.

Нажатием кнопки «Пуск» выполнение текущей подпрограммы продолжается.

## 2.2. Управление одной степенью подвижности манипулятора

Самое простое движение — изменение конфигурации манипулятора на один шаг двигателя. Для этого необходимо сформировать соответствующее управляющее слово (см. табл. 1), управляющий импульс  $t$  и задержку  $t$  (см. рис. 4).

Управляющее слово предварительно записывается в аккумулятор ЦП, затем выполняется алгоритм программы перемещения манипулятора (рис. 6) на один шаг (STEP). Текст программы приведен в табл. 2. Для отработки заданного числа шагов подпрограмма STEP включается в программу MOVS (рис. 7, табл. 3). Скорость вращения вала ШД в данном случае постоянная и определяется величиной  $t$  (подпрограммой TIME).

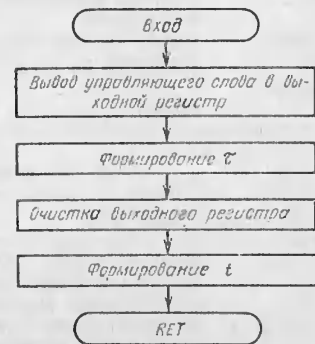


Рис. 6. Алгоритм программы поворота вала двигателя на один шаг

ТАБЛИЦА 2

МЕТКА	МНЕМОКОД	КОММЕНТАРИЙ
STEP:	OUT 30	ВЫВОД УПРАВЛЯЮЩЕГО СЛОВА В ВЫХОДНОЙ РЕГИСТР
	CALL SIGN	ФОРМИРОВАНИЕ $\tau$
	XRA A	ОЧИСТКА ВЫХОДНОГО РЕГИСТРА
	OUT, 30	
	CALL TIME	ФОРМИРОВАНИЕ $t$
	RET	
-----		
SIGN:	MVI E,---	ЗАГРУЗКА РЕГИСТРА E ПАРАМЕТРОМ
		ЗАДЕРЖКИ $\tau$
SIGN1:	DCR E	ЗАДЕРЖКА $\tau$
	JNZ SIGN1	
	RET	
-----		
TIME:	LXI B,----	ЗАГРУЗКА ПАРЫ РЕГИСТРОВ B,C ПАРАМЕТРОМ ЗАДЕРЖКИ $t$
		ЗАДЕРЖКА $t$
	CALL DELB	
	RET	

Чтобы осуществить вращение вала ШД с переменной скоростью, в ОЗУ микроЭВМ необходимо зарезервировать две ячейки скорости (задержка  $t$ ). Управляющая программа при получении очередного управляющего слова должна поместить текущее значение скорости из этих ячеек в регистры B и C. Для этого нужно заменить подпрограмму TIME на подпрограмму TIMV (как в табл. 4).

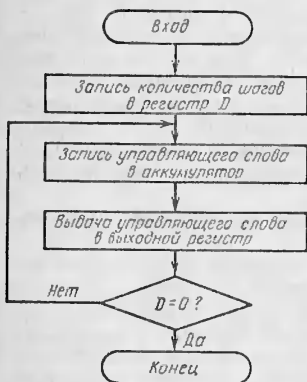


Рис. 7. Алгоритм программы поворота вала двигателя на заданный угол

ТАБЛИЦА 3

МЕТКА	МНЕМОКОД	КОММЕНТАРИЙ
MOV5:	MVI D,---	ЗАГРУЗКА В РЕГИСТР D КОЛИЧЕСТВА ШАГОВ
MOV51:	MVI A,---	ЗАГРУЗКА В АККУМУЛЯТОР УПРАВЛЯЮЩЕГО СЛОВА
	CALL STEP	ВЫДАЧА СИГНАЛА УПРАВЛЕНИЯ
	DCR D	ПОСЛЕДНИЙ ЛИ ШАГ?
	JNZ MOV51	
	RST1	СТОП

ТАБЛИЦА 4

МЕТКА	МНЕМОКОД	КОММЕНТАРИЙ
TIMV:	LXI H,---	ЗАГРУЗКА В ПАРУ РЕГИСТРОВ H,L АДРЕСА 1-Й ЯЧЕЙКИ ОЗУ, ГДЕ ХРАНИТСЯ ТЕКУЩЕЕ ЗНАЧЕНИЕ СКОРОСТИ
	MOV C, M	ЗАГРУЗКА ТЕКУЩЕГО ЗНАЧЕНИЯ СКОРОСТИ В ПАРУ РЕГИСТРОВ B,C
	INX H	
	MOV B, M	
	CALL DELB	ЗАДЕРЖКА $t$
	RET	

Заменив в программе STEP подпрограмму TIME на TIMV и организовав требуемое изменение текущего значения скорости, получим непрерывное движение манипулятора с переменной скоростью.

Часто требуется управлять манипулятором, используя пульт ручного управления или некоторое внешнее устройство, позволяющее перемещать исполнительный механизм.

Реализуется такой режим с помощью подпрограмм STEP. Для записи управляющего слова в аккумулятор можно использовать переключатели входного устройства микроЭВМ (адрес 20H).

Текст программы MANUAL приведен в табл. 5.

ТАБЛИЦА 5

МЕТКА	МНЕМОКОД	КОММЕНТАРИЙ
MANUAL:	IN 20	ЗАГРУЗКА В АККУМУЛЯТОР УПРАВЛЯЮЩЕГО СЛОВА ИЗ ВХОДНОГО РЕГИСТРА
	CALL STEP	ВЫДАЧА СИГНАЛА УПРАВЛЕНИЯ
	JMP MANUAL	

Предварительно в подпрограмму TIME необходимо занести значение параметра  $t$ , а в подпрограмму SIGN — значение параметра  $\tau$ .

Результатом выполнения этой программы будет вращение вала двигателя (двигателей) до тех пор, пока на входном регистре установлено соответствующее управляющее слово.

### 2.3. Позиционное и контурное управление учебным манипулятором

#### 2.3.1. Позиционный режим управления

Задача позиционного управления (рис. 8) заключается в изменении конфигурации манипулятора при перемещении из некоторого начального положения в заданную конечную точку без наложения каких-либо ограничений на траекторию движения рабочей точки манипулятора.

Допустим, нужно переместить звенья манипулятора из положения 1 в положение 2. Это можно осуществить относительным поворотом звеньев на углы  $\varphi_1$  и  $\varphi_2$ , которые определяются числом шагов каждого двигателя.  $\alpha_1$  и  $\alpha_2$  — соответственно углы отклонения звеньев в положении 1 относительно начального положения 0. Допустим,  $\alpha_1$  соответствует +25 шагам двигателя 1, а  $\alpha_2$  — -30 шагам двигателя 2. В положении 2 эти углы примут соответственно значения  $\alpha'_1$  — +10 шагов, а  $\alpha'_2$  — -20 шагов относительно положения 0.

В таком случае количество шагов, которое необходимо сделать, чтобы повернуть звенья на углы  $\varphi_1$  и  $\varphi_2$ , можно определить следующим образом:

$$\varphi_1 = \alpha'_1 - \alpha_1; \quad \varphi_1 = +10 - (+25) = -15;$$

$$\varphi_2 = \alpha'_2 - \alpha_2; \quad \varphi_2 = -20 - (-30) = +10.$$

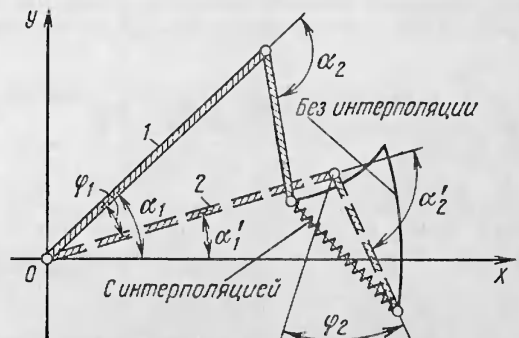


Рис. 8. Пример расчета траектории

Из приведенного примера видно, что знак полученного результата показывает направление вращения звена, следовательно, его можно поставить в соответствие с управляющим словом.

Программа, реализующая движение манипулятора по заданным точкам траектории, приведена в табл. 6. Скорость движения звеньев постоянна и определяется параметрами подпрограммы TIME. Обобщенные координаты точек  $\alpha_1$  и  $\alpha_2$ , пересчитанные в число шагов двигателей, хранятся в защищенной области ОЗУ, начиная с адреса 0A00H.

2.3.2. Управление манипулятором в режиме обучения  
Наиболее простой и естественный способ получения массива обобщенных координат  $\alpha_1$  и  $\alpha_2$  — это метод обучения. Суть его в том, что мы перемещаем манипулятор по заданным точкам, записывая в массив, организованный в ОЗУ по адресам 0A00H — 0AFFH.

Для управления манипулятором в режиме обучения используются клавиши 0...3 микроЭВМ, которым ставятся в соответствие управляющие слова 01H, 04H, 10H и 40H (табл. 1). Клавиша F используется для записи координат точек позиционирования в массив, организованный в ОЗУ по адресам 0A00H — 0AFFH.

Определить нажатую клавишу и выбрать соответствующую

ТАБЛИЦА 6

МЕТКА	МНМОКОД	КОММЕНТАРИЙ
LOOPP:	LXI SP, 0BA0 LXI B, 0A00	УСТАНОВИТЬ НАЧАЛЬНЫЙ АДРЕС СТЕКА ЗАГРУЗИТЬ В ПАРУ РЕГИСТРОВ В,С НАЧАЛЬНЫЙ АДРЕС МАССИВА ЗНАЧЕНИЙ $\alpha$
LOOPP1:	PUSH B	СОХРАНИТЬ АДРЕС ОЧЕРЕДНОГО ЗНАЧЕНИЯ $\alpha_2$ В СТЕКЕ
	LXI H, 0BA2	УСТАНОВИТЬ В ПАРЕ РЕГИСТРОВ H,L АДРЕС ТЕКУЩЕГО ЗНАЧЕНИЯ $\alpha_2$
	CALL DECL	ОПРЕДЕЛИТЬ ТРЕБУЕМОЕ ОТКЛОНЕНИЕ 2-ГО ЗВЕНА
	MOV A, E	ЗАГРУЗИТЬ В АККУМУЛЯТОР УПРАВЛЯЮЩЕЕ СЛОВО
	CALL MOVE	ВЫПОЛНИТЬ ЗАДАННОЕ ПЕРЕМЕЩЕНИЕ 2-ГО ЗВЕНА
	POP B	ПОСЛЕДНЕЕ ЛИ ЗНАЧЕНИЕ $\alpha$ ?
	CALL FIN	
	PUSH B	СОХРАНИТЬ АДРЕС ОЧЕРЕДНОГО ЗНАЧЕНИЯ $\alpha_1$ В СТЕКЕ
	DCX H	УСТАНОВИТЬ В ПАРЕ РЕГИСТРОВ H,L АДРЕС ТЕКУЩЕГО ЗНАЧЕНИЯ $\alpha_1$
	CALL DECL	ОПРЕДЕЛИТЬ ТРЕБУЕМОЕ ОТКЛОНЕНИЕ 1-ГО ЗВЕНА
	MOV A, E	ЗАГРУЗИТЬ В АККУМУЛЯТОР УПРАВЛЯЮЩЕЕ СЛОВО
	RLC	
	RLC	
	RLC	
	CALL MOVE	ВЫПОЛНИТЬ ЗАДАННОЕ ПЕРЕМЕЩЕНИЕ 1-ГО ЗВЕНА
	POP B	ПОСЛЕДНЕЕ ЛИ ЗНАЧЕНИЕ $\alpha$ ?
	CALL FIN	
	JMP LOOPP1	ПОВТОРИТЬ ДЛЯ СЛЕДУЮЩЕЙ ТОЧКИ ПОЗИЦИОНИРОВАНИЯ
DECL:	LDA B	ЗАГРУЗИТЬ В АККУМУЛЯТОР ОЧЕРЕДНОЕ ЗНАЧЕНИЕ $\alpha$
	SUB M	ОПРЕДЕЛИТЬ ЧИСЛО ШАГОВ ДВИГАТЕЛЯ ( $\varphi$ )
	JZ WAIT	ЕСЛИ $\varphi = 0$ , ТО ПЕРЕЙТИ К СЛЕДУЮЩЕМУ ЗНАЧЕНИЮ
	JM INV	ОПРЕДЕЛИТЬ НАПРАВЛЕНИЕ ВРАЩЕНИЯ ЗВЕНА
	MVI E, 04	ЗАГРУЗИТЬ В РЕГИСТР E УПРАВЛЯЮЩЕЕ СЛОВО
DECL1:	MOV D, A	ЗАГРУЗИТЬ В РЕГИСТР D ЗНАЧЕНИЕ $\varphi$ ЗАГРУЗИТЬ ОЧЕРЕДНОЕ ЗНАЧЕНИЕ $\alpha$ ПО АДРЕСУ ТЕКУЩЕГО ЗНАЧЕНИЯ $\alpha$
	LDA B	
	MOV M, A	
	RET	
	RET	

ПРОДОЛЖЕНИЕ ТАБЛИЦЫ 6

INV:	MVI E, 01	ЗАГРУЗИТЬ В РЕГИСТР E УПРАВЛЯЮЩЕЕ СЛОВО
	CMA A	ДЕШИФРОВАТЬ ЧИСЛО ШАГОВ $\varphi$
	INR A	
	JMP DECL1	
MOVE:	PUSH PSW	СОХРАНИТЬ УПРАВЛЯЮЩЕЕ СЛОВО В СТЕКЕ
	CALL STEP	ВЫДАТЬ СИГНАЛ УПРАВЛЕНИЯ В ВЫХОДНОЙ РЕГИСТР
	POP PSW	ВОССТАНОВИТЬ В АККУМУЛЯТОРЕ УПРАВЛЯЮЩЕЕ СЛОВО
	DCR D	ПОСЛЕДНИЙ ЛИ ШАГ ДВИГАТЕЛЯ ?
	JNZ MOVE	
	RET	
FIN:	INR C	ПОСЛЕДНИЙ ЛИ АДРЕС МАССИВА ? ( END- ПОСЛЕДНИЙ + 1 АДРЕС МАССИВА $\alpha$ )
	MOV A, C	
	CPI END	
	JZ RST1	
	RET	
WAIT:	LXI D, 0001	ЗАГРУЗИТЬ РЕГИСТРЫ D,E ПАРАМЕТРАМИ ОТСУТСТВИЯ ДВИЖЕНИЯ
	RET	

ний ей код управления можно, используя подпрограмму KIND, а обработать заданный шаг — при помощи подпрограмм STEP. Одновременно сделанный шаг записывается в регистр В или С. В итоге в регистре В будет записано отклонение (число шагов) от начального положения для двигателя 1, а в регистре С — для двигателя 2. Далее микроЭВМ ожидает очередного нажатия какой-либо клавиши.

При нажатии клавиши F происходит снятие защиты ОЗУ, передача информации из регистров В и С в две последовательные ячейки памяти в области 0A00H — 0AFFH и фиксируется следующий адрес памяти для записи обобщенных координат.

Алгоритм программы реализации режима обучения приведен на рис. 9.

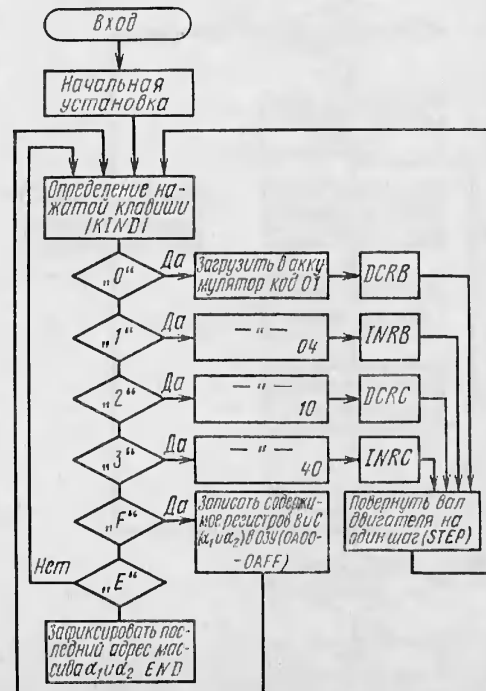


Рис. 9. Алгоритм программы, реализующий режим обучения

### 2.3.3. Контурное управление учебным манипулятором

Контурное управление в общем случае характеризуется заданием траектории и скорости на ней рабочей точки манипулятора. Как правило, траектория описывается в декартовом пространстве и для управления в этом режиме требуется выполнить обратное преобразование координат, заключающееся в пересчете точек траектории декартова пространства в обобщенные (углы поворотов звеньев).

На рис. 8 обозначены декартовы  $(X_1, Y_1)$  и обобщенные координаты  $(\alpha_1, \alpha_2)$  для начальной точки А линейной траектории, а также  $(X_2, Y_2)$  и  $(\alpha'_1, \alpha'_2)$  для конечной точки В.

Например, перемещение по прямой от точки А к точке В может быть рассчитано следующим образом.

Прямая АВ разбивается на  $p$  равных отрезков с  $p - 1$  узловыми точками. Для каждой точки определяют значения декартовых, а затем обобщенных координат. Накладывается условие, чтобы манипулятор перемещал рабочий орган между двумя соседними точками на прямой АВ за один и тот же интервал времени.

Если  $p$  достаточно велико, то манипулятор, проходя по заданным точкам, будет обрабатывать прямолинейное движение с необходимой скоростью и точностью.

Выполнение обратного преобразования координат требует наличия в микроЭВМ подпрограмм с обратными тригонометрическими функциями, т. е. нужен значительный объем памяти.

Поэтому в лабораторном практикуме исследуется линейная интерполяция в обобщенных координатах, которая требует только операции деления.

Общий алгоритм интерполяции можно описать последовательностью действий.

1. Определяем координату, по которой осуществляется меньшее количество шагов. Она будет называться основной.

2. Определяем, сколько шагов по неосновной координате приходится на один шаг по основной.

3. Обработаем один шаг по основной координате и соответствующее число шагов по другой координате.

4. Если обработаны не все шаги по основной координате, то переходим на п. 3.

5. Выход.

Следует отметить, что остаток от деления числа шагов по неосновной координате на число шагов по основной учитывается на первых интервалах интерполяции в виде добавления по одному шагу по неосновной координате соответствующее число раз.

Например,  $\varphi_1=9$ ;  $\varphi_2=4$ , тогда:  $9/4=2+1/4$ , остаток равен единице, на первом интервале интерполяции по неосновной координате будет отработано три шага, на остальных  $3^x$  интервалах — по два.

ТАБЛИЦА 7

МЕТКА	МНЕМОКОД	КОММЕНТАРИЙ
LOOPC:	LXI SP, OBA0 LXI B, OAO0	УСТАНОВИТЬ НАЧАЛЬНЫЙ АДРЕС СТЕКА ЗАГРУЗИТЬ В ПАРУ РЕГИСТРОВ B,C НАЧАЛЬНЫЙ АДРЕС МАССИВА ЗНАЧЕНИЙ $\alpha$
LOOPC1:	PUSH B LXI H, OBA2 CALL DECL MOV A, E STA OBA4 MOV A, D STA OBA5 POP B CALL FIN PUSH B	СОХРАНИТЬ АДРЕС ОЧЕРЕДНОГО ЗНАЧЕНИЯ $\alpha_2$ В СТЕКЕ УСТАНОВИТЬ В ПАРЕ РЕГИСТРОВ H,L АДРЕС ТЕКУЩЕГО ЗНАЧЕНИЯ $\alpha_2$ ОПРЕДЕЛИТЬ ТРЕБУЕМОЕ ОТКЛОНЕНИЕ 2-ГО ЗВЕНА ЗАГРУЗИТЬ В ЯЧЕЙКУ ПАМЯТИ OBA4H УПРАВЛЯЮЩЕЕ СЛОВО ДЛЯ 2-ГО ЗВЕНА СОХРАНИТЬ В ЯЧЕЙКУ ПАМЯТИ OBA5H ЧИСЛО ШАГОВ ДВИГАТЕЛЯ 2-ГО ЗВЕНА ПОСЛЕДНЕЕ ЛИ ЗНАЧЕНИЕ $\alpha$ ?
	DCX H CALL DECL MOV A, E RLC RLC RLC STA OBA3 LDA OBA5 MOV E, A LDA OBA6	СОХРАНИТЬ АДРЕС ОЧЕРЕДНОГО ЗНАЧЕНИЯ $\alpha_1$ В СТЕКЕ УСТАНОВИТЬ В ПАРЕ РЕГИСТРОВ H,L АДРЕС ТЕКУЩЕГО ЗНАЧЕНИЯ $\alpha_1$ ОПРЕДЕЛИТЬ ТРЕБУЕМОЕ ОТКЛОНЕНИЕ 1-ГО ЗВЕНА ЗАГРУЗИТЬ В ЯЧЕЙКУ ПАМЯТИ OBA3H УПРАВЛЯЮЩЕЕ СЛОВО ДЛЯ 1-ГО ЗВЕНА
	MVI B, 00 CMP B JZ LMOV XRA A CMP D JZ LMOV CMP E JZ LMOV CALL PSTER	ЗАГРУЗИТЬ В РЕГИСТР E ЧИСЛО ШАГОВ ДВИГАТЕЛЯ 2-ГО ЗВЕНА ЗАГРУЗИТЬ В АККУМУЛЯТОР УКАЗАТЕЛЬ РЕЖИМА ДВИЖЕНИЯ ЕСЛИ ДВИЖЕНИЕ БЕЗ ИНТЕРПОЛЯЦИИ, ТО ВЫПОЛНИТЬ ДВИЖЕНИЕ К ЗАДАННОЙ ТОЧКЕ ЕСЛИ КОЛИЧЕСТВО ШАГОВ ОДНОГО ИЗ ДВИГАТЕЛЕЙ РАВНО НУЛЮ, ТО ВЫПОЛНИТЬ ДВИЖЕНИЕ К ЗАДАННОЙ ТОЧКЕ
LOOPC2:	PUSH D CALL FSTEP CALL TWOMOV POP D XRA A CMP E JZ CNT DCR E	СОФОРМИРОВАТЬ БАЗОВЫЕ ПАРАМЕТРЫ ИНТЕРПОЛЯЦИИ ОПРЕДЕЛИТЬ КОЛИЧЕСТВО ШАГОВ НА ИНТЕРВАЛЕ ИНТЕРПОЛЯЦИИ ВЫПОЛНИТЬ ПЕРЕМЕЩЕНИЕ
CNT:	DCR B JNZ LOOPC2 JMP LOOPC3	ЕСЛИ ПЕРЕМЕЩЕНИЕ НА ИНТЕРВАЛЕ ИНТЕРПОЛЯЦИИ ВЫПОЛНЕНО, ТО УМЕНЬШИТЬ СЧЕТЧИК ИНТЕРВАЛОВ ИНТЕРПОЛЯЦИИ НА ЕДИНИЦУ ВЫПОЛНЕНО ЛИ ДВИЖЕНИЕ ПО ВСЕМ ИНТЕРВАЛАМ ИНТЕРПОЛЯЦИИ

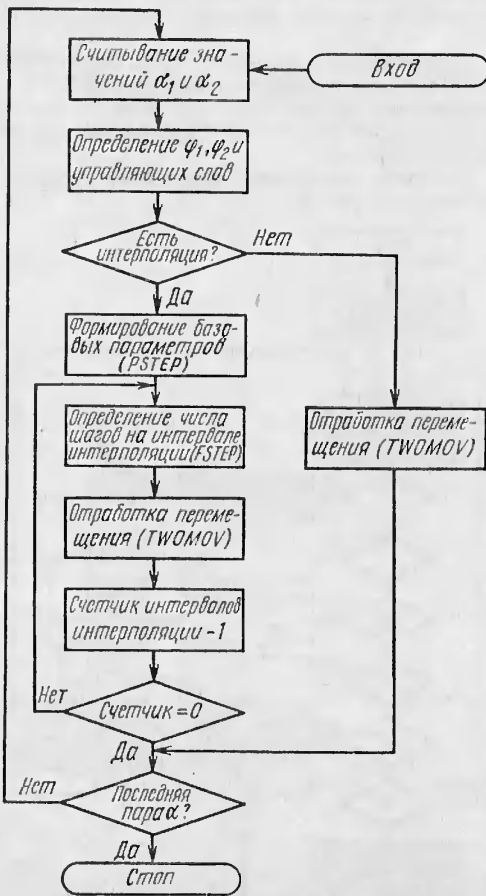


Рис. 10. Алгоритм программы, реализующий контурный режим управления

ПРОДОЛЖЕНИЕ ТАБЛИЦЫ 7

LMOV: CALL TWOMOV	ПОСЛЕДНИЙ ЛИ АДРЕС МАССИВА ?
LOOPC3: POP B	
CALL FIN	
JMP LOOPC1	
.....	
TWOMOV: LDA OBA3	ПОВТОРИТЬ ДЛЯ СЛЕДУЮЩЕЙ ТОЧКИ ПОЗИЦИОНИРОВАНИЯ
CALL MOVE	ЗАГРУЗИТЬ В АККУМУЛЯТОР УПРАВЛЯЮЩЕЕ СЛОВО ДЛЯ 1-ГО ЗВЕНА
LDA OBA4	ВЫПОЛНИТЬ ПЕРЕМЕЩЕНИЕ 1-ГО ЗВЕНА
MOV D, E	ЗАГРУЗИТЬ В АККУМУЛЯТОР УПРАВЛЯЮЩЕЕ СЛОВО ДЛЯ 2-ГО ЗВЕНА
CALL MOVE	ЗАГРУЗИТЬ КОЛИЧЕСТВО ШАГОВ ДВИГАТЕЛЯ 2-ГО ЗВЕНА В РЕГИСТР D
RET	ВЫПОЛНИТЬ ПЕРЕМЕЩЕНИЕ 2-ГО ЗВЕНА

Для управления в режиме интерполяции используется программа LOOPC. Эта программа за исключением ряда дополнений соответствует программе LOOPP. Режим движения задается содержимым ячейки с адресом OBA6.

Алгоритм программы управления манипулятором с интерполяцией в обобщенных координатах приведен на рис. 10, а текст — в табл. 7.

Подпрограмма PSTEP обращается к подпрограмме деления двух положительных чисел, определяет основную координату и число шагов по неосновной координате.

Подпрограмма FSTEP определяет величину перемещения по каждой координате между двумя узлами интерполяции.

3. Последовательность выполнения работы

В ОЗУ микроЭВМ вводятся: программы SIGN, TIME, STEP, MANUAL, задаются значения  $t$  и  $i$ . Используя переключатели входного устройства, микроЭВМ выводит нищущий узел в начальную точку;

программа MOV5. Исследуются движения манипулятора с фиксированной скоростью.

Используя подпрограмму TIMV, исследуются режимы перемещения звеньев с переменной скоростью.

В ОЗУ вводятся программа LOOPP, составленная по рис. 9 и исследуется режим обучения.

Вводятся программы для контурного движения. Исследуется движение с интерполяцией в обобщенных координатах.

Каждому обучающемуся предлагается начертить определенную геометрическую фигуру с заданными параметрами.

В целом, описанный лабораторный практикум требует около восьми часов учебного времени, причем студенты сами вводят и отлаживают программы.

Эффективность и наглядность обучения могут быть значительно повышены, если УМПК-80М будет дополнена контроллером кассетного магнитофона; дополнительным блоком ОЗУ с долговременным хранением информации на 4 Кбайт; встроенными подпрограммами тригонометрических функций: SINX, COSX, TGX, ARCTGX; подпрограммой деления целых чисел с остатком; подпрограммой нахождения  $\sqrt{X}$ .

Телефон 289-51-75, Москва

ЛИТЕРАТУРА

1. Микропроцессоры. Кн. 3. Средства отладки, лабораторный практикум и задачник: Учеб. для вузов / Н. В. Воробьев, В. Л. Горбунов, А. В. Горячев и др.; Под ред. Л. Н. Преснухина. — М.: Высшая школа, 1986. — 351 с.
2. Преснухин Л. Н., Панфилов Д. И., Романенко О. А., Шарони С. Г. Микропроцессорная лаборатория по изучению микропроцессорных комплексов с фиксированным набором команд // Микропроцессор-

ные средства и системы — 1985. — № 1. — С. 77—81.

3. Дискретный электропривод с шаговыми двигателями / Под общ. ред. М. Г. Чиликина — М.: Энергия, 1981. — 624 с.
4. Горбачев В. С., Полубояринов О. Д. Управляющие микропроцессорные устройства роботов: Учебное пособие. — М.: Высшая школа, 1987.

Статья поступила 23.01.88

УДК 681.3.06—181.4

В. С. Горбачев, О. Д. Полубояринов, М. В. Зуев, А. В. Гительсон

MINI-VAL: ЯЗЫК АВТОНОМНОГО ПРОГРАММИРОВАНИЯ УЧЕБНОГО РОБОТА «РОБКО-01»

В настоящее время во всем мире широкое распространение получили учебные мини-роботы. Для обучения они более целесообразны, чем промышленные роботы (ПР), так как обеспечивают полную безопасность, имеют меньшие габаритные размеры, массу и стоимость.

Один из их недостатков, как правило, — отсутствие адекватности при программировании и управлении учебными и промышленными роботами. Это объясняется тем, что для учебных роботов в качестве управляющих устройств (УУ) используются (ПЭВМ), в то время как ПР имеют специализированные УУ и пульта управления, ориентированные на данный ПР и определенный класс задач, для учебных роботов эти же средства очень упрощены (с ограниченными функциями и возможностями). Аппаратные и структурные ограничения УУ учебных роботов приводят к тому, что реализации промышленных языков и систем программирования ПР затруднительна и требует значительных финансовых затрат.

Аналогичные недостатки присущи учебному мини-роботу «Робко-01» (им управляет ПЭВМ «Правец-82»). Язык ROBASIC содержит мало инструкций и не позволяет в полной мере имитировать реальные режимы управления. Для обучения используется клавиатура микроЭВМ с самыми элементарными функциями задания движений — в обобщенной системе координат.

Для расширения возможностей манипулятора «Робко-01» разработаны интерфейс сопряжения «Робко-01» с ДВК2М и пультом ручного управления от УУ Сфера-36, а также создан язык программирования Mini-VAL, представляющий

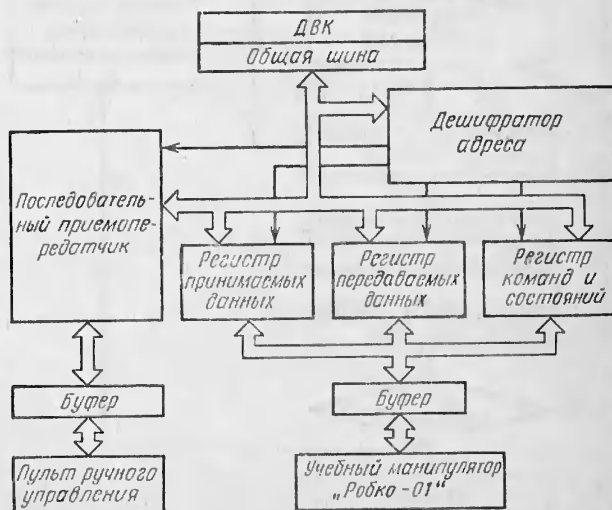


Рис. 1. Функциональная схема интерфейса сопряжения ДВК2М и «Робко-01»

собой выборку наиболее часто используемых и важных операторов распространенного языка программирования PP VAL. Версия этого языка используется в серийном отечественном роботе РМ-01 с УУ Сфера-36.

С помощью этих аппаратных и программных средств можно изучать все основные режимы работы робота РМ-01, используя «Робко-01», писать и отлаживать программы на языке Mini-VAL, исследовать и изучать вопросы управления роботами без дорогостоящего ПР.

Интерфейс (рис. 1) сопряжения ДВК2М и «Робко-01» представляет собой плату «Электроника 60». На плате расположены дешифратор, буферные регистры данных, адресов и состояний, усилители и модуль последовательного приемопередатчика для связи с пультом ручного управления.

Язык программирования Mini-VAL реализован как интерпретатор.

1. Операторы определения положения:  
 HERE {#} — запись текущего положения манипулятора;  
 LOCATE <имя точки> = <имя точки> — присвоение значения точки;  
 SHIFT <положение> <параметры> — сдвиг системы координат.

2. Операторы ветвления и переходов:  
 CALL <имя программы> — передача управления;  
 RETURN <параметры> — возврат из подпрограммы;  
 JUMP <метка> — переход внутри программы;  
 IF <оператор отношения> THEN <метка> — условный переход.

3. Операторы управления манипулятором:  
 GO <имя точки> — перемещение в точку с интерполяцией в обобщенных координатах;  
 GOS <имя точки> — перемещение в точку с линейной интерполяцией;  
 GONEAR <имя точки>, <расстояние> — перемещение схвата на требуемое расстояние от заданной точки;  
 GOREADY — перемещение в начальное положение;  
 MOVE <параметры> — перемещение с интерполяцией в обобщенных координатах и задание движений в базовой СК;

MOVES <параметры> — задание и перемещение в базовой СК;

MOVE JOINT <сочленения>, <угол> — изменение одной обобщенной координаты;

TMOVES <параметры> — перемещение с заданием движения и линейной интерполяцией в СК схвата.

4. Операторы установки скорости:  
 SPEED <величина скорости> — установка базовой скорости;

SPEED NEXT <величина скорости> — установка скорости текущего перемещения.

5. Операторы управления схвата:

CLOSE — закрыть схват;

OPEN — открыть схват.

6. Операторы общего назначения:

SET <переменная> = <арифметическое выражение> — присвоение значения переменной;

PRINT <текст> — вывод на дисплей;

C — <текст> — комментарий.

На рис. 2. приведена укрупненная блок-схема интерпретатора языка Mini-VAL.

Система программирования учебного мини-робота «Робко-01» с языком Mini-VAL реализована в ОС RT-11 V5.02 и написана на языке PASCAL. При необходимости набор операторов языка можно расширить до полной версии VAL, а также реализовать связи учебного манипулятора и адаптировать язык к любой отечественной персональной ЭВМ.

Телефон 289-51-75, Москва

Статья поступила 23.01.88

УДК 681.327

К. В. Львов, И. А. Гирба

## ПОДКЛЮЧЕНИЕ К ЭВМ «ЭЛЕКТРОНИКА ДЗ-28» УСТРОЙСТВ С ПОСЛЕДОВАТЕЛЬНЫМ ИНТЕРФЕЙСОМ ИРПС

Для подключения к ЭВМ «Электроника ДЗ-28» (далее ДЗ-28) периферийного оборудования, принимающего и передающего информацию по последовательному интерфейсу [1], необходимо иметь приспособление, которое преобразует 8-разрядный код в последовательный и наоборот. Такие устройства иногда выполняются на микросхемах средней интеграции с фиксированным алгоритмом обмена [2, 3]. В них не предусмотрена возможность изменять формат данных, проверка паритета, число стоповых битов для различных последовательных устройств ввода-вывода (УВВ).

Другой вариант — использовать специальную программируемую БИС, не имеющую ограничений по перечисленным выше параметрам. Схемы ИРПС [4—6] реализованы на БИС универсального синхронно-асинхронного приемопередатчика (УСАПП) с управлением от микропроцессора, но БИС УСАПП можно использовать и в системе с ЭВМ (рис. 1).

На элементах D1, D2 выполнен буферный каскад, подключающий двунаправленную шину данных микросхемы D13 к двум однонаправленным шинам входа и выхода ЭВМ. Микросхемы D3, D4 уменьшают нагрузку на шину управления ЭВМ. Генератор (D5.3, D5.4, D5.5) вырабатывает импульсы с периодом 6,51 мкс для синхронизации приемника и передатчика D13. Скорость обмена — 9600 бит/с. Микросхема D6 — дешифратор старшей тетрады Y13 шины управления ЭВМ. С помощью D6 можно адресовать 16 программируемых УСАПП, т. е. подключать к каналу ЭВМ несколько устройств с интерфейсом ИРПС.

Микросхема D7 — дешифратор младшей тетрады X13 шины управления ЭВМ, выделяющей пять команд

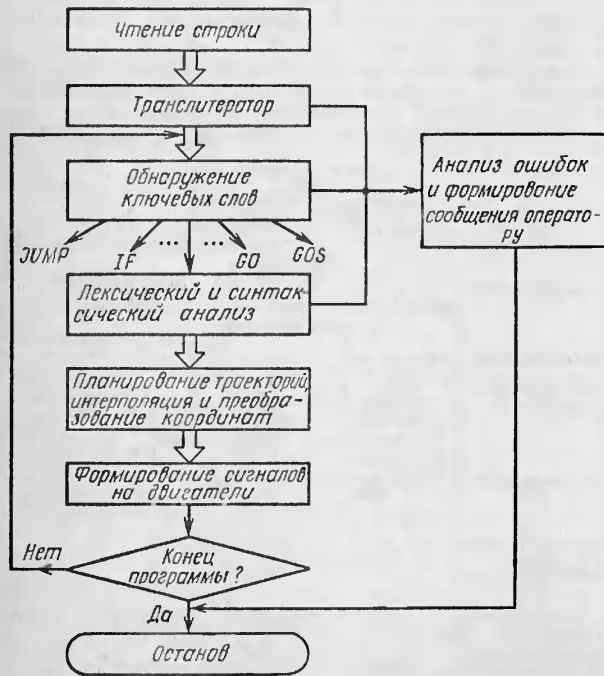
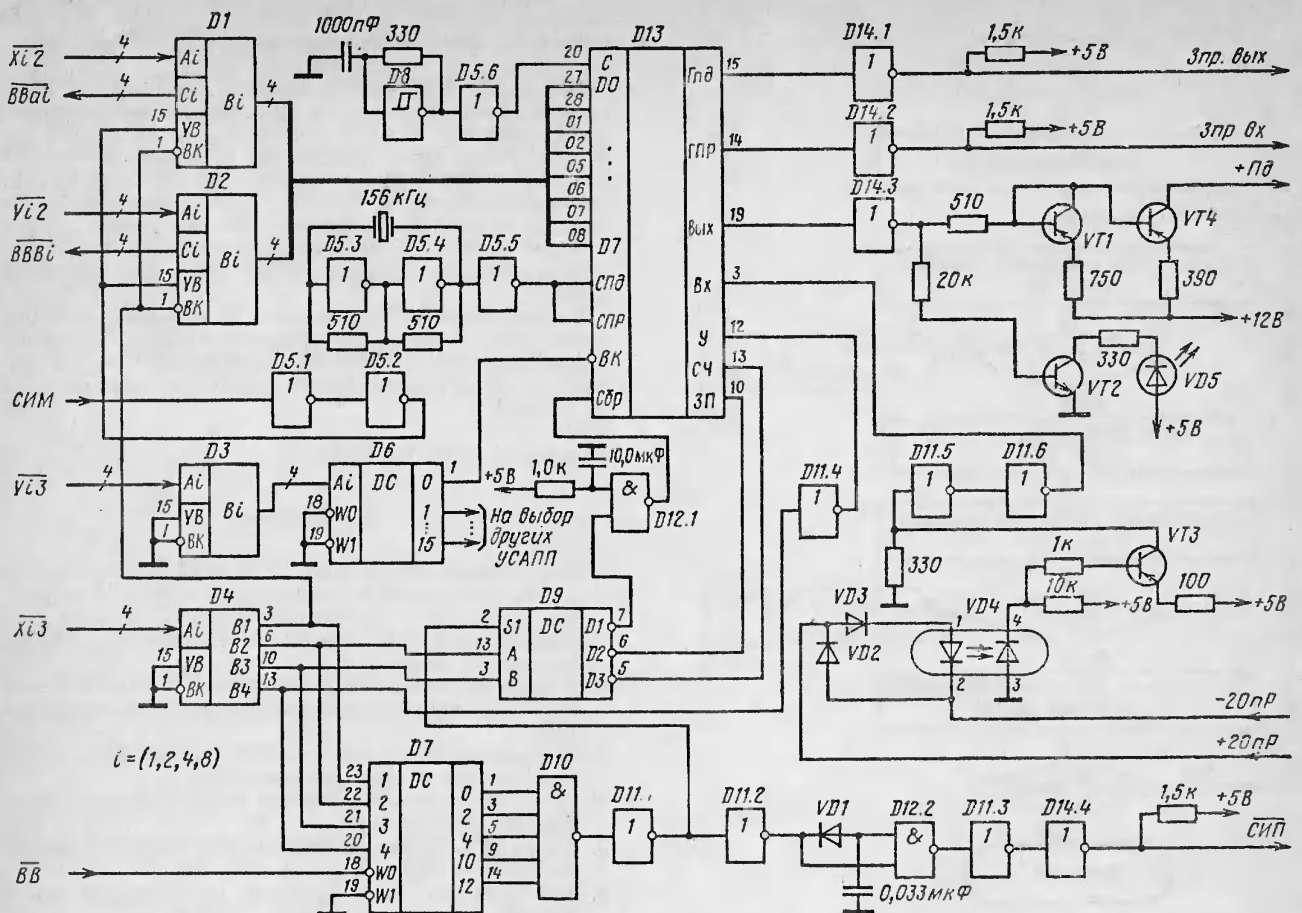


Рис. 2. Блок-схема интерпретатора языка Mini-VAL



DT...D4 - K589A116; D5, D11 - K155ЛН1; D6, D7 - K155ИД3; D8 - K155ТЛ1; D9 - K155ИД4; D10 - K155ЛА2; D12 - K155ЛАЗ; D13 - КР580ИК51; D14 - K155ЛН5; VD1...VD3 - КД521Г; VD4 - АОД101Д; VD5 - АЛ307ВМ; VT1, VT3, VT4 - КТ502; VT2 - КТ503

Рис. 1. Схема приспособления на базе БИС УСАПП КР580ИК51 для подключения к микроЭВМ «Электроника ДЗ-28» устройств с ИРПС

при обращении к приспособлению, при которых вырабатывается сигнал подтверждения приема информации СИП в ЭВМ.

На микросхеме задержки подтверждения приема СИП, D11.3, D14.4 улучшают форму импульса СИП, причем микросхема D14.4 — с открытым коллектором, т. е. предусмотрена возможность объединения этих сигналов от нескольких устройств по принципу монтажного ИЛИ. Микросхема D8 — генератор, вырабатывающий импульсы (1,6 МГц) для синхронизации работы УСАПП. Элемент D9 обеспечивает обращение к УСАПП при чтении, записи информации, при установке D13 в исходное состояние.

Код на шине управления младшей тетрады X̄i3 используется для выбора шинных формирователей D1, D2; код на шине X83 — для обращения к соответствующим регистрам УСАПП.

Данные передаются и принимаются по «Токовой петле 20 мА» с использованием на приемной стороне оптронной развязки VD4. Передача индицируется на светодиоде VD5.

Транзисторы VT1, VT4 образуют генератор выходного тока. Режим работы КР580ИК51 программируют запись управляющих байтов из ЭВМ [7]. Коды «ДЗ-28» по шине данных Ȳi2, X̄i2 при настройке УСАПП — 0101 (Режим) и 1308 (Команда). УСАПП настраивают на асинхронный

0000	0413 1008	CLR R8	ОАНУЛИТЬ РЕГИСТРЫ R8 И R12
00002	0413 1012	CLR R12	
00004	1308 0315	MOV#0315, S8	ЗАГРУЗИТЬ В S8 КОЛИЧЕСТВО ЦИКЛОВ ДЛЯ АНАЛИЗА СОСТОЯНИЯ ПЕРЕДАТЧИКА
00006	1106 0810	SWA R8, R10	ЗАГРУЗИТЬ В R10 АДРЕС РАБОЧЕЙ ЯЧЕЙКИ
00009	1408 0912	SOBZ R12	ЕСЛИ ПЕРЕДАТЧИК НЕ ГОТОВ, ПЕРЕЙТИ НА ИНДИКАЦИЮ
00010	1400 0111	INPD 0111	СЧИТАТЬ СЛОВО СОСТОЯНИЯ ПЕРЕДАТЧИКА
00012	1415 0710	BB15 7, 6, R10	МЛАДШИЙ РАЗРЯД РАВЕН 0 ?
00014	1402 0007	BR 00008	ЕСЛИ НЕ РАВЕН , ПОВТОРИТЬ АНАЛИЗ
00016	1403 0006	BR 00023	ЕСЛИ ПЕРЕДАТЧИК ГОТОВ , ПЕРЕЙТИ НА ВЫВОД БАЙТА ДАННЫХ
00018	0705 0705	DIG 5 , DIG 5	ИНДИКАЦИЯ: ПЕРЕДАТЧИК НЕ ГОТОВ
00020	0515	STOP	
00021	1402 0004	BR 00018	
00023	1106 0810	SWA R8, R10	ЗАГРУЗИТЬ В R10 АДРЕС БАЙТА ДАННЫХ ДЛЯ ВЫВОДА
00025	1401 0105	OUTO 0105	ВЫВОД БАЙТА ДАННЫХ
00027	1308 0315	MOV#0315, S8	ЗАГРУЗИТЬ В S8 КОЛИЧЕСТВО ЦИКЛОВ ДЛЯ АНАЛИЗА ГОТОВНОСТИ ПРИЕМНИКА
00029	1106 0810	SWA R8, R10	ЗАГРУЗИТЬ В R10 АДРЕС РАБОЧЕЙ ЯЧЕЙКИ
00031	1408 0912	SOBZ R12	ЕСЛИ ПРИЕМНИК НЕ ГОТОВ , ПЕРЕЙТИ НА ИНДИКАЦИЮ
00033	1400 0111	INPD 0111	СЧИТАТЬ СЛОВО СОСТОЯНИЯ ПРИЕМНИКА
00035	1415 1410	BB1C 6, R10	РАЗРЯД Д3=1?
00037	1402 0007	BR 00031	ЕСЛИ НЕ РАВЕН , ПОВТОРИТЬ АНАЛИЗ
00039	1403 0006	BR 00046	ЕСЛИ РАВЕН , ПРОДОЛЖИТЬ ИНДИКАЦИЮ: ПРИЕМНИК НЕ ГОТОВ
00041	0707 0707	DIG 7 , DIG 7	
00043	0515	STOP	
00045	1402 0004	BR 00041	
00046	1400 0103	INPD 0103	СЧИТАТЬ ОТВЕТНЫЙ БАЙТ
00048	1106 0810	SWA R8, R10	ЗАГРУЗИТЬ В R10 АДРЕС БАЙТА ДАННЫХ

Рис. 2. Программа работы ПС

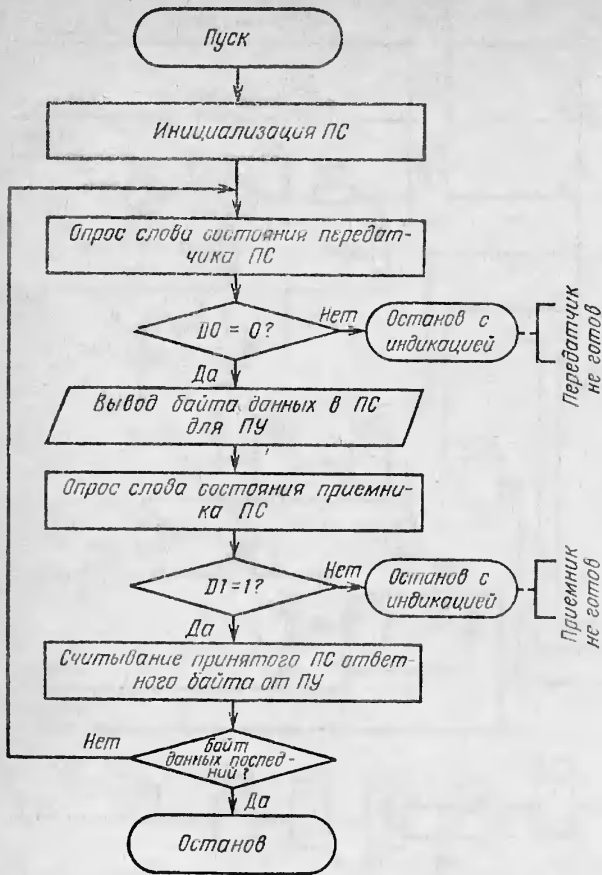


Рис. 3. Алгоритм работы приспособления ПС

режим 1:16 с двумя стоповыми битами без контроля четности, нечетности с длиной передаваемой и принимаемой информации в 8 бит. В коде команды используются биты

«Передача, прием информации возможен», «Запрос готовности приемника и передатчика».

При передаче или приеме данных «ДЗ-28» может обращаться к приспособлению по прерыванию (выводы готовности приемника УСАПП (сигнал Зпр. вх.), передатчика (сигнал К пр. вых.) подключены ко входам ПР8, ПР4 «ДЗ-28») или по опросу готовности соответствующих разрядов слова состояния УСАПП. Приспособление работает (рис. 2) в дуплексном режиме: посылка данных, прием ответной посылки данных с программным опросом «ДЗ-28» готовности передатчика (разряд 0) и приемника (разряд 1) УСАПП.

В представленном примере (рис. 3) «ДЗ-28» — источник информации, а периферийное устройство (ПУ) — приемник. Схема приспособления ИРПС к «ДЗ-28» позволяет реализовать и другие режимы работы изменением кодов настроек УСАПП.

Телефон 22-52-23, Ярославль

#### ЛИТЕРАТУРА

1. Нормативный материал НМ МПК по ВТ 10—78. Система малых электронных вычислительных машин. Интерфейс для радиального подключения устройств с последовательной передачей информации ИРПС.— М.: Изд-во стандартов, 1978.— 5 с.
2. Устройство последовательного обмена 15ВВВ-60/9600—003. Техническое описание и инструкция по эксплуатации. 1981.
3. Блок интерфейсный 15КС-4—039. Техническое описание и инструкция по эксплуатации. 1983.
4. Искра 226. Блок интерфейсный функциональный Искра 015—85. Паспорт. 1984.
5. Kubart R. Technician der microrechner 13. Radifernsehen electronic.— 1978.— Н 8.— S. 505—506.
6. Shragai M. General-purpose microcomputer board Wireless World. September.— 1983.— Vol. 89.— PP. 70—72.
7. ОСТ 11.348.917—82. Микросхемы интегральные полупроводниковые. Серия КР580. Руководство по применению.

Статья поступила 25.11.87

УДК 681.326.34

В. И. Джиган

## ЛИНИЯ СВЯЗИ УПРАВЛЯЮЩЕЙ МИКРОЭВМ С МОДУЛЯМИ АНТЕННОЙ РЕШЕТКИ

При передаче информации пространственно-разнесенным объектам, например исполнительным механизмам технологического оборудования, объединенным в сеть ЭВМ, параллельный обмен информацией сопряжен с различного рода трудностями [1], которые во многом преодолеваются при организации последовательного обмена. В большинстве случаев при этом скорость передачи информации не уменьшается, поскольку передаваемая в любой момент времени информационная посылка, как правило, — это результат вычислений, проводимых при передаче предыдущей посылки.

В данной работе на примере управления антенной решеткой рассматривается организация линии связи между

микроЭВМ [2] на основе КМДП БИС серии К588 и модулями антенной решетки. Информация передается и принимается в коде Манчестер-II [1, 3] с помощью БИС контроллеров оконечных устройств мультиплексной линии связи К588ВГ6 [4]. Для этой цели в состав внешних устройств микроЭВМ вводятся передатчик (рис. 1) последовательного кода (Пр.ПК), а в состав модулей антенной решетки (АР) — цифровые устройства (рис. 2), представляющие собой приемники последовательного кода (Пр.ПК). Конфигурация Пр.ПК может претерпевать изменения, обусловленные числом элементов АР, разрядностью используемых весовых коэффициентов, конструктивным исполнением модулей (числом прием-

ников в одном модуле) или особенностями других объектов, для передачи которым управляющего воздействия можно использовать представленную линию связи.

Рассмотрим особенности передачи и приема информации в АР с широкими функциональными возможностями. В простых случаях конфигурация элементов линии связи может изменяться в сторону очевидного упрощения, обусловленного форматами конкретных передаваемых посылок. Это способствует широкому использованию представленной линии связи в качестве основы для многих применений.

Для общности Пр.ПК построен исходя из следующих предположений: в модуле (объекте управления) расположен один приемник, одновременно функционируют приемная и передающая АР с одинаковым числом излучателей и одинаково направленными диаграммами. Для кругового обзора применяются АР с переключаемыми антенными элементами. В качестве управляемых весовых коэффициентов



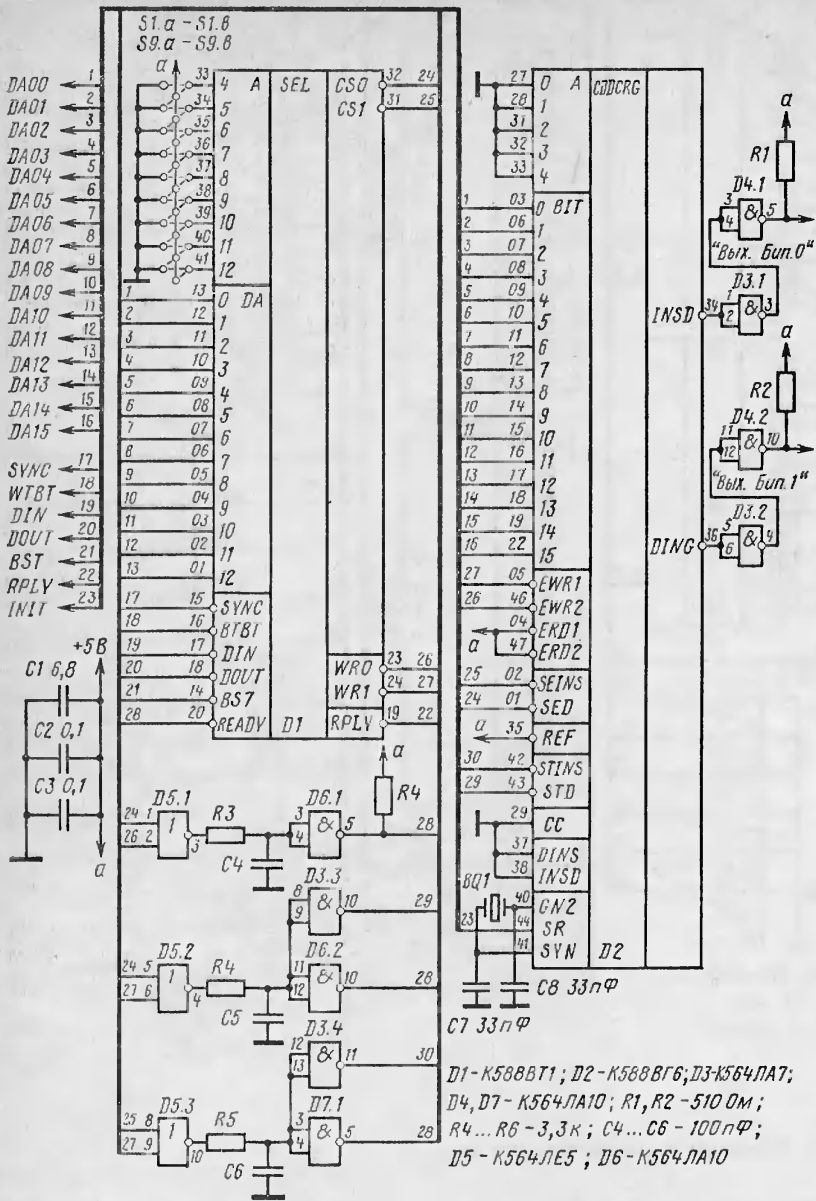


Рис. 1. Передатчик последовательного кода

используются пятиразрядные фазовращатели. К внешней помеховой обстановке AP адаптируется исходя из принципа возмущения весовых коэффициентов [5].

Эти особенности антенной системы определяют формат передаваемых микроЭВМ данных и конфигурацию Пр.ПК. Для переключения антенных элементов в каждый модуль AP передается код управления переключателями в формате

$$(1 \text{ KГ}_2\text{KГ}_1\text{KГ}_0 \text{ XXX XXX XXX XXX})_2,$$

где 1 — признак кода грани (подрешетки) с антенными элементами;  $\text{KГ}_2\text{KГ}_1\text{KГ}_0$  — трехразрядный код, обе-

спечивающий управление AP с числом граней, не превышающим восьми; X — значение разряда безразлично. Формат передачи кодов весовых коэффициентов имеет вид

$$0/1A_5A_4A_3A_2A_1A_0 \times$$

$$\times \Phi_1\Phi_2\Phi_3\Phi_4\Phi_0 \text{ XXX }_2,$$

где 0, в 15-м разряде означает, что передаются данные в виде кода весового коэффициента; 0/1 в 14-м разряде — что значение весового коэффициента невозможно или возмущено, соответственно:  $A_5...A_0$  — код адреса модуля AP;  $\Phi_1... \Phi_0$  — код весового коэффициента. Такой формат позволяет независимо формировать код весового коэф-

фициента в младшем байте передаваемого числа, а код адреса модуля с управляющими признаками — в старшем байте. Признак в 14-м разряде введен для запрещения возмущения весовых коэффициентов при приеме кода в модуле передающей AP, поскольку возмущение — источник паразитной модуляции сигналов, что в приемной адаптивной AP без доступа к сигналам в модулях AP неизбежно. Таким образом, данный формат управляющего воздействия позволяет оставить фазовое распределение передающей AP неизменным в процессе адаптации приемной AP и изменяться только при выводе нового фазового распределения, полученного в результате выполнения алгоритма адаптации или при сканировании диаграммы направленности. Число адресуемых модулей при этом не превышает 64.

Временные диаграммы работы приемника и передатчика последовательного кода представлены на рис. 3. Передатчик последовательного кода (см. рис. 1) построен на основе БИС К588ВГ6, работающей в режиме «контроллер». Этот режим обеспечивается при подаче на вход «Условие контроллер/оконечное устройство» сигнала низкого уровня. Данные, принимаемые в шины адреса-данных (DA) микроЭВМ по входам ВГ10...ВГ15 в параллельном двоичном коде, появляются на выходах «Вых. бип. 0» INSD и «Вых. бип. 1» DINS в последовательном коде Манчестер-II после запуска передатчика по входам «Пуск команд» STINS или «Пуск данных» STD. Передаваемая посылка состоит из 16 информационных битов, трех битов синхронизации и бита четности. Посылки типа «Команда» и «Данные» различаются по форме синхронизации [3]. Скорость передачи информации при частоте кварцевого резонатора BQ1 12 МГц внутреннего генератора импульсов равна 1 Мбит/с. Однако при изменении этой частоты до 20 МГц скорость передачи увеличивается почти в два раза. Для повышения нагрузочной способности выходов сигналов INSD, DINS используются элементы D4.1, D4.2 К564ЛА10 с открытым стоком.

Прием данных с шины DA микроЭВМ и запуск на передачу в последовательном коде осуществляют БИС К588ВГ6 и БИС селектора адреса (СА) К588ВТ1 (D1). СА функционирует под воздействием сигналов управления шиной DA: обмен (SYNC), признак вывода байтовой информации (WTBT), признак кода адреса внешнего устройства (BS7), ввод данных (DIN), вывод данных (DOUT), ответ (RPLY). Для передачи кода грани AP и управляющего воздействия БИС К588ВГ6 формируется посылка типа «данных». При выводе микроЭВМ данных по адресу

$$(1 \text{ 11A12 A11 A10 A9 A8} \times$$

$$\times \text{A7 A6 A5 A4 I 11X})_2,$$

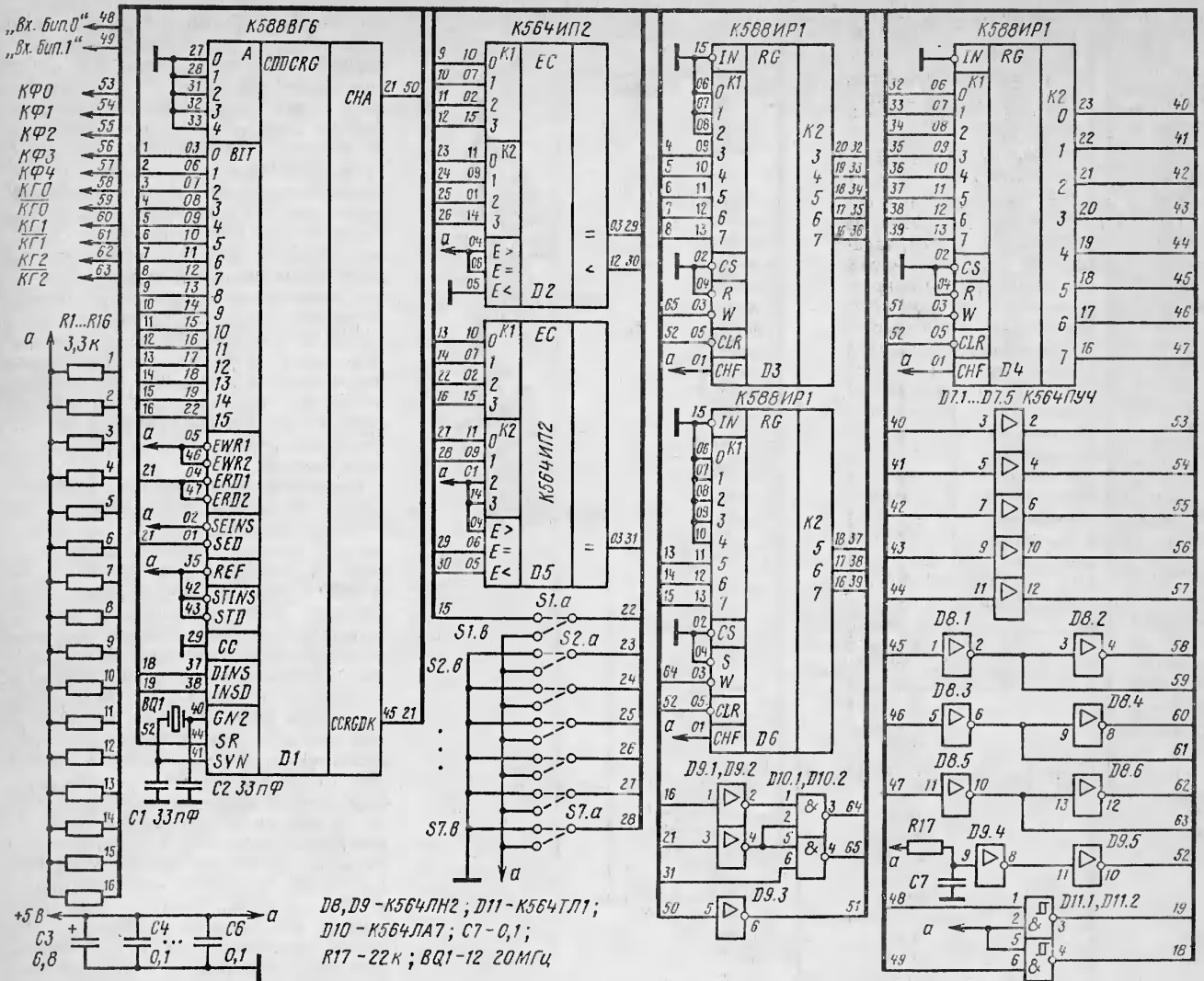


Рис. 2. Приемник последовательного кода

где A12...A4 — значения рядов шины адреса (A), БИС СА формирует на выходе выборки кристалла CS0 сигнал низкого уровня. При этом на выходах сигналов записи WR1 и WR0 вырабатываются низкие уровни при выводе полного 16-разрядного слова или на выходе WR0 — при выводе старшего байта, а на выходе WR1 — при выводе младшего байта. Сигналы CS0 поступают на вход «Выбор данных» (SED), а сигналы WR1 и WR0 — на входы «Разрешение записи» младшего (EWR1) и старшего (EWR2) байтов соответственно. По сигналам WR0, WR1, CS0 с помощью элементов D5.1, D5.2, D6.1, D6.2 формируется сигнал ответа RPLY. С помощью цепочек R3, C4 и R4, C5 формирование сигнала READY задерживается относительно сигнала DOUT на время записи информации во внутренний регистр данных БИС

K588BГ6. По окончании записи полного слова при передаче кода грани или младшего байта при передаче кода управляющего воздействия (после записи в старший байт кода адреса модуля) на выходе элемента D3.3 формируется сигнал низкого уровня, по которому запускается передатчик по входу «Пуск данных» STD. Передаваемые по линии связи данные принимаются каждым модулем AP. Сигналы «Бип. 0» и «Бип. 1» после формирования с помощью триггеров Шмитта D11.1, D11.2 и инвертирования поступают на входы INSD, DINS, соответственно, БИС K588BГ6 приемников последовательного кода (см. рис. 2). Во время дешифрации принимаемого кода на выходе «Условие незаполненности регистра декодера» CCRGDK вырабатывается сигнал высокого уровня,

который становится низким после дешифрации последнего бита послышки. Низким уровнем этого сигнала, поступающим на входы SED и «Разрешения считывания» ERD1, ERD2, разрешается считывание данных с шины BИT0...BИT15. Числа на этой шине появляются в инверсном коде по отношению к числам, представляющим данные в центральном процессоре микроЭВМ. Это объясняется тем, что шина DA микроЭВМ [2] — инверсная, а входы BИT0...BИT15 БИС K588BГ6 Прд. ПК — неинвертирующие. Поэтому во внутренний регистр данных БИС K588BГ6 информация записывается в инверсном коде и после дешифрации в приемнике считывается с выходов BИT0...BИT15 также в инверсном коде. Если принята послышка соответствует коду грани AP, то в разряде BИT15 появляется сигнал низкого уровня. По этому сигналу после

стробирования сигналом CCRGDK на элемент D10.1 записывается информация разрядов BIT14...BIT12 в регистр кода грани AP K588IP1. Во время передачи кода грани готовятся данные для передачи в первый модуль AP (штриховые линии, рис. 3).

Если принятая посылка соответствует управляющему воздействию, то старший байт посылки сравнивается с помощью схемы сравнения D2, D5 K564ИП2 с кодом адреса модуля, задаваемым перемычками S2...S7. Адрес, задаваемый на входах K2 схемы сравнения, определяется в инверсном коде. Уровень Лог.1 на входе третьего разряда шины K2 обеспечивает появление сигнала высокого уровня на выходе схемы сравнения (элемент D5) только при приеме кода управляющего воздействия. Перемычка S1 устанавливается в положение «а» в модуле передающей AP и в положение «в» — в модуле приемной AP. Поскольку на входе второго разряда шины K2 элемента D5 схемы сравнения задан уровень Лог.1, то сигнал высокого уровня на выходе схемы сравнения вырабатывается только при совпадении кодов адреса модуля независимо от состояния разряда BIT14 в приемной AP. Выходной сигнал схемы сравнения служит для записи в регистр D3 кода фазы. Код разрядов BIT7...BIT3 записывается в модуле приемной AP при любом состоянии разряда BIT14 и не записывается в модуле передающей AP при уровне разряда BIT14, равном нулю. Код в регистр D3 записывается сигналом с выхода схемы сравнения, который стробуруется сигналом CCRGDK на элементе D10.2.

После подготовки и передачи кодов управляющего воздействия во все остальные элементы AP (см. штриховые линии на рис. 3) данные с регистров D3 и D6 переписываются в выходной регистр D4. Одновременная перезапись фазового распределения во всех модулях необходима для уменьшения искажения диаграмм направленности AP. Сигнал перезаписи формируется так. При выводе кода числа

$$(1 \ 111 \ 1XX \ XXX \ XXX \ XXX)_2$$

по адресу, определяемому как

$$\begin{matrix} (1 \ 11A_{12} \ A_{11} \ A_{10} \ A_9 \ A_8 \times \\ \times \ A_7 \ A_6 \ A_5 \ A_4 \ 1 \ 100)_2. \end{matrix}$$

на выходе CS1 CA (см. рис. 1) вырабатывается сигнал низкого уровня, выводимое число записывается во внутренний регистр команд БИС K588BG6 по входам EWR1, EWR2 и «Выбор команд» SEINS, осуществляются формирование сигнала READY — с помощью элементов D5.3, D7.1, цепочки R5, C6 и запуск на передачу по входу «Пуск команд» STINS — сигналом с выхода элемента D3.4.

Принято Пр.ПК слово команды декодируется и значение его пяти старших

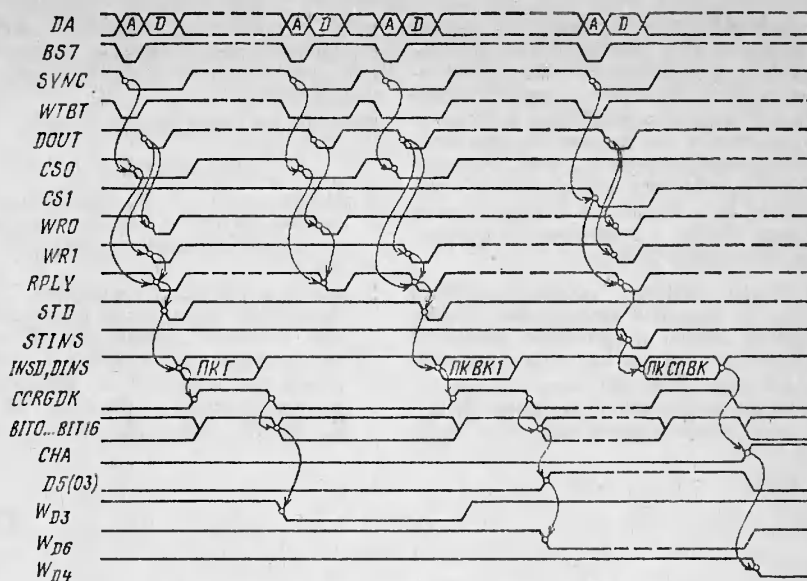


Рис. 3. Временная диаграмма работы передатчика и приемника последовательного кода.

А — адрес, D — данные, ПКГ — передача кода грани, ПК ВК — передача кода весового коэффициента 1-го модуля, ПК СП — передача кода сигнала пересанси весовых коэффициентов

битов сравнивается со значением кода, заданным на шине А БИС K588BG6 (см. рис. 2). Поскольку слово команды передается в инверсном коде аналогично слову данных, то после декодирования содержимое пяти старших разрядов регистра команд равно нулю и на выходе «Контроль адреса» СНА вырабатывается сигнал высокого уровня, так как на входах А4...А0 БИС K588BG6 заданы сигналы низкого уровня. Сигналом СНА, инвертируемым элементом D9.3, производится перезапись содержимого регистров D3 и D6 в выходной регистр D4. Выходные шины регистров Пр.ПК находятся в активном состоянии, поскольку на входах чтения R регистров присутствует сигнал низкого уровня. Поэтому сигналы на выходах регистров появляются с задержкой, равной задержке чтения после записи информации. Запись в регистры осуществляется сигналом низкого уровня по входам записи W, поскольку на входах выборки кристаллов CS постоянно присутствуют сигналы низких уровней. Перезапись кодов в регистр D4 — одновременная во всех модулях AP, так как шина А всех элементов D1 (рис. 2) коммутируется одинаково. Возможна и другая комбинация сигналов этой шины при условии передачи кода соответствующего числа в пяти старших разрядах посылки команды.

Для обеспечения входных токов ключей, управляющих дискретными фазовращателями и переключателями, сигналы с выходной шины регистра D4 усиливаются элементами D7 и D8,

соответственно. Фазовращатели и переключатели — это своего рода цифро-аналоговые преобразователи. Для управления переключателями часто требуются сигналы в прямом и инверсном кодах, формируемые элементами D8.

В исходное состояние БИС K588BG6 (работающая в режиме Пр.ПК) устанавливается сигналом INIT системной магистрали. В Пр.ПК начальная установка БИС K588BG6 и регистров K588IP1 осуществляется по включению питания сигналом с выхода формирователя (элементы D9.4, D9.5 и цепочка R17, C7).

Конфигурация Пр.ПК может изменяться в зависимости от назначения AP и ее характеристик. Число элементов AP определяет разрядность схемы сравнения; разрядность весовых коэффициентов — число используемых разрядов регистров D3, D4; пространственная форма AP (наличие или отсутствие подрешеток) — наличие или отсутствие регистра D6 и число его используемых разрядов. При размещении в одном конструктиве нескольких модулей AP возможно применение одного элемента D1 на требуемое число модулей. При всех подобных изменениях необходимо учитывать также изменение форматов передаваемых посылок.

Поскольку при связи микроЭВМ с модулями AP передача информации односторонняя, а требование гальванической развязки передатчика и приемников последовательного кода необязательно (Прд.ПК и Пр.ПК запитываются от одного источника), то в линии отсут-

ствуется трансформаторная связь, обычно применяемая в линиях связи, в которых передаются сообщения в коде Манчестер-II. Поэтому потребляемая Пр.ПК мощность резко снижается из-за отсутствия в его составе формирователя трансформаторной линии. Это особенно важно при большом числе модулей АР. Мощность, потребляемая одним Пр.ПК в динамическом режиме, не превышает 50 мВт.

Использование последовательного способа передачи информации обеспечивает низкое потребление мощности элементами линии связи, а использование кода Манчестер-II — высокую помехозащищенность этой линии. Приведенная конфигурация элементов линии

связи, а также ее возможные модификации можно применить в устройствах управления АР и других объектах управления.

Телефон 534-09-08, Москва

#### ЛИТЕРАТУРА

1. Шевкопляс Б. В. Микропроцессорные структуры. Инженерные решения. — М.: Радио и связь, 1986. — 264 с.
2. Джиган В. И., Лужецкий С. Г. МикроЭВМ управления фазированной антенной решеткой. — В кн.: Архитектура, схемотехника и математическое обеспечение микропроцессорных систем. Межвуз. сб. — М.: МИЭТ, 1986, с.131—136.

3. Williams D. Industrial controller joins the MIL—STD, 1553 BUS—Electronic Design, — 1982 — V.30. — N 21. — P. 205—211.
4. Горевой В. В., Кособрюхов В. А., Таратын И. А., Хвощ С. Т. Комплекты интерфейсных БИС серии К588 для мультиплексного канала // Электронная промышленность. — 1986. — Вып. 2. — С. 3—6.
5. Cantoni A. Application of orthogonal perturbation sequences to adaptive beamforming // IEEE Trans. on Antennas and Propagation. — 1980. — V AP-28. — N 2. — P. 191—202.

Статья поступила 12.11.87

## ОПЫТ ПРИМЕНЕНИЯ ПЭВМ «ЭЛЕКТРОНИКА БК-0010»

УДК 681.03.06

А. П. Казанцев, Л. Н. Майоров, А. Б. Данилов

### УКАЗАТЕЛЬ ИНФОРМАЦИИ И ИНТЕРФЕЙС ЦВЕТНОГО ТЕЛЕВИЗОРА ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА БК-0010»

Для реализации эффективного интерфейса пользователя с микроЭВМ «Электроника БК-0010»\* разработаны указатель информации типа джойстик и интерфейс цветного телевизора.

При диалоге пользователя с ЭВМ целесообразно применение указателя информации вместо клавиатуры. Это повышает скорость ввода информации в машину, способствует устойчивости диалога. Взаимодействие пользователя с машиной сводится к выбору из определенного меню необходимых процедур. Меню может быть организовано в виде текста из отдельных строк, обозначающих определенную процедуру, либо в виде транспарантов с информацией о назначении.

Указатель информации (УИ) — двухкоординатный механизм, задающий в прямоугольной системе координат положение видимой на экране отметки (точки, курсора). Механизм, расположенный в корпусе, сопряжен с ручкой и содержит два потенциометра. В зависимости от положения ручки меняются значения сопротивлений, которые по электронной схеме преобразуются в электрические сигналы координат указания (рис. 1). При манипуляции ручкой машина получает пропорциональные сопротивлениям потенциометров координаты для указания объектов, при нажатии кнопки на ру-

коятке поступает сигнал для отметки выбранного объекта на экране. Двухкоординатный механизм и плата электроники крепятся к массивному основанию, где также установлен разъем для подключения кабеля, соединяющего устройство с ЭВМ. Все устройство закрывается крышкой с прямоугольным отверстием, которое ограничивает секторы перемещений ручки.

Ждущие мультивибраторы по запуску от машины генерируют ТТЛ-импульсы высокого уровня (выводы 3 микросхем таймеров), их длительность пропорциональна сопротивлению соответствующего координатного потенциометра (X или Y). Конденсатор С1 и потенциометр R1 вместе с регистром R3 образуют времязадающее RC-звено мультивибратора.

Электрическое сопряжение УИ с «Электроникой БК-0010» осуществляет-

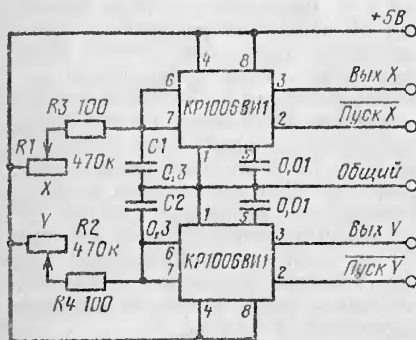


Рис. 1. Принципиальная схема указателя графической информации.

\* Казанцев А. П., Данилов А. Б. Интерфейс пользователя ПЭВМ «Электроника БК-0010». Микропроцессорные средства и системы, 1989, с. 57.

```

ERROR: TRAP 205 ;Переход на обработку ошибки (ФСКАЛ)
;
; *** При входе: в R1 - 0(K) или 1(V) ***
BEGIN: TST R1 ;Проверить аргумент
BLT ERROR ;<0?
CMP R1,#1
BGT ERROR ;>1?
; *** Аргумент правильный! ***
INC R1 ;Биты: #0 - пуск X, #1 - пуск Y
CLR R0 ;Сброс счетчика
; *** 177714 - адрес порта ввода и порта вывода ***
MOV R1,#177714 ;Пуск!
MOV R1,#177714 ;Задержка
CLR #177714 ;Конец пуска
COUNT: INC R0 ;Цикл счета длительности
BIT R1,#177714 ;Ввод из порта
BEQ COUNT ;Импульс не кончился?
; *** Целое в R1 превращается в "плавающий" форму ***
MOV R0,R1
JMP #133352 ;Превращение и выход из подпрограммы

```

Рис. 2. Программа, обслуживающая указатель информации

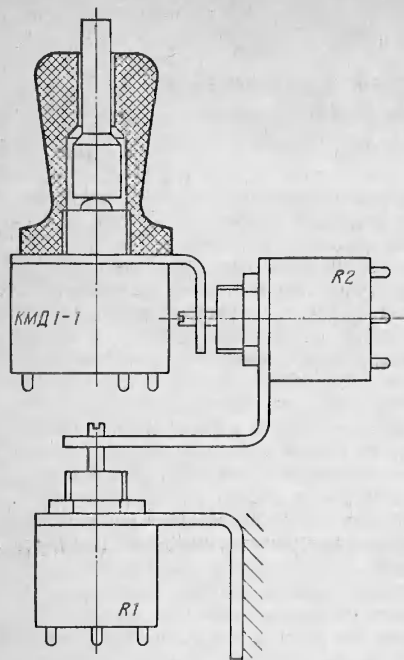


Рис. 3. Эскиз механизма указателя

ся через штатный разъем ввода-вывода машины. Возможно подключение выходов X и Y к разрядам 0 и 1 порта ввода, а линий «Пуск X» и «Пуск Y» — к 0 и 1 порта вывода соответственно. При таком подключении работает драйвер (рис. 2). Кнопка КМД1—1 (рис. 3) может также подключаться к порту ввода, например к разряду 2. Для питания схемы УИ используется шина микроЭВМ «Электроника БК-0010» +5 В, которая выводится на контакты А8, А9, В8, В9 разъема портов через установленную на плате машины перемычку. Шина «Общий» выведена на контакты А11, А18, А19, В11, В18, В19.

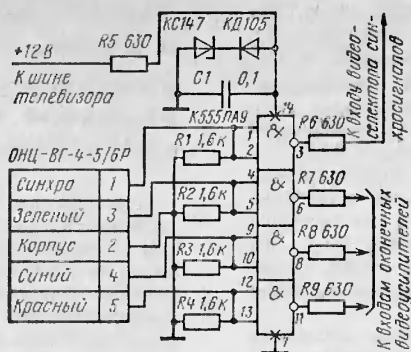


Рис. 4. Схема сопряжения микроЭВМ с цветным телевизором

Программа, обслуживающая УИ, запускает каждый мультивибратор в отдельности и определяет длительность импульсов. Это процедура-функция, встроенная в ФОКАЛ-БК-0010.

Для определения состояния кнопки выполняется считывание разряда 2 из порта ввода. На языке ФОКАЛ это можно сделать функцией GX.

Первый координатный потенциометр R1 угольником прикреплен к основанию; R2 закреплен на оси КПП. Рукоятка, зажимая угольник, навинчивается на микровыключатель КМД1—1 нарезанной в ее теле резьбой М 8×0,5. В полости свободно размещен толкатель (кнопка), передающий усилие нажима на микровыключатель.

Рукоятка и толкатель вытачиваются из текстолита, эбонита или плотной древесины. Оси рукоятки и обоих потенциометров должны пересекаться в одной точке. Закрепить угольники на осях потенциометров проще всего пайкой, используя в качестве флюса ортофосфорную кислоту. Сектор движения по каждой координате ограничивается углом 90°, поэтому номинал потенциометров выбран втрое больше рабочего диапазона.

Интерфейс цветного телевизора. На печатной плате микроЭВМ «Электроника БК-0010» можно установить элементы связи с RGB-монитором. Цветной телевизор с раздельными оконечными видеосигналами (УПИМЦТ-61, ЗУСЦТ-П-51, 4УПИЦТ-51) можно использовать как видеомонитор (рис. 4). Основное достоинство предложенного варианта — переключенные телевизионного приемника в режим видеомонитора, происходит при стыковке разъемного соединения с машиной. Это достигается использованием логической микросхемы с открытыми коллекторными выходами, которая при разъединении разъема за счет резисторов на входе схемы (R1... , R5) отключается от цепей телевизора и не оказывает на него никакого влияния. Выходные сопротивления снижают логические уровни микросхемы до уровня входных сигналов блоков телевизора. Применение микросхем серии К555 позволяет питать схему от телевизионного приемника через простейший стабилизатор (D1, D2, R5).

Схема размещается на корпусе телевизора (все прежние соединения разъема «Магнитофон» нужно ликвидировать). Вход «Видео» блока селектора синхронимпульсов лучше разорвать коммутирующей группой разьема (необходимо заменить разъем в телевизоре) — устраняется «подрагивание» изображения в режиме монитора из-за шумов по радиоканалу.

*Кооператив «Информ» продает указатель графической информации и ПО. Тел. 497-86-52, Москва (Майоров Леонид Николаевич с 19 до 21 ч.)*

Статья поступила 09.06.87

УДК 681.3.06

А. Г. Водяник

## АДАПТАЦИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ ФОДОС К МИКРОЭВМ «ЭЛЕКТРОНИКА БК-0010»

Для расширения круга задач, решаемых на микроЭВМ «Электроника БК-0010» [1] в учебном заведении, резидентная часть монитора ОС ФОДОС адаптирована к БК-0010. Разработан вариант локальной сети шинной структуры, объединяющий несколько БК-0010 с более мощной главной микроЭВМ. Устройства главной микроЭВМ разделяются пользователями БК-0010. Предоставляемых возможностей достаточно для работы с интерпретатором языка БЕЙСИК, макроассемблером, текстовыми редакторами, компоновщиком, прикладными программами,

построенными с помощью компиляторов языков Паскаль и Фортран, в получении навыков обращения с ОС. Предлагаемое решение применимо, если учебное заведение располагает несколькими БК-0010 и комплексом на базе микроЭВМ «Электроника 60» (или ДВК2) или имеет комплектный класс технических средств [2].

Адаптация резидентной части монитора заключается в замене драйвера системного устройства драйвером сети DX (сохранено имя драйвера НГМД) и решению вопросов, связанных с выводом на терминал. Небольшой рабочий объем ОЗУ БК-0010 (16 Кбайт) обусловил применение минимальной версии монитора DXMNSJ. BL. Драйвер сети обеспечивает доступ к виртуальным устройствам DX0 ; DX1 ;...DX7 ;. Отображение задается исполняемой

О. Е. Киселев

## УНИВЕРСАЛЬНЫЙ ТЕКСТОВЫЙ РЕДАКТОР ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА БК-0010»

Персональные (бытовые) компьютеры очень часто используются в качестве пишущей машинки для ввода, хранения и редактирования различных текстов. Для решения задач на ЭВМ «Электроника БК-0010» разработан универсальный текстовый редактор. Он позволяет вводить текст и корректировать его на экране, т. е. изменять любой символ редактируемого фрагмента, раздвигать символы в строке для вставки новых букв, убирать символы со сжатием строки, вносить, удалять и дублировать строки текста. Редактор включает развитые средства быстрого перемещения курсора по экрану и листания текста в обоих направлениях. Если курсор находится на первой строке экрана, то при нажатии на клавишу ↑ изображение сдвигается вниз и вверх экрана появляется предыдущая строка текста (если есть). Аналогичное ролирование вверх происходит при нажатии на клавишу ↓, когда курсор находится в последней или предпоследней строке экрана. Любую группу строк (в том числе одну строку или весь текст) можно отметить как блок, а затем целиком скопировать на другое место или удалить. Для отметки текущей строки как начала или конца блока необходимо нажать всего на две клавиши. Можно выполнить и обратную операцию — установить курсор на строку, которая ограничивает блок. Специально для форматирования текста служат операции разрезания текущей строки по позиции курсора и склейки двух строк в одну. При этом, если результирующая строка получается слишком длинной, то она вновь разрезается на две: в первой оставляется столько слов (не символов!), сколько помещается; все остальное переносится на новую строку.

Находящийся в памяти текст состоит из строк переменной длины (максимальный размер — 63 символа); хвостовые пробелы не хранятся. Но пользователь при редактировании об этом не подозревает — он может перемещать курсор по экрану без всяких ограничений и вставлять любые символы в любом месте строки. У человека, таким образом, складывается впечатление работы с текстом, состоящим из строк равной длины, как при корректировке обычного печатного текста. Эта черта отличает данную программу от других широко известных экранных редакторов (например, K52). Реализация в ней концепции непосредственного редактирования значительно повышает удобство работы пользователя.

Кроме описанных выше функций обработки текстов, редактор позволяет сохранить текст на магнитной ленте, считать ранее введенный текст и добавить его в конец того, который уже находится в памяти. С помощью операций блочной пересылки добавленный фрагмент можно перенести в любое место. Универсальный драйвер позволяет распечатать текст на любом печатающем устройстве, реализующем протокол ИРПР. Эта функция опробована на устройствах ТПУ, УВВПЧ-30—004, ТС 7180, D100 (комплекса ДВК). Причем на последних двух устройствах можно печатать строчные и прописные буквы русского и латинского алфавитов.

Программа редактора текстов написана в перемещаемом коде и занимает чуть больше 3 Кбайт оперативной памяти. Весь остальной объем ОЗУ БК-0010 (13 Кбайт) может использоваться для хранения редактируемого текста — до 8 печатных (через 2 интервала) страниц по 1800 символов.

420008, Казань, ул. Ленина, 18, Казанский Государственный университет им. В. И. Ульянова-Ленина, химический факультет, тел. 39-87-94

в главной микроЭВМ программой управления сетью. Так, DXO: может соответствовать системному устройству SY: главной микроЭВМ, DX1:, DX2:, ... DX7: — файлам DK: DX1. VOL, DK: DX2. VOL, ... DK: DX7. VOL. Емкость виртуального устройства — 512 информационных блоков. Если эквивалент устройства имеет меньшую емкость, то недоступная область изолируется фиктивным файлом F.BAD.

Каждому монитору, работающему в сети, присвоен номер и выделен свой рабочий файл на виртуальном устройстве DXO:. Для i-го монитора это SWAP<sub>i</sub>. SYS и ему доступны для записи файл SWAP<sub>i</sub>. SYS и устройство DX<sub>i</sub>:. Все остальные устройства доступны только для чтения. Таким образом исключается одновременная модификация данных несколькими мониторами.

МикроЭВМ БК-0010 не имеет регистров состояния и данных терминала. Функции терминала выполняются подпрограммой вывода символов, расположенной в ПЗУ. Чтобы обеспечить работоспособность программ, обращающихся к регистрам терминала, создана подпрограмма, конструктивно объединенная с драйвером сети. Регистры терминала имитируются словами ОЗУ. Получив очередной символ из слова-регистра данных, подпрограмма запоминает содержимое системной области связи и обращается к подпрограмме вывода символов. Затем она восстанавливает область связи, анализирует слово-регистр состояния и, если прерывание от терминала разрешено, имитирует очередное прерывание. Дополнительные затраты времени на выдачу символа не превышают 200 мкс.

В разработанном варианте сети используется главная микроЭВМ «Электроника 60». Шина сети состоит из 33 проводников. «Электроника 60» подключается к шине через интерфейс И2, а БК-0010 — через порты ввода-вывода. Возможно подключение до семи БК-0010. Для общего накопителя на гибких дисках это число следует считать предельным. При оснащении главной микроЭВМ более емким накопителем на жестких дисках число БК-0010 может быть увеличено до 10...15. Скорость передачи данных — 4 Кбайт/с, что почти в четыре раза выше, чем в звездобразной сети комплекта класса на основе интерфейса ИРПС.

В Мариупольском металлургическом институте построена сеть, содержащая четыре БК-0010. В качестве общего системного устройства успешно применялся электронный квазидиск [3] емкостью 256 блоков, превышающий ИГМД по быстродействию и надежности. Описанные средства удобны в работе, их легко задействовать в условиях вуза или техникума силами инженеров и преподавателей.

341004, Мариуполь, Донецкая область, ул. Заозерная, 98/Б

### ЛИТЕРАТУРА

1. Косенков С. М., Полосин А. И., Сцепицкий З. А. и др. Бытовая персональная микроЭВМ «Электроника БК-0010» // Микропроцессорные средства и системы. — 1985. — № 1. — С. 22—25.
2. Фролов Г. И., Косенков С. М., Шахнов В. А. и др. Комплексный класс технических средств на базе микроЭВМ «Электроника БК-0010» и ДВК2МШ // Микропроцессорные средства и системы. — 1986. — № 4.
3. Сорокин Ю. Ю., Лаврентьев В. В., Максимиак С. П., Субач В. В. «Электронный диск» для микроЭВМ «Электроника 60М» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 92.

Сообщение поступило 16.04.87

Статья поступила 01.02.88

## ЭФФЕКТИВНОЕ ХРАНЕНИЕ ДАННЫХ В ЯЗЫКЕ ФОКАЛ ПЭВМ «ЭЛЕКТРОНИКА БК-0010»

Версия языка высокого уровня ФОКАЛ для ПЭВМ «Электроника БК-0010» позволяет самостоятельно создавать разнообразные эффективные программы. Язык прост в изучении, имеет функции для работы с символьной и графической информацией, внешними устройствами. Один из существенных недостатков ФОКАЛа — неэкономное использование памяти при хранении данных (любая переменная занимает 8 байт). С увеличением числа переменных падает скорость вычислений.

Для устранения этого недостатка в рамках интерфейса внешних функций (ИВФ)\* создана система «МАССИВЫ 2». Она представляет собой совокупность функций, организующих запись, хранение и считывание из ОЗУ данных в форматах «байт», «целое», «вещественное».

По сравнению с ФОКАЛом для указанных типов данных память расходуется эффективнее в 8, 4 и 2 раза соответственно. Скорость работы программы не снижается с ростом числа переменных. В ФОКАЛ-программе перед первым обращением к системе необходимо с помощью функции F.D (IMENSION) зарезервировать внутреннюю память ИВФ под данные различных типов. Для этого в списке аргументов функции указывают число элементов массивов каждого типа в последовательности «байт», «целое», «вещественное», например F.D (100, 500, 200). После исполнения функции F.D ее значение равно числу байтов, оставшемуся свободным для последующего резервирования.

Запись и чтение данных в системе выполняются функциями, имена которых совпадают с типами данных: F.V (YTE), F.I (NTEGER), F.R (EAL). Первый и второй аргументы вычисляются, как арифметические выражения, и задают номер элемента массива и его записываемое значение. Если второй аргумент отсутствует, то происходит считывание, и функция принимает значение указанного элемента. Запись F.V (2+3, FCHR(—1)) означает, что элементу байтового массива с номером пять будет присвоено значение символа, считанного с клавиатуры. Для вызова элемента необходимо выполнить функцию F.V(5). В случае ошибки при резервировании памяти или обращении к несуществующему элементу любого из массивов выдается стандартное сообщение ФОКАЛа: «нет места для переменных».

Поскольку используемая область памяти непосредственно ФОКАЛом не обслуживается, предусмотрена функция F.L (IBRARY), осуществляющая запись и считывание массивов с магнитофона, и функция F.C (LEAR), стирающая данные каждого из типов отдельно или все вместе.

Система функций «МАССИВЫ 2» перемещается, занимает около 0,7 Кбайт, допускает работу с версией ИВФ, поддерживающей режим «расширенной памяти».

107005, Москва, 2-я Бауманская ул., д. 5, МВТУ  
им. Н. Э. Баумана, кафедра МТ-7, Полянскому П. В.

Сообщение поступило 18.06.87

\* Казанцев А. П. Интерфейс внешних функций интерпретатора ФОКАЛ-БК-0010 // Микропроцессорные средства и системы.— 1987.— № 4.— С. 86.

## ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ ПЭВМ «ЭЛЕКТРОНИКА БК-0010»

Разработан набор процедур-функций, встроенных в интерпретатор ФОКАЛ-БК-0010 посредством интерфейса внешних функций [1]. Это быстро работающие программы в кодах, подготовленные на ассемблере.

Для работы под управлением интерпретатора ФОКАЛ реализована процедура-функция «закраски окон» в течение одной трети секунды одним из семи цветов (красным, зеленым, синим, желтым, пурпурным, голубым, черным). Графическая информация и текст выводятся на закрашенное окно (фон) четырьмя цветами (красным, зеленым, синим и черным). Можно быстро изменить цвета фона и изображения в любом окне. Функция закраски имеет вид

F.C (COLOR, LEFT, RIGHT, BOTTOM, TOP).

Здесь F.C — зарезервированное имя функции; COLOR, LEFT, RIGHT, BOTTOM, TOP — аргументы функции, где COLOR — номер цвета закраски; LEFT — левая граница окна; RIGHT — правая граница; BOTTOM — нижняя граница; TOP — верхняя граница.

Все аргументы в обращениях к функции могут быть выражениями языка ФОКАЛ. Можно выбрать следующие коды: 0 — красный, 1 — зеленый, 2 — синий, 3 — желтый, 4 — пурпурный, 5 — голубой, 6 — черный. Эти коды закрашивают окно полностью. Код 7 меняет цвет окна и изображения в нем: красный на черный и наоборот, зеленый на синий и наоборот, желтый на бледно-синий (но не голубой), пурпурный на бледно-зеленый. Голубой цвет не изменяется.

Пример 1. Весь экран объявляется окном и закрашивается красным цветом.

```
1.10 COMMENT * Закраска экрана красным цветом
1.20 SET RED = 0
1.30 SET LEFT = 0
1.40 SET RIGHT = 705
1.50 SET BOTTOM = 0
1.60 SET TOP = 239
1.70 EXECUTE F.C (RED, LEFT, RIGHT, BOTTOM, TOP)
1.80 QUIT
```

Пример 2. После исполнения программы в первой части экрана появятся прямоугольные транспаранты синего цвета с красными надписями в середине.

```
2.05 C * Выключение индикации курсора, очистка экрана
2.10 X FCHR(154);X FCHR(12)
2.12 C * Изображение колонки транспарантов
2.15 F I=0,40,120;X F.C(2,200,239,69+I,99+I)
2.20 X FCHR(145);C * Включить красный цвет
2.22 C * Нанесение надписей на транспаранты
2.25 F I=4,4,16;S J=1/4;X FK(26,I-1);T " ";X FCHR(47+J);T " "
2.30 q
```

Указатель графической информации. Если применяется указатель информации, например джойстик, то ускоряется процесс взаимодействия человека с ЭВМ. Диалог при этом сводится к выбору посредством указателя необходимых процедур из предложенного на экране меню. Меню может быть организовано в виде списка процедур или «приборного табло» — рядов транспарантов с краткими надписями или пиктограммами. Пользователь легко манипулирует ручкой джойстика, и ЭВМ получает координаты для указания объектов на экране монитора, а человек видит перемещаемый по экрану курсор, указывающий

на позиции списка процедур или на транспаранты табло. При нажатии кнопки на рукоятке в ЭВМ поступает сигнал для отметки выбранного объекта.

Для обслуживания джойстика имеется функция F.J (COORDINATE), встроенная в ФОКАЛ.

COORDINATE — аргумент функции. Для координаты X он должен иметь значение 0, для Y — значение 1. Аргумент может быть выражением. Функция при обращении к ней принимает значение координаты, заданной аргументом функции. Два последовательных обращения сначала с аргументом 0, а затем 1 дают пару координат точки, которая должна быть показана на экране.

Пример 3. Управление перемещением курсора по знакам на экране в зависимости от положения рукоятки джойстика. Переменные K и B 3.20 — масштабные коэффициенты:

```
3.10 S H=F.J(0);S Y=F.J(1)
3.20 S JK=K/K;S YK=Y/L
3.30 X FK(KK,YK)
3.40 G 3.10
```

Из-за погрешностей задатчиков координат в джойстике перемещающийся по экрану курсор «дрожит». Эти погрешности практически полностью ликвидируются цифровой фильтрацией по простому алгоритму сглаживания методом «скользящего окна» и получается плавное перемещение курсора.

Музыкальная функция. Для индикации состояний процессов, происходящих в ЭВМ, могут служить музыкальные мелодии. В некоторых случаях музыкальные сообщения могут заменять текстовые.

Музыкальная функция, встроенная в ФОКАЛ, имеет вид F.M (MELODY),

где F.M — имя функции MELODY — ее аргумент, который задает порядковый номер мелодии из резидентной библиотеки. Аргументом может быть любое выражение, принимающее значения 0, 1, 2 и 3 (четыре мелодии). Встроенная функция представляет собой программу подачи звуковых сигналов на основе музыкального звукоряда. Одна из них (мелодия 3) состоит из последовательности нот, генерируемых случайно. Шестнадцать первых нот этой последовательности имеют длительность 1/4, а последняя нота — целая. Таким образом получается законченная «случайная» мелодия.

Постоянные мелодии (0, 1 и 2) записаны и отлажены с помощью музыкального редактора (специальной инструментальной программы). Прогрывающая программа из редактора была модернизирована и явилась основой встроенной музыкальной функции. Функция состоит из двух взаимосвязанных частей: проигрывающей программы и резидентной библиотеки мелодий. При обращении к функции проигрывающая программа исполняет мелодию, указанную аргументом.

Пример 4.

```
4.10 X F.A(0); C * Мелодия 1
4.20 X F.A(1); C * Мелодия 2
4.30 X F.A(2); C * Мелодия 3
4.40 X F.A(3); C * Мелодия 4 (случайная)
```

Пример 5.

```
5.05 C * Бесконечная случайная мелодия
5.10 S MELODY=3
5.20 X F.A(ME); F J=1,100;
5.30 G 5.20
```

Описанные функции являются элементарными процедурами, используя которые, можно построить на микроЭЭМ «Электроника БК-0010» выразительный интерфейс пользователя. Он эффективен при построении и использовании систем, предназначенных для непрограммирующих профессионалов.

142292, Московская обл., Пущино,  
Институт биофизики АН СССР;

тел.: 923-96-68 доб. 2-93, Москва,  
3-90-01 доб. 2-97, Пущино

## ЛИТЕРАТУРА

1. Казанцев А. П. Интерфейс внешних функций интерпретатора ФОКАЛ-БК-0010 // Микропроцессорные средства и системы. — 1987. — № 4. — С. 86.
2. Казанцев А. П., Майоров Л. Н., Данилов А. Б. Указатель информации и интерфейс цветного телевизора // Микропроцессорные средства и системы, данный номер, С. 54.

Статья поступила 09.06.87

УДК 681.32

П. В. Полянский

## «ЭЛЕКТРОНИКА БК-0010» В СИСТЕМАХ ИССЛЕДОВАНИЯ ОБЪЕКТОВ С РАСПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ

Для повышения стабильности проплавления при электроно-лучевой сварке необходимо в ходе технологического процесса контролировать распределение плотности тока электронного луча. Для решения этой задачи создана автоматизированная система исследований на базе ПЭВМ «Электроника БК-0010».

Выбор данной ПЭВМ обусловлен достаточной надежностью, подтвержденной при эксплуатации в производственных условиях, легкостью подключения внешних устройств через параллельный программируемый порт, отсутствием необходимости использовать дополнительные инструментальные средства при программировании данной ПЭВМ. Следует специально отметить возможность графической интерпретации результатов измерений. Это особенно важно при исследовании объектов с распределенными параметрами, где полезную информацию несет пространственное изменение характеристик объекта, оценить которое исследователю удобнее всего по предъявляемой «картинке» [1].

В разработанной системе источником первичной информации служит круговой зондовый датчик (рис. 1), представляющий собой более совершенный вариант дифференциального датчика, использующего принцип «прямого края» [2]. Для получения информации о распределении плотности тока электронный луч необходимо отклонить на пластине датчика в нескольких различных направлениях с одновременным считыванием показаний датчика в память ПЭВМ, где они должны пройти функциональные преобразования.

Луч можно уводить из зоны сварки не более чем на 1—2 мс [3]. За это время сварочная ванна не успевает заметно остыть. Исходя из требуемой геометрической точности и принятой схемы эксперимента за указанное время необходимо выполнить около 1000 измерений. Быстродействие ПЭВМ «Электроника БК-0010» не позволяет справиться с такой задачей программным путем. Поэтому процесс получения исходной информации возложен на систему сбора данных (ССД) «Луч» (рис. 2), реализованную аппаратно и управляемую от ПЭВМ. ССД осуществляет быстрое отклонение луча, синхронное с ним считывание показаний



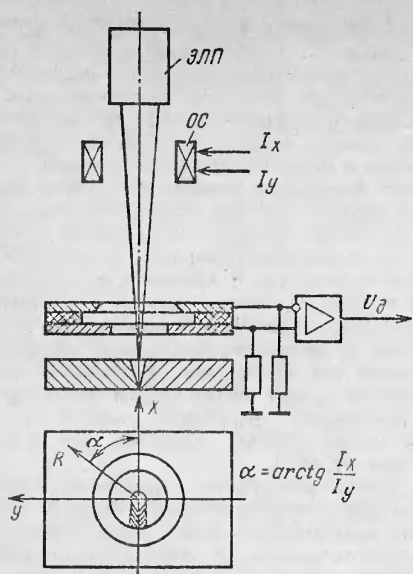


Рис. 1. Функциональная схема датчика распределения тока в электронном луче

датчика и запоминание их в буферном ОЗУ для последующей передачи в ПЭВМ. За 1 мс ССД проводит 1024 измерения. Элементной базой устройства служат микросхемы К1107ПВ1 (АЦП), К541РУ2 (ОЗУ), К544Д2 (ОУ), К572ПА1А (ЦАП X, Y), К1108ПА1А (ЦАП R), микросхемы серии К155 (устройство местного управления).

Особенность созданной ССД — отсутствие ручных органов управления. Все функционально важные параметры настраиваются от управляющей ПЭВМ. Для отклонения луча в нужном направлении на ПЭВМ задаются знаки и амплитуды токов, подаваемых на отклоняющую систему. Чтобы не происходило потери точности аналого-цифрового преобразования при различных уровнях входного сигнала, коэффициент передачи усилителя сигнала датчика программно устанавливается от ПЭВМ так, чтобы максимальное значение ожидаемого сигнала совпадало с допустимой границей входного напряжения АЦП. Из буферного ОЗУ ССД в ПЭВМ может быть считан любой фрагмент данных, что улучшает соотношение сигнал/шум и значительно экономит память ПЭВМ.

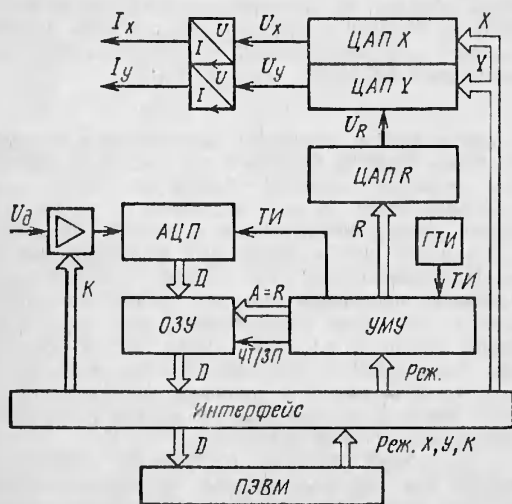


Рис. 2. Структурная схема аппаратной части системы сбора данных «Луч»

Использование в системе исследования ПЭВМ совместно с программно управляемой ССД позволяет решить сразу несколько существенных задач. Во-первых, резко ограничивается возможность нарушения технологического процесса из-за ошибок оператора, так как все команды проходят программный контроль и явно ошибочные данные в аппаратуру не передаются. Во-вторых, управление системой ведется только через меню режимов, что снижает требования к программистской квалификации исследователя. В-третьих, ПЭВМ осуществляет многоступенчатую программную фильтрацию данных, их последующее функциональное преобразование и выдачу результата в графической форме, отвечающей физической сущности эксперимента.

Наиболее интересно в данной системе решен вопрос борьбы с помехами. Из-за специфики процессов при электронно-лучевой сварке сигнал с датчика сильно зашумлен. Аппаратные фильтры оказываются малоэффективными, так как параметры полезного сигнала заранее неизвестны. Дополнительными высокочастотными помехи вносит и канал аналого-цифрового преобразования, причем характеристики помех могут меняться во времени в результате теплового дрейфа параметров микросхем. Поэтому борьба с помехами осуществляется программно.

Исследователь имеет в своем распоряжении режим работы, в котором он может задать любую последовательность алгоритмов фильтрации из предложенных по меню и сравнить сигналы до и после обработки, которые предъявляются ему в двух программных окнах на экране дисплея. Причем, первый шаг фильтрации — считывание из буферного ОЗУ ССД только той части данных, в которой появление полезного сигнала наиболее вероятно. Лучшая для конкретной помеховой обстановки последовательность фильтров запоминается и автоматически задействуется при последующих измерениях.

В настоящий момент в меню входят следующие процедуры фильтрации: рекуррентное усреднение с апертурами 2, 4, 8; рекуррентные медианные фильтры с апертурами 3, 5, 7 [4]; градиентный фильтр с перестраиваемым порогом и апертурой 5; процедура численного дифференцирования. Последняя связана с особенностями выделения полезного сигнала датчиков данного типа [5]. Все фильтры написаны на языке ПЭВМ, а их параметры подобраны так, чтобы исключить использование операций с плавающей запятой.

Программное обеспечение системы создано на базе интерфейса внешних функций (ИВФ) [6]. Эта программная система позволяет дополнить язык ФОКАЛ произвольными функциями, написанными на языке ПЭВМ. В качестве внешних функций оформлены все процедуры обмена сигналами с аппаратурой, переформатирования данных из вещественного формата в байтовый и наоборот, записи и считывания данных с магнитофона. В них широко использованы внутренние подпрограммы обработки текстов и пакет плавающей запятой интерпретатора языка ФОКАЛ [7], подпрограммы пускового монитора [8]. Собственно на языке ФОКАЛ написан диалог с оператором, выполнены оформление экрана и расчет режимов управления аппаратурой.

Такое решение позволяет быстро осуществлять измерение и первичную обработку данных, экономно расходовать память ПЭВМ (система оперирует с 4К отсчетами при каждом измерении, которые размещены в памяти в байтовом формате). При этом сохраняется возможность мобильной корректировки прикладных программ, созданных на ФОКАЛе. Система в программном смысле остается открытой и позволяет обрабатывать новые алгоритмы преобразования данных (например, фильтрации или распознавания) на ФОКАЛе. Для практического использования полученное решение может быть перепрограммировано на языке ПЭВМ и включено в состав внешних функций.

Опыт разработки и практического использования данной системы показал, что при наличии специализированной аппаратной системы сбора данных и программном обеспечении, сочетающем язык высокого уровня (ФОКАЛ или БЕЙСИК) и язык ПЭВМ. На ПЭВМ «Электроника БК-0010» возможно создание высокоэффективных систем исследова-

дования, в том числе и объектов с распределенными параметрами, где велики объемы исходных данных и требования к скорости обработки и виду представления информации.

107005, Москва, 2-я Бауманская ул., д. 5, МВТУ им. Н. Э. Баумана, кафедра МТ-7, Полянскому П. В.

## ЛИТЕРАТУРА

1. Казанцев А. П. МикроЭВМ для информационно-измерительных систем // Материалы 5-го Всесоюзного симпозиума по модульным информационно-вычислительным системам. — Кишинев, 1985. — С. 80—81.
2. Майоров Л. Н. Измерение характеристик луча при электронно-лучевой сварке // Вопросы атомной науки и техники. Сер. Сварка в ядерной технологии. — 1984. — Вып. 1 (12). — С. 26—28.
3. Баня Е. Н., Киселевский Ф. Н., Кистерская Л. Д. Выбор времени сканирования в системах автоматического направления электронного пучка по стыку // В республ. межвед. научно-техн. сб.: Автоматизация производственных процессов в машиностроении и приборостроении — 1976. — Вып. 15. — С. 11—14.
4. Быстрые алгоритмы в цифровой обработке изображений / Т. С. Хуанг, Дж.-О. Эклунд, Г. Дж. Нуссбаумер и др.; Под ред. Т. С. Хуанга; Пер. с англ. — М.: Радио и связь, 1984. — С. 156—212.
5. Franz K. Kirner. Albert Schuler High Resolution Method for Determination of Current — Density — Distribution in Electron Beams of Arbitrary Shape // 3rd International Colloquium on Welding and Melting by Electrons and Laser Beam. — Lyon. 5—9 September. — 1983. France, p. 45—52.
6. Казанцев А. П. Интерфейс внешних функций интерпретатора ФОКАЛ — БК-0010 // Микропроцессорные средства и системы. — 1987. — № 4. — С. 86.
7. Программное обеспечение СМ ЭВМ. Операционная система с разделением функций РАФОС. Диалоговая система программирования ДИАСП. Описание языка. — 1980. Т. 5, кн. 5, ч. 3.
8. Математическое обеспечение микроЭВМ «Электроника БК-0010». Руководство системного программиста. — 1985. Т. 1, кн. 2.

Статья поступила 18.06.87

УДК 681.3.06

В. Т. Монахов

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПЭВМ СЕРИИ «ЭЛЕКТРОНИКА БК-0010»

В течение 1987 г. московские любители ПЭВМ и их коллеги в других городах продолжали пополнять программное обеспечение четырех модификаций ПЭВМ «Электроника БК-0010» (далее, БК-0010). Число программ для БК-0010 достигло полутысячи.

Общая характеристика программ 1987 г. — значительное расширение использования машинного языка вместо языков высокого уровня и лучшее их документирование. Правда, в силу ограниченности ОЗУ БК-0010 большинство инструкций к программам реализовано в виде отдельных файлов. Это иногда приводит к их разрозненному обращению среди владельцев ПЭВМ, но потери значительно меньше, чем при полном отсутствии документации. Снизить число случаев разъединения программ и инструкций к ним можно «стандартизацией» их именования, при которой документация обозначается тем же именем, что и основная программа, но с добавлением окончания DOS (или ДОК).

**Системные программы.** Ассемблер БК-0010 (С. В. Шмытов, А. Н. Сомов, С. А. Кумандин) получился простым и удобным в обращении при сохранении всех достоинств и мощностей ассемблера МАКРО-11 (по аналогии с которым он создан). Будь у профессиональных программистов столь же простая версия, язык ассемблера значительно шире применялся бы в практике программирования.

Программа ассемблера названа МИКРО и включает в себя четыре модуля: монитор, транслятор, компоновщик и текстовый редактор. Компактность МИКРО (его объем 4,5 Кбайт) позволяет одновременно создавать на БК-0010 ассемблерные тексты (до 8 Кбайт) и оттранслированные (объектные и/или загрузочные) модули (до 2 Кбайт). При компоновке можно создавать загрузочные модули размером до 16,5 Кбайт. В программе организован неограниченный обмен текстами или их частями, а также объектными и загрузочными модулями между ОЗУ и магнитофоном. По общему признанию, последние версии ассемблера (МИКРО10 и МИКРО8К) — одни из лучших программ, созданных для БК-0010.

Вторая крупная разработка — несколько версий **интерпретатора языка программирования Форт**. К сожалению, этот мощный язык высокого уровня в нашей стране пока не получил широкого распространения, и литература по нему весьма ограничена и почти недоступна широкой аудитории.

Между тем, Форт позволяет создавать компактные, даже по сравнению с машинным языком, программные продукты. Форт использует обратную польскую форму записи, знакомую многим по программируемому микрокалькуляторам, и может быть одинаково интересен начинающим программистам и профессионалам. По простоте освоения и эффективности Форт не отличается от Паскаля, но в отличие от него занимает в памяти незначительное место, позволяя даже в загружаемом виде создавать на БК-0010 интересные и практически полезные программы. Особенно удобен Форт для программирования логических и динамических игр, модулей управления роботами и другими внешними приборами и т. п. Достоинство Форта — легкая переносимость его программ на другие типы ЭВМ.

Первые версии программ **микроотладчика машинных программ** содержали ошибки и работали неустойчиво. Программы GROT, GPOT, OTL, OTL9 — усовершенствованные модификации отладчика, надежность и удобство работы с ними заметно повышены.

Создан еще один **реассемблер**, названный ДИЗАКС. С его помощью можно полностью восстановить текст ассемблерной программы, включая метки и псевдокоманды. Тексты пригодны для ввода и дальнейшей обработки на ассемблере МИКРО8К.

Таким образом, на сегодняшний день мы располагаем достаточно полным комплектом средств для успешного и удобного программирования в машинных командах и на ассемблере БК-0010.

С апреля 1987 г. появились компьютеры с **встроенным БЕЙСИКом БК-0010**. Однако более близкое ознакомление с ним несколько омрачило радость по поводу долгожданной разработки. И дело оказалось не только в ряде досадных ошибок, обнаруженных в трансляторе этого языка, но и в том, что использование БЕЙСИКа БК вдвое уменьшило фактическое ОЗУ ПЭВМ. С целью повышения скорости исполнения программ транслятор БЕЙСИКа создает в ОЗУ некое подобие объектного кода, который занимает почти столько же памяти, что и текст программы. Фактически на БЕЙСИКе БК можно писать программы, которые вместе с данными занимают объем не более 7 Кбайт в режиме полного экрана и 14 Кбайт — в режиме расширенной памяти.

Для разработки программ, содержащих большое число цифровых или текстовых данных, пришлось вернуться к Фокалу, который, к счастью, сохранен в новых модификациях ПЭВМ. С этой целью была создана программа **СФМ** (немногим более 2 Кбайт) **расширения Фокала символьными функциями и массивами**. Она позволяет вводить сим-

вольные переменные и константы с клавиатуры и с магнитной ленты, сравнивать символичные величины между собой, вырезать из них фрагменты, модифицировать (с клавиатуры или программно), определять их длину, численное значение набора цифр и произвольных арифметических выражений, входящих в их состав, выводить значения символических величин на экран, принтер или магнитную ленту, стирать из памяти, производить поиск символической величины по начальному фрагменту ее значения. Кроме того, программа СФМ снабжена функциями пересчета чисел в различные системы счисления, работы с массивами упакованных целых и вещественных чисел, перенумерации строк фокальных программ, компоновки фокальной программы из отдельных блоков, хранящихся на магнитной ленте. В программе зарезервирована возможность запрограммировать в машинных командах десять дополнительных функций любого назначения. Удобство эксплуатации СФМ состоит также в том, что она образует вместе с фокальной частью и всеми массивами, включая массив символических величин, нераздельное целое, ввод-вывод которого на магнитную ленту осуществляется обычными средствами Фокала.

Рядом полезных функций дополнен **интерпретатор языка БЕЙСИК ДВК, адаптированный для БК-0010**. Программа EXFLIBR, реализующая эту функцию, дополняет БЕЙСИК графическими функциями, имеющимися в Фокале, оператором КЕУ динамического ввода, функциями прямого обращения к ячейкам памяти ПЭВМ, а также функцией обращения из БЕЙСИКА к любым подпрограммам, написанным в машинных командах (в том числе подпрограммам монитора, системы диагностики и интерпретатора Фокала). Предполагается интерпретатор БЕЙСИКА ДВК и программу EXFLIBR включать в комплект поставки ПО ПЭВМ.

Для обработки текстовых материалов в 1987 г. разработано **несколько экранных текстовых редакторов**, среди которых — модифицированный ЭКРЕД, который можно использовать на любой модификации ПЭВМ серии БК-0010, ЭДАСП (адаптация для БК-0010 профессионального редактора EDIK), а также МИКРО8К, описанный выше.

Тексты, созданные в ЭКРЕДе или МИКРО8К, можно затем прочитать в любых других редакторах и даже в Фокале по оператору WRITE. Тексты, написанные в ЭДАСПе, можно читать только в этом редакторе. Максимальную емкость имеет ЭКРЕД — 12...13,5 Кбайт, или 8...9 полных экранов.

Достаточно развитая **информационно-поисковая система** для БК-0010 — программа BASE, позволяющая отыскивать в базе данных, введенной в ОЗУ, требуемую информацию по нескольким ее признакам, оформленным в виде закодированного поискового образа. База в каждой статье может содержать до 22 параметров, по которым можно вести ее поиск.

База данных, удобная для создания комбинированных словарей, например типа англо-русского и русско-английского, — это программа на Фокале ARRAS, написанная с применением системы СФМ. В этой программе, рассчитанной более чем на 1000 словарных статей, поисковым образом может служить не только целое слово, а любой его начальный фрагмент, даже одна буква. При поиске на экран последовательно выводятся все словарные статьи, начинающиеся с заданного поискового образа. В программе реализован пермутационный принцип: если поисковый образ набирается латинскими буквами, то база реализует англо-русский словарь, а если русскими — то русско-английский.

Аналогичная (по мере 700 слов) программа (SLOWA) создана на БЕЙСИКе БК. Более простой принцип словаря с поиском по первому слову словарной статьи реализован в программе MBD (малая база данных), написанной в машинных командах.

Ряд системных программ предназначен для облегчения написания программ в машинных командах. В программе ДМВК1-В, написанной на «вильнюсском» БЕЙСИКе, приведено описание системных переменных

БК-0010; в программе DOCGDEF изложен способ организации вычислений с использованием пакета подпрограмм обработки чисел с плавающей десятичной точкой; программа DUMPB облегчает распечатку содержимого ячеек памяти ПЭВМ с БЕЙСИКом БК, а PATHC — ввод машинных подпрограмм в такую ПЭВМ.

Для облегчения разработки фокальных программ создан ряд программных модулей: MODSA — для организации спецалфавита, MOD15 — побитовое кодирование информации, MOD90 — стирание части экрана и полиэкранный вывод информации на экран, MOD94 — ввод большого массива чисел с клавиатуры с контролем правильности ввода, MOD96 — ввод химической формулы вещества с клавиатуры, MOD98 — ввод текста с клавиатуры с редактированием ошибочных символов, MOD101 — построение таблицы на экране.

Для чтения текстовых файлов при отсутствии текстового редактора, на котором они были созданы, можно воспользоваться программами МОНСТР и ЧТЕНИЕ. Последняя пригодна для любых текстов и почти не увеличивает размер файла (ее объем всего 100<sub>6</sub> байтов), но зато первая (3 Кбайт) позволяет быстро листать текст не только вперед, но и назад.

Музыкальные редакторы пополнились программой MELOMAN. По сравнению с описанной ранее программой МЕЛОМАН она располагает рядом дополнительных возможностей, а, главное, выводит на экран обычный нотный стан с общепринятой, музыкальной нотацией.

Создана одна из первых программ для синтеза речи — ГОЛОС2.

Из других системных программ можно назвать КОPIR и СОPY — программы ускоренного копирования файлов с одной магнитной ленты на другую через ПЭВМ, BANK — база данных с поиском информации по цифровому коду, KORFOC — программа корректировки фокальных программ, не считываемых нормально с магнитной ленты, РЕДЭК — экранный редактор с электронной лупой, ФОРТРЕД — экранный редактор для интерпретатора языка Форт.

Среди прикладных программ 1987 г. прежде всего назовем пакет программ статистического анализа, включенный заводом-изготовителем БК-0010 в официальное ПО ПЭВМ серии БК-0010. Этот пакет (на Фокале) разработан на кафедре «Микроэлектроника» Минского радиотехнического института. В состав пакета входят семь программ: REG — регрессионный анализ, обеспечивающий вычисление коэффициентов многомерной полиномиальной модели методом наименьших квадратов, отбрасывание статистически незначимых членов, оценку адекватности и работоспособности модели и графическое представление данных; PROG — статистическое прогнозирование, обеспечивающее аппроксимацию временного ряда одной из стандартных функций роста; DESC — дескриптивный анализ; HYP — проверка статистических гипотез; PLAN — планирование статистического эксперимента; FUR — Фурье-анализ — для выявления в процессе диалога периодических закономерностей и корреляций во временных рядах; GRAD, представляющая собой базу данных и графический редактор. Пакет снабжен полным комплектом документации.

Второй пакет прикладных программ — EXAMPANALYS (написан на Фокале), включающий в себя 10 математических программ: FOUR — прямое и обратное преобразование Фурье комплекснозначной функции, у которой мнимая и действительная части заданы в виде массивов числовых значений; TEUL — суммирование N членов бесконечного ряда; LINSYS — решение системы линейных уравнений методом главных элементов; MRIB — нахождение первых N корней алгебраического уравнения методом Рыбакова; SHTURM — решение обыкновенных дифференциальных уравнений, в частности краевой задачи Штурма-Лиувилля; GAUSS10 — численное интегрирование по Гауссу — Кристоффелю; CAUCHY — вычисление интеграла в смысле главного значения от функции, имеющей особенность; DETS — численное дифференцирование; FMCG —

поиск минимума функции нескольких переменных методом сопряженных градиентов; INTDIF — интегрирование системы обыкновенных дифференциальных уравнений методом Рунге-Кутты 4-го порядка. Этот пакет, разработанный в МФТИ, — первый программный продукт, поступивший в открытую продажу.

В 1987 г. появились программы (на БЕЙСИКе БК): TIMUR и SIMPSO — для интегрирования по Симпсону, ROMBIN — для интегрирования по Ромбергу, PLATA — для оптимизации размещения элементов на печатной плате, а также несколько игровых, сделанных на других языках программирования программ (BOM, HOCKEY, INVASI, SNAKE1, GUESSB, TOWER, RULET и др.).

И все же большинство наиболее интересных игровых программ этого года было создано или адаптировано в машинных командах: GARDEN — уборка урожая, пожираемого вредителями; ВИДЕОСПОРТ, включающая в себя футбол, теннис и сквош; КЛАД и LOAD RUNNER — поиски сокровищ в лабиринте, охраняемом сторожами; Форт и АТАКА — военные действия против неприятеля, SHERIF — динамическая приключенческая игра, XONIX и ANTIX — динамические игры на закрасивание «территории», ТЕТРИС — высокодинамичная игра типа лентинго; ПИФ и PAC — HALL — стрельба по движущимся целям и др.

Из логических игр следует упомянуть программу GO, написанную на Форте и позволяющую играть в Го с компьютером. Программирование этой древней китайской игры на ПЭВМ можно считать уникальным достижением, ибо до недавнего времени даже японские специалисты считали, что составить программу игры в Го принципиально невозможно из-за астрономически большого числа вариантов оптимального хода в начальной стадии игры.

Из демонстрационных программ, в которых пользователь выступает в роли пассивного наблюдателя, можно назвать программу РЕКЛ информационно-рекламного типа с отличной динамической графикой и бегущей строкой, а также программу TESTGRAF, позволяющую демонстрировать разнообразные чертежи и рисунки, хранящиеся на магнитной ленте в виде фокального файла данных.

Названия здесь программы далеко не исчерпывают всего многообразия программ, разработанных в прошлом году. Сбор их и каталогизация еще только начинаются. Кроме того, в связи с организацией закупки программ у населения через центры информатики многие интересные программы авторы не распространяли широко, и о них мало кто знает. Будем надеяться, что центры информатики дополнят этот обзор данными, которыми они располагают.

Тенденция развития программ 1988 г. — дальнейшее повышение их качества и сложности. Возросла доля программ, написанных в машинных командах и на ассемблере.

Из системных программ интересны программы расширения Фокала — XFOSCAL, ФОНД, МИФ и СФМ2.

Программа XFOSCAL (автор С. Зильберштейн, г. Киров) имеет встроены экранный редактор, дополнительные операторы вызова спрайтов на экран и исполнения музыкальных вставок на Фокале [1].

Программа ФОНД (автор Г. В. Прис, Москва) дополняет Фокал целым рядом новых операторов: выбора и ввода программ в машинных командах с магнитной ленты, вывода их на ленту, обращения к подпрограммам в машинных командах из программ на Фокале и стирания их из памяти, а также сдвига блока системного дополнения Фокала по любому заданному адресу. Имеются также операторы вычисления арифметических выражений и вывода на экран результатов вычисления с точностью до девяти значащих цифр [2]. Основное достоинство ФОНДа состоит в возможности создания расширяемой библиотеки подпрограмм (в машинных командах), которые

хранятся на магнитной ленте и, при необходимости, вводятся в ОЗУ для непосредственного исполнения в качестве оверлеев или для формирования самостоятельной программы с заданным набором операций на Фокале и в машинных командах.

Программа МИФ, или МИНИФОНД — это упрощенный вариант ФОНДа, предназначенный в основном для создания игровых и обучающих программ. В ней отсутствует математическое расширение Фокала, но включены более развитые средства для отладки и сборки программ из фокальных модулей, экранного редактирования текстов, повышения выразительности и динамичности создаваемых программ. Расширенные версии программы МИФ представлены в файлах МИФ+ПП и МИФ+ДОК1.

Программа СФМ2 (автор В. Т. Монохов) по сравнению с описанной выше программой СФМ дополнена функцией FFP удлинения символьной переменной, функцией FFK программирования ключей клавиатуры и функцией FFW вывода модулей на Фокале на магнитную ленту для последующего использования их для сборки программ, включающих дополнительные функции. Особенность версии СФМ2 — наличие дистрибутивного комплекта на ассемблере ПЭВМ «Электроника БК-0010», который значительно облегчает ее модификацию.

В 1988 г. были модифицированы некоторые распространенные программы: реассемблер АНТИАС удачно дополнен операцией изменения содержимого ячеек памяти без выхода из программы и оператором поиска заданной машинной команды (файл ДИЗКОР2), база данных BASE теперь позволяет выводить на экран многострочные поисковые образы, текстовый редактор ЭДАСП дополнен драйвером принтера УВВПЧ-30 — 004 (файл EDASPL), улучшена работа графического редактора ГРЕДО (файл ГРЕДЗ), музыкальный редактор МЕЛОМАН (файл MEL2.1) теперь имеет более развитую систему нотной записи и работает более устойчиво, чем прежние версии.

Разработаны несколько специальных программ печати содержимого памяти ПЭВМ на принтере типа УВВПЧ-30 — 004 (файлы PRINT, ТПРИНТ, ФОПЕЧ, КОПЕЧ и др.) и типа «Роботрон К6312М» (файл ПРИНТРОБ). Большинство созданных ранее экранных текстовых редакторов также дополнено операцией вывода текста на принтер. Программа TIMER реализует таймерную функцию ПЭВМ, которая отсутствует в ПЭВМ с БЕЙСИКом БК-0010.

Сбор и каталогизация программного обеспечения ПЭВМ серии «Электроника БК-0010» только еще начинается. Этому содействует создание Межотраслевого специализированного фонда алгоритмов и программ для ПЭВМ (МСФАП ПЭВМ) при «ЭЛЕКС» (тел. 206-63-63). Основные задачи фонда — сбор и государственная регистрация программ для ПЭВМ, организация их экспертизы и испытаний, подготовка рекомендаций по использованию и продаже программных продуктов. В числе прочих фонд принимает от авторов и программы для ПЭВМ серии «Электроника БК-0010». Работа ведется по авторскому договору, обеспечивающему защиту авторских прав и выплату вознаграждения за созданный программный продукт.

Телефон 529-75-73, Москва, Монохов В. Т.

## ЛИТЕРАТУРА

1. Фокал или БЕЙСИК? // Наука и жизнь. — 1988. — № 4. — С. 113.
2. Прис Г. В. Супер-Фокал, или Фонд БК // Информатика и образование. — 1988. — № 5. — С. 94—95.

Статья поступила 20.11.88

А. С. Ефимов, С. В. Котов

## ПЭВМ «ЭЛЕКТРОНИКА БК-0010» В КАЧЕСТВЕ ТЕРМИНАЛА

Для работы ПЭВМ в составе вычислительных комплексов разработана программа эмуляции терминала типа VT-52 с аппаратной поддержкой.

Программа размещается в ОЗУ (адреса 30000...37777, загрузка с ленты) или ПЗУ (адреса 120000...127777). Для отображения поступающих данных на экране используются подпрограммы из драйвера экрана монитора БК-0010Ш (ПЗУ по адресам 100000...117777), которые работают с форматами 32 или 64 символа в строке, а не 80, как VT-52. Это оправдано, так как аппаратно выделенных 512 точек на строку недостаточно для удовлетворительного отображения 80 символов.

Эмулятор отображает не всю страницу VT-52 (24×80 символов), а часть ее, указанную пользователем, т. е. окно размерами 24×31 или 24×63. Невидимая часть страницы хранится в памяти и может быть выведена на экран при перемещении окна. Серьезным недостатком БК яв-

ляется нестандартная клавиатура, специализированная для работы с драйвером ТВ монитора БК.

Программа обрабатывает прерывания с клавиатуры и линии связи. Коды с клавиатуры (в режиме OFF ZINE) или линии (ON ZINE) запоминаются в буфере для исполнения на экране, так как скорость выдачи символов на экран драйвером ТВ монитора мала. Передача кодов терминалу прекращается при заполнении буфера на линии по коду DC3, а возобновляется по коду «DC1» после обработки кодов из буфера, которые дублируются при изменении состояния сигнала готовности терминала к обмену с ЭВМ (RXRD). При обработке кода программа производит изменения в странице памяти и в отображаемом окне, если это необходимо.

Соприжение терминала с ЭВМ осуществляется стандартным интерфейсом ИРПС. Для размещения программы в адресном пространстве 120000...127777 используются две микросхемы К573РФ2, в качестве адресных регистров — микросхемы К555ТМ9. Аппаратная часть размещается в корпусе блока ИРПС.

Телефон 35-14-66, Новосибирск

Сообщение поступило 14.06.88

## АВТОМОБИЛЬНАЯ ЭЛЕКТРОНИКА

С этого номера журнала мы начинаем публиковать материалы, посвященные автомобильной электронике. Возрастающий интерес к данной тематике обусловлен принципиальной возможностью улучшения характеристик автомобиля с помощью микроэлектронных средств. Применение МП и МП-систем в автомобиле осуществляется для управления различными узлами (особенно двигателем) и создания сервиса.

В подборке статей показаны два подхода к использованию МП-средств: применение специально разработанных БИС для управления бензиновым двигателем и построение систем контроля электроаппаратуры зажигания на МП-средствах общего применения.

В этом номере даны технические характеристики процессора КМ1823ВУ1, формирователя импульсов зажигания КМ1823АГ1 и контроллера системы зажигания КМ1823ВГ1. Корректор угла опережения зажигания по параметру детонации КМ1823ВГ3, а также варианты реализации на основе этих БИС электронных блоков и систем управления двигателями будут рассмотрены в последующих номерах.

Публикуемые материалы полезны широкому кругу автолюбителей, работникам автомобильной промышленности, включая и предприятия автосервиса, студентам недавно созданной специальности «Автомобильная электроника».

А. И. Овчаренко, П. В. Пушко, В. Е. Хилобок, В. И. Зверев

## МИКРОПРОЦЕССОРНАЯ СИСТЕМА КОНТРОЛЯ ДАТЧИКОВ ЗАЖИГАНИЯ

Автоматизированная система контроля (АСК) параметров датчиков зажигания (ДЗ) реализована на базе технических средств МикроДАТ (рис. 1).

В настоящее время в практике производства подлежат контролю асинхронизм и угол замкнутого состояния контактов, характеристики центробежного и вакуумного регуляторов ДЗ. Все контролируемые параметры представлены в частотно-временной форме.

АСК выполняет следующие функции:

управления частотой вращения электродвигателем (М) в супервизорном режиме. Уставка в виде байта выдается на один из каналов преобразователя код-напряжения, выходной сигнал которого является уставкой для тиристорного регулятора частоты вращения (в качестве датчика частоты вращения используется тахогенератор);

управления разрежением при поступлении кодовых уставок на блок ключей, формирующий позиционные

сигналы «Увеличение разрежения», «Уменьшение разрежения» и «Останов» (используется для стабилизации разрежения на входе вакуумного регулятора контролируемого ДЗ). В разрыв вакуумной магистрали включен электромагнитный клапан (ЭК), отключающий магистраль от насоса при проверке герметичности ДЗ (исключается влияние течи насоса на результат контроля);

измерения частотно-временных параметров ДЗ с помощью универсального программируемого аналого-цифрового преобразователя (АЦП) косвенным методом. Временные интервалы преобразуются способом последовательного счета по двум каналам: в одном канале — всегда период следования одноимпульсного датчика положения (ДП), а во втором, в зависимости от измеряемого параметра, — временные интервалы.

На рис. 2 приведены временные диаграммы сигналов ДЗ, ДП и катушки зажигания (КЗ) для четырехсекторных ДЗ. Разрежение измеряется автоматическим способом усилителя и преобразователя напряжение-код. Измеренные значения параметров сравниваются с заданными границами в микроЭВМ. Технологическая и измерительная информация представляется в виде графиков и таблиц на экране видеотерминала (при необходимости выводится на

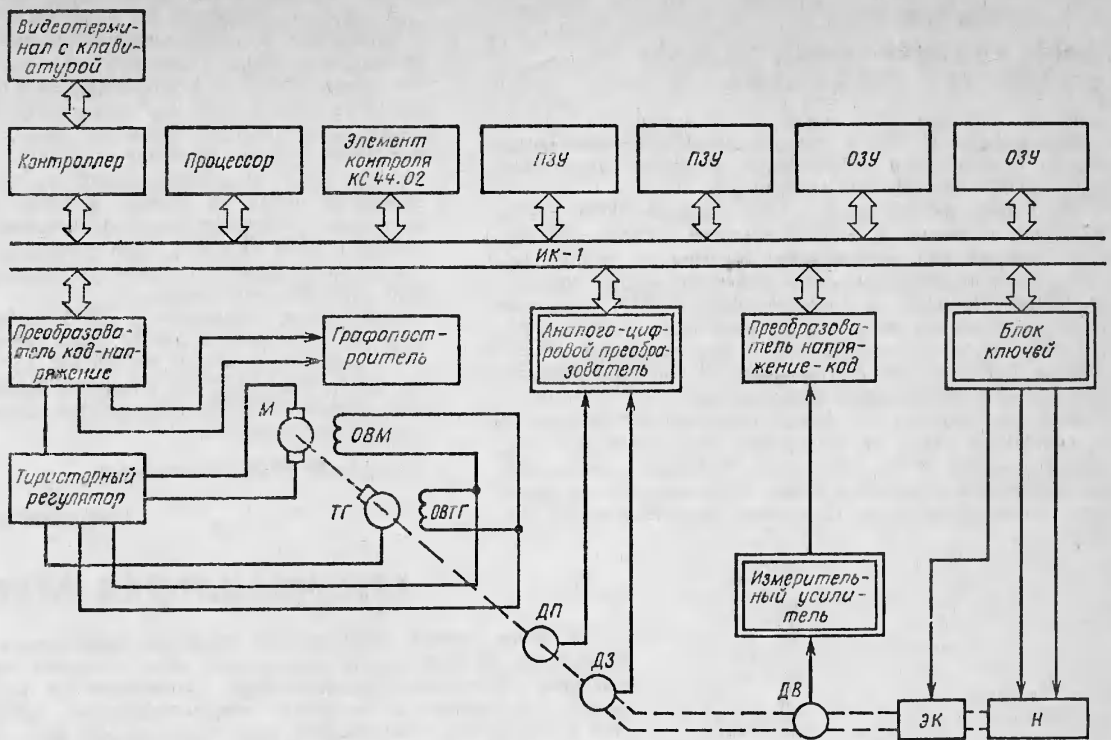


Рис. 1. Структурная схема АСК (двойной линией обозначены специализированные блоки)

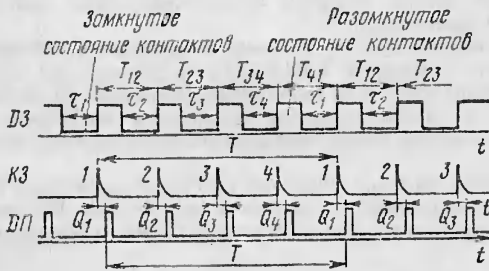


Рис. 2. Временные диаграммы входных информационных сигналов АСК

цифровая печать). Для регистрации графиков предусмотрено подключение двухкоординатного графопостроителя.

Программное обеспечение хранится в ПЗУ. Программа пользователя составлена на языке БЕЙСИК. Подпрограммы измерения параметров ДЗ и управления графикой реализованы в системе команд микропроцессора КР580ИК80. Экспериментальный образец стенда испытан и передан в эксплуатацию.

310002, Харьков, ул. Фрунзе, д. 21, ХПИ, кафедра «Информационно-измерительной техники», тел. 40-00-83

Статья поступила 23.05.88

УДК 621.317

А. И. Овчаренко

## ПРИМЕНЕНИЕ ТАЙМЕРА КР580ВИ53 ДЛЯ ИЗМЕРЕНИЯ ВРЕМЕННЫХ ИНТЕРВАЛОВ ПРИ ИССЛЕДОВАНИИ ДАТЧИКОВ ЗАЖИГАНИЯ

В данной статье описана архитектура измерительного преобразователя (ИП) параметров датчиков зажигания на основе программируемого таймера КР580ВИ53, а также процедуры обмена с микропроцессором (МП).

Структура ИП приведена на рис. 1. Внешними информационными сигналами ИП являются: ДЗ — датчика зажигания, ДЧВ — датчик частоты вращения распредвала, ДП — датчика положения распредвала. Предполагается

(рис. 2), что все три указанных сигнала жестко синхронизированы по частоте за счет механической связи между датчиками. В целях обеспечения отсчета в угловых градусах выбирается следующее соотношение частот сигналов ДП и ДЧВ:

$$f_{\text{дчв}} = 360 \cdot 10^6 f_{\text{дп}}, \text{ где } K = \{0,1\}.$$

Программируемый таймер (ПТ) преобразует временные интервалы в цифровые эквиваленты. Наличие в ПТ трех

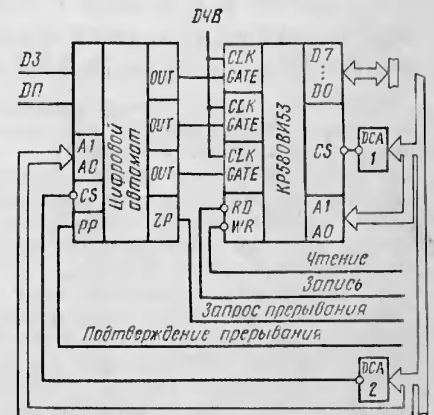


Рис. 1. Структура ИП

Процедуры организации преобразований в ИП

Запрос прерывания	Значение идентификатора (номер канала ПТ)	Оцифрованный временной интервал	Параметр ДЗ
P1	0	P0→P1	УОЗ по первому импульсу ДЗ
P2	1	P0→P2	УРСК по второму импульсу ДЗ
P3	2	P0→P3	АС между первым и вторым импульсами ДЗ
P4	0	P3→P4	УРСК по третьему импульсу ДЗ
P5	1	P3→P5	АС между вторым и третьим импульсами ДЗ
P6	2	P5→P6	УРСК по четвертому импульсу ДЗ
P7	0	P5→P7	АС между третьим и четвертым импульсами ДЗ
P8	1	P7→P8	УРСК по первому импульсу ДЗ
P0	2	P7→P0	АС между четвертым и первым импульсами ДЗ

каналов\* дает возможность осуществить это преобразование параллельно (временные интервалы поступают на входы GATE, а квантуемая их частота — на входы CLK ПТ). Обмен информацией между ПТ и МП базируется на дешифрации адреса (BADR1), на который настроен дешифратор DCA1. При этом два младших разряда адреса (A0, A1) определяют номер канала (0...2), к которому осуществляется обращение. Во всех трех каналах ПТ используется режим 1 и преобразование в двоично-десятичных кодах.

Цифровой автомат (ЦА) формирует временные интервалы, соответствующие параметрам датчиков зажигания (асинхронизму — АС, углу разомкнутого (или замкнутого) состояния контактов — УРСК или УЗСК, углу опережения зажигания — УОЗ); сигналы запросов прерывания P0...P8; идентифицирует номера импульса ДЗ (на рис. 2 показан случай четырех искровых датчиков зажигания). Обмен информацией между ЦА и МП осуществляется как аппаратно (сигналы запроса прерывания ZP и подтверждения прерывания PP), так и программно с помощью дешифрации адреса BADR2, на который настроен дешифратор DCA2. При этом два младших разряда адреса определяют состояние ЦА (табл. 1).

Таблица 1

Адресное поле ИП

Адрес	Состояние ЦА
BADR2	Установка элементов памяти в нулевое состояние
BADR2+1	Идентификация номера импульса ДЗ
BADR2+2	Запуск ЦА
BADR2+3	Останов ЦА

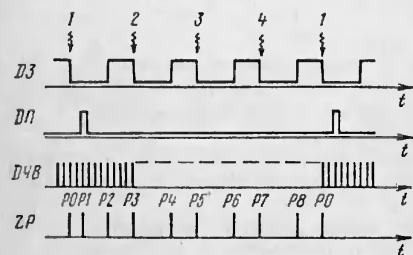


Рис. 2. Временные диаграммы входных информационных сигналов (ДЗ, ДП, ДЧВ) и сигналов запросов прерывания (P0...P8).

\* Микропроцессоры и микроЭВМ в системах автоматического управления. / Под ред. С. Т. Хвоща.— Л.: Машиностроение, 1987.

Сущность функционирования преобразователя в цикле измерения поясняется рис. 3. Необходимость операции

OUT BADR2+1,0 связана с идентификацией четвертого импульса ДЗ (см. рис. 2) и организацией первооче-

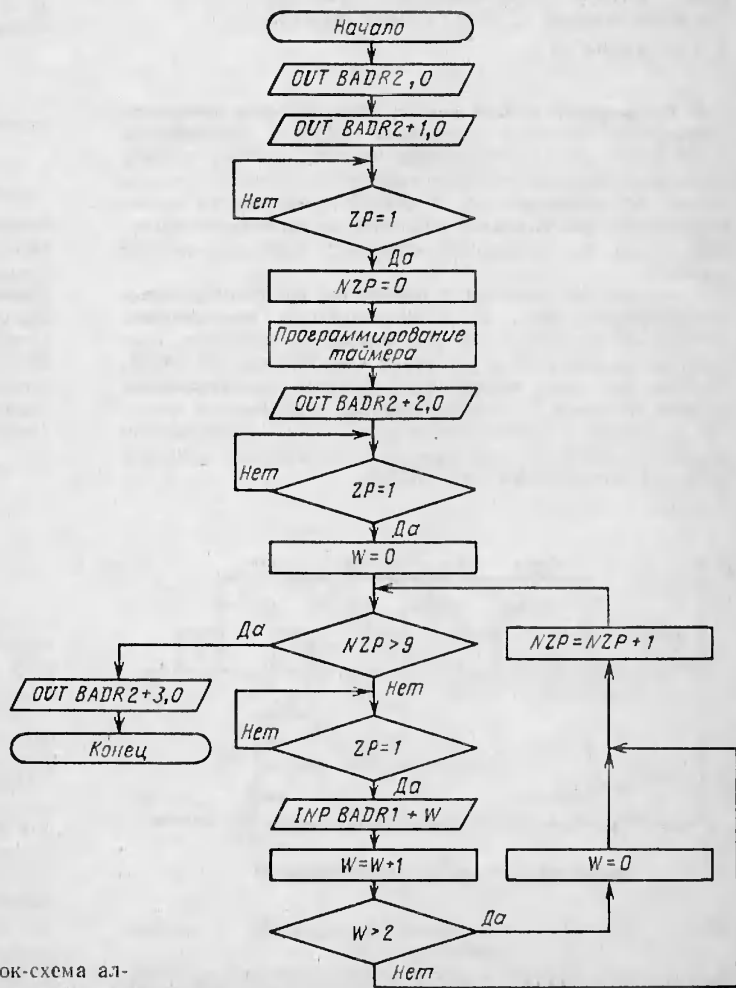


Рис. 3. Блок-схема алгоритма цикла измерения

редного по времени формирования запроса прерывания P8. После приема этого запроса и подтверждения прерывания МП осуществляет программирование ПТ и обязательно до начала формирования запроса прерывания P0 запуск ЦА (OUT BADR2+2,0).

Собственно преобразование начинается с момента формирования прерывания P0, после приема и подтверждения которого идентификатору W номера канала ПТ присваивается значение 0. Далее алгоритм приобретает циклический характер, описываемый табл. 2. После каждого чтения (INP BADR1+W) устанавливается счетчик соответствующего канала (на рис. 2 не показано).

Критическими с точки зрения быстродействия МП являются две процедуры алгоритма: программирование ПТ (должна быть закончена за время, меньшее интервала P8→0) и чтение УОЗ

(интервал P0→1) при больших значениях УОЗ. Расчеты и эксперимент показали, что при тактовой частоте МП КР580ИК80А  $f=2$  МГц,  $f_{дз} \leq 4000$  об/мин,  $УОЗ \leq 30^\circ$  и двухбайтном формате слов ПТ скорость выполнения критических участков ассемблированной программы вполне достаточна для правильной и надежной работы ИП. Отметим, что процедура оцифровки и ввода параметров ДЗ, указанных в табл. 2, занимает не более двух периодов сигнала ДП. По окончании цикла измерения могут быть вычислены косвенно измеряемые параметры ДЗ:  $УЗСК_i = 90^\circ - УРСК_i$ ;  $УОЗ_{i/i \neq 1} = 90^\circ - [AC_{(i,i+1)} - УОЗ_i]$ ;  $\overline{УОЗ} = \frac{1}{4} \sum_{i=1}^4 УОЗ_i$ .

Особенность применения ПТ в рассмотренном ИП — использование всех трех каналов. Это объективно не явля-

ется необходимым (можно обойтись двумя), но позволяет снизить требования к быстродействию МП либо при фиксированном быстродействии расширить сверху частотный диапазон исследований. При использовании только двух каналов ПТ и достаточном быстродействии МП третий канал может быть использован для широтно-импульсного управления двигателем, имитирующим распределительный вал. Применение только одного канала предполагает измерения в непрерывающихся циклах, соответствующую потерю скорости преобразования и существенно ужесточает требования к стабильности частоты вращения распределительного вала.

310002, Харьков, ул. Фрунзе, д. 21, ХПИ, кафедра «Информационно-измерительной техники», тел. 40-00-83

Статья поступила 23.05.88

УДК 681.325.5—181.4

А. И. Овчаренко, В. Е. Хилобок

## ПРЕОБРАЗОВАТЕЛЬ ПАРАМЕТРОВ ДАТЧИКОВ-РАСПРЕДЕЛИТЕЛЕЙ ЗАЖИГАНИЯ С ИНТЕРФЕЙСОМ КТС ЛИУС-2

В предыдущей работе рассмотрены вопросы измерения параметров датчиков зажигания (ДЗ) с применением КР580ВИ53 для случая многоимпульсного датчика частоты вращения, сигналом которого квантуются соответствующие временные интервалы ДЗ. В данной статье описан преобразователь, реализующий квантование постоянной частотой, т. е. не требующий сложного многоимпульсного датчика.

Контроль ДЗ сводится к измерению время-импульсных характеристик (рис. 1) и последующим вычислениям. Измерение характеристик ДЗ можно организовать программно во внутренних регистрах микропроцессора (МП). Однако при этом требуется аппаратная идентификация номера импульса ДЗ, долговременная стабильность частоты вращения и значительное время для осуществления контроля. Поэтому предлагается аппаратное решение задачи с применением КР580ВИ53.

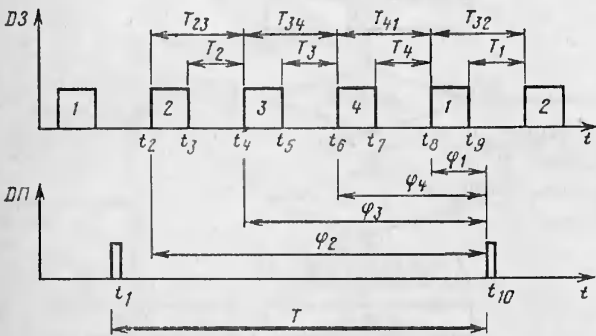


Рис. 1. Временные диаграммы сигналов ДЗ и датчика положения (ДП):

$T_{ij}$  — временной интервал между соседними импульсами зажигания,  $j=i+1, \text{ mod } 4, i=1...4$ ;  $T_i$  — время замкнутого состояния контактов ДЗ;  $\varphi_i$  — время опережения зажигания относительно импульса ДП;  $T$  — период следования импульсов ДП

Структура преобразователя и его подключение к интерфейсной магистрали ИК-1 КТС ЛИУС-2 приведены на рис. 2. Рассмотрим функционирование преобразователя. На этапе загрузки производится настройка каналов КР580ВИ53 на режим программируемой задержки (режим 0) и запись в счетчики каналов начальных значений (табл. 1).

Таблица 1

Операции загрузки КР580ВИ53

Операции	Данные	Изменяемая часть АЗ...А7 адреса
Управляющее слово канала 0 и сброс первого триггера	30H	ВН
Управляющее слово канала 1	70H	3Н
Управляющее слово канала 2	В0H	3Н
Младший байт канала 0	FFH	0Н
Старший байт канала 0	FFH	0Н
Младший байт канала 1	FFH	1Н
Старший байт канала 1	FFH	0Н
Младший байт канала 2	FFH	2Н
Старший байт канала 2	FFH	2Н

Процесс загрузки в соответствии с архитектурой интерфейса ИК-1 сопровождается формированием сигнала ОТВ при совпадении неизменяемой части адреса А7...А4 с настройкой дешифратора.

По окончании загрузки начинается собственно измерение временных интервалов. Операции обмена на этом этапе перечислены в табл. 2.

Процесс измерения заключается в следующем. После запуска D-триггера подготавливается к срабатыванию и при подаче импульса ДП устанавливается в единичное состояние. При этом открываются оба ключа и квантовая частота  $f_{кв}$  поступает на входы CLK всех трех каналов КР580ВИ53; на вход GATE канала 0 поступают временные интервалы  $T_{ij}$  —  $T_i$ , на вход GATE канала 1 — временные интервалы  $T_i$  (время замкнутого состояния контактов), а также первый  $t_2-t_1$  и последний  $t_{10}-t_9$  интервалы, на вход GATE канала 2 — интервал  $T$ . Процессы квантования временных интервалов и чтения результатов в



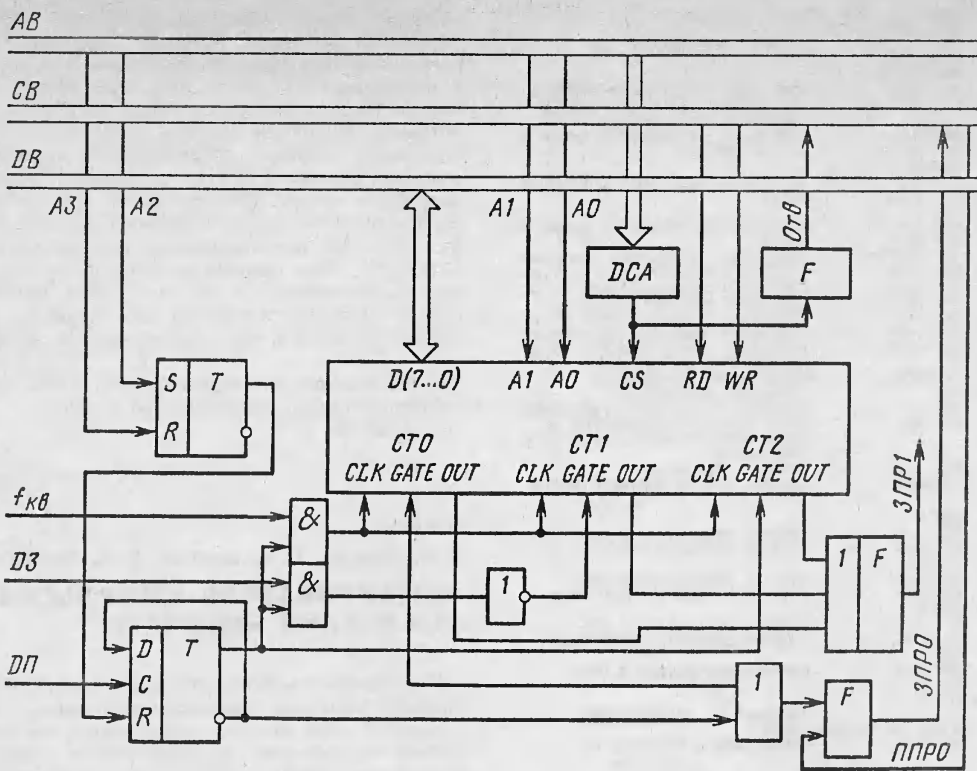


Рис. 2. Структурная схема преобразователя

Таблица 2

Операции чтения из КР580ВИ53

Операции	Изменяемая часть А3...А0 адреса
Запуск	7Н
Младший байт канала 0	0Н
Старший байт канала 0	0Н
Младший байт канала 1	1А
Старший байт канала 1	1А
Младший байт канала 2	2Н
Старший байт канала 2	2Н

Таблица 3

Информация в ОЗУ

Адрес ячейки ОЗУ	Код	Адрес ячейки ОЗУ	Код
4003Н, 4004Н	$(t_2 - t_1) f_{кв}$	400ДН, 400ЕН	$(t_7 - t_6) f_{кв}$
4005Н, 4006Н	$(t_3 - t_2) f_{кв}$	400ФН, 4010Н	$(t_8 - t_7) f_{кв}$
4007Н, 4008Н	$(t_4 - t_3) f_{кв}$	4011Н, 4012Н	$(t_9 - t_8) f_{кв}$
4009Н, 400АН	$(t_5 - t_4) f_{кв}$	4013Н, 4014Н	$(t_{10} - t_9) f_{кв}$
400ВН, 400СН	$(t_6 - t_5) f_{кв}$	4015Н, 4016Н	$(t_{10} - t_1) f_{кв}$

каналах 0 и 1 чередуются, т. е. при квантовании в канале 0 из канала 1 осуществляется чтение и наоборот. Эти процессы управляются с помощью импульсов запроса прерывания ЗПРО по фронтам и срезам сигналов второго ключа и инверсного выхода D-триггера. Наличие сигнала ЗПРО интерпретируется МП, как необходимость чтения из соответствующего канала (0 или 1) с последующей записью в этот канал начального значения (FFFFH).

Выходы каналов ОУТ0...ОУТ2 объединены элементом ИЛИ с целью формирования сигнала ЗПР1, свидетельствующего о переполнении одного из каналов, т. е. о несоответствии уровня  $f_{кв}$  квантовому временному интервалу (аварийный останов).

Длительность этапа измерений равна T и завершается вторым импульсом ДП (момент  $t_{10}$ ), который устанавливает D-триггер в импульсное состояние. На линии ЗПРО формируется десятый по счету импульс. В ответ МП выставляет на линии А3 Лог. 0, устанавливая R, S и D-триггеры

Таблица 4

Алгоритмы вычисления угловых характеристик ДЗ

Характеристика	Алгоритм вычисления
Асинхронизм	$A_{ij} = T_i \cdot 360 / T - 90$ , где $T_{12} = T - T_{23} - T_{34} - T_{41}$ , $T_{23} = t_4 - t_2$ , $T_{34} = t_6 - t_4$ , $T_{41} = t_8 - t_6$ , $T = t_{10} - t_1$
Угол замкнутого состояния контактов	$U_i = 360 T_i / T$ , где $T_1 = T_{12} - t_9 + t_8$ , $T_2 = t_4 - t_3$ , $T_3 = t_5 - t_5$ , $T_4 = t_2 - t_7$
Угол опережения зажигания	$F_i = \begin{cases} \varphi_i 360 / T - 90 (i+1) \bmod 4, & \text{если } i > 1 \\ \varphi_i 360 / T, & \text{если } i = 1 \end{cases}$ где $\varphi_1 = t_{10} - t_8$ , $\varphi_2 = t_{10} - t_1$ , $\varphi_4 = t_{10} - t_6$

```

10 ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ ТАЙМЕРА КР580ВИ53
20 ORG 3000H
30 DI ;ЗАПРЕТ ПЕРЕРЫВАНИЯ
40 PUSH B
50 PUSH H
60 MVI B,10 ;ЗАГРУЗКА СЧЕТЧИКА ПЕРЕРЫВАНИЙ
70 LXI H,4000H ;УСТАНОВКА НАЧАЛЬНОГО АДРЕСА
80 ;ХРАНЕНИЯ УПРАВЛЯЮЩЕГО СЛОВА В
90 MOV M,M ;ЗАГРУЗКА УПРАВЛЯЮЩЕГО СЛОВА В
100 OUT 0B0H ;СТ.0, СБРОС
110 INX H
120 MOV A,M ;ЗАГРУЗКА УПРАВЛЯЮЩЕГО СЛОВА В
130 OUT 0B3H ;СТ.1
140 INX H
150 MOV A,M ;ЗАГРУЗКА УПРАВЛЯЮЩЕГО СЛОВА В
160 OUT 0B3H ;СТ.2
170 MVI A,0FFH ;УСТАНОВКА НАЧАЛЬНОГО ЗНАЧЕНИЯ
180 OUT 0B0H ;ЗАГРУЗКА МЛАДШЕГО БАЙТА СТ.0
190 OUT 0B0H ;ЗАГРУЗКА СТАРШЕГО БАЙТА СТ.0
200 OUT 0D1H ;ЗАГРУЗКА МЛАДШЕГО БАЙТА СТ.1
210 OUT 0D1H ;ЗАГРУЗКА СТАРШЕГО БАЙТА СТ.1
220 OUT 0D2H ;ЗАГРУЗКА МЛАДШЕГО БАЙТА СТ.2
230 OUT 0D2H ;ЗАГРУЗКА СТАРШЕГО БАЙТА СТ.2
240 EI ;РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЯ
250 IN 0D7H ;ПУСК НА ПРЕОБРАЗОВАНИЕ
260 INT: HLT ;ОЖИДАНИЕ ПЕРЕРЫВАНИЯ
270 DCR B ;ДЕКРЕМЕНТ СЧЕТЧИКА ПЕРЕРЫВАНИЙ
280 JNZ INT ;ЕСЛИ ЦИКЛ ПРЕОБРАЗОВАНИЯ НЕ
290 ;ЗАВЕРШЕН, ТО ЖДАТЬ СЛЕДУЮЩЕГО
300 ;ПЕРЕРЫВАНИЯ
310 DI ;ЗАПРЕТ ПЕРЕРЫВАНИЯ
320 IN 0B6H ;СБРОС, ЧТЕНИЕ МЛАДШЕГО БАЙТА
; СТ.1

330 INX H
340 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
350 IN 0D1H ;ЧТЕНИЕ СТАРШЕГО БАЙТА СТ.1
360 INX H
370 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
380 IN 0D2H ;ЧТЕНИЕ МЛАДШЕГО БАЙТА СТ.2
390 INX H
400 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
410 IN 0D2H ;ЧТЕНИЕ СТАРШЕГО БАЙТА СТ.2
420 INX H
430 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
440 POP H
450 POP B
460 RET
470 ;ПОДПРОГРАММА ОБРАБОТКИ СИГНАЛА ЗПР0
480 MOV A,B ;ЗАЯСНЕНИЕ В АККУМУЛЯТОР
; СОСТОЯНИЯ
490 CPI 10 ;СЧЕТЧИКА ПЕРЕРЫВАНИЙ
500 JZ R5 ;ЕСЛИ 1-Е ПЕРЕРЫВАНИЕ, ТО
510 CPI 9 ;ПЕРЕЙТИ НА R5Т0
520 JZ ST1 ;ЕСЛИ 2-Е ПЕРЕРЫВАНИЕ, ТО
530 CPI 8 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.1
540 JZ ST1 ;ЕСЛИ 3-Е ПЕРЕРЫВАНИЕ, ТО
550 CPI 7 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.0
560 JZ ST1 ;ЕСЛИ 4-Е ПЕРЕРЫВАНИЕ, ТО
570 CPI 6 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.1
580 JZ ST0 ;ЕСЛИ 5-Е ПЕРЕРЫВАНИЕ, ТО
590 CPI 5 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.0
600 JZ ST1 ;ЕСЛИ 6-Е ПЕРЕРЫВАНИЕ, ТО
610 CPI 4 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.1
620 JZ ST0 ;ЕСЛИ 7-Е ПЕРЕРЫВАНИЕ, ТО
630 CPI 3 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.0
640 JZ ST1 ;ЕСЛИ 8-Е ПЕРЕРЫВАНИЕ, ТО
650 CPI 2 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.1
660 JZ ST0 ;ЕСЛИ 9-Е ПЕРЕРЫВАНИЕ, ТО
670 JZ ST0 ;ПЕРЕЙТИ К ОБМЕНУ С СТ.0
680 EI ;РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЯ
690 RS: RST 0 ;ВОЗВРАТ ИЗ ПОДПРОГРАММЫ
700 ST1: IN 0D1H ;ЧТЕНИЕ МЛАДШЕГО БАЙТА СТ.1
710 INX H
720 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
730 IN 0D1H ;ЧТЕНИЕ СТАРШЕГО БАЙТА СТ.1
740 INX H
750 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
760 MVI A,0FFH ;УСТАНОВКА НАЧАЛЬНОГО ЗНАЧЕНИЯ
770 OUT 0B3H ;ЗАГРУЗКА МЛАДШЕГО БАЙТА СТ.1
780 OUT 0D1H ;ЗАГРУЗКА СТАРШЕГО БАЙТА СТ.1
790 EI ;РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЯ
800 RST 0 ;ВОЗВРАТ ИЗ ПОДПРОГРАММЫ
810 ST0: IN 0D0H ;ЧТЕНИЕ МЛАДШЕГО БАЙТА СТ.0
820 INX H
830 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
840 IN 0D0H ;ЧТЕНИЕ СТАРШЕГО БАЙТА СТ.0
850 INX H
860 MOV M,A ;ЗАПИСЬ РЕЗУЛЬТАТА В ОЗУ
870 MVI A,0FFH ;УСТАНОВКА НАЧАЛЬНОГО ЗНАЧЕНИЯ
880 OUT 0B0H ;ЗАГРУЗКА МЛАДШЕГО БАЙТА СТ.0
890 OUT 0D0H ;ЗАГРУЗКА СТАРШЕГО БАЙТА СТ.0
900 EI ;РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЯ
910 RST 0 ;ВОЗВРАТ ИЗ ПОДПРОГРАММЫ
920 END

```

Рис. 3. Подпрограмма инициализации таймера КР580ВИ53

в нулевое состояние. По окончании описанного этапа в ОЗУ хранится информация (табл. 3).

На третьем этапе МП вычисляет угловые характеристики ДЗ (табл. 4).

На рис. 3 приведена подпрограмма инициализации КР580ВИ53, написанная в системе команд МП КР580ИК80. Текст подпрограммы содержит достаточно подробные ком-

ментарии, однако некоторые ее фрагменты требуют дополнительных пояснений. Пара HL используется в качестве регистра адреса хранения управляющих слов и результатов измерения в ОЗУ. Начальный адрес 4000H выбран произвольно. Регистр В выполняет функции счетчика прерываний, который обнуляется десятим импульсом по линии ЗПР0. Начальное значение, загружаемое в каналы КР580ВИ53, выбрано равным FFFFH (необходимо использовать всю разрядную сетку). Базовый адрес преобразователя равен D0H, однако может быть изменен настройкой дешифратора. Пуск на преобразование производится по команде INOD 7H. Это связано с необходимостью активизации только дешифратора и линии A2. Для КР580ВИ53 в этом случае выполняется команда «нет операции» и буфер шины данных остается в высокоимпедансном состоянии.

310002, Харьков, ул. Фрунзе, д. 21, ХПИ, кафедра «Информационно-измерительной техники», тел. 40-00-83

Статья поступила 23.06.88

УДК 621.327

В. В. Лангуев, С. А. Болотов, В. А. Трусов

## ФОРМИРОВАТЕЛЬ ИМПУЛЬСОВ ЗАЖИГАНИЯ КМ1823АГ1

Формирователь импульсов зажигания КМ1823АГ1 вырабатывает импульсы управления электронным коммутатором первичной цепи катушки зажигания с постоянной скважностью и импульсы, используемые в резервном режиме работы контроллера микропроцессорной системы зажигания с постоянным углом опережения зажигания.

Состав формирователя (рис. 1):

формирователь коротких импульсов (ФКИ) по каждому фронту углового импульса, поступающего на вход SYN микросхемы (на его выходах — две серии импульсов: по фронту и срезу углового импульса);

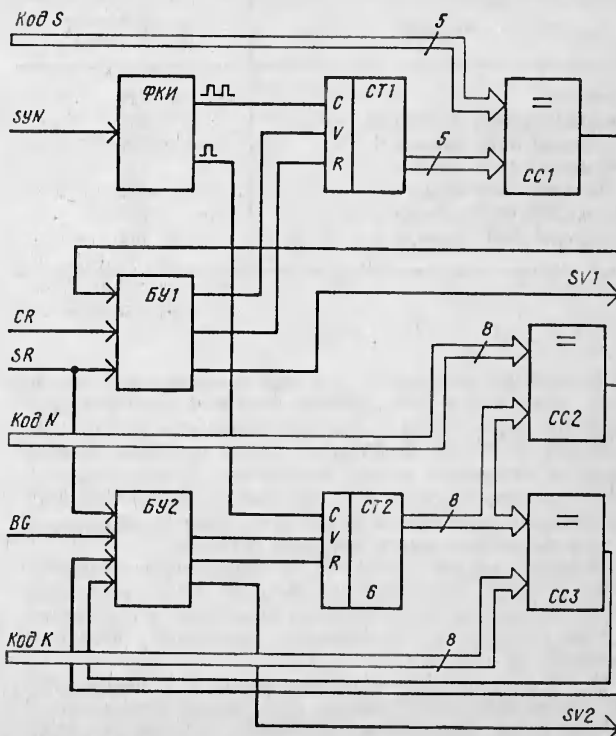


Рис. 1. Структурная схема микросхемы КМ1823АГ1

счетчик СТ1, схема сравнения СС1 и блок управления (БУ1) для выдачи импульсов зажигания с постоянной скважностью; БУ2, СТ2, СС2 и СС3 для создания резервных импульсов зажигания.

Условное графическое обозначение микросхемы приведено на рис. 2.

#### Назначение выводов микросхемы КМ1823АГ1

1..8	Входы программирования кода N	D0...D7
9...12	Входы программирования кода S	D16...D19
13	Вход начальной установки SR	
14	Общий ОУ	
15...22	Входы программирования кода K	D8...D15
23	Вход импульса начала отсчета BG	
24	Выход резервного импульса зажигания SV2	
25	Вход импульса зажигания CR	
26	Выход импульса зажигания постоянной скважности SV1	
27	Вход угловых импульсов SYN	
28	Напряжение питания U <sub>cc</sub>	

При подаче на вход SR импульса отрицательной полярности происходит начальная установка формирования. Импульсы зажигания с постоянной скважностью формируются по входному сигналу, определяющему момент зажигания и подаваемому на вход CR микросхемы. По этому сигналу на выходе микросхемы устанавливается Лог. 0, т. е. подается срез импульса зажигания с постоянной скважностью. Одновременно счетчик СТ1 обнуляется по сигналу из БУ1 и снимается сигнал блокировки счета (начинается подсчет импульсов, поступающих на счетный вход с выхода ФКИ). Содержимое СТ1 сравнивается с кодом S (на выходе СС1 Лог. 1). БУ1 выдает сигнал, блокирующий работу СТ1 (на выходе SV1 Лог. 1, фронт импульса зажигания с постоянной скважностью). С появлением на входе CR следующего импульса, определяющего момент зажигания, описанный процесс повторяется.

Между входными сигналами, определяющими момент зажигания, вырабатывается постоянное число угловых импульсов, поэтому скважность на выходе SV1 будет зависеть только от величины кода S.

Код S 5-разрядный, значения четырех младших разрядов определяются логическими уровнями на входах D16...D19 микросхемы, значение пятого равно инверсному значению четвертого разряда (вход D19).  $S = Z/2q - 1$ , где Z — число УИ, подаваемых на вход SYN между двумя соседними импульсами начала отсчета (не превышает 150), q — требуемая скважность импульсов зажигания (близка к трем).

Резервные импульсы зажигания формируются при подаче импульса начала отсчета на вход BG микросхемы. БУ2 обнуляет СТ2 и снимает с него сигнал блокировки счета, СТ2 считает импульсы, поступающие из ФКИ, и сравнивает с величиной кода K. При достижении СТ2 этого значения сигнал на выходе СС2 переключается

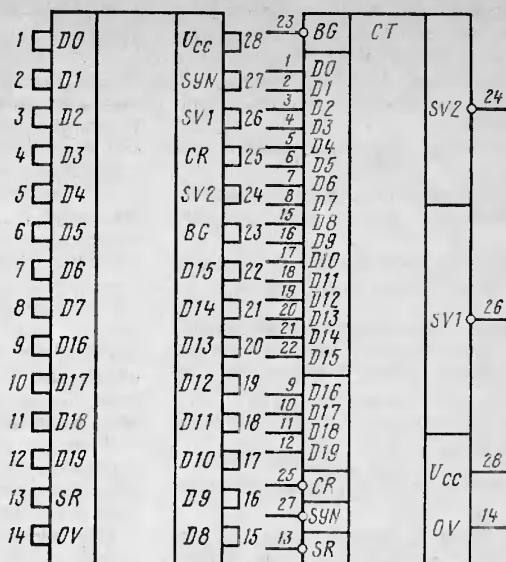


Рис. 2. Условное графическое обозначение микросхемы КМ1823АГ1

в Лог. 1, БУ2 устанавливает на выходе SV2 микросхемы Лог. 0, формируя срез резервного импульса зажигания, который определяет момент зажигания. СТ2 продолжает счет импульсов до сравнения с кодом K. При этом по сигналу Лог. 1, поступающему с выхода СС3, БУ2 обнуляет СТ2 и устанавливает на выходе SV2 Лог. 1 (создается фронт резервного импульса зажигания). Далее СТ2 продолжает счет импульсов и работа схемы повторяется. Числовое значение восьмизначного кода K равно числу угловых импульсов, подаваемых на вход между двумя соседними импульсами начала отсчета.  $N = K - \Theta \cdot Z / 180$ , где  $\Theta$  — требуемый угол опережения зажигания.

Вычисленные значения кодов S, N, K должны быть переведены в двоичную форму.

#### Основные характеристики микросхемы КМ1823АГ1

Диапазон рабочих температур, °С	-60...+100
Напряжение питания U <sub>cc</sub> , В	5±10%
Ток потребления в диапазоне температур -60...+100 °С, мА, не более	70
Выходной ток низкого уровня, мА, при U <sub>OL</sub> =0,4 В, не менее	0,8
Выходной ток высокого уровня, мА, при U <sub>OH</sub> =(U <sub>cc</sub> -0,4) В, не менее	0,4

Телефон 77-44-32, Минск

Статья поступила 29.03.88

УДК 621.327

В. В. Лангуев, А. И. Ольшак, С. Ю. Гаврилюк

## ПРОЦЕССОР КМ1823ВУ1

Процессор содержит следующие функциональные блоки (рис. 1):

арифметическое устройство для выполнения операций алгебраического сложения 8-разрядного операнда, поступающего из регистра-аккумулятора и 2-разрядного со знаком, поступающим с шины данных;

регистр-аккумулятор, осуществляющий хранение результата сложения;

буферный регистр для временного хранения кода угла опережения зажигания;

счетчик угла опережения зажигания для определения заданного угла поворота коленчатого вала двигателя с помощью подсчета угловых импульсов (их число задается кодом, переписываемым в счетчик из буферного регистра);

таймер, в заданном интервале которого измеряется часть вращения коленчатого вала двигателя;

программируемый делитель для формирования сигнала стробирования регистра-аккумулятора и сигнала декремента счетчика повторений (вы-

рабатывается при поступлении на его вход заданного числа угловых импульсов), определяемого по информации на программных входах;

счетчик повторений заданного числа выходных импульсов программируемого делителя, который выдает сигнал для блока управления при переполнении;

блок управления электроклапаном ЭПХХ для создания управляющего сигнала в соответствии с программой, хранящейся во внешнем ПЗУ;

делитель частоты с коэффициентом деления, равным восьми, понижающий входную тактовую частоту до значения внутренней тактовой частоты;

блок управления узлами процессора,

вырабатывающий сигналы записи для таймера, регистра-аккумулятора, счетчика/повторения, блока управления электроклапаном ЭПХХ, а также сигналы управления таймером и счетчиком адреса;

блок внешней коррекции опережения зажигания;

счетчик адреса, формирующий шестизначный код адреса для внешнего ПЗУ.

Условное графическое обозначение микросхемы приведено на рис. 2.

#### Назначение выводов микросхемы КМ1823ВУ1

- 1 — Выход внутренней тактовой частоты С
- 2 — Выход управления циклом измерения ST
- 3 — Вход угловых импульсов SYN
- 4 — Вход импульсов «Начало отсчета» BG
- 5 — Вход корректора GOCR
- 6 — Выход импульса зажигания CR
- 7...12 — Выходы адреса А5...А0
- 13 — Выходы управления электропневмоклапаном ЭПХХ CO
- 14 — OV
- 15 — Выход внутреннего генератора GN2
- 16 — Вход подтверждения конца преобразования АЦП CN
- 17 — Вход «Установка в начальное состояние» SR
- 18 — Вход внутреннего генератора GN1
- 19 — Вход старшего разряда регистра таймера D8
- 20...27 — Входы данных D7...D0
- 28 — U<sub>cc</sub>

#### Принцип вычисления угла опережения зажигания

Оптимальная работа двигателя внутреннего сгорания обеспечивается управлением углом зажигания в зависимости от нагрузки двигателя, характеризующейся абсолютным давлением во впускном трубопроводе и частотой вращения коленчатого вала двигателя. Диапазон изменения давления разбивается на заданное число участков, каждому из которых соответствует изменение угла

зажигания от частоты вращения коленчатого вала двигателя (рис. 3).

Частота вращения коленчатого вала двигателя определяется подсчетом угловых импульсов за фиксированный интервал времени. Число угловых импульсов, формируемых за один оборот коленчатого вала двигателя, всегда постоянно. Таким образом, каждому угловому импульсу, поступившему в течение измерительного интервала, соответствует приращение скорости на величину

$$\Delta n = 60 \cdot 10^6 / Z \cdot T_{\text{И}} (\text{мин}^{-1}),$$

где Z — число угловых импульсов, поступающих за один оборот вала двигателя, T<sub>И</sub> — длительность измерительного интервала в микросекундах.

В то же время, появление каждого углового импульса соответствует повороту коленчатого вала двигателя на угол  $\Delta\varphi = 360^\circ / Z$ .

Угол опережения зажигания (УОЗ) в этом случае удобно задавать числом угловых импульсов. Учитывая это, график зависимости угла зажигания от частоты вращения коленчатого вала двигателя можно разбить на линейные участки, в пределах которых изменения постоянны. Каждый линейный участок можно задать таким образом, чтобы приращению частоты вращения, выраженному целым числом угловых импульсов, соответствовало приращение угла зажигания, также выраженное целым числом угловых импульсов. Например, для участка характеристики (рис. 3), расположенного в зоне 0...400 мин<sup>-1</sup>, приращению частоты вращения на 4Δn (Δn=25 мин<sup>-1</sup>) соответствует приращение угла зажигания на Δφ.

Угол зажигания определяется суммированием в течение измерительного ин-

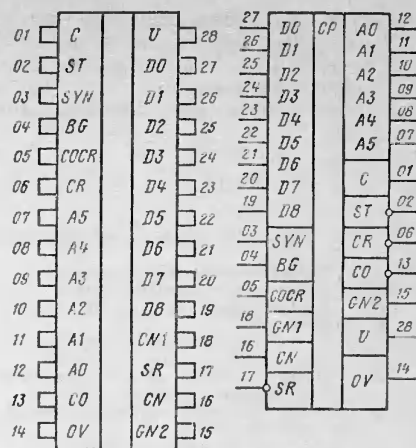


Рис. 2. Условное графическое обозначение БИС КМ1823ВУ1

тервала приращений угла зажигания, начиная со значения, соответствующего нулевой частоте вращения коленчатого вала двигателя.

Преимущество данного принципа — отсутствие необходимости интерполяции для повышения точности вычисления УОЗ, совмещение во времени процессов измерения частоты вращения вала двигателя и вычисления значения УОЗ, соответствующего измеренной частоте.

#### Форма представления информации

Описанный принцип вычисления УОЗ реализуется аппаратными средствами процессора, работающего по жесткому алгоритму. Информация о характеристиках управления поступает в процессор из внешнего ПЗУ, при этом каждому линейному участку характеристики ставятся в соответствие А, М и N — целые числа, определяющие значение приращений угла зажигания (в пределах ±3Δφ), число угловых импульсов, при поступлении которых изменяется величина УОЗ (в пределах 1...15), и число приращений значения УОЗ на данном участке характеристики (в пределах 1...16).

Таким образом, величина участка по частоте вращения определяется как произведение NΔn, а максимальное приращение УОЗ для этого участка — AN.

Для кодирования всей информации, описывающей зависимость величины УОЗ от частоты вращения, можно использовать либо 32, либо 64 байта, при этом, в нулевой байте размещается константа, определяющая длительность измерительного интервала (СТ), а в первом — код начального значения угла опережения зажигания А<sub>0</sub>.

Так как счетчик УОЗ процессора 8-разрядный, а формирование сигнала, определяющего момент зажигания, про-

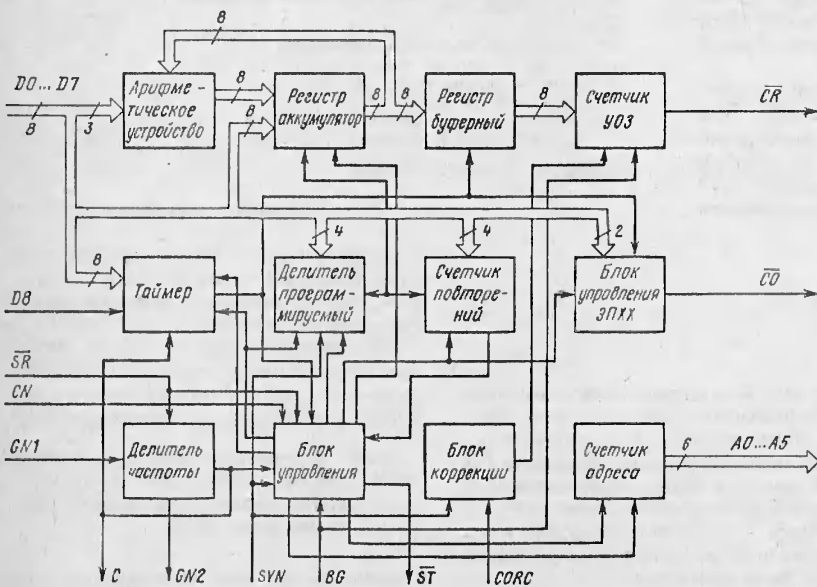


Рис. 1. Структурная схема БИС КМ1823ВУ1

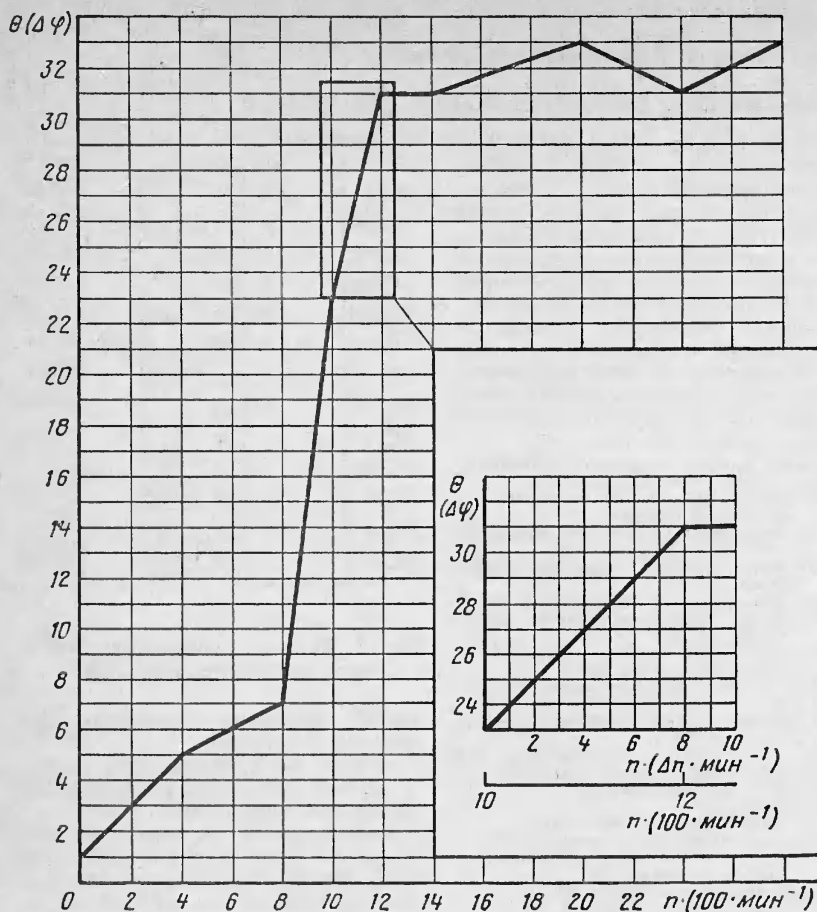


Рис. 3. Пример графика зависимости угла опережения зажигания

исходит при его переполнении, то  $A_0 = 256 - \frac{Z}{L} + F_0$ , где  $Z$  — число угловых импульсов;  $L$  — импульсов зажигания, формируемых за один оборот коленчатого вала двигателя;  $F_0$  — табличное начальное значение УОЗ, выраженное через  $Z$ .

Таймер (13-разрядный) считает импульсы внутренней тактовой частоты, однако возможно программирование только девяти старших разрядов, поэтому код длительности временного интервала рассчитывается по формуле:  $ST = T_H / 128 t_T$ , где  $t_T$  — период следования импульсов входной тактовой частоты БИС процессора.

Программирование старшего разряда производится с помощью подключения входа D9 микросхемы к шине питания либо общей шине устройства, а остальные восемь разрядов кодом, поступающим в процессор из внешнего ПЗУ по шине данных.

Так как в БИС процессора используются инвертирующие входные вентили, полученные значения кодов  $A_0$  и  $ST$  должны быть проинвертированы. Для кодирования каждого участка ха-

рактеристики управления УОЗ используются два байта ПЗУ.

Информация записывается в соответствии со схемой рис. 4.

Коды управления электроклапаном ЭПХХ

- 00 — Не используется
- 01 — Включение ЭПХХ
- 10 — Выключение ЭПХХ
- 11 — Сохранение предыдущего состояния

#### Работа процессора

При подаче напряжения питания возникает сигнал  $\overline{SR}$ , по которому блок управления вырабатывает сигналы установки в нуль счетчика адреса и таймера (на выходах  $A_0 \dots A_5$  и  $\overline{ST}$  — низкий уровень).

После снятия сигнала начальной установки блок управления переписывает информацию с шины данных в таймер (код  $ST$ ), т. е. осуществляется программирование длительности измерительного интервала и схема переходит в состояние ожидания сигнала «Подтверждение конца преобразования АЦП».

Если этот сигнал не поступает на вход  $\overline{CN}$  за время следования двух импульсов

начала отсчета, то входящая в состав блока управления схема контроля работоспособности АЦП принудительно завершает ожидание.

При поступлении сигнала или при принудительном завершении ожидания по срезу следующего тактового импульса блок управления  $С01$  выдает сигнал инкрементирования счетчика адреса. На шине данных устанавливается информация, выбранная из первого байта ПЗУ и соответствующая коду начального значения УОЗ ( $A_0$ ). По следующему тактовому импульсу сигнал записи информации с шины данных поступает в регистр аккумулятора  $RGA$ . После окончания импульса записи подается сигнал инкрементирования счетчика адреса — на шине данных возникает информация, содержащая коды числа  $N$  и управления ЭПХХ. По следующему тактовому импульсу блок управления фор-

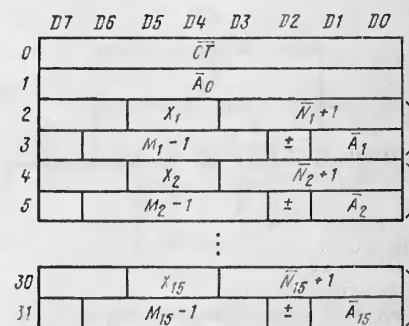


Рис. 4. Размещение информации в ПЗУ

мирует сигнал записи в счетчик повторений и блок управления ЭПХХ.

При поступлении углового импульса блок управления формирует сигнал, разрешающий работу программируемого делителя и таймера; начинается отсчет измерительного интервала и на выходе  $\overline{ST}$  появляется высокий логический уровень.

Заданное число угловых импульсов, определяется кодом на программных входах программируемого делителя, формирующего на своем выходе импульс стробирования регистра-аккумулятора. При этом в регистр-аккумулятор заносится информация с выхода арифметического устройства (производится операция алгебраического суммирования). Кроме того, импульс с выхода программируемого делителя приходит на счетный вход счетчика повторений и вызывает изменение его состояния. Описанный процесс повторяется до переполнения счетчика повторений.

Сигнал переноса, возникающий при этом, поступает в блок управления, который инкрементирует счетчик адреса. По следующему тактовому импульсу формируется сигнал записи в счетчик повторений и блок управления ЭПХХ;

## УСТРОЙСТВО ВВОДА-ВЫВОДА КМ1823ВВ1

счетчик адреса вновь инкрементируется и информация из следующего байта ПЗУ передается по шине данных на программируемый делитель и арифметическое устройство. Таким образом, осуществляется переход к следующему участку характеристики управления УОЗ. Далее процесс повторяется до окончания измерительного интервала.

На выходе таймера устанавливается импульс, вызывающий перезапись сформированного в регистре-аккумуляторе кода УОЗ в буферный регистр, задание состояния блока управления ЭПХХ в выходной триггер этого блока, а также переход схемы в исходное состояние, обнуляя счетчик адреса и таймер.

На выходах А0...А5 и СТ устанавливается низкий логический уровень. Далее весь процесс повторяется.

Угол поворота коленчатого вала двигателя в счетчике УОЗ вычисляется

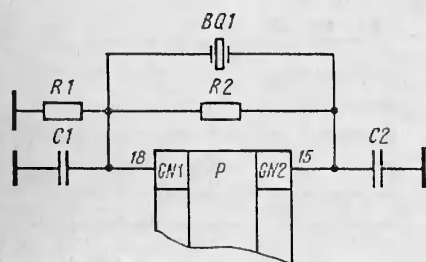


Рис. 5. Схема включения внутреннего генератора

независимо от остальной части схемы.

По импульсу начала отсчета на вход ВГ информация из буферного регистра переписывается в счетчик УОЗ, который подсчитывает угловые импульсы. При его переполнении вырабатывается сигнал переноса на выходе СR, управляющий моментом искрообразования.

## Основные характеристики микросхемы КМ1823ВУ1

Диапазон рабочих температур, °С	-60...+100
Напряжение питания $U_{CC}$ , В	$5 \pm 10\%$
Ток потребления в диапазоне рабочих температур мА, не более	70
Выходной ток низкого уровня, мА (при $U_{OD}=0,4$ В), не менее	0,8
Выходной ток высокого уровня, мА (при $U_{OH}=(U_{CC}-0,4)$ В), не менее	0,4
Частота тактового сигнала, кГц	4200

В качестве внутреннего генератора в БИС используется КМОП-инвертор, вход и выход которого соединены с выводами GN1 и GN2 (рис. 5).

При необходимости использования внешнего генератора его сигналы подаются на вход GN1 (выход GN2 остается неподключенным).

Телефон 77-44-32, Минск

Статья поступила 29.03.88

Примечание. X — любое состояние

## Назначение выводов микросхемы КМ1823ВВ1

- 01 — Выход сигнала зажигания CR2
- 02 — Вход концевых выключателя положения дроссельной заслонки SA
- 03 — Выход импульса зажигания первого канала COSW1
- 04 — Выход импульса зажигания второго канала COSW2
- 05 — Выход импульса выборки канала SESW
- 06 — Выход сигнала ограничения частоты МКCR2
- 07 — Выход импульса опроса преобразователя «напряжение-время» RQ
- 08 — Вход импульса зажигания CR1
- 09 — Вход угловых импульсов SYN1
- 10 — Выход угловых импульсов удвоенной частоты SYN2
- 11 — Выход импульса начала отсчета BG2
- 12 — Вход импульса начала отсчета BG1
- 13 — Вход тактовых импульсов C
- 14 — Общий 0V
- 15 — Вход начальной установки SR
- 16 — Вход импульса цикла измерений ST
- 17 — Вход импульса ответа преобразователя «напряжение-время» AN
- 18...23 — Выходы разрядов адреса А5...А0
- 24...27 — Входы разрядов данных D3...D0
- 28 — Напряжение питания  $U_{CC}$

## Состав УВВ (рис. 2):

формирователь НО, вырабатывающий сигнал НО, запаздывающий на 15 периодов сигнала угловых импульсов удвоенной частоты для обеспечения вычисления и отслеживания процессором КМ1823ВУ1 углов зажигания, запаздывающих относительно входного сигнала НО;

формирователь задержанного НО, устанавливающий сигнал, сдвинутый на полпериода с выхода формировате-

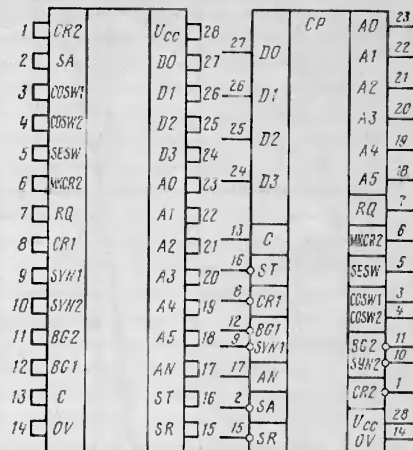


Рис. 1. Условное графическое обозначение микросхемы КМ1823ВВ1

ля НО, необходимый для функционирования процессора;

элемент ИЛИ, передающий сигналы с выходов НО и задержанного НО на выход ВГ2 микросхемы;

формирователь угловых импульсов удвоенной частоты, создающий по каждому фронту УИ импульсы длительностью в один период тактовой частоты (для повышения точности вычисления углов поворота коленчатого вала двигателя);

блок формирования адресов, генерирующий адреса внешнего ПЗУ;

регистр данных для хранения констант программирования, считываемых из внешнего ПЗУ;

блок анализа частоты, определяющий порог превышения частотой угловых импульсов заданного значения и блокирования сигналов управления электронным коммутатором катушек зажигания;

## Размещение констант А и Б в ПЗУ

Разряды адреса ПЗУ						Разряды шины данных			
A5	A4	A3	A2	A1	A0	D3	D2	D1	D0
0	0	1	0	1	1	A3	A2	A1	A0
0	0	1	0	1	0				
0	0	1	0	0	1	A7	A6	A5	A4
0	0	1	0	0	0				
0	0	0	1	1	1	B3	B2	B1	B0
0	0	0	1	1	0				
0	0	0	1	0	1				
0	0	0	1	0	0	X	X	B5	B4

Примечание. X — любое состояние

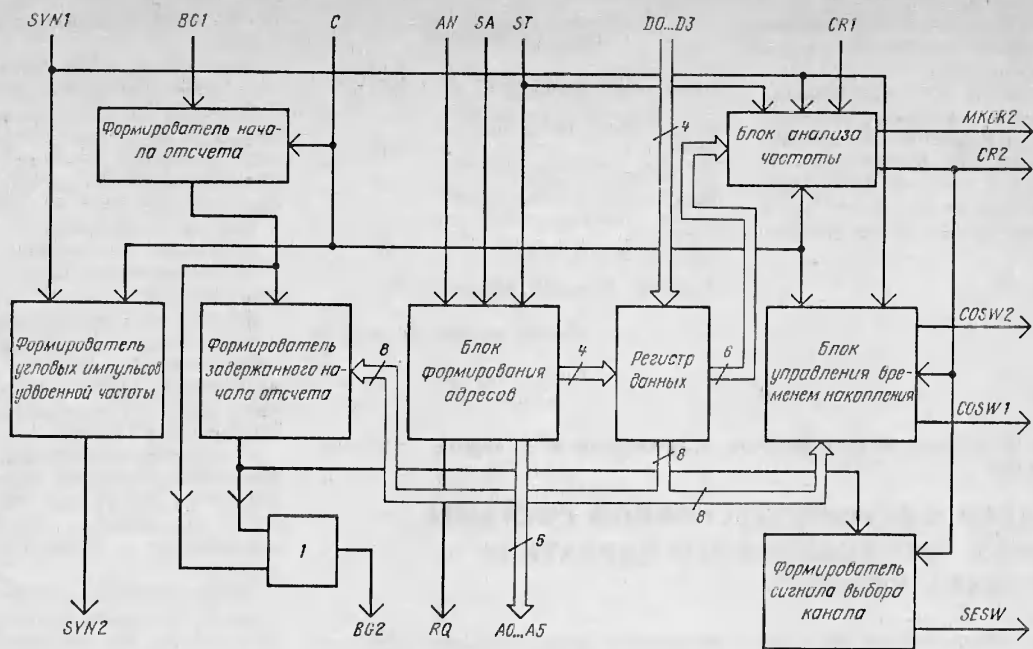


Рис. 2. Структурная схема микросхемы КМ1823ВВ1

блок управления временем накопления энергии в катушке зажигания;

формирователь сигнала выбора канала, управляющий двухканальным электронным коммутатором.

Начальная установка УВВ осуществляется при поступлении импульса длительностью не менее одного периода тактовой частоты на вход SR (при появлении на входе С тактовых импульсов микросхема готова к работе); программирование выполняется после подачи на вход ST сигнала цикла измерения (на выходах А0...А5 формируются адреса ячеек ПЗУ, в которых хранятся коды констант, соответствующих заданному типу двигателя внутреннего сгорания). Информация из ПЗУ тетрадами заносится в 14-разрядный регистр данных, где хранятся две константы (см. таблицу): 8-разрядный код числа угловых импульсов, вырабатываемых за один период сигнала НО, подаваемого на вход ВG1 (А), и код числа (В), определяющего частоту УИ (выходные сигналы управления электронным коммутатором первичных цепей катушек зажигания блокируются при превышении этого порога).

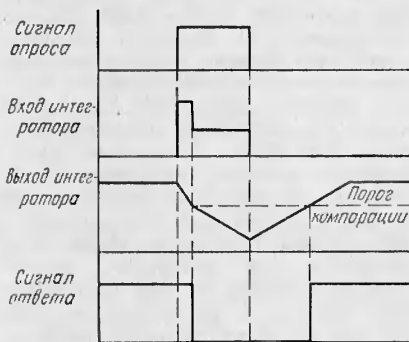


Рис. 4. Временные диаграммы работы преобразователя «напряжение-время»

$$A = 256 - Z, B = \frac{30 \cdot 10^6}{n_{\text{макс}} T Z}$$

где  $n_{\text{макс}}$  — наибольшая частота вращения коленчатого вала двигателя сгорания;  $T$  — период тактовой частоты;  $Z$  — число УИ, подаваемых за один период входного сигнала НО.

После окончания программирования микросхемы на выходе RQ формируется сигнал опроса преобразователя «напряжение-время», работающего по принципу двойного интегрирования (рис. 3).

Временные диаграммы, поясняющие работу преобразователя совместно с микросхемой КМ1823ВВ1, представлены на рис. 4.

По ответному сигналу преобразователя на входе AN блок формирования адресов начинает отсчет временного интервала (длительность равна 152 периодам тактовой частоты), который опре-

деляет продолжительность прямого интегрирования преобразователя. По окончании интервала прямого интегрирования сигнал опроса преобразователя «напряжение-время» снимается и начинается интервал обратного интегрирования.

После окончания обратного интегрирования преобразователь снимает сигнал ответа, подаваемый на вход AN микросхемы, и в блоке формирования адресов фиксируется 6-разрядный код, пропорциональный длительности временного интервала, сформированного преобразователем «напряжение-время». Этот код является адресом строки ПЗУ, в которой хранится характеристика управления углом опережения зажигания в зависимости от частоты вращения коленчатого вала двигателя внутреннего сгорания.

Предусмотрен резервный режим работы микросхемы при отсутствии сигнала ответа преобразователя «напряжение-время». В этом случае на выходах А0...А5 формируется код 001100, соответствующий средней величине временного интервала, созданного преобразователем.

Блок анализа частоты сравнивает величины частоты УИ и константы В. При превышении частотой УИ значения, заданного константой В, блок анализа частоты блокирует прохождение следующих сигналов: зажигания на выход CR2, выбора канала на выход SESW и COSW1. COSW2 на выход блока управления временем накопления энергии.

Сигналы CR2 и SESW управляют одно- или двухканальным коммутатором первичной цепи катушки зажига-

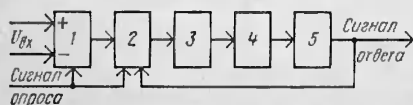


Рис. 3. Структурная схема преобразователя «напряжение-время»:

1 — нормирующий усилитель, 2 — аналоговый коммутатор, 3 — интегратор, 4 — компаратор напряжения, 5 — выходной ключ

ния, который самостоятельно распоряжается временем накопления энергии в катушке зажигания.

Блок управления временем накопления обеспечивает управление выходными ключами коммутатора с одновременным управлением временем накопления энергии в катушке зажигания. Время накопления энергии  $t_{\text{н}} = 7T_{\text{УИ}} + 2$  (мс), где  $T_{\text{УИ}}$  — период угловых импульсов.

#### Основные характеристики микросхемы КМ1823ВВ1

Диапазон рабочих температур, °С	-60... +100
Напряжение питания $U_{\text{СС}}$ , В	$5 \pm 10\%$
Ток потребления в диапазоне температур -60...+100 °С, мкА, не более	70
Выходной ток низкого уровня, мА, при $U_{\text{ОЛ}} = 0,4$ В и менее	0,8
Выходной ток высокого уровня, мА, при $U_{\text{ОН}} = (U_{\text{СС}} - 0,4)$ В, не менее	0,4
Тактовая частота, кГц	520

Телефон 77-44-32, Минск

Статья поступила 29.03.88

УДК 621.327

В. А. Бобков, В. В. Лангуев, С. А. Болотов, А. И. Ольшак, В. А. Трусов, С. Ю. Гаврилюк

## КОНТРОЛЛЕР МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ ЗАЖИГАНИЯ АВТОМОБИЛЬНОГО ДВИГАТЕЛЯ «ЭЛЕКТРОНИКА МС2713»

Контроллер «Электроника МС2713» построен на базе БИС серии К1823 и рассчитан на работу в составе системы зажигания автомобильного двигателя (рис. 1).

На входы контроллера поступают сигналы от датчиков начала отсчета, угловых импульсов, температуры охлаждающей жидкости и положения дроссельной заслонки. Датчик разрежения (входящий в состав контроллера) с помощью вакуумного патрубка соединяется с впускным трубопроводом двигателя. Индукционные датчики начала отсчета и угловых импульсов (14.3847) устанавливаются на двигателе таким образом, чтобы в их магнитном поле проходили маркерный зуб начала отсчета на маховике двигателя и зубья шестерни маховика соответственно. Датчик начала отсчета синхронизирует работу контроллера и двигателя и генерирует одиночный импульс в момент прохождения в его магнитном поле маркерного зуба. Взаимное положение датчика начала отсчета и маркерного зуба на маховике такое, что импульс начала отсчета формируется всегда в определенной фазе работы двигателя (например, в верхней мертвой точке поршня одного из цилиндров). Датчик угловых импульсов используется при вычислении частоты вращения и угла поворота коленчатого вала двигателя. На выходе датчика вырабатываются импульсы при прохождении в его магнитном поле зубьев шестерни маховика. С помощью полупроводникового датчика температуры охлаждающей жидкости (19.3828) определяются условия температурной коррекции характеристики управления углом опережения зажигания. Датчик положения дроссельной заслонки (концевой выключатель на карбюраторе) регистрирует ее состояние. По этой информации определяются режимы управления эко-

номайзером принудительного холостого хода.

На выходах контроллера формируются сигналы управления двухканальным коммутатором первичных цепей катушек зажигания (КЗ1 и КЗ2). Сигнал зажигания (СЗ) управляет временем накопления энергии в КЗ и устанавливает момент искрообразования. Сигнал выбора канала (ВК) указывает канал, коммутирующий первичную цепь (КЗ1 или КЗ2). С помощью двухвыводных катушек зажигания осуществляется бесконтактное распределение высокого напряжения на свечи соответствующих цилиндров. Искра возникает в паре цилиндров, но поскольку в одном из них происходит такт сжатия, а во втором — такт выпуска, то вторая искра не оказыва-

ет никакого влияния на работу двигателя.

Электромагнитный клапан экономайзера принудительного холостого хода (1902.3741) перекрывает канал топливopодачи в системе холостого хода карбюратора в режиме торможения двигателем при закрытой дроссельной заслонке. Питание на него подается с выхода контроллера в соответствии с заданным алгоритмом управления. Функциональные блоки контроллера (рис. 2):

преобразователи сигналов датчиков начала отсчета и УИ, реорганизующие квазисинусоидальный сигнал индукционного датчика в прямоугольные импульсы с логическими уровнями интегральных МОП схем;

преобразователь сигнала датчика положения дроссельной заслонки, формирующий логический сигнал с уровнями интегральных КМОП схем в соответствии с состоянием контактов датчика;

преобразователь порогового типа сигнала полупроводникового датчика температуры, вырабатывающий сигнал Лог. 1 или Лог. 0 в зависимости от достижения датчиком заданного порогового значения;

полупроводниковый датчик разрежения тензорезистивного типа, определяющий нагрузку на двигатель по величине разрежения во впускном трубопроводе двигателя и создающий на своем выходе постоянное напряжение, пропорциональное величине разрежения;

преобразователь «напряжение-время»;

устройство ввода-вывода; процессор КМ1823ВУ1;

ПЗУ (2К×8 бит) для хранения закодированных характеристик управления углом зажигания, клапаном ЭПХХ и служебных констант, необхо-

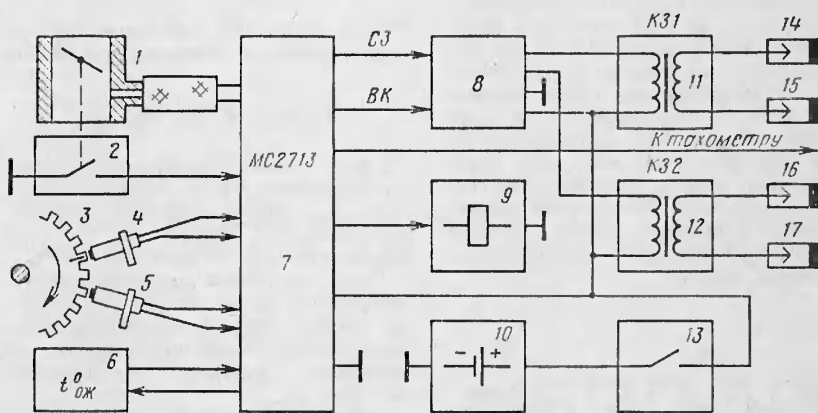


Рис. 1. Структурная схема микропроцессорной системы зажигания автомобильного двигателя:

1 — впускной трубопровод двигателя, 2 — датчик положения дроссельной заслонки, 3 — маховик двигателя, 4 — датчик начала отсчета, 5 — датчик угловых импульсов, 6 — датчик температуры охлаждающей жидкости, 7 — контроллер «Электроника МС2713», 8 — коммутатор, 9 — электромагнитный клапан экономайзера принудительного холостого хода, 10 — аккумуляторная батарея автомобиля, 11, 12 — катушки зажигания КЗ1 и КЗ2, 13 — выключатель зажигания, 14...17 — свечи зажигания.



димых для функционирования процессора и УВВ;

формирователь импульса зажигания (ФИЗ) КМ1823АГ1;

мощный выходной ключ управления ЭПХХ, переключающий электромагнитный клапан с активным сопротивлением обмотки  $33 \text{ Ом} \pm 10\%$  и имеющий защиту выхода от короткого замыкания на общий провод и от перегрузки по току;

выходные ключи сигналов СЗ и ВК для согласования выходов контроллера со входами коммутатора и защиты выходов микросхем от импульсных перенапряжений и помех;

блок питания, стабилизирующий напряжение питания всех узлов контроллера на уровне  $\pm 5 \text{ В} \pm 5\%$  во всем диапазоне питающих напряжений и температур окружающей среды.

Характеристика управления углом зажигания — зависимость величины угла зажигания от частоты вращения коленчатого вала и величины разрежения во впускном трубопроводе двигателя (вместе со служебными константами) занимает 1 Кбайт информационной емкости ПЗУ. В ПЗУ хранятся две такие характеристики — для прогретого двигателя (температура охлаждающей жидкости выше  $65^\circ\text{C}$ ) и непрогретого (ниже  $65^\circ\text{C}$ ). Логический уровень с выхода преобразователя сигнала датчика температуры охлаждающей жидкости поступает на старший (А10) разряд адреса ПЗУ и выбирает характеристику. Пять следующих разрядов адреса ПЗУ (А9...А5) формируются УВВ по результату измерения величины разрежения во впускном трубопроводе двигателя и позволяют выбрать из ПЗУ любую из 32 страниц, в каждой из которых хранится информация, описывающая зависимость изменения величины угла зажигания от частоты вращения коленчатого вала двигателя, соответствующую измеренному разрежению. Разрешающая способность контроллера составляет 20 мм рт. ст. по разрежению во впускном трубопроводе и  $25 \text{ мин}^{-1}$  — по частоте вращения коленчатого вала двигателя. Угол зажигания изменяется с дискретностью, равной или кратной половине периода входных угловых импульсов.

**Работа контроллера.** Сигналы от датчиков поступают на входы УВВ, где удваивается частота угловых импульсов (для повышения точности вычисления угла зажигания и угла поворота коленчатого вала двигателя), сдвигается импульс начала отсчета в сторону запаздывания на 15 удвоенных угловых импульсов (это необходимо для обеспечения вычисления процессором углов, запаздывающих относительно начала отсчета) и формируется второй импульс начала отсчета, сдвинутый относительно первого на  $180^\circ$  по углу поворота коленчатого вала двигателя. Входной импульс начала отсчета используется УВВ для синхронизации контроллера с двигателем и

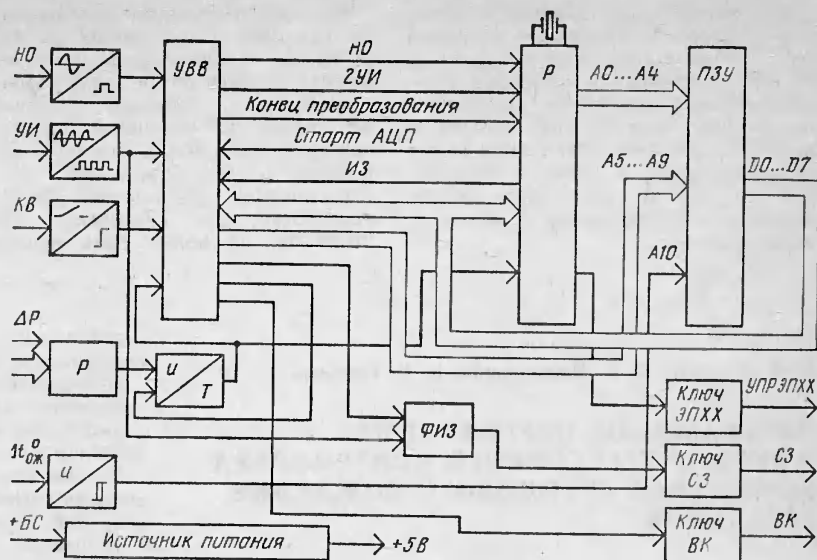


Рис. 2. Структурная схема контроллера «Электроника МС2713»

обеспечения правильной подачи сигнала выбора канала коммутатора первичных цепей катушек зажигания. В процессоре использован метод определения угла поворота коленчатого вала двигателя, основанный на счете угловых импульсов, начиная с момента появления импульса начала отсчета. У четырехцилиндрового четырехтактного двигателя рабочий цикл занимает два оборота коленчатого вала и одинаковые такты у цилиндров поочередно повторяются через  $180^\circ$ , поэтому необходим второй импульс.

Начало измерения разрежения и вычисления величины угла зажигания синхронизировано с импульсом начала отсчета для исключения влияния на результат вычисления пульсаций разрежения во впускном трубопроводе и неравномерностей вращения, вызываемых периодической сменой тактов работы цилиндров двигателя. По сигналу начала отсчета процессор формирует сигнал «Старт АЦП», по которому УВВ запускает преобразователь «Напряжение-время» и начинается процесс изменения величины напряжения на выходе датчика разрежения в цифровой код. Конец процесса характеризуется сигналом «Конец преобразования» (на адресных входах А5...А9 устанавливается адрес страницы ПЗУ). Процессор вычисляет значение угла зажигания, соответствующее текущей частоте вращения коленчатого вала двигателя, и одновременно определяет состояние электроклапана ЭПХХ, соответствующее режиму работы двигателя. Вычисленное значение угла зажигания запоминается, на выходе управления ЭПХХ устанавливается соответствующий логический уровень, процессор переходит в режим ожидания следующего импульса начала отсчета.

Далее работа контроллера повторяется.

Вычисление угла поворота коленчатого вала двигателя и формирование импульса зажигания, определяющего момент разрыва первичной цепи катушки зажигания, производится аппаратной процессора независимо от работы остальной части схемы.

По импульсу начала отсчета в счетчик угла поворота коленчатого вала, входящий в состав процессора, загружается дополнительный код угла зажигания, вычисленный в предыдущем цикле работы контроллера, и начинается счет угловых импульсов. При переполнении счетчика на выходе процессора формируется импульс зажигания, который через УВВ поступает в ФИЗ. ФИЗ по импульсу зажигания вырабатывает сигнал постоянной скважности, подаваемый через ключ на выход контроллера. Импульс зажигания используется УВВ при создании сигнала выбора канала. Ключи коммутатора переключаются в момент появления импульса зажигания, а правильность переключения подтверждается сигналом начала отсчета, что повышает помехоустойчивость контроллера.

Основные характеристики контроллера «Электроника МС2713»

Диапазон измерения частоты вращения коленчатого вала двигателя, $\text{мин}^{-1}$ . . . . .	25...8000
Диапазон измеряемого вакуумметрического давления, мм рт. ст. . . . .	0...620
Диапазон управления углом опережения зажигания, град . . . . .	-20...+70
Диапазон рабочих напряжений питания, В . . . . .	6...16
Использование перенапряжения по бортовой сети, В . . . . .	-28...+150
Диапазон температур окружающей среды, $^\circ\text{C}$ . . . . .	-45...+85
Потребляемая мощность, Вт, не более . . . . .	5

Контроллер имеет защиту от перемены полярности питающего напряжения и замыкания контактов разъема на корпус автомобиля, сохраняет работоспособность при подаче питающего напряжения +18 В при наличии в бортовой сети автомобиля импульсных перенапряжений и помех с амплитудой до 150 В, не требует обслуживания и регулировки в процессе эксплуатации.

Конструктивно контроллер выполнен на печатной плате размерами 150×160 мм с односторонним расположением элементов и двухсторонней металлизацией, которая закреплена на шасси и защищена стальным кожухом. Габаритные размеры контроллера — 171×162×43 мм.

Контроллер «Электроника МС2713» разработан для автомобиля «ВАЗ 21083-02», но может быть применен

для других типов двигателей в аналогичных системах зажигания при условии замены ПЗУ, в котором должны быть закодированы характеристики и параметры данного двигателя.

Телефон 77-44-32, Минск

Статья поступила 20.06.88

УДК 681.3.06

В. В. Дудихин, Е. Г. Меньшикова, Н. В. Смирнов

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОПРОЦЕССОРНОГО КОНТРОЛЛЕРА КОМПЛЕКСА ГРУППОВОГО ВОЖДЕНИЯ ТРАКТОРОВ

Одним из примеров внедрения микропроцессорной техники в сельское хозяйство может служить разработка системы управления комплексом группового вождения тракторов (КГВТ) на базе МП КР1816ВЕ35.

При использовании данного комплекса два серийных трактора (трактор-лидер с трактористом и трактор-дублер) работают под управлением одного тракториста-оператора, производительность труда которого превышает практически вдвое.

Требуемые параметры движения трактора-дублера: дальность и угол визирования (угол между продольной осью трактора и направлением кабеля, соединяющего тракторы) задаются оператором с пульта-задатчика, расположенного на тракторе-лидере. В ходе работы параметры поддерживаются автоматически с большой точностью. Продольное отклонение между тракторами от заданной дальности не должно превышать 0,8 м, а поперечное 0,15 м.

Связь между тракторами осуществляется с помощью телефонного кабеля в режиме последовательного обмена командами (его длина может достигать 30 м).

Известны варианты построения подобных систем управления на основе аналоговых вычислительных устройств [1, 2]. Однако при использовании микропроцессорной техники уменьшается стоимость электронной аппаратуры в 1,5 раза, увеличивается надежность системы, облегчается модификация и настройка аппаратуры на любые типы тракторов.

Предлагаемый контроллер состоит из микросхемы КР1816ВЕ35, соединенной восьмиразрядной шиной данных с интерфейсом адреса данных, ПЗУ программ и констант (емкость ПЗУ 2К×8), ОЗУ (емкость 20×4) и схемы сопряжения с задатчиками режима.

Программное обеспечение построено по модульному принципу. Основной модуль управляет работой КГВТ, отвечает за ввод измеренной информации и ее преобразование, управляет распределением памяти при хранении, вырабатывает команды управления и индикации. В его состав входят подпрограммы:

инициализации и запуска таймера, определяющего временной интервал вычислений;

задания начальных значений, сброса всех передаточных функций и флагов;

ввода измеренной информации, считывания из внешней памяти 12-разрядных чисел в смещенном двонном коде, перевод их в дополнительный код, увеличение разрядности чисел до 16;

преобразования информации и расчета параметров движения трактора-дублера, исходя из текущих величин и табличных данных (таблицы хранятся в ПЗУ);

реализации цифровых фильтров и фильтрации сигнала слома полурам трактора-дублера;

вычисления сигналов ошибки;

формирования и коррекции сигналов рассогласования;

выработки и выдачи команд управления дублером и сигналов индикации для тракториста-оператора.

Вспомогательный модуль содержит пакет сервисных и вычислительных подпрограмм выполнения арифметических операций, пересылки и коррекции данных:

умножения 16-разрядных чисел со знаком;

сложения чисел любой длины, выбираемых из ОЗУ;

получения дополнительного кода;

оценки попадания 16-разрядного параметра в заданные границы;

масштабирования и устранения избыточных разрядов;

умножения и деления чисел любой длины на постоянный множитель.

Разработка ПО контроллера проводилась в несколько этапов. После масштабирования исходного алгоритма и получения структурной схемы программы для кодирования была составлена вспомогательная моделирующая программа на языке Фортран, смоделирован алгоритм на микроЭВМ МЕРА-60.

Кодирование основной программы выполнялось с использованием кросс-системы программирования МИКРОС 48, работающей в операционной системе RT-11. В ее состав кроме ассемблера и компоновщика, входит программно-логическая модель (ПЛМ), способная выполнять некоторые функции отладки. С помощью ПЛМ (задавая входные данные, аналогичные данным вспомогательной программы) основная программа была отлажена и оттестирована (ее объем более 1500 строк ассемблера МП КР1816ВЕ35, что превышает шесть страниц памяти в ПЗУ).

Опыт разработки ПО встраиваемого микропроцессорного контроллера реального времени показал, что в программах, объем которых превышает 500 операторов, основной вопрос — создание достаточно мощных комплексов для совместной отладки программного и аппаратного обеспечения.

113556, Москва, ул. Болотниковская, д. 1, корп. 4, кв. 178, Смирнов Николай Викторович, тел. 110-93-720, 333-72-68р.

## ЛИТЕРАТУРА

1. Гевейлер Н. Н., Ефимов Ю. С., Рыбачук О. С., Марцинкевич Е. В., Чубарь А. Ф. Алгоритм управления тракторами «Кировец» К701 при дублерном вождении / Механизация и электрофикация сельского хозяйства. — 1986. — № 11.
2. А. С. 923397 СССР, МКИ<sup>3</sup> А01В69/04. Устройство выработки команд управления системы дублерного вождения тракторов А. Ф. Чубарь, Е. В. Марцинкевич, Е. И. Стогов и др. — Огубл. 1982, Бюл. № 16.

Статья поступила 22.02.88

Таблица 4

Параметр	Обозначение	Норма	
		не менее	не более
Напряжение питания в режиме считывания, В	$U_{CC1}$	—	7
Напряжение питания в режиме программирования, В	$U_{CC2}$	—	21
Напряжение сигнала входной информации, В	$U_{IH}$ , $U_{IL}$	0,3,	—, 7

параметры микросхемы К573РФ3 при  $T = +25 \pm 10^\circ C$  приведены в табл. 3, предельные значения допустимых электрических режимов эксплуатации — в табл. 4.

Входы и выходы в микросхеме совмещены, поэтому передача данных осуществляется в мультиплексном режиме. Три старших разряда используются для программирования кода микросхемы, что обеспечивает выборку одной из восьми микросхем на общую магистраль без дополнительного дешифратора.

В микросхеме используются четыре управляющих сигнала:

«обмен» — обеспечивает запись адреса во входной регистр адреса;

«чтение-программирование» — определяет режим работы;

«разрешение чтения» — используется для выдачи считанной информации на общую магистраль при наличии сигнала «обмен»;

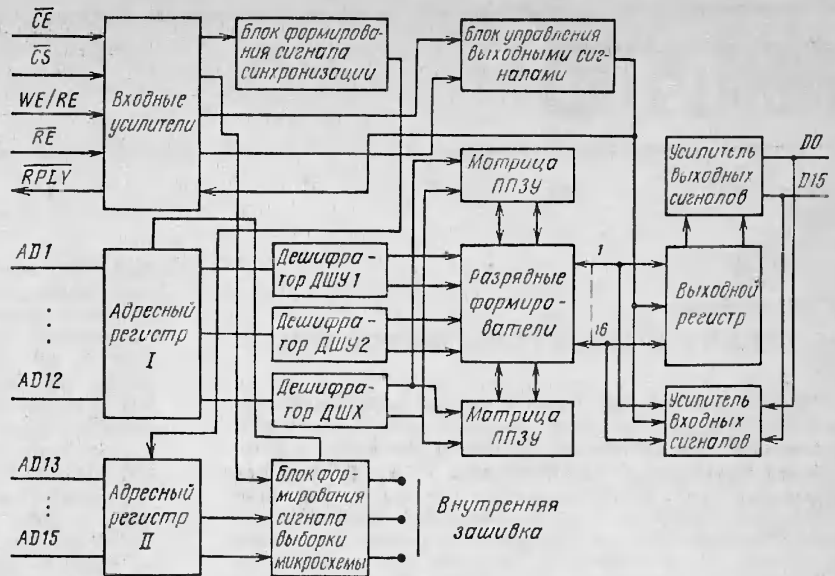


Рис. 2. Электрическая структурная схема РПЗУ К573РФ3

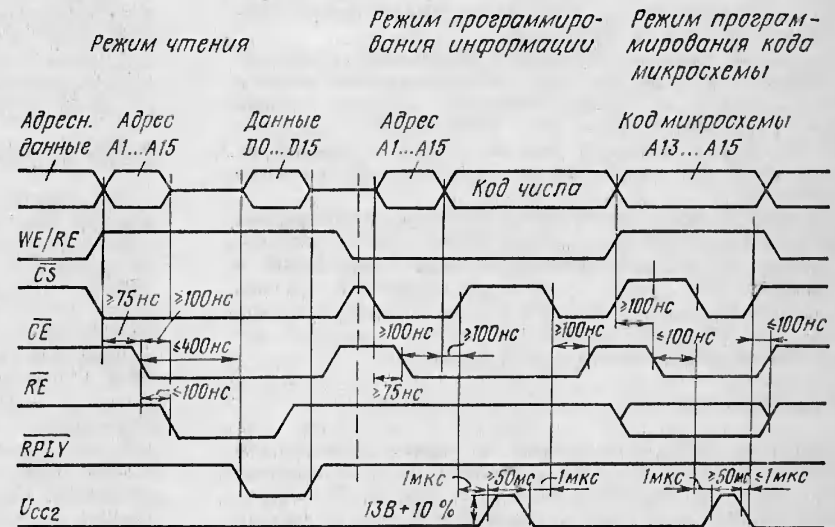


Рис. 3. Временные диаграммы работы РПЗУ К573РФ3

Таблица 3

Параметр	Обозначение	Норма	
		не менее	не более
Выходное напряжение Лог. 0, В	$U_{OL}$	—	0,5
Выходное напряжение Лог. 1, В	$U_{OH}$	2,4	—
Ток потребления, мА	$I_{CC}$	—	40
Динамический ток потребления, мА	$I_{CCD}$	—	80
Входной ток Лог. 0, мкА	$I_{IL}$	—	1,0
Входной ток Лог. 1, мкА	$I_{IH}$	—	1,0
Время выборки разрешения, нс	$t_{CO}$	—	400
Время сохранения выходной информации после сигнала чтения, нс	$t_{ROH}$	—	100
Время записи информации, мс*	$t_W$	—	50

«ответ» — сопровождает информацию, поступающую из РПЗУ на общую магистраль. Кроме этих сигналов предусмотрен сигнал «ВК», обеспечивающий дополнительную выборку.

Временная диаграмма работы РПЗУ в режимах «чтение», «программирование» и «программирование кода микросхемы» приведена на рис. 3. Время хранения информации — 60 мин, число циклов перезаписи  $\geq 10$ .

\* Допускается запись информации до трех раз

УДК 681.3.06

Э. М. Пройдаков

## ВВЕДЕНИЕ В МОНИТОРЫ МИКРОЭВМ

На первом этапе разработки прототипа микропроцессорной системы важное место занимает проектирование и разработка (или адаптация) программы-монитора, в дальнейшем именуемой просто монитором. Попытка разработчиков обойти этот этап (или даже просто уйти от понимания проблемы) приводит к непродуманному заимствованию чужих архитектурных и программных решений со всеми присущими им недостатками. Зачастую это мотивируется намерением использовать существующее на других микроЭВМ программное обеспечение (ПО). Однако при таком подходе теряется главное преимущество микропроцессорной техники — возможность создать архитектуру, адекватную области применения.

В статье вводится необходимая терминология, рассматривается структура монитора, методологические вопросы его создания, а также некоторые вопросы техники программирования мониторов. Предлагаемый материал может быть полезен при разборе любого из существующих мониторов микроЭВМ и разработке оригинального монитора.

Учитывая, что в нашей стране наиболее разнообразные архитектуры микропроцессорных систем (МПС) разработаны на основе микропроцессоров типа КР580ИК80А и К1810ВМ86/88 [1—3], все примеры приводятся для них. Следует отметить, что изучение и (или) создание монитора для МПС является хорошим введением в системное программирование для этого класса машин.

### Назначение монитора

Монитор МПС предназначен, в первую очередь, для выполнения в ней функций инженерного пульта, который существует на больших и мини-ЭВМ для целей тестирования аппаратуры ЭВМ и задания режимов ее работы (в качестве инженерного пульта больших ЭВМ сейчас все чаще используются микроЭВМ).

Таким образом, монитор выполняет роль отладочного средства, позволяющего взаимодействовать с МПС, проверять и отлаживать аппаратные средства, запускать и отлаживать программы. Например, распечатав память после заполнения ее тестовыми последовательностями типа 55Н и ААН, можно выявить грубые ошибки в схеме дешифрации адресов, ошибки разводки печатной платы и др. Выполняя программу по шагам с помощью монитора, можно посмотреть и изменить содержимое регистров, ячеек памяти, портов ввода-вывода и т. д.

Так как монитор выполняет отладочные операции, а защита памяти в большинстве 8-, 16-разрядных МП отсутствует, то размещение монитора в РПЗУ/ПЗУ защищает его от стирания или порчи со стороны отлаживаемых программ. Незащищенной является небольшая (до 256 байт) область рабочей памяти и стека монитора, которые размещаются в ОЗУ. Очевидно, что начальный тест ОЗУ и его обрамление в мониторе использовать ОЗУ не должны.

Монитор МПС называется резидентным, если он записан в ПЗУ или РПЗУ. Во всех других случаях он является загрузаемым. Существуют мониторы

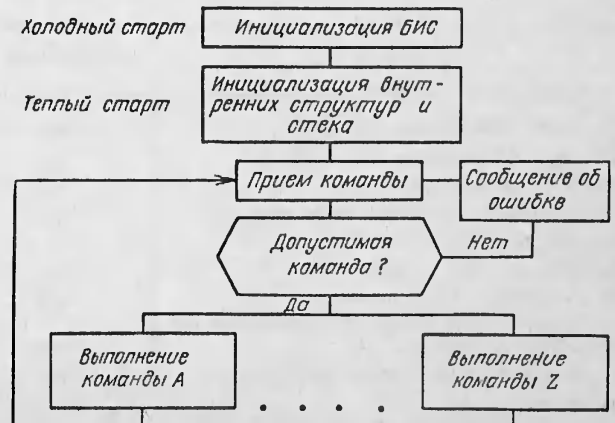
(СМ1800), состоящие из резидентной и загружаемой частей. Большинство мониторов являются полностью резидентными программами.

Физически монитор — это программа объемом от 256 байт до 32 Кбайт. Стандартный объем монитора для систем на базе МП КР580ИК80А — 2 Кбайт, на МП К1810ВМ86 — 8 Кбайт. Наблюдается тенденция к росту размеров мониторов. При тех же функциональных возможностях объем монитора для систем на МП К1810ВМ86 будет в два раза большим из-за того, что для этого МП средняя длина команды в системных программах — 4 байт, а не 2,1 байт, как в МП КР580ИК80А. Объем монитора, обеспечивающего поддержку графического ввода-вывода, может достигать 32 Кбайт и более (в МПС на МП КР580ИК80А для этого вводится страничная организация памяти).

Программирование мониторов традиционно ведется на Микроассемблерах из-за требований экономии памяти РПЗУ/ПЗУ. Однако сейчас многие мониторы, особенно для систем на базе МП К1810ВМ86, пишутся на машинноориентированных языках высокого уровня, таких как ПЛ/М-86 (монитор СМ1810) и Си.

### Работа монитора

При запуске МПС обычно сначала инициализируется монитор. Этот момент начального запуска называют «холодным стартом». В момент холодного старта происходят: настройка БИС параллельного и последовательного интерфейсов, таймера, контроллера прерываний, контроллера дисков и др.; тестирование доступного объема ОЗУ; настройка скорости обмена с дисплеем или линией передачи данных, если «подкачка» программ производится из другой ЭВМ. В ряде МПС, имеющих несколько дисплеев, в момент холодного старта определяется, какой из них будет выполнять роль системной консоли, т. е. устройства, с которого пользователь будет вводить команды и на которое выводится результат исполнения этих команд. Для этого все дисплеи опрашиваются программой инициализации до момента нажатия на одном из них некоторой клавиши, например пробела или «U». Если в составе МПС есть дисководы, то производится их проверка и загрузка ОС.



Последовательность работы простого монитора

Отметим, что монитор не является частью ОС, хотя есть исключения, которые относятся к загружаемым мониторам.

Для выхода в монитор по разного рода ошибкам (обычно нефатальным и нестандартным ситуациям), в нем существует точка запуска, называемая «теплым стартом», при котором не выполняются многие действия холодного старта, например не происходит перинициализация БИС. В процедуре теплового старта, как правило, заново инициализируется стек и внутренняя структура данных монитора.

Мониторы микроЭВМ содержат подпрограммы, выполняющие обмен данными между микроЭВМ и периферийными устройствами, а также необходимые для этого операции управления внешними устройствами. Эти подпрограммы носят название драйверов ввода-вывода. К слову драйвер добавляется название соответствующего устройства, например: драйвер устройства печати, драйвер клавиатуры, драйвер вывода на экран дисплея, драйверы кассетного магнитофона и др. Некоторые устройства имеют несколько драйверов — по одному на каждую выполняемую на них операцию (драйвер чтения с диска, драйвер записи на диск).

По характеру работы монитор можно отнести к интерпретаторам. Он принимает с клавиатуры однобуквенную команду и от одного до трех ее параметров, исполняет команду, если она правильно задана, и снова ожидает ввода очередной команды (см. рисунок).

В ряде случаев наличия развитого монитора достаточно для разработки прикладного ПО, например для однокристальных ЭВМ типа K1816BE51, используемых в качестве встраиваемых контроллеров. При постановке ОС драйверы ввода-вывода монитора и некоторые другие его подпрограммы используются в базовой системе ввода-вывода BIOS адаптируемой ОС, что делает ОС независимой от конкретной архитектуры МПС.

В любом случае следует стремиться разместить монитор в старших адресах памяти системы, так как младшие адреса обычно заняты ОС, системными таблицами, областью параметров, транзитными программами. В 8-разрядных микроЭВМ для этого требуются некоторые дополнительные аппаратные средства из-за необходимости передавать управление по сигналу «сброс» на адрес начала монитора, а не на адрес 0000H, как это заложено в архитектуре 8-разрядных МП. В частности, размещение в начале памяти монитора в ЭВМ CM1800 вызвало трудности при постановке ОС1800 и ДОС1800, а память, занимаемую монитором, разработчикам пришлось сделать затеняемой, т. е. замаскированной ОЗУ в процессе загрузки ОС.

Монитор в затеняемой памяти — это частный случай страничной памяти, когда большая физическая память отображается на реальное адресное пространство страницами от 2 до 64К. Страничная организация памяти снимает ограничения на размер монитора в 8-разрядных микроЭВМ и позволяет значительно расширить его средствами графического ввода-вывода.

Переход на новую страницу осуществляется обычно посредством записи ее номера в порт (регистр) переключения страниц. В мониторе функцию перехода на новую страницу или возврата на предыдущую выполняет драйвер переключения страниц. Если размер страницы равен максимально адресуемой памяти, то при переходе на новую страницу возникает проблема переписи в нее программ и данных (страницы такого объема называют также банками памяти, а память — банкируемой. Эта терминология пришла от разработчиков аппаратуры мини-ЭВМ).

В МП K1810BM86 управление после сброса передается на последние 16 байт адресуемой памяти, что определяет местоположение монитора в конце памяти.

#### Команды монитора

Минимально необходимый набор команд монитора состоит всего из трех команд:

S aaaa — просмотр и изменение памяти;  
X [R n] — просмотр и изменение регистров;

G [aaaa] [,bbbb]

запуск программы с контрольными точками.

Число команд в максимальном наборе ограничивается 26 буквами латинского алфавита. Однако в этом случае внутри одной команды часто создаются подкоманды. Обычный набор команд монитора в дополнение к перечисленным включает в себя:

D [aaaa] [, eeee] — дамп памяти;  
A [aaaa] — однопроходный ассемблер;  
L [aaaa] [, eeee] — однопроходный дисассемблер;  
T [nnnn] — пошаговое выполнение команд;  
Ga aaa, eeee, bb — заполнение памяти константой;  
Irr и Opp, bb — ввод-вывод через порты;  
R <имя> — загрузка файла;  
Sa aaa, eeee, cccc — сравнение двух участков памяти;  
Ma aaa, eeee, cccc — пересылка участка памяти;  
H pppp [, mppm] — шестнадцатеричная арифметика (сумма и разность двух чисел);  
Qdn-lm — переназначение устройств ввода-вывода;  
Ka aaa, eeee — вычисление контрольной суммы области памяти;  
Z [nn] — запуск тестов.

Обозначения: aaaa — начальный адрес; eeee — конечный адрес; [ ] — необязательный operand; rr — номер порта; cp — физическое имя устройства; ln — логическое имя устройства; pppp, mppm — целые числа; Rp — номер или обозначение регистра; bb — байт данных.

Подробное описание основных команд монитора приведено в книге М. Дамке [4].

Так как командная строка монитора обычно не длиннее 16 символов, то средства для ее редактирования, как правило, не предоставляются либо даются простейшие. Однако для ряда систем удобно иметь прямо в мониторе построчный редактор в драйвере ввода с клавиатуры. Объем такого редактора, аналогичного по функциям редактору командной строки ОС ISIS-II, около 512 байт.

Единый подход ко всему спектру отладочных средств требует, чтобы команды монитора были подмножеством набора команд программного отладчика, а последний — подмножеством набора команд внутрисхемного эмулятора, как это реализовано, например, в MON86-DEBUG-88 ГИСЕ фирмы ИНТЕЛ. Это обеспечивает более быстрое освоение указанных средств и, что очень важно, психологический комфорт при работе в среде отладочных средств.

В ряде МПС удобнее загружать ОС (или интерпретаторы с языков БЕЙСИК, Форт и др.) из затеняемого или страничного РПЗУ. Это обеспечивает более быструю загрузку и позволяет отказаться от системных дисков, что добавляет 2—3 дорожки к полезной емкости диска. При наличии микросхем РПЗУ емкостью 8К×8 и более возможно хранение в РПЗУ нескольких ОС (например, CP/M-80 и ISIS-II). Для загрузки ОС (или БЕЙСИКа) из РПЗУ в состав команд монитора вводится соответствующая команда.

Для использования в ряде ситуаций возможности монитора в ОС должна быть реализована команда для выхода в монитор, а в мониторе — команда возврата в ОС. Чтобы можно было разбираться с причинами разного рода зависаний (ситуаций, когда машина не реагирует ни на какие команды с клавиатуры), в микроЭВМ желательна возможность выхода в монитор от кнопки внешнего прерывания, при этом сохраняются контекст (содержимое регистров и стек) программы, вызвавшей зависание. К сожалению, в большинстве выпускаемых ПЭВМ (типа IBM PC XT) существует только кнопка RESET (сброс), после нажатия которой контекст уничтожается. Программный отладчик не всегда выручает.

Монитор должен индцировать возникновение программных прерываний, обработка которых не была предусмотрена. Некоторые уровни прерываний используются самим монитором. Для холодного старта в МП KP580HK80A используется программное прерывание PSTO, для теплового

старта — RST1 и для пошагового режима — RST7. В простых учебных или отладочных мониторах удобно во время холодного старта записывать в ОЗУ программ пользователя код OFFH, который является машинным кодом команды RST7. Это позволяет быстро обнаруживать неправильные передачи управления в свободные области ОЗУ.

### Структура монитора

Монитор должен быть хорошо структурирован. Это необходимо как для быстрой отладки, так и, что гораздо важнее, для хорошей читаемости его исходного текста, сопровождения, модификации и (или) адаптации.

Системное программное обеспечение называется открытым, если его исходные тексты либо исходные тексты его модифицируемых модулей предоставляются пользователям. Это делается в следующих случаях:

1. Разработчики не могут взять на себя бремя сопровождения, модификации и (или) адаптации программного продукта.

2. Открытая система быстрее распространяется и легче осваивается пользователем (например, CP/M-80). Пользователь, прочитав и проработав исходные тексты некоторой системы, становится единомышленником разработчика, даже если он переписал заново 100 % текста.

Мониторы принадлежат к классу открытых систем. По сравнению с разработкой обычных программ открытые системы требуют заметно больших усилий и более высокой квалификации.

Можно выделить следующие уровни построения монитора:

- инициализации (холодный и теплый старты);
- главной программы (ввод и разбор команд);
- семантических подпрограмм (выполнение команд);
- вспомогательных подпрограмм (например, сравнение двух адресов);
- драйверов ввода-вывода.

Для возможности обращения к драйверам из других программ команды переходов размещаются в векторе переходов по фиксированным адресам (обычно в начале монитора). Это делается для того, чтобы изменения, вносимые в монитор, не сказались на остальном ПО.

Отметим, что при изменении конфигурации аппаратных средств МПС, в первую очередь, изменяются программа инициализации и драйверы ввода-вывода. Для повышения мобильности и структурированности монитора при его программировании широко применяются векторы переходов, таблицы настройки и байты конфигурации устройств ввода-вывода (IOBYTE).

### Разработка монитора

Монитор часто является первой программой, разрабатываемой для МПС. Можно выделить следующие основные схемы его разработки:

1. Разработка на инструментальной системе. Отладка с помощью логического анализатора и (или) внутрисхемного эмулятора (в худшем случае с помощью аппаратного пульта).
2. Заимствование готового монитора с минимальной адаптацией к прототипу создаваемой системы.
3. Создание монитора методом раскрутки. Простейший монитор, до 256 байт кодируется вручную. Затем уже с его помощью разрабатываются все более развитые версии. При этом в составе прототипа удобно иметь программатор РПЗУ.

Перенос монитора с инструментальной системы на прототипную может быть организован следующими способами:

- через память (энергонезависимую, общую);
- через аппаратуру шинного либо внутрисхемного эмулятора;

пересылкой программ по последовательному, реже параллельному интерфейсу;

записью в РПЗУ/ПЗУ и установкой его на плате прототипа. В этом случае на время отладки (особенно для сложных систем) на плате прототипа рекомендуется поставить разъем с нулевым усилием, чтобы упростить процесс установки РПЗУ.

Конструктивно монитор всегда размещают на плате модуля центрального процессора, но для встраиваемых систем возможна реализация монитора в отладочной клавиатуре, которая предназначена для наладчиков и нужна только для поиска неисправностей.

### Мониторы мультимикропроцессорных систем (ММС)

Подавляющее большинство ММС разработано как системы с разделением функций процессоров, в которых за каждым процессором жестко или динамически закреплено выполнение определенной системной функции (системный процессор, процессор ввода-вывода и т. д.) или решение определенных задач (задачи).

Немногие примеры разработанных ММС показывают, что имеются два основных способа реализации мониторов для них.

Первый — это реализация монитора на базе процессора ввода-вывода, так как ему обычно доступна вся общая память ММС. Работа такого монитора должна поддерживаться аппаратурой ММС, например для режима трассировки других процессоров. Этот подход был использован в мониторе ММС «Истра-01» [6].

Второй подход, использованный для ММС на основе МП КР580ИК80А, заключается в том, что один и тот же монитор, записанный в РПЗУ, объемом 2 Кбайт устанавливается на всех процессорных платах ММС. Каждый из этих мониторов может работать асинхронно в режиме ведущего или ведомого монитора. Ведущим монитором считается тот, который при включении системы обнаружил, что дисплей подключен к его процессорной плате и готов работать с ним (например, было нажатие определенной клавиши), в противном случае монитор объявляет себя ведомым. Обмен информацией (посредством посылок сообщений) между ведущим и ведомыми мониторами осуществляется либо через общую память, либо общие порты ввода-вывода. Так как второй подход по идеологии близок к мультимикропроцессорным ОС реального времени (РРХ-86, ОС, СФП 1800/1810), то для 16-разрядных МП в ММС монитор используется совместно с записанным в РПЗУ ядром такой ОС либо заменяется интерактивным отладчиком этой ОС [7].

Так как наилучшим способом изучения монитора является чтение его исходного текста, то в учебных целях целесообразно издать исходные тексты наиболее типичных мониторов, редакторов текстов, отладчиков и простых ОС типа CP/M-80 и RMX-80.

Телефон 283-97-64, Москва

### ЛИТЕРАТУРА

1. Уокерли Дж. Архитектура и программирование микроЭВМ.— М.: Мир, Т. 2,— 1984.— 360 с.
2. Микропроцессоры: системы программирования и отладки /Под ред. В. А. Мясникова, М. Б. Игнатьева.— М.: Энергоатомиздат, 1985.— С. 272.
3. Каган Б. М., Сташин В. В. Микропроцессоры в цифровых системах.— М.: Энергия, 1979.— С. 193.
4. Дамке М., Операционные системы микроЭВМ.— М.: Финансы и статистика, 1985.— С. 150.
5. Шелкунов Н. Н., Дианов А. П. Техника программирования 16-разрядных микроконтроллеров // Микропроцессорные средства и системы.— 1987.— № 2.
6. Воробьев А. Д., Кнеллер Э. Г., Пац В. Б. ПЭВМ «Истра»: архитектура, технические характеристики // Микропроцессорные средства и системы.— 1987.— № 1.— С. 15.
7. MC-86 User's Guide. Manja GMB.— BRD, 1983.

Статья поступила 7.07.87

## СПОСОБЫ ПОСТРОЕНИЯ И ИССЛЕДОВАНИЯ ИНТЕРФЕЙСОВ УСТРОЙСТВ ВВОДА И ОТОБРАЖЕНИЯ ИНФОРМАЦИИ МИКРОКОНТРОЛЛЕРОВ НА ПРИМЕРЕ БИС КР580ВВ79

Микропроцессорные системы в зависимости от их функционального назначения используют различные устройства ввода и отображения (УВО) информации. В отличие от универсальных микроЭВМ, для которых характерно подключение функционально законченных УВО через стандартные интерфейсы, в микроконтроллерах обычно применяются специализированные клавиатуры и дисплеи на основе знаковосинтезирующих индикаторов (ЗСИ). Подключение таких УВО требует разработки специфического схемотехнического и программного обеспечения. Их сложность определяется принципом действия, видом и числом элементов, применяемых для ввода и отображения информации, а также набором реализуемых функций.

Освободить микропроцессор (МП) от организации работы клавиатуры и дисплея позволяют аппаратные интерфейсы. Выигрыш от их применения становится особенно ощутимым (благодаря использованию типовых схемных решений и упрощений драйверов) при построении интерфейсов на основе БИС КР580ВВ79, КБ1835ВГ1-4, К1806ВП1-092, К1806ВП1-093 и др.

Понять способы организации таких интерфейсов можно на примере конкретной схемы, иллюстрирующей возможности обслуживания различных типов клавиатуры и дисплеев\*.

Этим целям служит модуль УМПК-80 МИ6\*\*, входящий в состав лаборатории изучения микропроцессорных систем «Пирамида» [4]. Модуль предназначен для исследования особенностей работы интерфейсов УВО, построенных на основе БИС КР580ВВ79 [5], и позволяет анализировать:

способы распределения функций интерфейса между его аппаратной и программной частями;

режимы работы БИС программируемого контроллера клавиатуры и индикации и управление ими;

возможные варианты схем сопряжения БИС с полупроводниковыми ЗСИ и различными типами клавиатур.

\* Алгоритмы программного обслуживания клавиатуры и дисплея, реализующие их программно, а также необходимые схемотехническое обеспечение подробно описаны в работе [1] и рассматриваться не будут.

\*\* Приведенная в статье методика может быть частично использована для исследований БИС КР580ВВ79, входящей в состав микроЭВМ УМПК-48/ВМ [2] или УМПК-86/ВМ [3].

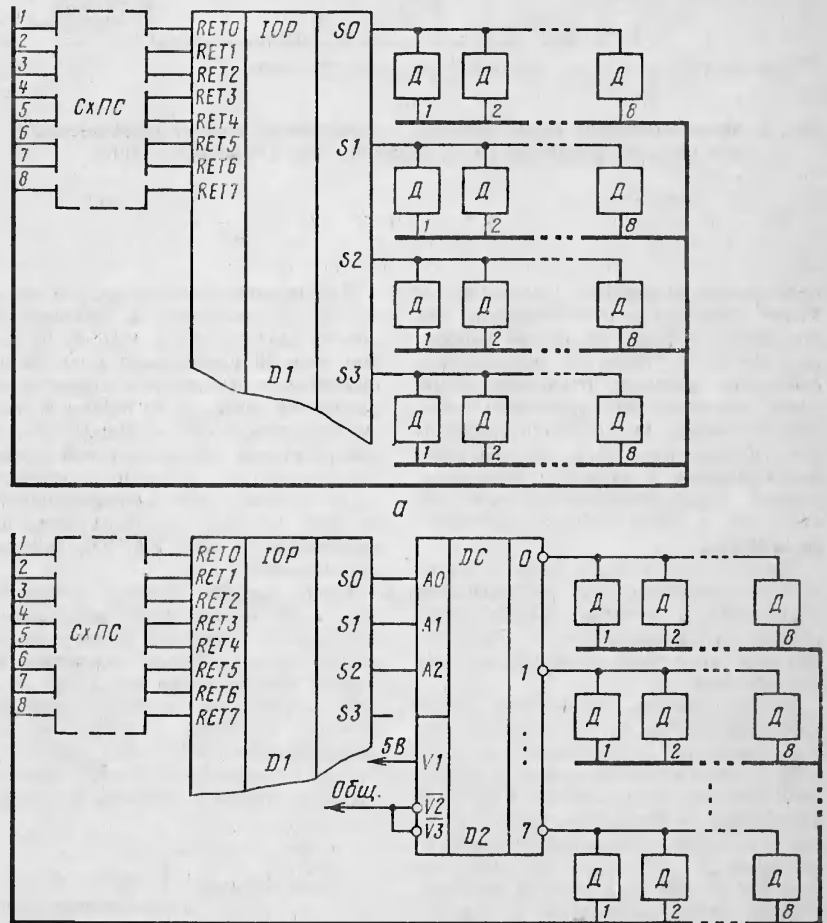
### Интерфейс клавиатуры

При вводе информации с клавиатуры необходимо обеспечить опрос и анализ состояния датчиков, устранение дребезга, определение кодов датчиков, буферирование введенной информации, сервис (выбор способов опроса и анализа состояний, формирование запросов прерываний и т. п.). Для упрощения интерфейсов УВО обычно применяют мультиплексный опрос датчиков клавиатуры и такое же отображение информации на дисплей. Это, в свою очередь, позволяет объеди-

нить функции регенерации изображения и опроса клавиатуры.

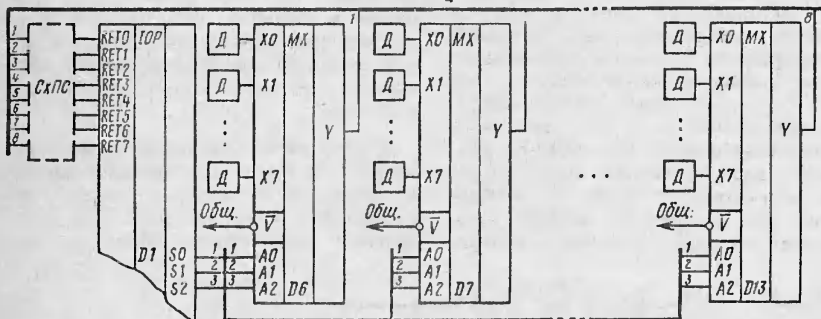
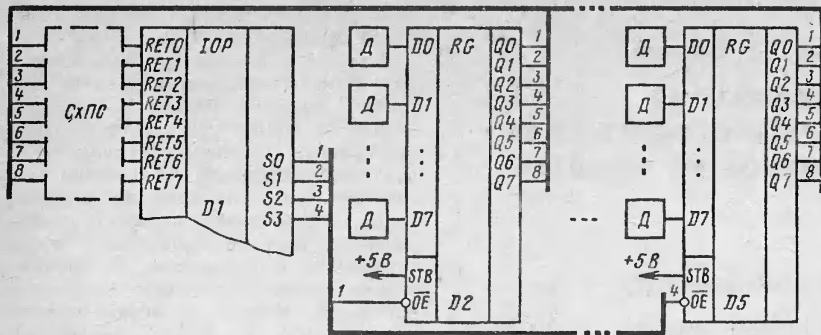
Датчики клавишного или сенсорного типа, фиксирующие нажатие или касание пальцем, являются основным элементом клавиатуры. Наиболее распространены механические клавишные датчики, изменяющие при нажатии свое сопротивление. Основные их достоинства — невысокая стоимость и возможность прямого подключения к интерфейсам, а недостатки — сравнительно низкая надежность и дребезг контактов. Наряду с механическими используются и другие клавишные датчики: емкостные (изменяющие при нажатии емкость), интегральные (например, работающие на эффекте Холла), механические со встроенными схемами антидребезга, оптические и другие.

В отличие от клавишных сенсорные датчики не имеют движущихся частей и «чувствуют» касание пальца по вносимой емкости или наводкам переменного напряжения, изменению со-



D1 - КР580ВВ79 ; D2 - К555ИД7

Рис. 1. Мультиплексный опрос матрицы с непосредственным включением датчиков при дешифрированном (а) и кодированном (б) сканировании



D1 - KP580BB79 ; D2... D5 - KP580HP82 ; D6... D13 - K555KP7

б

Рис. 2. Мультиплексный опрос матрицы с электронной коммутацией датчиков при дешифрированном (а) и кодированном (б) сканировании

противления сенсорного элемента и т. п. Такие датчики изготавливаются, как правило, в виде элементов топологии печатных плат. К недостаткам сенсорных датчиков относятся отсутствие механической обратной связи, затрудняющее определение опротоном факта срабатывания датчика, необходимость в схемах сопряжения, слабая помехоустойчивость, чувствительность к загрязненности сенсорного элемента.

Мультиплексный опрос датчиков обеспечивается по сигналам сканирования, которые формируются самими БИС интерфейсов клавиатуры или внешними схемами под их управлением.

В интерфейсах на основе БИС KP580BB79 опрос датчиков клавиатуры и регенерация изображения на дисплее осуществляются под управлением счетчика сканирования. Командой «Установка режимов работы клавиатуры и дисплея» может устанавливаться один из двух видов сигналов на выходах SO...S3 (два режима сканирования): дешифрированный и кодированный. В первом случае на выходах формируются сигналы, непосредственно используемые для сканирования, во втором — на них выводится двоничный код номера сканируемой линии, и для получения сигналов сканирования требуются внешние схемы дешифрации.

Для мультиплексного опроса датчики Д объединяются в матрицу одним из двух способов (рис. 1, 2). Выбор способа соединения, а также необходимость применения схемы преобразования сигналов по уровню и (или) длительности СхПС определяются типом датчиков. Максимальный размер обслуживаемой матрицы составляет 4x8 датчиков для дешифрированного (рис. 1,а, 2,а) и 8x8 датчиков для кодированного (рис. 1,б, 2,б) режимов сканирования.

Кроме выбора режима сканирования пользователь имеет возможность программно (командой «Программирование синхронизации») устанавливать частоту сканирующих импульсов, задавая коэффициент деления предварительного делителя частоты счетчика синхронизации БИС. На вход этого счетчика с вывода CLC БИС подается внешняя тактовая частота, а сигналы

с выходов используются внутренними устройствами, в частности счетчиком сканирования. После сброса по сигналу CLC коэффициент деления автоматически устанавливается равным 31. В дальнейшем он может быть изменен пользователем в диапазоне  $2 \leq k \leq 31$ , однако так, чтобы частота на выходе предварительного делителя, равная  $f_{CLK}/k$ , не превышала 100 кГц (например, для внешней тактовой частоты 2 МГц, типичной для МП БИС KP580BM80A, коэффициент деления может изменяться от 20 до 31). Период импульсов на линиях сканирования составляет  $512 \cdot k \cdot T_{CLK}$  для кодированного и  $256 \cdot k \cdot T_{CLK}$  для дешифрированного режимов сканирования, а их длительность равна  $64 \cdot k \cdot T_{CLK}$ . Состояние датчиков сканируемой строки матрицы проверяются в строго определенные моменты (рис. 3).

Частота сканирования клавиатуры должна выбираться в зависимости от параметров применяемых датчиков и схемы преобразования сигнала. Например, для механических клавишных датчиков она определяется исходя из времени дребезга контактов с учетом того, что длительность цикла проверки на дребезг (время задержки внутренней схемы устранения дребезга клавиатуры БИС) равна удвоенному периоду сканирующих импульсов (т. е. период сканирования должен быть больше, чем половина длительности дребезга).

Практически исследовать временные диаграммы обоих режимов работы счетчика сканирования БИС KP580BB79 можно, наблюдая осциллограммы сигналов на выходах счетчика сканирования (выводы SO...S3) при выполнении программы 1 на модулях УМПК-80/МИ6 (рис. 4) и УМПК-80/ВМ.

Анализ информации о состоянии датчиков, полученный в ходе опроса матрицы, и подготовка информации о сработавших датчиках для ввода в микроЭВМ проводятся различными способами (рис. 5) в зависимости от требований, предъявляемых к работе клавиатуры.

Контроллер клавиатуры и дисплея KP580BB79 позволяет программно выбирать один из следующих режимов работы клавиатурной части: 1. Сканирование клавиатуры с обнаружением двухклавишных сцеплений. 2. Сканирование клавиатуры с обнаружением N-клавишных сцеплений. 3. Сканирование

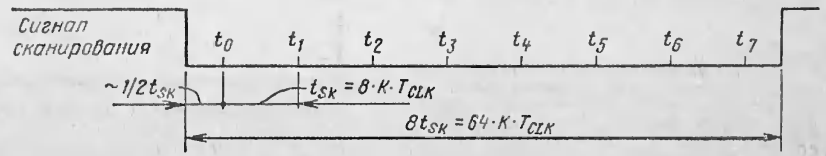


Рис. 3. Проверка состояния датчиков сканируемой строки матрицы:  $t_{sk}$  — время сканирования одного датчика;  $t_i$  — момент проверки состояния датчика сканируемой строки, подключенного ко входу RET<sub>i</sub>



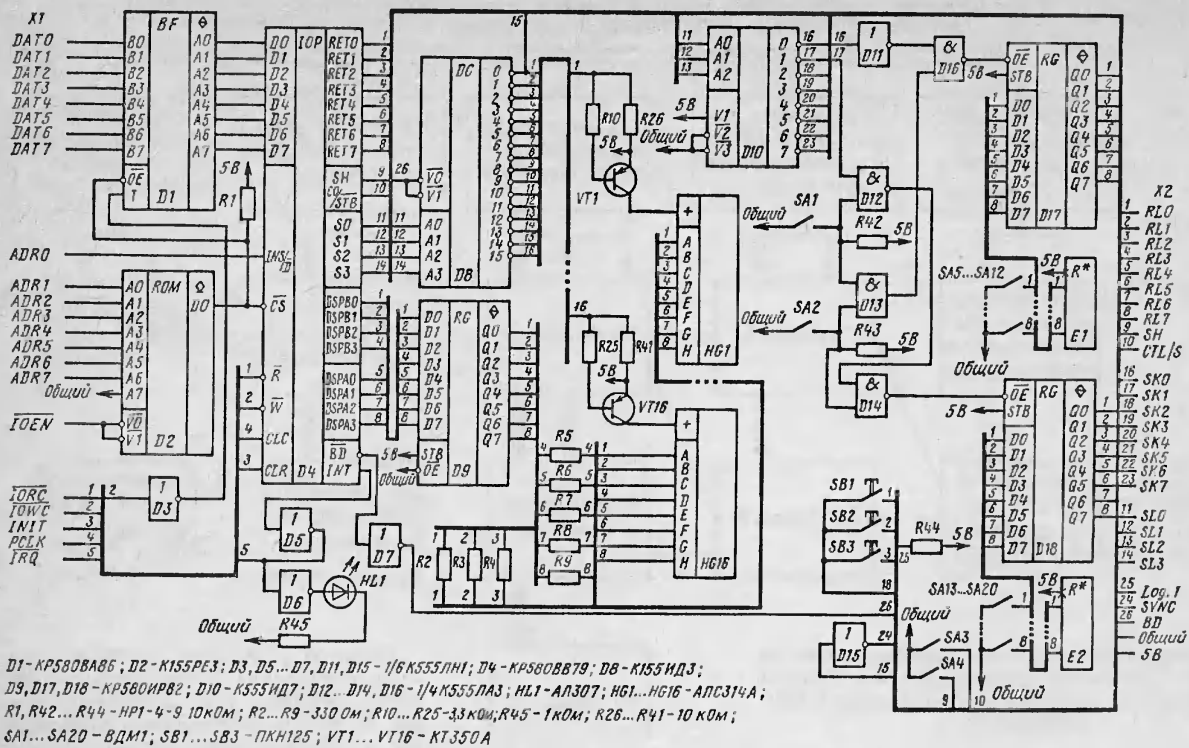


Рис. 4. Принципиальная электрическая схема модуля УМПК-80/МИ6

----- ПРОГРАММА 1 -----

; для ИССЛЕДОВАНИЯ ВРЕМЕННЫХ ДИАГРАММ СИГНАЛОВ СКАНИРОВАНИЯ  
; ВХОД У ОСЦИЛЛОГРАФА ПОДКЛЮЧАЕТСЯ К ВЫВОДАМ SO...S3 БИС, А  
; ЕГО синхронизация осуществляется СИГНАЛОМ SYNC ( X2 )  
; МОДУЛЯ УМПК-80/МИ6.

START: MVI A,СОМО ;СОМО - КОМАНДА "УСТАНОВКА РЕЖИМОВ  
РАБОТЫ КЛАВИАТУРЫ И ДИСПЛЕЯ"

; D7, D6, D5 = 0, 0, 0  
; РЕЖИМЫ ДИСПЛЕЙНОЙ ЧАСТИ БИС  
D4 = 0 - ВВОД СЛЕВА  
= 1 - ВВОД СПРАВА СО СДВИГОМ  
D3 = 0 - ВОСЬМИСИМВОЛЬНЫЙ ДИСПЛЕЙ  
= 1 - ШЕСТНАДЦАТИСИМВОЛЬНЫЙ ДИСПЛЕЙ  
; РЕЖИМЫ КЛАВИАТУРНОЙ ЧАСТИ БИС  
D2, D1 = 0, 0 - РЕЖИМ 1  
= 0, 1 - РЕЖИМ 2  
= 1, 0 - РЕЖИМ 4  
= 1, 1 - РЕЖИМ 5  
; РЕЖИМЫ СЧЕТЧИКА СКАНИРОВАНИЯ  
D0 = 0 - КОДИРОВАННОЕ СКАНИРОВАНИЕ  
= 1 - ДЕШИФРИРОВАННОЕ СКАНИРОВАНИЕ

OUT P79+1 ;ЗАПИСЬ В РЕГИСТР КОМАНД БИС  
MVI A,СОМ1 ;СОМ1 - КОМАНДА "ПРОГРАММИРОВАНИЕ  
синхронизации"

; D7, D6, D5 = 0, 0, 1  
; D4, D3, D2, D1, D0 - КОД КОЭФФИЦИЕНТА ДЕЛЕНИЯ К

OUT P79+1 ;  
RST 1 ;ОСТАНОВ ДЛЯ НАБЛЮДЕНИЯ ОСЦИЛЛОГРАММ

; В ДАННОЙ ПРОГРАММЕ РЕЖИМЫ РАБОТЫ КЛАВИАТУРЫ И ДИСПЛЕЯ В КОМАНДЕ  
;СОМО УСТАНОВЛИВАЮТСЯ ПРОИЗВОЛЬНО (НАПРИМЕР СОМО = 0ВН ИЛИ 09Н),  
;А КОД КОЭФФИЦИЕНТА ДЕЛЕНИЯ В КОМАНДЕ СОМ1 МОЖЕТ ИЗМЕНЯТЬСЯ В  
;ПРЕДЕЛАХ ОТ 14Н(МИН) ДО 19Н(МАХ).

вание клавиатуры с обнаружением N-клавишных сцеплений и ошибок. 4. Сканирование матрицы датчиков. 5. Ввод по стробу.

Режимы 1...4 предназначены для обслуживания клавиатуры в виде матрицы сенсорных или клавишных датчиков и отличаются, в основном, способами обработки информации о ее состоянии (см. рис. 5) Режим 5 служит для ввода по стробу данных в параллельном коде от любого входного устройства, в том числе функционально законченной клавиатуры.

Буферирование введенной информации, подготовленной для выдачи в микроЭВМ после анализа состояния матрицы, необходимо вводить из-за асинхронности моментов фиксации срабатывания датчиков и считывания данных о них МП БИС. Для фиксации кодов сработавших датчиков используют регистр или ОЗУ, организованное как стек, а для фиксации состояния всей матрицы — обычное ОЗУ. Глубину стека определяют исходя из разности скоростей ввода и вывода данных. Извещение микроЭВМ о наличии готовых к вводу данных в интерфейс клавиатуры обеспечивает схема формирования запроса прерывания и (или) регистр состояния буферного ОЗУ. Последний, при отсутствии формируемых интерфейсом прерываний, может использоваться для организации ввода данных в микроЭВМ по опросу как в произволь-

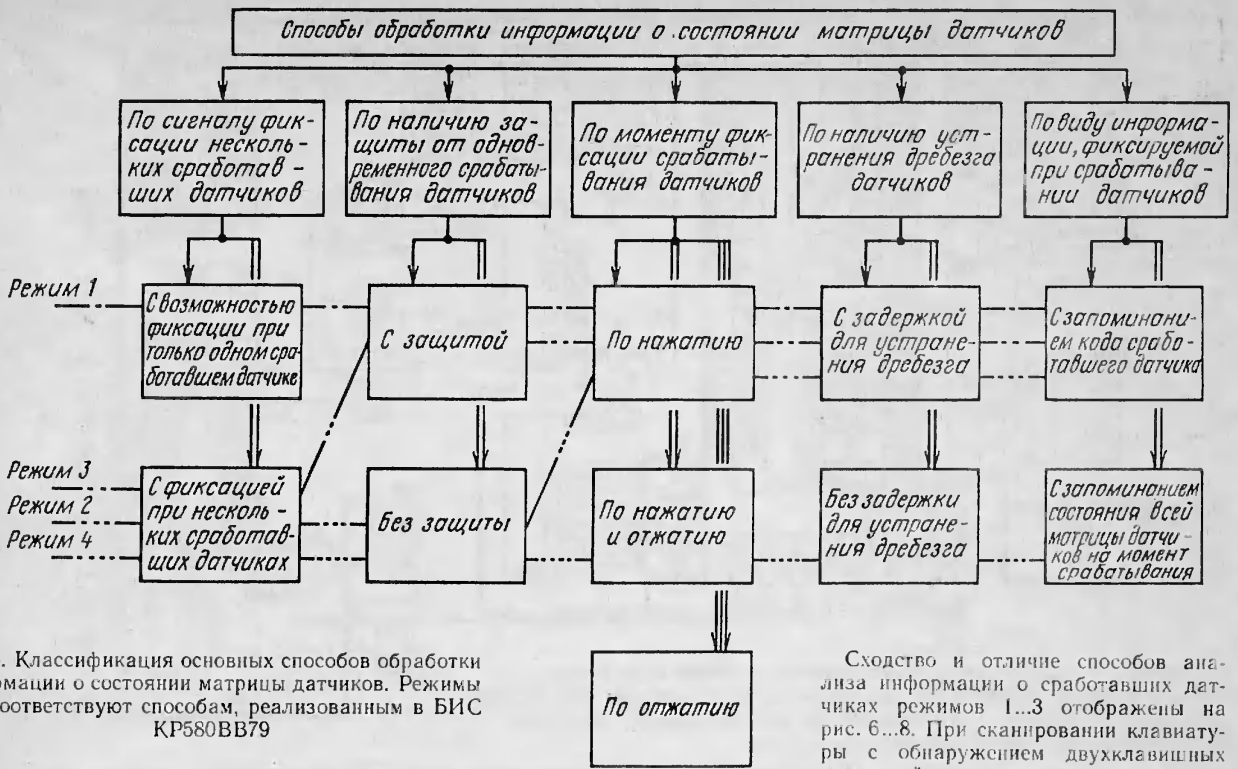


Рис. 5. Классификация основных способов обработки информации о состоянии матрицы датчиков. Режимы 1...4 соответствуют способам, реализованным в БИС КР580ВВ79

Сходство и отличие способов анализа информации о сработавших датчиках режимов 1...3 отображены на рис. 6...8. При сканировании клавиатуры с обнаружением двухклавишных сцеплений запрещен ввод кода сработавшего датчика, если такие датчики уже есть в матрице. Этот режим исключает возможность ошибочного одновременного нажатия на две или несколько клавиш. Режим сканирования клавиатуры с обнаружением N-клавишных сцеплений допускает ввод кода сработавшего датчика независимо от

ные моменты, так и по прерываниям от задатчика интервалов времени.

В БИС КР580ВВ79 для этих целей служат: буферное ОЗУ (8×8 бит), которое работает как стек FIFO на восемь символов в режимах 1...3, 5 и как ОЗУ датчиков в режиме 4; регистр слова состояния стека — ОЗУ датчиков; выход запроса прерывания, формируемого схемой интерфейса клавиатуры. Доступ к буферному ОЗУ возможен после аппаратного сброса или выполнения команды «Чтение стека — ОЗУ датчиков» (табл. 1, 2).

В режимах 1...3 код датчика заносится в стек только один раз после определения достоверного срабатывания, независимо от того, как долго датчик остается в сработавшем положении. Информация о таких датчиках кодируется в виде 8-разрядного слова,

где D0...D2 — номер возвратной линии (т. е. столбца) матрицы, D3...D5 — номер сканируемой строки матрицы, к которой подключен сработавший датчик; D6 и D7 — состояния входов SH и CO/STB БИС на момент фиксации срабатывания. Наличие этих входов у БИС дает возможность ставить в соответствие каждому из датчиков клавиатуры четыре кода и, таким образом, увеличивать число идентифицируемых программой символов до 256.

Для регламентации обмена данными между стеком БИС и микроЭВМ можно пользоваться словом состояния. Его разряды D0...D3 отражают число кодов в стеке, причем D3=1 указывает на то, что в стеке находится восемь символов, и если данные из него не будут считаны, то информация о следующих сработавших датчиках будет утеряна. Об ошибочных попытках чтения данных из пустого стека и ввода в полный стек очередного кода свидетельствуют флаги переполнения D4=1 и переноса D5=1. После чтения слова состояния разряды D4 и D5 автоматически сбрасываются.

Одновременно с записью первого кода в стек устанавливается запрос прерывания на выходе INT БИС. Запрос снимается во время чтения очередного кода из стека и устанавливается вновь после окончания чтения, если стек еще не пуст. Кроме этого, возможен аппаратный или программный сброс запроса прерывания (табл. 2).

Таблица 1

Сигналы на входе БИС			Выполняемая операция
INS/D	R	W	
1	1	0	Запись команды Чтение слова состояния БИС Запись данных в ОЗУ отображения Чтение данных из стека — ОЗУ датчиков или ОЗУ отображения
1	0	1	
0	1	0	
0	0	1	

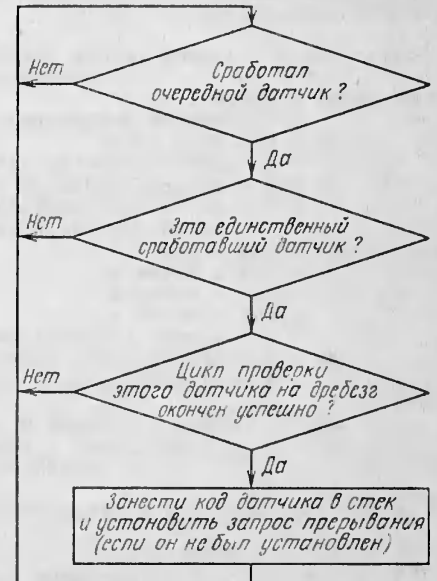


Рис. 6. Алгоритм обработки информации о состоянии датчиков в режиме сканирования клавиатуры с обнаружением 2-клавишных сцеплений

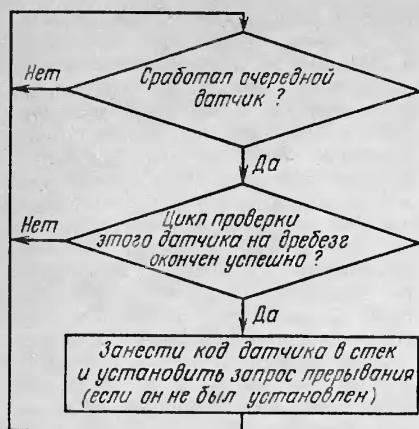


Рис. 7. Алгоритм обработки информации о состоянии датчиков в режиме сканирования клавиатуры с обнаружением N-клавишных сцеплений

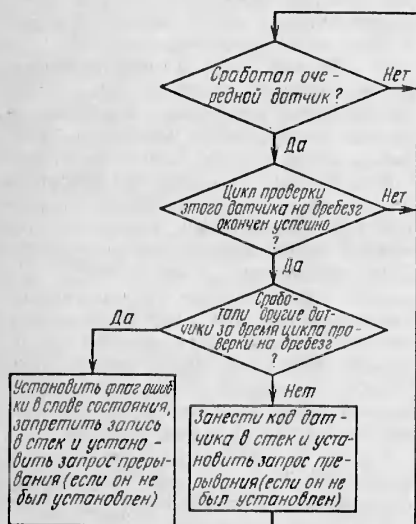


Рис. 8. Алгоритм обработки информации о состоянии датчиков в режиме сканирования клавиатуры с обнаружением N-клавишных сцеплений и ошибок

того, есть ли такие датчики в матрицах или нет. Его можно использовать там, где уже нажатые клавиши не блокируют ввод остальных (например, в клавиатурах игровых автоматов). Сканирование клавиатуры с обнаружением N-клавишных сцеплений и ошибок является компромиссным вариантом. В этом режиме срабатывание очередного датчика до окончания цикла проверки на врезе приводит к тому, что следующий датчик считается ошибкой и дальнейший ввод кодов прекращается. Устанавливаются флаг ошибки (разряд D6=1 в слове состояния) и запрос прерывания. Сброс запроса прерывания и флага ошибки, а также возобновление записи в стек осуще-

	Сброс		Команда «Сброс»			
	Аппаратный $T_{CLR} > 6T_{CLK}$	Программный (команда «Установка режимов»)	1100NN10	1100NN00	1101NN00	1100NN01 или 1101NN01
Режим работы БИС	Кодированное сканирование, режим I клавиатуры, 16-символьный дисплей	Определяется командой	X	X	X	X
Коэффициент деления K	31	X	X	X	X	X
Счетчик сканирования	Сбрасывается	X	X	X	X	Сбрасывается
Источник чтения	Стек — ОЗУ датчиков	X	X	X	X	X
Адрес чтения	0	0	X	X	0	0
Стек	Пустой	Пустой	Пустой	X	X	Пустой
Слово состояния	0	0	0	X	X	0
Запрос прерывания	0	0	0	X	X	0
Адрес записи	0	0	X	X	0	0
ОЗУ отображения	X	X	X	X	NN	NN
Код очистки на выходах DSPAi и DSPBi	0	X	X	NN	NN	NN
Маска гашения	0	X	X	X	X	X
Маска запрета записи	0	X	X	X	X	X

Примечание. X — остается без изменения

ствляются с помощью команды «Сброс» (или после аппаратного сброса). При необходимости информация, накопленная в стеке до появления ошибки, может быть считана до выполнения сброса.

Ознакомиться с режимами работы 1...3 клавиатурной части БИС можно, воспользовавшись программами 2 и 3. Они позволяют исследовать спо-

собы анализа и кодирования информации о сработавших датчиках, порядок установки и сброса запроса прерывания и разрядов слова состояния, а также все команды БИС, используемые для работы с этими режимами. Для исследования режимов работы интерфейсов клавиатуры на основе БИС КР580ВВ79 модуль УМК-80/МИБ имеет блок переключа-

```

; ----- ПРОГРАММА 2 -----
;
; ПРОГРАММА ПРЕДНАЗНАЧЕНА ДЛЯ ИССЛЕДОВАНИЯ РЕЖИМОВ 1 И 2
; КЛАВИАТУРНОЙ ЧАСТИ. ДЛЯ ИМИТАЦИИ ОДНОВРЕМЕННОГО СРАБАТЫВАНИЯ
; ДАТЧИКОВ МОЖНО ВОСПОЛЬЗОВАТЬСЯ СЛЕДУЮЩИМ ПРИЕМОМ: ЗАПРЕТИТЬ
; ВЫБОРКУ РЕГИСТРОВ D17, D18 ЗАМКНУВ ПЕРЕКЛЮЧАТЕЛЬ SA2 (SA1
; ОСТАЕТСЯ РАЗОМКНУТЫМ). ЗАМКНУТЬ НЕСКОЛЬКО ПЕРЕКЛЮЧАТЕЛЕЙ В
; НУЛЕВОЙ И/ИЛИ В ПЕРВОЙ СТРОКЕ. ЗАТЕМ РАЗРЕШИТЬ ВЫБОРКУ РЕГИСТРОВ
; D17, D18. ВО ВСЕХ ДРУГИХ СЛУЧАЯХ - SA1, SA2 РАЗОМКНУТЫ.
;
START: MVI A, COM0 ; УСТАНОВИТЬ РЕЖИМ 1 ИЛИ 2 КЛАВИАТУРНОЙ
      OUT R79+1 ; ЧАСТИ
      MVI A, COM2 ; COM2 - КОМАНДА "ЧТЕНИЕ СТЕКА - ОЗУ
                ; ДАТЧИКОВ"
;
; D7, D6, D5 = 0, 1, 0
; D4 = 0 - ЧТЕНИЕ БЕЗ АВТОИНКРЕМЕНТИРОВАНИЯ
;         = 1 - ЧТЕНИЕ С АВТОИНКРЕМЕНТИРОВАНИЕМ
; D3 - НЕ ИСПОЛЬЗУЕТСЯ
; D2, D1, D0 - АДРЕС ЯЧЕЙКИ, ЧИТАЕМОЙ МП БИС
;
      OUT R79+1 ;
M1: IN R79+1 ; ЧТЕНИЕ СЛОВА СОСТОЯНИЯ БИС КР580ВВ79
   OUT PORT ; ЗАПИСЬ СЛОВА СОСТОЯНИЯ В ВЫХОДНОЙ
           ; ПОРТ МИКРОЭВМ
;
      RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ
M2: IN R79 ; ЧТЕНИЕ ДАННЫХ ИЗ СТЕКА - ОЗУ ДАТЧИКОВ
   OUT PORT ; И ЗАПИСЬ ИХ В ВЫХОДНОЙ ПОРТ МИКРОЭВМ
   RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ
   JMP M1
; ИССЛЕДОВАТЬ ПОРЯДОК СБРОСА ЗАПРОСА ПРЕРЫВАНИЯ МОЖНО НАБЛЮДАЯ ЗА
; СОСТОЯНИЕМ ВЫВОДА INT БИС КР580ВВ79 ПО СВЕТОДИОДУ HL1 МОДУЛЯ
; УМПК-80/МИ6 ПРИ ВЫПОЛНЕНИИ КОМАНДЫ С МЕТКОЙ M2 ПО МАШИННЫМ
; ЦИКЛАМ. В ДАННОЙ ПРОГРАММЕ КОДЫ КОМАНД: COM0 = 0FH (РЕЖИМ 1),
; COM1 = 0AH (РЕЖИМ 2); COM2 = 40H.

```

```

; ----- ПРОГРАММА 3 -----
;
; ПРОГРАММА СЛУЖИТ ДЛЯ ИССЛЕДОВАНИЯ РЕЖИМА 3 КЛАВИАТУРНОЙ ЧАСТИ,
; КОТОРЫЙ УСТАНОВЛИВАЕТСЯ ИЗ РЕЖИМА 2 С ПОМОЩЬЮ КОМАНДЫ COM7. ДЛЯ
; ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТОВ ИСПОЛЬЗУЕТСЯ ТОТ ЖЕ ПРИЕМ, ЧТО И В
; ПРОГРАММЕ 2. УБЕДИТЬСЯ, ЧТО УСТАНОВЛЕННЫЙ ПРИ ОБНАРУЖЕНИИ ОШИБКИ
; В РЕЖИМЕ 3 ЗАПРОС ПРЕРЫВАНИЯ СБРАСЫВАЕТСЯ ТОЛЬКО КОМАНДОЙ COM6,
; МОЖНО ИСКЛЮЧИВ ИЗ ПРОГРАММЫ КОМАНДЫ С МЕТКАМИ M2 И M3.
;
START: MVI A, COM0 ; УСТАНОВКА РЕЖИМА 2 КЛАВИАТУРНОЙ ЧАСТИ
      OUT R79+1
M1: MVI A, COM7 ; COM7 - КОМАНДА "СБРОС ПРЕРЫВАНИЯ -
                ; УСТАНОВКА РЕЖИМА 3 КЛАВИАТУРНОЙ ЧАСТИ"
;
; D7, D6, D5 = 1, 1, 1
; D4 = 1 - УСТАНОВКА РЕЖИМА 3
;         = 0 - ОТМЕНА РЕЖИМА 3
; D3, D2, D1, D0 - НЕ ИСПОЛЬЗУЮТСЯ
;
      OUT R79+1
      MVI A, COM2 ; УСТАНОВКА ИСТОЧНИКА ЧТЕНИЯ СТЕКА - ОЗУ
      OUT R79+1 ; ДАТЧИКОВ
      RST 1 ; ОСТАНОВ
      IN R79+1 ; ЧТЕНИЕ СЛОВА СОСТОЯНИЯ БИС
      OUT PORT ; И ЗАПИСЬ ЕГО В ВЫХОДНОЙ ПОРТ МИКРОЭВМ
      RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ
M2: MVI A, COM6 ; COM6 - КОМАНДА "СБРОС"
;
; D7, D6, D5 = 1, 1, 0
; D4 = 0 - ЗАДАНИЕ КОДА ОЧИСТКИ ДЛЯ ВЫХОДОВ DSP
;         = 1 - ОЧИСТКА ОЗУ ОТОБРАЖЕНИЯ
; D3, D2 = 0, X - КОД ОЧИСТКИ 00H
;         = 1, 0 - КОД ОЧИСТКИ 20H
;         = 1, 1 - КОД ОЧИСТКИ 0FFH
; D1 = 1 - СБРОС РЕГИСТРА СЛОВА СОСТОЯНИЯ И ЗАПРОСА
;         ; ПРЕРЫВАНИЯ
; D0 = 1 - ПРОГРАММНЫЙ СБРОС (СМ. ТАБЛ. 2)
;
M3: OUT R79+1
   IN R79 ; ЧТЕНИЕ ДАННЫХ ИЗ СТЕКА - ОЗУ ДАТЧИКОВ
   OUT PORT ; И ЗАПИСЬ ИХ В ВЫХОДНОЙ ПОРТ МИКРОЭВМ
   RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ
   JMP M1
;
; В ДАННОЙ ПРОГРАММЕ КОДЫ КОМАНД: COM0 = 0AH, COM7 = 0FH,
; COM2 = 40H, COM6 = 0C2H.

```

телей, в котором SA5...SA10 имитируют нулевую, SA13...SA20 — первую и SB1...SB3 — вторую строки матрицы датчиков. Нулевая и первая строки подключаются с помощью регистров D17 и D18, датчики второй строки включены непосредственно в матрицу. Сигналы сканирования матрицы датчиков клавиатуры формирует дешифратор D10. Сигналы выборки на регистры D17 и D18 подаются от схемы управления D12...D14 в зависимости от положения переключателей SA1, SA2. Сигналы SH и CO/STB имитируются переключателями SA3 и SA4.

При сканировании матрицы датчиков (режим 4) в каждой из восьми ячеек буферного ОЗУ хранится информация об одной из ее строк, причем все разряды ячеек отображают состояние соответствующих датчиков. Доступ к ОЗУ датчиков возможен после аппаратного сброса или выдачи команды «Чтение стека — ОЗУ датчиков». Команда задает также способ доступа (произвольный или автоинкрементный) к ОЗУ датчиков и адрес ячейки, из которой будут читаться данные. Первый способ подразумевает выдачу команды каждый раз при необходимости установить адрес новой ячейки и следить за состоянием датчиков одной строки. Опрос ОЗУ датчиков вторым способом предполагает выдачу только одной команды, в которой указывается адрес ячейки и установлен флаг автоинкрементирования. При каждом обращении к ОЗУ датчиков адрес ячейки автоматически увеличивается, что даст возможность последовательно просматривать содержание всех или части ячеек. При сбросе устанавливается адрес нулевой ячейки ОЗУ датчиков и произвольный способ доступа (см. табл. 2).

Режим 4 предполагает программное определение датчика, изменившего состояние (функциональные группы датчиков лучше размещать в одной или последовательных строках матрицы). БИС инвертирует сигналы со входов RET0...RET7 перед записью в ОЗУ датчиков; разряды D0...D5 слова состояния в этом режиме не используются и принимают произвольное значение (D4=D5=0, D0...D3 — номер строки матрицы, которая сканировалась в момент чтения слова состояния), а разряд D6=1, если хотя бы один датчик в матрице находится в состоянии Лог. 0. После выдачи команд, сбрасывающих слово состояния, достоверная информация с разряда D6 может быть получена только через интервал времени, необходимый для опроса всех строк матрицы.

Программы обслуживания интерфейса клавиатуры в режиме 4 сложнее, поэтому его применение оправдано лишь в тех случаях, когда требуется определять моменты нажатия и отжатия клавиш, а также время нахождения в каждом положении (например, при организации автоматического повторения ввода, если клавиша не от-

пушена через определенное время).

Интерфейс клавиатуры при опросе матрицы в режиме 4 сверяет текущее состояние датчиков сканируемой строки с информацией о состоянии, полученной во время предыдущего сканирования и хранящейся в ОЗУ датчиков, а также обновляет эту информацию. Состояния датчиков сканируемой строки проверяются последовательно (см. рис. 4), а запись новой информации о состоянии выполняется одновременно с проверкой датчика, подключенного к входу RET7. При обнаружении первого датчика, изменившего свое состояние, устанавливается запрос прерывания и после фиксации состояния текущей строки дальнейшая запись в ОЗУ датчиков запрещается. Запрос снимается и запись в ОЗУ датчиков возобновляется автоматически при чтении данных из любой ячейки, если в последней команде «Чтение стека — ОЗУ датчиков» был сброшен флаг автоинкрементирования (т. е. при произвольном доступе), или командой «Сброс прерывания», если этот флаг был установлен. При возобновлении записи интерфейс зафиксирует переключение только тех датчиков, информация на выходах которых к этому моменту отлична от информации в соответствующих ячейках ОЗУ датчиков. Например, изменение состояния датчика, переключившегося два раза за время запрета записи, обнаружено не будет. С учетом этого условия минимальная длительность сигнала на выходе датчика должна превышать время выполнения подпрограммы обслуживания прерывания.

Работа интерфейса в режиме 4 имеет ряд особенностей, затрудняющих анализ информации по прерываниям. Так, наличие в одной строке двух датчиков, изменивших до опроса свое состояние, приведет к генерации одного запроса прерывания. Однако если такие датчики расположены в разных строках матрицы, то будут сформированы два запроса. Исключение составляет нулевая строка матрицы, в которой множественные изменения состояния датчиков до опроса приводят к формированию такого же числа запросов прерывания и запретов записи в ОЗУ датчиков. В этом случае информативным является только первое прерывание, так как данные о состоянии строки вводятся сразу же и могут быть считаны подпрограммой обслуживания. Опрос ОЗУ датчиков по следующим прерываниям (при отсутствии других изменений) не дает новой информации. Кроме того, возможна установка запроса прерывания при сканировании нулевой строки после подачи напряжения питания при отсутствии каких-либо изменений в матрице датчиков, что необходимо учитывать при разработке программы начальной установки БИС. Входы SH и CO/STB в режиме 4 не работают.

Если необходимо использовать всю матрицу датчиков, то наиболее удобным является алгоритм, отводящий некоторую область ОЗУ микроЭВМ под копию содержимого ОЗУ датчиков БИС. Информация из ОЗУ датчиков читается по прерыванию, причем каждый байт сверяется с копией. При обнаружении расхождений данные в ОЗУ микроЭВМ обновляются, определяются датчики, изменившие состояние, информация о них кодируется (например, разряды D0...D2 — номер датчика в строке, D3...D5 — номер строки, D6=1 — датчик включился, а D6=0 — датчик выключился) и заносится в специальный буфер для дальнейшей обработки. После просмотра всего ОЗУ датчиков устройство выходит из подпрограммы обслуживания прерывания. Недостаток такого алгоритма — возможность холостых вызовов при множественных изменениях в нулевой строке.

Формат представления данных, особенности анализа информации о датчиках, изменивших свое состояние, порядок установки и сброса запросов

прерывания, а также работы со словом состояния в режиме 4 клавиатурной части БИС можно исследовать с помощью программы 4.

Ввод информации по строку (режим 5) дает возможность осуществить прием и хранение в буферном ОЗУ БИС данных от любого внешнего устройства. В этом режиме буферное ОЗУ организовано как стек. Функционирование ОЗУ, доступ к нему, формирование запроса прерывания, а также использование информации разрядов D0...D5 слова состояния осуществляются аналогично режимам 1...3.

Информация от внешнего устройства в режиме 5 принимается со входов RET0...RET7 по стробирующему импульсу, подаваемому на вход CO/STB. По его срезу информация фиксируется внутренними схемами БИС, а по фронту — перезаписывается в стек. Время между срезом и фронтом должно быть не менее  $T_{CLC}$ . Данные в процессе записи в буферное ОЗУ инвертируются.

Для исследования порядка записи

```

;
; ----- ПРОГРАММА 4 -----
;
; ПРИ ИССЛЕДОВАНИИ МНОЖЕСТВЕННЫХ ИЗМЕНЕНИЙ В НУЛЕВОЙ СТРОКЕ МАТРИ-
; ЦЫ ДАТЧИКОВ, В КОМАНДЕ SOM2 УСТАНАВЛИВАЕТСЯ РЕЖИМ ДОСТУПА К
; СТЕКУ - ОЗУ ДАТЧИКОВ БЕЗ АВТОИНКРЕМЕНТИРОВАНИЯ, АДРЕС - 000.
; ЧИСЛО ДАТЧИКОВ, ИЗМЕНИВШИХ СВОЕ СОСТОЯНИЕ ЗА ВРЕМЯ ЗАПРЕТА ЗАПИ-
; СИ В СТЕК - ОЗУ ДАТЧИКОВ БУДЕТ ОПРЕДЕЛЯТЬ ЧИСЛО ЗАПРОСОВ ПРЕРЫ-
; ВАНИЯ. ПРИ УСТАНОВЛЕННОМ АВТОИНКРЕМЕНТНОМ СПОСОБЕ ДОСТУПА К
; СТЕКУ - ОЗУ ДАТЧИКОВ (РЕЖИМ 4) ЗАПРОС ПРЕРЫВАНИЯ СБРАСЫВАЕТСЯ
; КОМАНДОЙ SOM0 ИЛИ SOM7. УБЕДИТЬСЯ В ЭТОМ МОЖНО УДАЛИВ ИЗ
; ПРОГРАММЫ КОМАНДЫ С МЕТКАМИ M2, M3.
;

```

```

MVI A, SOM2 ; УСТАНОВИТЬ ИСТОЧНИК ЧТЕНИЯ СТЕКА -
OUT P79+1 ; ОЗУ ДАТЧИКОВ И СПОСОБ ДОСТУПА К НЕМУ
M1: IN P79+1 ; ЧТЕНИЕ СЛОВА СОСТОЯНИЯ БИС
OUT PORT ; И ЗАПИСЬ ЕГО В ВЫХОДНОЙ ПОРТ МИКРОЭВМ
RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ
IN P79 ; ЧТЕНИЕ ДАННЫХ ИЗ СТЕКА - ОЗУ ДАТЧИКОВ
OUT PORT ; И ЗАПИСЬ ИХ В ВЫХОДНОЙ ПОРТ МИКРОЭВМ
RST 1
M2: MVI A, SOM0 ; ИЛИ SOM7 - СБРОС ЗАПРОСА ПРЕРЫВАНИЯ
M3: OUT P79+1
RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ СОСТОЯНИЯ ВЫХОДА
; INT ИНТЕРФЕЙСНОЙ БИС
;
; JMP M1
;

```

; В ДАННОЙ ПРОГРАММЕ КОДЫ КОМАНД: SOM0 = 0СН, SOM2 = 4ОН,  
; SOM6 = 0С2Н, SOM7 = 0ЕОН.

```

;
; ----- ПРОГРАММА 5 -----
;
; ДЛЯ ИМИТАЦИИ ВНЕШНЕЙ КЛАВИАТУРЫ ВОСПОЛЬЗУЙТЕСЬ СЛЕДУЮЩИМ ПРИЕ-
; МОМ: ЗАМКНУТЬ ПЕРЕКЛЮЧАТЕЛИ SA1, SA2 ПРИ ЭТОМ РЕГИСТР D17 ВЫБРАН
; ПОСТОЯННО. СТРОБИРУЮЩИЙ СИГНАЛ НА ВХОД СО/STB ИНТЕРФЕЙСНОЙ БИС
; ПОДАЕТСЯ С ПЕРЕКЛЮЧАТЕЛЯ BA3.
;

```

```

START: MVI A, SOM0 ; УСТАНОВИТЬ РЕЖИМ 5 КЛИАВИАТУРНОЙ ЧАСТИ
OUT P79+1
M1: IN P79+1 ; ЧТЕНИЕ СЛОВА СОСТОЯНИЯ БИС
OUT PORT ; И ЗАПИСЬ ЕГО В ВЫХОДНОЙ ПОРТ МИКРОЭВМ
RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ
IN P79 ; ЧТЕНИЕ ДАННЫХ ИЗ СТЕКА - ОЗУ ДАТЧИКОВ
OUT PORT ; И ЗАПИСЬ ИХ В ВЫХОДНОЙ ПОРТ МИКРОЭВМ
RST 1 ; ОСТАНОВ ДЛЯ ИНДИКАЦИИ
;
; JMP M1
;

```

; В ДАННОЙ ПРОГРАММЕ КОД КОМАНДЫ SOM0 = 0ЕН

данных в стек клавиатурной части БИС в режиме стробируемого ввода можно воспользоваться программой 5.

Разработка схемы клавиатуры и ее интерфейса на основе БИС КР580ВВ79 состоит, как правило, из трех этапов. На этапе 1 с учетом требований к организации диалога с оператором определяются число датчиков в клавиатуре, способ их разбиения на функциональные группы и режим работы клавиатурной части БИС. Одновременно выбирается тип датчиков, исходя из требований к размещению, условиям эксплуатации, надежности и эргономике. На этапе 2 определяются размер матрицы, размещение в ней датчиков (с учетом требований к кодам датчиков и разбиения на функциональные группы) и режим сканирования БИС. Затем, исходя из принципиальной схемы датчика, выбирается способ организации мультиплексного опроса матрицы и, если необходимо, на функциональном уровне прорабатывается схема преобразования сигналов датчиков. На этапе 3 создается принципиальная электрическая схема клавиатуры и интерфейса.

Для непосредственного включения можно применять датчики, устанавливающие на выходе сигнал только во время опроса соответствующей строки матрицы и не влияющие на его состояние в остальное время (рис. 9). Простейшие клавишные механические датчики можно подключать в матрицу непосредственно при условии формирования сигналов сканирования логическими элементами с открытым коллектором (рис. 9, а). Использование разделительных диодов (рис. 9, б) или электронной коммутации (рис. 9, в) позволяет избежать влияния двух или более одновременно замкнутых в столбце клавиш на выходы схемы формирования сигналов сканирования (замыкание выхода с Лог. 1 на выход с Лог. 0). Непосредственное включение интегральных клавишных переключателей в матрицу возможно, если они имеют выход с открытым коллектором или третьим состоянием и вход выборки, как например,

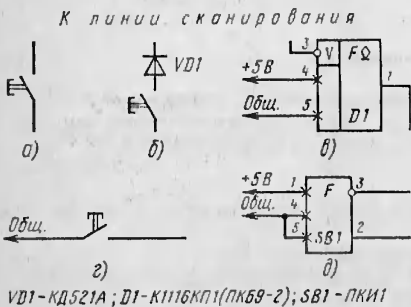
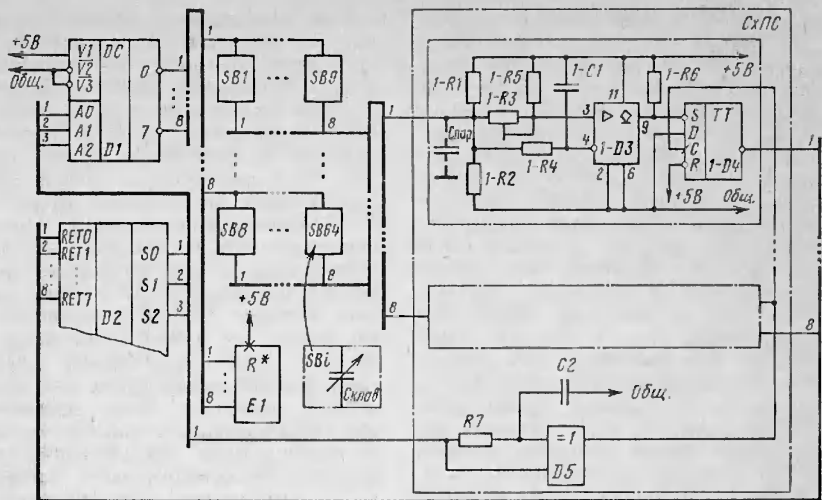
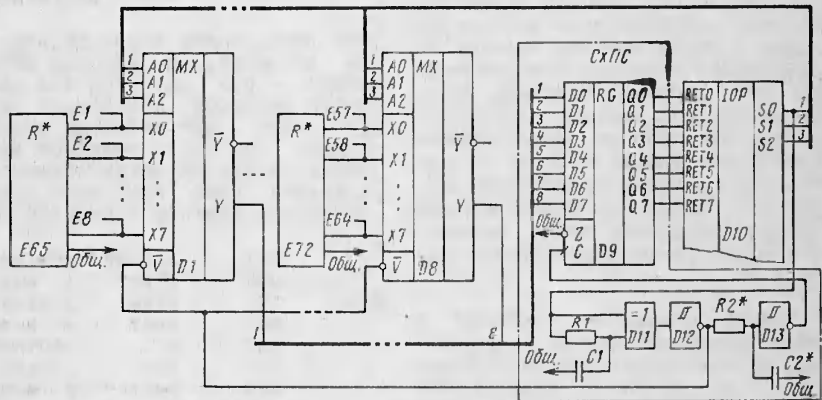


Рис. 9. Клавишные датчики: а, б, в — для непосредственного включения в матрицу, г, д — для электронной коммутации



С1-0,068мкФ; С2-390пФ; D1-K555ИД7; D2-КР580ВВ79; D3-K554СА3; D4-1/2К555ТМ2; D5-1/4К555П16; E1-НР1-4-9-1кОм; R1, R2, R5-1/8НР1-4-9-560Ом; R3-22кОм; R4-91кОм; R6-1/8НР1-4-9-1кОм; R7-360Ом

Рис. 10. Подключение емкостных клавишных датчиков



D1... D8-K555КП7; D9-K555ИР23; D10-КР580ВВ79; D11-1/6К555П15; D12, D13-1/6К555ТМ2; E1... E8-сенсорные датчики; E55... E72-НР1-4-9-22кОм; R1-1кОм; R2-560Ом; С1-180пФ; С2-22пФ

Рис. 11. Подключение сенсорных датчиков

переключатель ПК69-2, построенный на основе микросхемы К1116КП1 (рис. 9, в), использующей эффект Холла. Если клавиши имеют обычный ТТЛ-выход, то возможна лишь электронная коммутация (рис. 9, д).

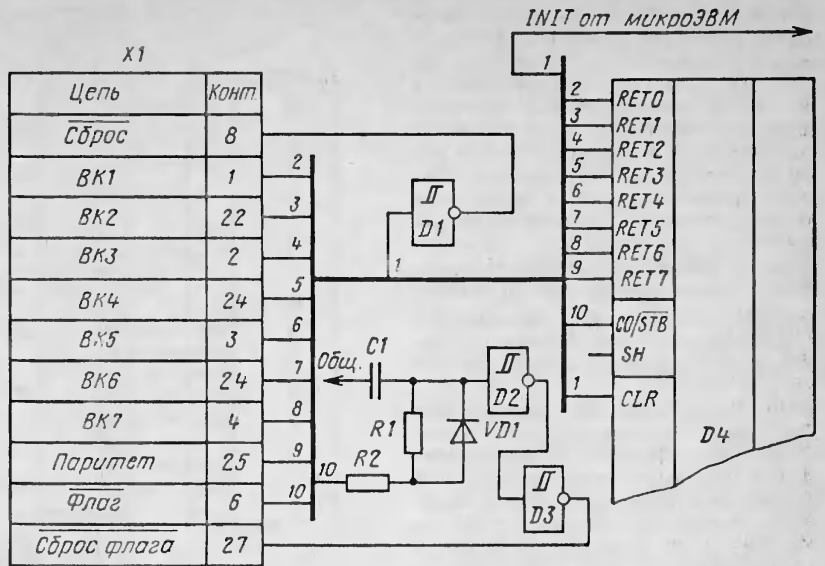
Более сложно организовать клавиатуру на основе емкостных клавишных датчиков (рис. 10). При нажатии емкость датчика  $C_{\text{клав}}$  увеличивается, что приводит к возрастанию величины перепада напряжения, передаваемого с выхода дешифратора формирования сигналов опроса матрицы D1 на ее возвратную линию через емкость  $C_{\text{клав}}$ . Малое приращение емкости датчика, а также наличие большой паразитной емкости проводников  $C_{\text{пар}}$  делают изменения перепада незначительными и требуют при-

менения чувствительной схемы преобразования сигналов. Замыкание обнаруживается с помощью дифференциального усилителя D3, сравнивающего усредненное интегрирующей цепью R4C1 напряжение возвратной линии с сигналом на этой же линии. Триггер D4 служит для запоминания сигнала на период опроса и сбрасывается в начале сканирования каждой строки матрицы. Резисторы R1 и R2 необходимы для привязки потенциала возвратной линии, R3 — регулировки чувствительности, E1 — увеличения перепада напряжения на выходах дешифратора D1.

Использование сенсорных датчиков также требует дополнительного аппаратного обеспечения. Наиболее простым оно будет для датчиков, определяю-

щих касание по вносимой емкости (рис. 11). Такие датчики обычно выполняются в виде площадок на печатной плате. Начальная емкость сенсора определяется суммой емкостей площадки и проводника, соединяющего ее со схемой. При касании емкость сенсора возрастает. Каждая из линий X0...X7 мультиплексоров D1...D8 является одним из входов многоходового логического элемента И (другие входы используются для выбора канала по адресу с линией A0...A2 и подачи сигнала выборки V). При отсутствии сигнала выборки емкость сенсоров E1...E64 разряжена резисторами E65...E72. В начале сканирования каждой строки схема D11, D12, R1, C1 формирует сигнал, разрешающий работу мультиплексоров (опрос сенсоров в этой схеме аналогичен показанному на рис. 2, б). Емкости опрашиваемых датчиков начинают заряжаться входным током соответствующих элементов. Длительность процесса заряда емкости до напряжения, соответствующего уровню Лог. 1, определяется емкостью сенсора. Через некоторое время (определяемое цепью R2, C2), достаточное для того, чтобы сенсоры, к которым не прикасались, успели зарядиться до этого напряжения, а сенсоры, к которым прикасались — нет, происходит запись информации с выходов мультиплексоров в регистр TD9. В связи с тем, что сенсоры имеют малую емкость, схема успевает сработать до начала опроса линии RET0 (см. рис. 3).

БИС KP580BB79 наряду с обслуживанием матрицы датчиков клавиатуры можно применять для опроса восьми каналов ввода медленно изменяющейся информации общего назначения. При этом схема такого интерфейса не будет отличаться от приведенных на рис. 2.



D1...D3 — 1/6K555TL2, D4 — KP580BB79

R1 — 10кОм, R2 — 560 Ом, C1 — 1мкФ, X1 — Вилка РП15 — 32ШБК

Рис. 12. Подключение блока клавиатуры 15BVB-97-006

Например, контроль за состоянием массива датчиков предаварийной ситуации может выполняться БИС, работающей в режиме 2 или 4. В первом случае будет выдаваться код сработавшего датчика, однако число одновременно включенных датчиков ограничено.

Режим 5 БИС можно использовать, например, для подключения функционально законченных клавиатур, осуществляющих выдачу кодов нажатых клавиш в параллельном коде (рис. 12).

В режиме 4 в отличие от режимов 1...3 и 5 ввод информации общего на-

значения и получение информации о состоянии клавиатуры при отсутствии в микроЭВМ системы прерываний или по прерываниям от системного таймера затруднено или невозможно. Это связано с запретом записи в ОЗУ датчиков при обнаружении изменений в матрице. Устранить этот недостаток можно с помощью специальной схемы (рис. 13) мультиплексного опроса датчиков клавиатуры или любых входных устройств в режиме стробирующего ввода. Запись данных в БИС осуществляется после подачи сигнала запуска и завершения

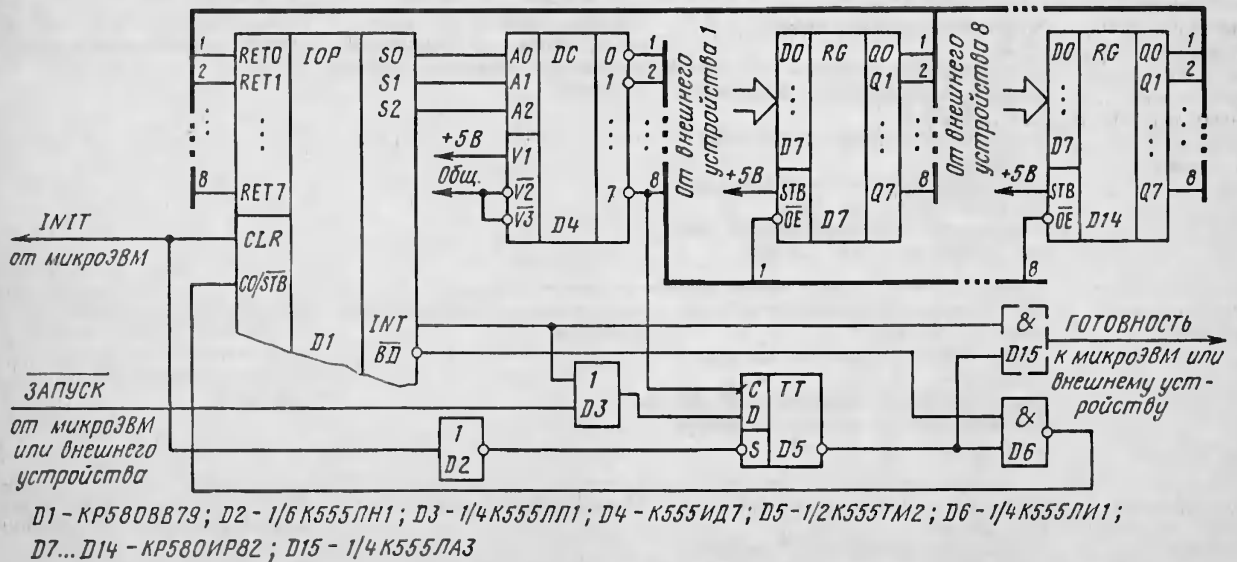


Рис. 13. Организация мультиплексного опроса при опробуемом вводе информации

чтения всей информации предыдущего опроса из стека. Схема управления D2... D6 при поступлении сигнала запуска опрашивает по очереди регистры D7 ... D14 (используется и непосредственное включение датчиков клавиатуры в матрицу) и формирует восемь раз строб записи в БИС. Окончание процесса ввода определяется программно по слову состояния (заполненный стек) или по сигналу готовности с элемента D15. К недостатку приведенного варианта схемы следует отнести необходимость использования сигнала  $\overline{BD}$  и, следовательно, невозможность работы в условиях полного запрета отображения.

При разработке схем интерфейсов необходимо учитывать, что входы  $\overline{RETO}$ ... $\overline{RET7}$ , SH, CO/STB имеют внутренние активные сопротивления для поддержания в состоянии Лог. 1. В режимах 1...3 ко входам SH и CO/STB можно подсоединить переключатели (с фиксацией или без нее), используемые для выбора регистров, алфавита и т. п. Следует помнить, что для работы некоторых схем интерфейсов после включения питания требуется не только выдача команд задания режима работы БИС, но и очистка стека или сброс прерываний.

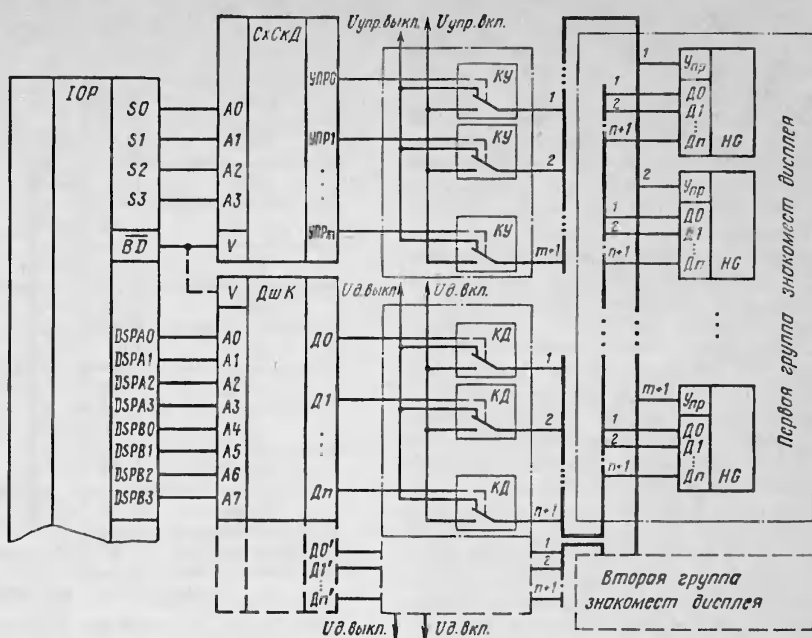


Рис. 14. Мультиплексное отображение информации

Таблица 3

Тип ЗСИ	Параметры питающих напряжений	Достоинства	Недостатки
Полупроводниковые	Напряжение питания — 2...5 В Максимальный постоянный ток сегмента — 4...25 мА Импульсный ток сегмента — 20...300 мА	Быстродействующие (время включения 10 нс), долговечность, высокая механическая стойкость, малые габариты, три цвета свечения	Высокая потребляемая мощность, большая стоимость
Вакуумные люминесцентные	Напряжение питания накала — 0,7...5,5 В Ток потребления накала — 50...100 мА Импульсное напряжение питания сетки — 20...70 В Импульсный ток потребления сетки — 2,5...45 мА Импульсное напряжение питания анодов сегментов — 20...70 В Импульсный ток потребления анодов сегментов — 0,1...1,3 мА	Высокая яркость, долговечность, малая потребляемая мощность, несколько цветов свечения	Несколько источников питания, низкая механическая стойкость
Вакуумные накаливаемые	Импульсное напряжение питания сегмента — 20...300 В Эффективное напряжение питания сегмента — 4,5...7,0 В Ток потребления сегмента — 20...50 мА Ток потребления сегмента в момент включения — 70...150 мА	Дешевизна, стабильность параметров	Инертность (время включения 25...50 мс), низкие вибростойкость и механическая прочность, высокое потребление, один цвет свечения
Газоразрядные сегментные постоянного тока	Напряжение поджига разряда — 190...220 В Напряжение поддержания разряда — 120...170 В Ток потребления сегмента — 25...40 мкА	Малая потребляемая мощность	Высокое питающее напряжение, низкая механическая прочность



## Интерфейс дисплея

Интерфейс дисплея должен выполнять следующие функции: регенерацию изображения, буферирование отображаемой информации, декодирование информации для ЗСИ, сервис (очистка дисплея, запрет отображения, установка порядка вывода информации на дисплей и способа доступа к буферному ОЗУ дисплея). Сложность интерфейса определяется видом отображаемой информации и физическим принципом действия ЗСИ [6]. Сравнительная характеристика различных типов ЗСИ (табл. 3), можно заключить, что наиболее просто с ТТЛ микросхемами сопрягаются полупроводниковые индикаторы.

При мультиплексном отображении информации (динамическая индикация) выбор знакоместа, на которое она будет выводиться в текущий момент времени, осуществляется сигналами сканирования (рис. 14) с помощью ключей управления (КУ), а включение необходимых сегментов — ключами данных (КД). Устойчивое изображение на дисплее получается при условии регенерации с частотой, большей 40 Гц.

В интерфейсах на основе БИС КР580ВВ79 мультиплексное отображение информации осуществляется под управлением счетчика сканирования. При дешифрированном режиме работы счетчика непосредственно на выходах БИС формируется четыре сигнала сканирования, что дает возможность использовать до четырех знакомест в одной группе. В случае кодированного сканирования в зависимости от запрограммированного режима работы дисплея можно получить с помощью внешней схемы формирования сигналов сканирования  $SxCK$  до восьми или шестнадцати таких сигналов. В первых двух случаях период сканирования дисплея совпадает с периодом сканирования клавиатуры и составляет, соответственно,  $256 \cdot k \cdot T_{CLK}$  и  $512 \cdot k \cdot T_{CLK}$ , а в третьем случае он превышает его в два раза и равен  $1024 \cdot k \cdot T_{CLK}$ .

Буферирование отображаемой информации при динамической индикации необходимо для хранения информации, выводимой МП БИС на дисплей и выдачи в нужный момент при сканировании. Максимальный объем отображаемой информации (число знакомест и сегментов зависит от организации буферного ОЗУ и представления информации о символах (двоичный или семисегментный коды, КОИ-7 и т. п.)). Число сканируемых знакомест ограничено предельно допустимыми режимами ЗСИ (при возрастании скважности сканирующих импульсов для сохранения приемлемого уровня яркости необходимо увеличивать питающие токи или напряжения индикаторов). Дальнейшее повышение разрядности дисплея достигается одновременным сканированием нескольких групп знакомест и

параллельной выдачей информации на них из буферного ОЗУ.

Для буферирования выводимой на дисплей информации в БИС КР580ВВ79 служит внутреннее ОЗУ отображения объемом 16 байт. С помощью команды «Гашение — запрет записи» пользователь может организовать запись в ОЗУ как целыми байтами, так и отдельными тетрадами (чтение всегда осуществляется только побайтно). Это позволяет подключать вторую группу знакомест дисплея, удваивая их число в каждом режиме сканирования с помощью дешифратора кодов ДшК (см. рис. 14).

Данные для дисплея выдаются из ОЗУ отображения на выходы DSPB0... DSPB3 (тетрада В, разряды D0...D3 ОЗУ) и DSPA0...DSPA3 (тетрада А, разряды D4...D7). Чтобы исключить влияние переходных процессов в дешифраторе кодов интерфейса или подсветки индикаторов при переключениях сканирующих импульсов на качество изображения, БИС формирует сигнал запрета отображения  $\overline{BD}$  (рис. 15, а). Если этот сигнал активен, то на выходы DSP БИС выдается код очистки дисплея. Его значение (00, 20H, 0FFH) устанавливается программно командой «Сброс», (см. табл. 2). При аппаратном сбросе устанавливается код очистки 00, а содержимое ОЗУ отображения не изменяется. На время подачи сигнала сброса на вход CLR БИС ее выходы  $\overline{BD}$  и DSP устанавливаются в состояние Log.0. При необходимости (например, для устранения мигания дисплея в случае изменения МП БИС всего содержимого ОЗУ отображения) БИС

позволяет запрещать отображение тетрад А и (или) В с помощью команды «Гашение — запрет записи». На погашенные тетрады постоянно выдается код очистки дисплея (рис. 15 б, в).

Кроме программирования кодов очистки для выходов DSP, команда «Сброс» позволяет быстро очищать дисплей, заполняя все ячейки дисплейного ОЗУ кодами 00, 20H (код пробела в КОИ-7) или 0FFH. Выбор кода очистки зависит от построения ключей данных интерфейса. Во время очистки ( $16 \cdot k \cdot T_{CLK}$ ) ОЗУ отображения не доступно. Это состояние индицируется разрядом D7=1 слова состояния БИС.

Временные диаграммы выдачи информации на дисплей, способы запрета отображения и задания кода очистки можно исследовать, применяя программу 6.

С помощью МП БИС можно записывать и читать данные в ОЗУ отображения (после установки соответствующего источника чтения) с организацией произвольного или автоинкрементного способа доступа. Адрес ячейки и способ доступа задаются последней выполненной командой «Запись в ОЗУ отображения» или «Чтение ОЗУ отображения», а источник чтения — последней выполненной командой «Чтение стека — ОЗУ датчиков» или «Чтение ОЗУ отображения». БИС позволяет при программировании режима работы задавать один из двух порядков отображения информации на дисплее: ввод слева (как на обычных алфавитно-цифровых дисплеях) или ввод справа со сдвигом (как у микрокалькуляторов). Заданный порядок учитывается только при автоинкрементной записи в ОЗУ отображения.

Для организации правильного вывода информации на дисплей необходимо учитывать установленные режимы работы счетчика сканирования и дисплейной части БИС, а также используемый способ доступа к ОЗУ отображения. Если режим работы БИС задан таким

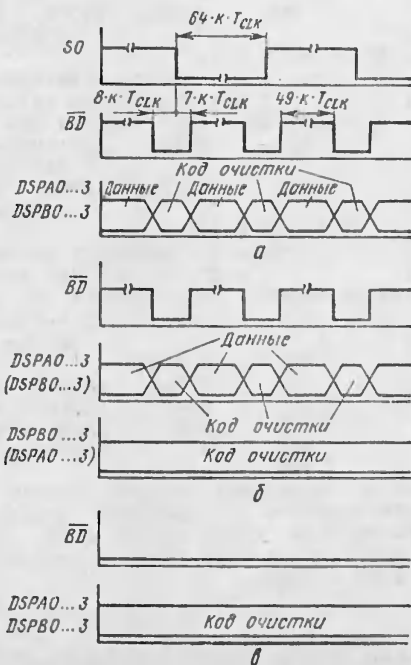


Рис. 15. Временные диаграммы выдачи информации на дисплей



Рис. 16. Порядок отображения информации на дисплее при вводе справа со сдвигом

образом, что формируется шестнадцать сигналов сканирования дисплея, то каждому знакоместу ставится в соответствие одна ячейка ОЗУ отображения. Соответствие будет однозначным (номер знакоместа совпадает с адресом ячейки ОЗУ), если не использовалась автоинкрементная запись при вводе справа со сдвигом. В противном случае соответствие нарушается (рис. 16). Из-за сложности определения адреса ячейки ОЗУ отображения конкретного знакоместа для правильного размещения информации на дисплее при ее обновлении удобнее вначале восстановить соответствие между ними, очистить дисплей выдчей командой «Сброс» или заново запрограммировать режим работы БИС. Следует отметить, что в случае работы БИС с формированием восьми сигналов сканирования, запись в ячейки с адресами  $k$  и  $k+8$  ( $k=0...7$ ) приведет к выводу информации на одно и то же знакоместо независимо от установленного порядка отображения, а информация, заносимая в любую из этих ячеек, запишется и в остальные. При дешифрованном сканировании дисплея каждому знакоместу поставлены в соответствие четыре ячейки ОЗУ отображения с адресами  $k$ ,  $k+4$ ,  $k+8$ ,  $k+12$  ( $k=0...3$ ).

Если знакоместа дисплея разбиты на несколько отдельных групп, то для вывода на них информации удобнее пользоваться произвольным способом доступа к ОЗУ отображения или вводом слева с автоинкрементным доступом.

Все режимы работы дисплейной части БИС, отличающиеся числом обслуживаемых знакомест, порядком отображения информации, способом доступа к ОЗУ отображения, а также метод записи в него отдельных тетрад можно исследовать с помощью программы 7.

Разработка схем дисплея и его интерфейса на основе БИС КР580ВВ79 начинается с выбора физического принципа действия ЗСИ, исходя из технических (потребляемой мощности, механической прочности, числа питающих напряжений, срока службы, стабильности параметров и т. п.) и эргономических (цвета свечения, контрастности, яркости, расстояния наблюдения, числа наблюдателей и т. п.) требований.

Одновременно по числу необходимых знакомест определяется структурная схема формирования сигналов сканирования дисплея (СхСкД) и скважность сканирующих импульсов. Далее по заданным скважности, яркости, цвету, размеру знака и привычности его начертания выбирается конкретный ЗСИ для мультиплексного отображения. Требуемая яркость должна обеспечиваться при заданной скважности, а соответствующие такой яркости напряжения не должны превышать предельно допустимых значений. Затем, исходя из вида отображаемой информации, числа сегментов ЗСИ и кодов символов, применяемых драйвером дисплея микроЭВМ, выбирают схему дешифратора

```

;
;
;----- ПРОГРАММА 6 -----
;
;ДЛЯ ИССЛЕДОВАНИЯ ВРЕМЕННОЙ ДИАГРАММЫ СИГНАЛА ВD И ПОРЯДКА
;УСТАНОВКИ КОДОВ ОЧИСТКИ ДЛЯ ВЫХОДОВ DSP БИС МОЖНО ВОСПОЛЬЗО-
;ВАТЬСЯ ДВУХЛУЧЕВЫМ ОСЦИЛЛОГРАФОМ. В ПЕРВОМ СЛУЧАЕ, ВХОДЫ X1 И
;X2 ОСЦИЛЛОГРАФА ПОДКЛЮЧАЮТСЯ К ВЫВОДАМ SO И VD, А ВО ВТОРОМ - К
;ВЫВОДАМ ВD И DSPVO. СИНХРОНИЗАЦИЯ ОСЦИЛЛОГРАФА ОСУЩЕСТВЛЯЕТСЯ ПО
;СИГНАЛУ SYNC (X2) МОДУЛЯ УМК-80/МИ6.
;
;
START: MVI A,СОМО ;УСТАНОВИТЬ КОДИРОВАННОЕ СКАНИРОВАНИЕ
        OUT P79+1 ; И ШЕСТНАДЦАТИСИМВОЛЬНЫЙ ДИСПЛЕЙ
        RST 1 ;ОСТАНОВ ДЛЯ НАБЛЮДЕНИЯ ОСЦИЛЛОГРАММ
;
; СИГНАЛОВ ВD И SO
M1: MVI A,СОМ6 ;ЗАПИСАТЬ КОД ОЧИСТКИ OFFH
     OUT P79+1 ; ДЛЯ ОЗУ ОТОБРАЖЕНИЯ
M2: MVI A,СОМ6 ;ЗАПИСАТЬ КОД ОЧИСТКИ 00 ДЛЯ
     OUT P79+1 ; ВЫХОДОВ DSP
     MVI A,СОМ5 ;СОМ5 - КОМАНДА "ГАШЕНИЕ - ЗАПРЕТ
; ОТОБРАЖЕНИЯ"
;
; D7,D6,D5 = 1,0,1
; D4 - НЕ ИСПОЛЬЗУЕТСЯ
; D3 = 1 - ЗАПРЕТ ЗАПИСИ В ТЕТРАДУ А
; D2 = 1 - ЗАПРЕТ ЗАПИСИ В ТЕТРАДУ В
; D1 = 1 - ГАШЕНИЕ ТЕТРАДЫ А
; D0 = 1 - ГАШЕНИЕ ТЕТРАДЫ В
;
        OUT P79+1
        RST 1 ;ОСТАНОВ ДЛЯ НАБЛЮДЕНИЯ ОСЦИЛЛОГРАММ
; СИГНАЛОВ ВD И DSPVO
; В ДАННОЙ ПРОГРАММЕ КОДЫ КОМАНД: СОМО = 08H,СОМ6 С МЕТКОЙ
; M1 = 0DCH, СОМ6 С МЕТКОЙ M2 = 0C0H. УСТАНОВИВШАЯ КОД КОМАНДЫ
; СОМ5 РАВНЫМ 0A2H ИЛИ 0A3H МОЖНО НАБЛЮДАТЬ, СООТВЕТСТВЕННО, ГА-
; ШЕНИЕ ТЕТРАДЫ А ИЛИ ПОЛНОЕ ГАШЕНИЕ ДИСПЛЕЯ.

```

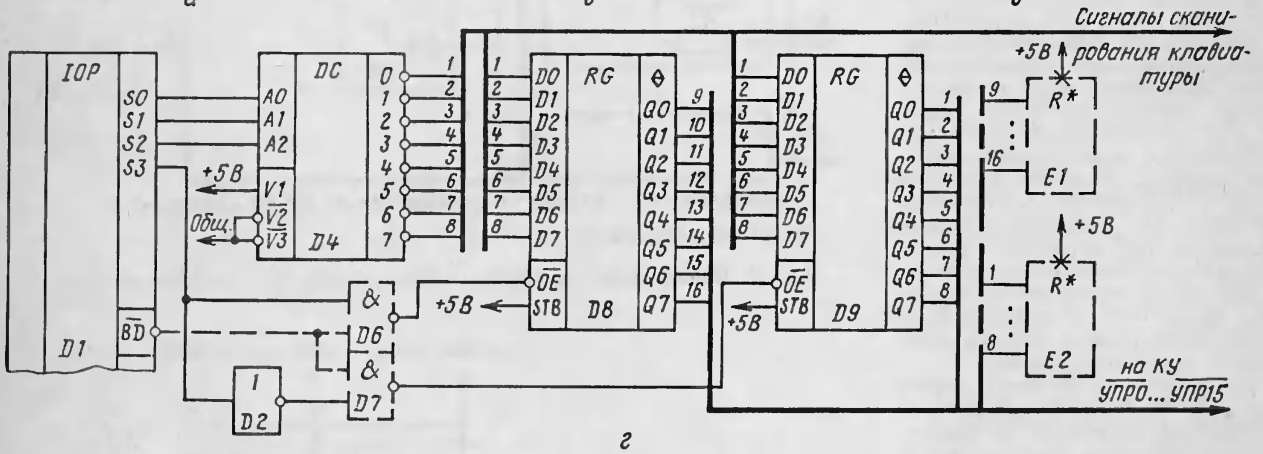
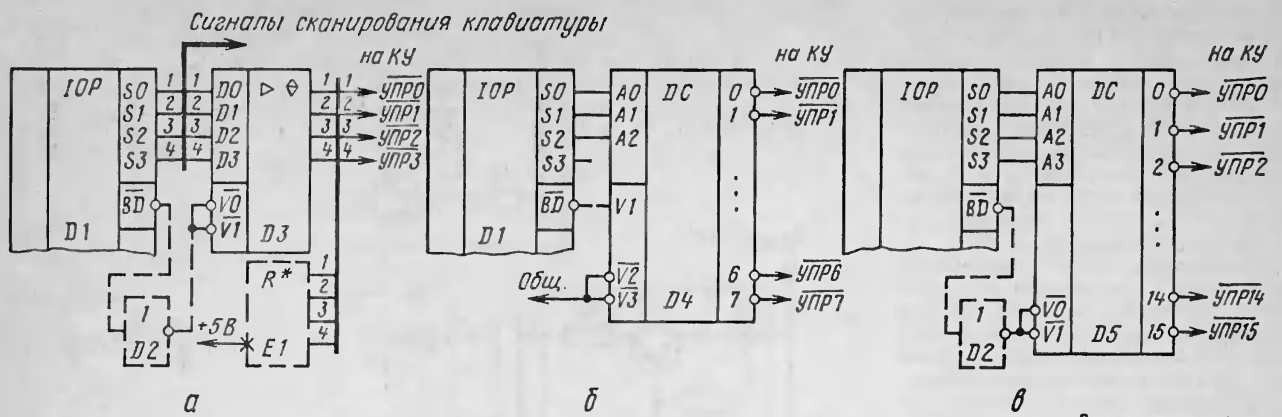
```

;
;----- ПРОГРАММА 7 -----
;
;ЭТА ПРОГРАММА ИСПОЛЬЗУЕТ ВХОДНОЙ ПОРТ МИКРОЭВМ ДЛЯ ЗАДАНИЯ
;КОДОВ, ЗАГРУЖАЕМЫХ В ОЗУ ОТОБРАЖЕНИЯ БИС.
;
START: MVI A,СОМО ;УСТАНОВКА КОДИРОВАННОГО СКАНИРОВАНИЯ
        OUT P79+1 ; ШЕСТНАДЦАТИСИМВОЛЬНОГО ДИСПЛЕЯ
;
; СО ВВОДОМ СЛЕВА
        MVI A,СОМ3 ;СОМ3 - КОМАНДА "ЧТЕНИЕ ОЗУ
; ОТОБРАЖЕНИЯ"
;
; D7,D6,D5 = 0,1,1
; D4 = 1 - ЧТЕНИЕ С АВТОИНКРЕМЕНТИРОВАНИЕМ
; D3,D2,D1,D0 - АДРЕС ЯЧЕЙКИ ДЛЯ ЧТЕНИЯ МП БИС
;
        OUT P79+1 ;УСТАНОВИТЬ ПРОИЗВОЛЬНЫЙ РЕЖИМ ЧТЕНИЯ
; ОЗУ ОТОБРАЖЕНИЯ С АДРЕСОМ - 000
;
        MVI A,СОМ4 ;СОМ4 - КОМАНДА "ЗАПИСЬ В ОЗУ
; ОТОБРАЖЕНИЯ"
;
; D7,D6,D5 = 1,0,0
; D4 = 1 - ЗАПИСЬ С АВТОИНКРЕМЕНТИРОВАНИЕМ
; D3,D2,D1,D0 - АДРЕС ЯЧЕЙКИ ДЛЯ ЗАПИСИ
;
        OUT P79+1 ;УСТАНОВИТЬ ПРОИЗВОЛЬНЫЙ РЕЖИМ ЗАПИСИ
; В ОЗУ ОТОБРАЖЕНИЯ С АДРЕСОМ - 000
;
        MVI A,СОМ5
        OUT P79+1 ;ОТМЕНИТЬ ВСЕ ЗАПРЕТЫ
M1: IN PORT ;ЧТЕНИЕ ВХОДНОГО ПОРТА МИКРОЭВМ
     OUT P79 ;ПОЛУЧЕННЫЙ КОД - В ОЗУ ОТОБРАЖЕНИЯ
     RST 1 ;ОСТАНОВ ДЛЯ ИНДИКАЦИИ
     JMP M2
;
;ВСЕ ОСТАЛЬНЫЕ РЕЖИМЫ РАБОТЫ ДИСПЛЕЙНОЙ ЧАСТИ БИС МОЖНО
;ИССЛЕДОВАТЬ С ПОМОЩЬЮ ПРОГРАММЫ 7 ПОСЛЕ СООТВЕТСТВУЮЩЕГО
;ИЗМЕНЕНИЯ КОДОВ КОМАНД СОМО, СОМ3...СОМ5. В ДАННОМ ВАРИАНТЕ
;ПРОГРАММЫ 7 КОДЫ КОМАНД: СОМО = 08H, СОМ3 = 60H, СОМ4 = 80H,
;СОМ5 = 0A0H.

```

кодов (ДшК). Учитывая определенные номиналы питающих напряжений и токов, а также вид (обычный выход, выход с открытым коллектором или с тремя состояниями) и нагрузочную способ-

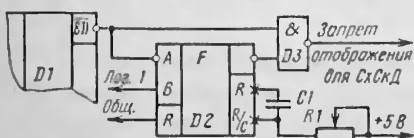
ность выходных сигналов СхСкД и ДшК, разрабатывают схемы ключей управления и данных. На заключительном этапе создается электрическая принципиальная схема дисплея и его



Д1 - КР580ВВ79; Д2 - 1/6К555ЛН1; Д3 - К155ЛП10; Д4 - К555ИД7; Д5 - К555ИД3; Д6, Д7 - К555ЛА3; Д8, Д9 - КР580ИР82(ИР83); Е1, Е2 - НР1-4-9

интерфейса, определяется частота сканирования дисплея, согласованная с частотой сканирования клавиатуры (как правило, компромиссное значение частоты находится легко). Порядок отображения и способ доступа к ОЗУ выбираются в зависимости от алгоритма обновления информации и организации дисплея (один массив знакомест, несколько разделных функциональных групп с независимым выводом, массив единичных индикаторов и т. п.).

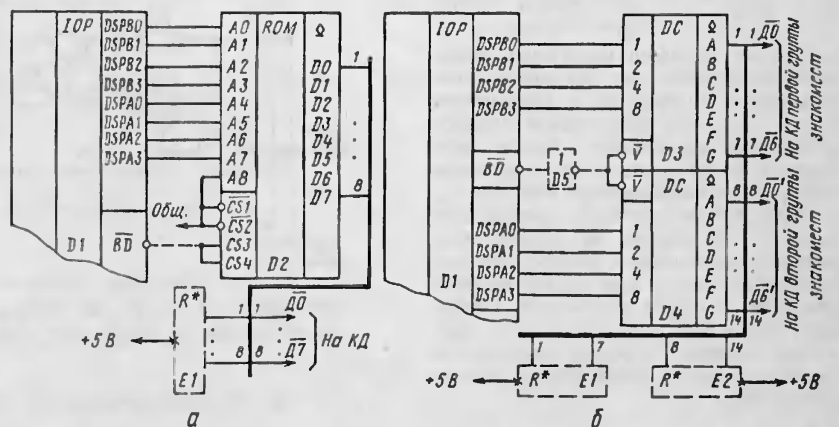
Структура схемы формирования сигналов сканирования дисплея и выбор режима работы БИС зависят, прежде всего, от числа сигналов (рис. 17, а-в). Если периоды сканирования дисплея и клавиатуры совпадают (например, для режимов БИС, соответствующих рис. 17, а, б), то можно использовать сигналы выходов СхСкД и для опроса клавиатуры.



Д1 - КР580ВВ79; Д2 - 1/2К555АГ3; Д3 - 1/4К555ЛА3; С1 - 0,1 мкФ; R1 - 47 кОм

Рис. 18. Схема управления яркостью свечения индикаторов

Рис. 17. Примеры построения схем формирования сигналов сканирования дисплея: а — дешифрованное сканирование ( $m=3$ ); б — кодированное сканирование и восьмисимвольный режим работы дисплея ( $m=7$ ); в, г — кодированное сканирование и шестнадцатисимвольный режим работы дисплея ( $m=15$ )



Д1 - КР580ВВ79; Д2 - КР556РТ5; Д3, Д4 - КР514ИД2; Д5 - К555ЛН1; Е1, Е2 - НР1-4-9

Рис. 19. Построение дешифраторов кодов

туры. Однако, если в СхСкД введено стробирование дешифраторов сигналом запрета отображения ВД для получения паузы между включением соседних зна-

комест (это требуется, например, в газоразрядных индикаторах ГИП-11, ИГП-17) или устранения подсветки сегментов ЗСИ при переключениях (на рис. 18

показано пунктиром), то сигналы сканирования дисплея и клавиатуры должны формироваться разными схемами. Исключения составляют схемы, в которых разделение сигналов сканирования осуществляется с помощью буферных элементов, одновременно выполняющих функцию усилителей (рис. 17, а, в). Изменяя длительность импульса стробирования СхСкД, можно управлять яркостью свечения ЗСИ (рис. 18).

Дешифраторы кодов применяются, как правило, в трех случаях: для упрощения программы драйвера дисплея (исключения таблицы перекодировки символов), при использовании в дисплее ЗСИ с числом сегментов одного знакоместа, большим восьми, а также при подключении к БИС двух групп цифровых ЗСИ (рис. 19). Для преобразования цифровой информации из двоичного или двоично-десятичного кода в семисегментный можно применять ИМС КР514ИД1, КР514ИД2, К514ПП1 или К155ПП4. При использовании других кодов дешифраторы удобно выполнять на основе ПЗУ КР556РТ4, КР556РТ5, К155РЕ3 и др. Требуемая выходная разрядность ДшК достигается параллельным подключением входов нескольких ПЗУ. Если перекодировка выполняется программно, то нужда в ДшК отпадает и сигналы с выходов БИС подаются непосредственно на КУ.

Структура ключей данных и управления зависит от требуемых уровней и нагрузочной способности их входных и выходных сигналов. Наиболее просто подключаются полупроводниковые ЗСИ с малым током потребления (см. рис. 5), так как вместо КД могут быть использованы ИМС с мощными выходами. Необходимость в ключах данных (рис. 20) возникает при возрастании тока, потребляемого одним сегментом из-за использования ЗСИ с большим потреблением или большего числа знакомест в дисплее. Резисторы R3 служат для выравнивания тока между сегментами ЗСИ и его ограничения.

При подключении вакуумных люминесцентных ЗСИ (рис. 21) для упрощения схемы интерфейса и уменьшения числа питающих напряжений анодное и сеточное напряжения выбирают равными, а накал питают постоянным или импульсным напряжением. Резистор R<sub>н</sub> служит для ограничения напряжения накала и получения на катоде положительного смещения, что обеспечивает лучшее записывание при подаче на анод или сетку нулевого потенциала, а резистор R2 — для предотвращения подсветки анодов сегментов выбранного знакоместа из-за токов утечки транзистора VT2.

Гораздо реже применяются вакуумные накаливаемые ЗСИ и газоразрядные сегментные индикаторы постоянного тока. Ключи для них выполняются по аналогичным схемам. При разработке схем ключей для вакуумных накаливаемых ЗСИ следует учитывать большое время включения из холодного состоя-

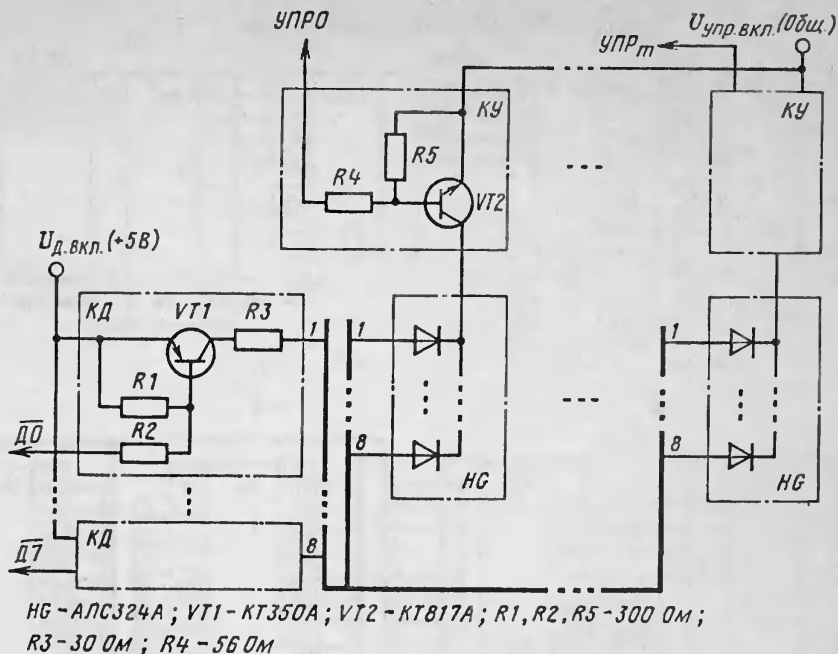


Рис. 20. Подключение полупроводниковых индикаторов с общим катодом

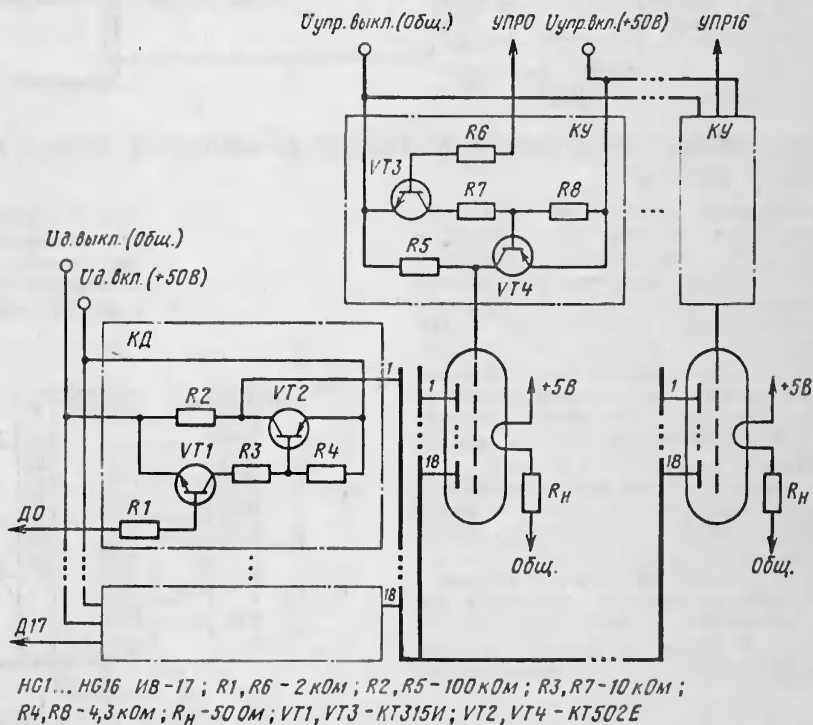


Рис. 21. Подключение вакуумных люминесцентных индикаторов

ния и возникающий при этом значительный бросок тока. Для устранения этих эффектов нити накаливания сегментов иногда держат разогретыми (но не светящимися), пропуская через них небольшой ток. Для газоразрядных ЗСИ напряжение поджига разряда обычно превышает напряжение его поддержа-

ния и подается лишь на короткий момент в начале сканирования каждого знакоместа. Для этого используется дополнительный ключ, формирующий импульс повышенного напряжения необходимой длительности на шине питания КУ под управлением одновибратора. Запуск одновибратора необходимо

осуществлять по фронту сигнала ВД БИС.

Применение модуля интерфейса клавиатуры и дисплея УМПК-80/МИ6 совместно с учебной микроЭВМ УМПК-80/ВМ (или модулей УМПК-48/ВМ, УМПК-86/ВМ) позволяет не только практически изучать особенности функционирования БИС КР580ВВ79 и интерфейсов на ее основе, но и исследовать рабочие характеристики вновь разрабатываемых клавиатур, дисплеев, схем сопряжения их с БИС и программ драйверов, а также значительно сократить срок создания УВО микроконтроллеров.

Телефон для справок: 532-99-75, Москва

#### ЛИТЕРАТУРА

1. Горбунов В. Л., Панфилов Д. И. Микропроцессоры. Книга 8. Лабораторный практикум.— М.: Высшая школа, 1984, с. 51—62.
2. Кушнир В. Е., Панфилов Д. И., Шаронин С. Г. Учебная микроЭВМ на основе однокристалльной ЭВМ КМ1816ВЕ48 // Микропроцессорные средства и системы.— 1986.— № 6.— С. 75—81.
3. Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г., Шаронин Ю. Г. Учебная микроЭВМ на основе микропроцессора КМ1816ВМ86 // Микропроцессорные средства и системы.— 1986.— № 5.— С. 16—22.

4. Преснухин Л. Н., Панфилов Д. И., Романенко О. А., Шаронин С. Г. Лаборатория по изучению микропроцессорных комплектов с фиксированным набором команд // Микропроцессорные средства и системы.— 1985.— № 1.— С. 77—81.
5. Самофалов К. Г., Викторов О. В., Кузнецов А. К. Микропроцессоры.— Киев, Техника, 1986.
6. Вуколов Н. И., Михайлов А. Н. Знакоинтегрирующие индикаторы. Справочник. М.: Радио и связь. 1986.

Статья поступила 16.03.88

#### РЕФЕРАТЫ СТАТЕЙ

УДК 621.3.049

Стоянов А. И., Хорошун В. С., Ашканов В. Н. Двухкристалльный микропроцессор серии К1831. Микропроцессорные средства и системы.— 1989.— № 3.— С. 2.

Описан двухкристалльный 16-разрядный микропроцессор серии К1831, обеспечивающий построение современных высокопроизводительных устройств, в том числе микроЭВМ и перспективных ЭВМ.

УДК 681.3.06

Ненашев М. А., Петрыкин Ю. С. Системы управления базами данных для персональных ЭВМ // Микропроцессорные средства и системы.— 1989.— № 3.— С. 7.

Предлагаемая работа является обзором наиболее известных и используемых СУБД для ПЭВМ IBM PC и совместимых с ней моделей. В работе приводятся основные характеристики СУБД, рекомендации по их применению для различных задач.

УДК 681.3.06

Христов П. В. Архитектура ЭСКОРТ-машины и ее аппаратная поддержка // Микропроцессорные средства и системы.— 1989.— № 3.— С. 13.

Рассматривается архитектура виртуальной машины инструментальной среды программирования ЭСКОРТ. Сформулированы требования к аппаратной поддержке интерпретатора ЭСКОРТ-программ.

УДК 681.3.06

Бельчинский И. Л. Перенос ядра ИНМОС между машинами различных архитектур // Микропроцессорные средства и системы.— 1989.— № 3.— С. 17.

Рассматривается реализация переноса ОС ИНМОС с машин типа СМ4 на ПЭВМ совместимых с IBM PC, приводятся возникшие при переносе проблемы и их решения.

УДК 681.3

Гейман Л. М. Этапы развития информатики как системы знаний // Микропроцессорные средства и системы.— 1989.— № 3.— С. 31.

Подробно рассмотрены этапы истории информатики в плане развития информационной символики и систем коммуникации. Отмечено влияние социальных и политических факторов, производственных отношений на формирование систем письменности в различных регионах Земли. Обосновано понимание информатики как неотъемлемой части естествознания и техники.

UDC 621.3.049

Stoyanov A. I., Khoroshunov V. S., Ashkanov V. N. Two-chip microprocessor type K1831. // Microprocessor devices and systems.— 1989.— N 3 — P. 2.

The two-chip 16-bit microprocessor type K1831 designed for application in high-performance microprocessor devices as well as in microcomputers is described.

UDC 681.3.06

Nenashev M. A., Petrykin Yu. S. Database support systems for personal computers. // Microprocessor devices and systems.— 1989.— N. 3 — P. 7.

The general description of some Database systems for IBM PC/XT microcomputer is given.

UDC 681.3.06

Khristov P. V. The architecture of ESCORT-machine and its hardware support. // Microprocessor devices and systems.— 1989.— N. 3 — P. 13.

The architecture of virtual machine in instrumental software environment ESCORT is shown. The specifications for hardware support of ESCORT code interpreter are formulated.

UDC 681.3.06

Belchinsky I. L. The INMOS kernel portability between computers of different architecture. // Microprocessor devices and systems.— 1989.— N. 3 — P. 17.

The process of UNIX-like INMOS operating system kernel transfer from PDP-11-compatible computer to IBM PC is described. Some problems which appeared are discussed and their solutions are explained.

UDC 681.3

Geyman L. M. The primary stages of informatics evolution as a knowledge system. // Microprocessor devices and systems.— 1989.— N 3 — P. 31.

The thorough examination of informatics evolution in the fields of information coding and communication system is made. The influence of social, political factors and relations of production onto language notation system in various Earth regions is outlined. The meaning of informatics as the integral part of science and technology is explained.

УДК 681.322.1

Гиглавы А. В., Немец А. В. Освоение технологии САПР БИС в КНР // Микропроцессорные средства и системы.— 1989.— № 3.— С. 34.

Приведен обзор процесса создания в КНР в 1985—1988 гг. совместными усилиями академических институтов, вузов, электронных заводов и НИИ передовых САПР БИС — основы для самостоятельного развития микроэлектроники.

УДК 681.3.06—181.4

Исаев А. Н., Блинные В. И., Кривошеев М. И., Сарьян В. К., Красносельский И. Н., Боловинцев Ю. М. Передача патентной информации в составе телевизионного сигнала // Микропроцессорные средства и системы.— 1989.— № 3.— С. 38.

Рассматривается актуальная задача циркулярной передачи патентной информации заинтересованным организациям с помощью разветвленной ТВ сети.

Технические решения, предлагаемые в статье, основаны на уже выполненных проработках и экспериментальных исследованиях и могут быть реализованы.

УДК 621.9.06

Григорьев Л. А., Прокопчик А. В., Разживин Б. А. Технологическая подготовка производства с помощью информационно-поисковой системы «Деталь» // Микропроцессорные средства и системы.— 1989.— № 3.— С. 41.

Рассмотрена ИПС на основе ПЭВМ «Искра 226», с помощью которой можно организовать работу по унификации деталей и их элементов, получить оценку параметров производства, повысить качество технологических разработок.

УДК 681.03.06.

Казанцев А. П., Майоров Л. Н., Данилов А. Б. Указатель информации и интерфейс цветного телевизора для микроЭВМ «Электроника БК-0010» // Микропроцессорные средства и системы.— 1989.— № 3.— С. 54.

Рассмотрены схемы, конструкция и обслуживающие программы указателя графической информации для ПЭВМ.

УДК 681.3.06

Пройдаков Э. М. Введение в мониторы микроЭВМ // Микропроцессорные средства и системы.— 1989.— № 3.— С. 78.

Приводится терминология, рассматривается структура монитора, методология его создания, а также некоторые особенности техники программирования мониторов.

UDC 681.322.1

Giglavay A. V., Nemetz A. V. The LSI CAD application in China. // Microprocessor devices and systems.— 1989.— N. 3.— P. 34.

The survey shows the process of LSI CAD development in China in 1985—88 as the result of joint effort of academic institutes, high school, research centers and electronics plants. These progressive CAD systems constitute the base for independent microelectronics development in China.

UDC 681.3.06—181.4

Isayev A. N., Blinnikov V. I., Krivosheev M. I., Saryan V. K., Krasnoselsky I. N., Korytov V. V., Bolovintsev Yu. M. Patent information transmission in composite TV signal. // Microprocessor devices and systems.— 1989.— N. 3 — P. 38.

The actual task of patent information broadcasting to specialized patent services using conventional TV net is discussed. The technical solutions proposed in the article are based on already performed experiments and may be used in practice.

UDC 621.9.06

Grigoriev L. A., Prokopchik A. V., Razhivin B. A. Technologic CAD using "PART" database management system. // Microprocessor devices and systems.— 1989.— N. 3 — P. 41.

The database management system running on "ISKRA 226" microcomputer which helps to unify assembly parts and their elements is described. The system performs manufacturing parameters estimation and enhances quality of technologic design.

UDC 881.03.06

Kazantsev A. P., Mayorov L. N., Danilov A. B. Graphic manipulator and colour TV interface for "Elektronika BK-0010" home computer. // Microprocessor devices and systems.— 1989.— N. 3 — P. 54.

The circuit diagram, construction and software handlers for graphic manipulator for BK-0010 computer are given.

UDC 681.3.06

Proydaikov E. M. The introduction to Monitor programs for microcomputers. // Microprocessor devices and systems.— 1989.— N. 3 — P. 78.

The glossary, structure of monitor program and its programming technique is in the scope of the article. Some peculiarities of monitors are discussed.

## Заместитель главного редактора С. М. Пеленов

Номер подготовили:  
Е. И. Бабич, Г. Г. Глушкова  
В. М. Ларионова  
Корректор Е. М. Кучерявенко  
Техн. редактор Г. И. Колосова

Адрес редакции журнала:  
103005, Москва, Малый  
Сухаревский пер., д. 9-а  
Тел.: 208-73-23; 208-19-94

Сдано в набор 24.04.89. Т — 09997  
Подписано к печати 06.06.89  
Формат 84×108<sup>1</sup>/<sub>16</sub>  
Офсетная печать  
Усл. печ. л. 10,08. Уч.-изд. л. 14,6  
Тираж 108750 экз.  
Заказ № 5642  
Цена 1 р. 10 к.

Орган Государственного  
комитета СССР  
по вычислительной технике и  
информатике

Набрано в ордена Трудового Красного  
Знамени Чеховском полиграфическом  
комбинате Государственного  
комитета СССР по печати.  
142300, г. Чехов, Московской обл.

Отпечатано в Московской  
типографии № 13 ПО «Периодика»  
Государственного комитета СССР  
по печати.  
107005, г. Москва, Денисовский пер., 30.  
Заказ 144.

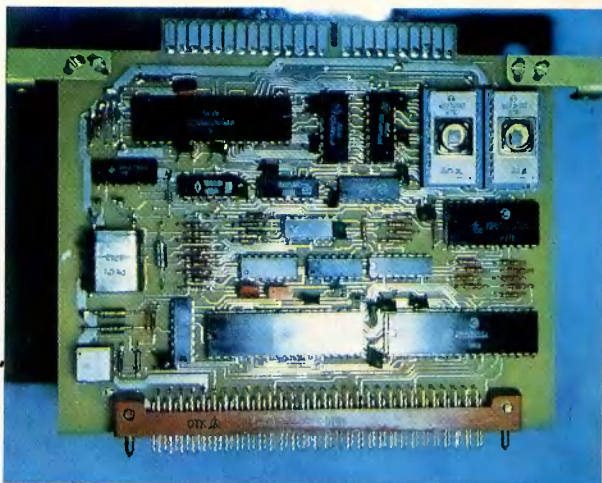
## ТЕХНОЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ МТК-40 и МТК-35

Контроллер МТК-40 — создан на базе микропроцессора КР580ВМ80А и предназначен для работы с технологическим оборудованием, контрольно-измерительными и испытательными приборами и другими системами обработки цифровой и аналоговой информации. Основное устройство контроллера МТК-40 — процессор, связанный с остальными подчиненными по отношению к нему функциональными элементами внутренней шины. Каждому блоку, по принципу «Общей шины», отводится своя зона в адресном пространстве процессора и обращение к ним с его стороны осуществляется по командам обращения к памяти.

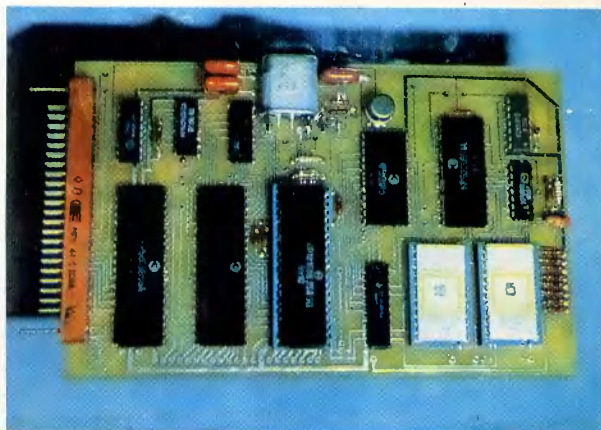
Контроллер МТК-40 использует внешние источники питания постоянного тока с напряжениями +5, +15, —15 В.

Контроллер МТК-35 (тоже на базе микропроцессора КР580ВМ80А) имеет встроенный таймер и последовательный интерфейс для связи с машиной более высокого уровня.

Для МТК-35 разработаны блок питания и плата клавиатуры и индикации.  
Телефон для заказов 468-81-75, Москва.



МТК-40



МТК-35

### Технико-технологические характеристики МТК-40

Тактовая частота, МГц . . . . .	2
Объем памяти, Кбайт	
ОЗУ . . . . .	1
ПЗУ . . . . .	4
Число двунаправленных аналоговых каналов ввода-вывода . . . . .	8
Число каналов таймера, доступных пользователю . . . . .	3
Число двунаправленных каналов дискретного ввода-вывода, по 3 байта	24
Число последовательных каналов	1
Разрядность ЦАП и АЦП, бит . . . . .	10
Время преобразования, мкс	
АЦП . . . . .	200
ЦАП . . . . .	5
Наработка, ч, не менее	
на сбой . . . . .	1 000
на отказ . . . . .	10 000
Срок службы, лет, не менее . . . . .	7
Потребляемый ток, А, не более	
от источника +5В . . . . .	1
от источника +15В . . . . .	0,2
от источника —15В . . . . .	0,1
Габаритные размеры, мм . . . . .	155×130×15
Масса, кг . . . . .	0,3

### Основные технические характеристики МТК-35

Объем памяти,	
ОЗУ, байт . . . . .	64
ПЗУ, Кбайт . . . . .	4
Число аналоговых входов . . . . .	8
Число дискретных входов-выходов . . . . .	24
Разрядность аналого-цифрового преобразования, бит . . . . .	10
Время срабатывания АЦП, мкс . . . . .	800

# ЮНИКОМ

## Совместный советско-австрийский коммерческий научно-производственный центр Институт атомной энергии им. И. В. Курчатова

БИМС ДВКЗ предназначен для разработки, отладки и эксплуатации прикладных программ, использующих графический ввод и отображение информации, на дисплейных вычислительных комплексах ДВКЗ.

Как любой интерпретатор, БИМС прост в освоении и удобен при разработке и отладке программ. Ошибки в программе могут вызвать диагностические сообщения, но никогда не нарушают работоспособность системы. Язык БИМС внешне похож на БЕЙСИК.

БИМС является открытой системой. При необходимости язык БИМС можно расширить, дополнив его новыми операторами и функциями для проведения специальных вычислений или для управления нестандартными внешними устройствами. Расширение БИМС проводится средствами Макроассемблера. Библиотека макроопределений и все необходимые файлы для генерации новых версий БИМС входят в комплект поставки.

**Операционная среда:**  
RT-11 (ОС ДВК).

**Документация:**

Описание языка — 89 с.

Руководство системного программиста — 22 с.

Руководство оператора — 33 с.

**Файлы:**

Предусмотрены чтение, запись и модификация текстовых и двоичных файлов в режимах как последовательного, так и произвольного доступа.

**Геометрия:**

БИМС ДВКЗ способен работать с двумерными геометрическими объектами: точками, окружностями, отрезками прямых и дугами. Любой из этих объектов хранится в памяти как одна переменная или один элемент массива. Над геометрическими элементами можно выполнять операции: сдвиг, масштабирование, поворот, выделение части, составление из частей, вывод на экран.

**Вас не удовлетворяет объем ОЗУ или быстроедействие дисковой памяти ДВКЗ? Делу поможет компактное кодирование и рациональная организация процесса программирования. Эти и другие возможности Вам предоставит**

**ГРАФИЧЕСКАЯ ИНТЕРПРЕТИРУЮЩАЯ СИСТЕМА**

### БИМС

**Именные выражения:**

Вместо явного указания имени любого объекта (переменной, массива, программного модуля) в БИМСе можно использовать специальную языковую конструкцию — именованное выражение. Это обычное строковое выражение, значение которого при работе вычисляется и интерпретируется как имя. Например, в строковой переменной может храниться имя другой переменной. Именованное выражение — мощное программное средство, позволяющее работать с развитыми структурами данных.

**Растровая графика:**

Предусмотрен ряд возможностей для обработки растровых фрагментов — прямоугольных областей экрана, содержащих произвольное изображение. Растровые фрагменты можно перемещать, сохранять в массивах и файлах, производить различные логические операции, составлять из частей и выделять части. Растровая графика БИМС ДВКЗ — быстроедействующая, она ориентирована на

методы графического диалога с использованием мультиспликации.

**Списки:**

Над строковыми массивами БИМС можно производить операции вставки и удаления элементов. Это дает возможность организовать списки, например, типа «стек» и «очередь».

**Параллельные процессы:**

БИМС допускает параллельное (одновременное) выполнение нескольких программ по алгоритмам, заданных в различных или в одном и том же программном модуле. Предусмотрена синхронизация процес-

сов: каждый процесс может объявить событие или «зависнуть» в ожидании события, которое может быть объявлено другим процессом или программой обработки прерывания.

**Модульная структура программ:**

Программа на языке БИМС состоит из одного или нескольких программных модулей. Каждый модуль имеет имя. Модуль может быть выполнен как подпрограмма. Модуль может быть загружен с диска в ОЗУ или выгружен — тогда занимаемая им память освобождается.

**Оверлейные программы:**

В оперативной памяти программный модуль хранится в скомпилированном виде. На диске он может быть как скомпилированным, так и в виде текста. Скомпилированный модуль примерно вдвое короче своего текстового представления. Он может быть загружен для выполнения в оперативную память как оверлейная программа. Поскольку такая загрузка не требует синтаксического анализа, она производится быстро — около 0,5 с. (для накопителей типа НГМД 6022).

**Программная обработка программ:**

БИМС имеет операторы и функции, позволяющие обрабатывать программные модули и получать любые их фрагменты в виде строк текста. Нет никакой проблемы в том, чтобы написать на БИМСе программу, которая распечатывает собственный листинг, или программу-редактор, модифицирующую модули непосредственно в оперативной памяти.

БИМС не содержит отдельного ядра, организующего диалог системы с оператором. Эта функция выполняется обычным модулем БИМС, который Вы можете изменить или переписать по своему вкусу.

**Компилятор:**

Обычно преобразование текста программ в вид, необходимый для выполнения, производится интерпретирующим ядром БИМС. Но в комплект поставки входит отдельная задача — компилятор, который может перерабатывать текстовые файлы программ на языке БИМС в готовую задачу операционной системы.

**СП «ЮНИКОМ» МОЖЕТ АДАПТИРОВАТЬ ПО ДОГОВОРЕННОСТИ МЕТОДОЛОГИЮ БИМС ДЛЯ ЭФФЕКТИВНОЙ РАЗРАБОТКИ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ IBM PC И ПРОФЕССИОНАЛЬНЫХ ЭВМ  
НАШ АДРЕС:**

**СОВМЕСТНОЕ СОВЕТСКО-АВСТРИЙСКОЕ  
ПРЕДПРИЯТИЕ «ЮНИКОМ»**

СССР, Москва, 123182, Максимова, 6. Тел. 196-47-58, 196-94-05  
Телекс 417618, р. сч. 60800991 в ВЭБ СССР.