

**ММП**

# МИКРО- ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

**4 | 1988**

ISSN 0233-4844

**Многофункциональный комплекс аппаратно-программных средств для разработки микроконтроллеров на основе семейства однокристалльных ЭВМ серии К1816**  
**Система автоматизированного проектирования микропроцессорных устройств, организованная на ПЭВМ типа ДВК, программно настраивается на различные типы секционированных микропроцессоров и учитывает конфигурацию проектируемых устройств**  
**Совместимость ПЭВМ различных типов с помощью аппаратно-программных сопроцессоров: анализ архитектур сопроцессоров, рекомендации по выбору и применению, описание сопроцессора БА 86М**  
**Пакет программ Паскаль/МТ+** обеспечивает разработку эффективных, структурированных программ для встроенных МП контроллеров  
**Графический адаптер для организации рабочей станции на базе ПЭВМ ЕС 1840:** технические характеристики, структурная схема



# СИСТЕМА ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА БАЗЕ МИКРОЭВМ, СОВМЕСТИМЫХ С IBM PC

На геологическом факультете МГУ создана система цифровой обработки изображений на базе микроЭВМ, совместимых с IBM PC. Один из вариантов системы показан на первой странице обложки. Аппаратная часть содержит несложный интерфейс ввода изображений с телекамеры или с электронного микроскопа. Программная часть системы позволяет эффективно проводить коррекцию, преобразование и анализ изображений, одномерное и двумерное преобразование Фурье, обработку стереопар и цветных изображений, реконструкцию трехмерной структуры по проекциям (томография). В качестве примера на второй странице обложки приведены изображения с экрана микроЭВМ при анализе морфологии шлифа горной породы: коррекция

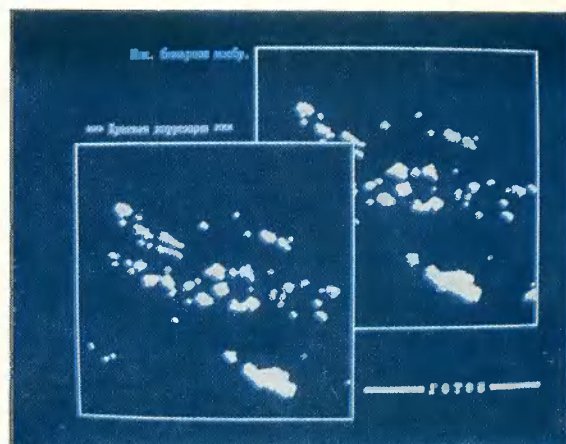
неравномерности освещения объекта (а), выделение интересующего минерала (б), удаление искаженных границей кадра частиц (в) и анализ морфологии выбранного минерала (г).

Основные области применения этой системы цифровой обработки изображений — лабораторные и научные исследования в материаловедении, биологии, медицине, геологии, физике, химии, при дефектоскопии и анализе отказов изделий микроэлектроники и т. д. Подробная информация о системе и примеры ее использования будут опубликованы в статье А. Ю. Сасова в следующем номере журнала.

**Справки по телефону: 939-35-87, Москва, Сасов А. Ю.**



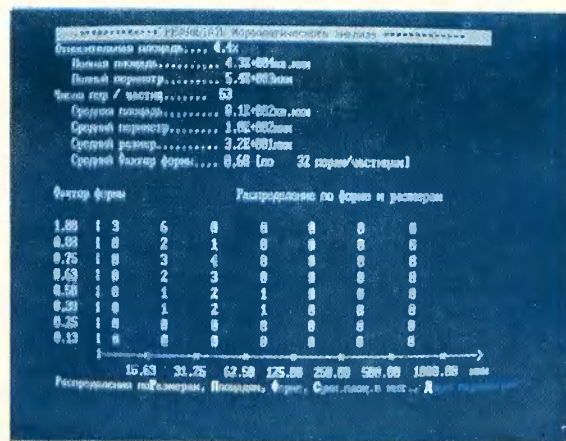
а



б



в



г

ОРГАН  
ГОСУДАРСТВЕННОГО  
КОМИТЕТА СССР  
ПО ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКЕ  
И ИНФОРМАТИКЕ

Издается с 1984 года

# МГ МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД

НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 4 / 1988 МОСКВА

## СОДЕРЖАНИЕ МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Периферийное оборудование  
микроЭВМ и  
микропроцессоров

## ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ Машинная графика

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Операционные системы  
микроЭВМ  
Обмен опытом

Языковые средства  
микроЭВМ

## ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ САПР

Измерительные системы

## УЧЕБНЫЙ ЦЕНТР

Справочная информация

Ершов А. П. — Колонка редактора	2
Кушнир В. Е., Панфилов Д. И., Шаронин С. Г. — Многофункциональный комплекс программно-аппаратных средств для семейства однокристальных ЭВМ серии К1816	3
Нунупаров Г. М. — Сопряжение микроЭВМ К1827ВЕ1 с внешним ППЗУ	6
Шубин С. Ф. — Сопряжение дисплея СМ 7204 с микроЭВМ «Электроника 60»	8
Динес В. Л., Ардашев Н. Н., Мартюхин В. А. — Устройство сопряжения микропроцессорной системы КТС ЛИУС-2 с дисплеем РИН-609	12
Грачев А. Н., Ильин Е. П., Хухлаев Е. В. — Использование дисплея «Графит» в качестве удаленного графического терминала ИМС КАПРИ	14
Иванов Е. А. — Совместимость ПЭВМ с помощью аппаратно-программных сопроцессоров	17
Кулаичев А. П. — Графический БЕЙСИК для ДВК-3	21
Гехман Б. И. — Диалоговая система на микроЭВМ «Электроника ДЗ-28» для подготовки графических иллюстраций	23
Злотник Е. М., Киркоров С. И., Стежко И. К. — Графический адаптер для ППЭВМ ЕС 1840	25
Големинов Н. Г., Званцев А. А. — Регенеративный графический дисплей	27
Долгушев В. Г., Коваленко В. А., Левшин А. В. — Автоматизация чертежно-графических работ на микроЭВМ «Электроника 60»	28
Глухов В. Н., Долбилов Л. А., Дудников Е. Е., Лоозе Й. — МикроDOS — адаптивная система программного обеспечения для 8-разрядных микроЭВМ	33
Анисимов А. А., Анисимов Г. А. — Режим разделения времени в системах с однозадачным монитором	37
Корнеев В. В., Тарков М. С. — Операционная система микромашиной вычислительной сети с программируемой структурой МИКРОС	41
Гайворонский М. А. — Простой способ адаптации развитых программных средств для работы со стандартным дисплеем микросистемы	44
Шилин А. Е. — Программный эмулятор команд расширенной арифметики	47
Брылев С. П. — Программирование встроенных микропроцессоров с применением компилятора Паскаль / МТ+	50
Вершигора Р. В. — Сопрограммная связь функций в языке Си	53
Филин А. Л. — Организация программ динамической структуры при разработке интегрированных систем с использованием языка высокого уровня	55
Гуржуенко И. В., Липидус В. Ю., Летник Л. А. — Кроссассемблер для МП КР580 на БЕЙСИКЕ	58
Кизуб А. В. — РИСК — реально используемая сетка конструктора	61
Баталов Б. В., Русаков С. Г., Савин В. В. — Пакет прикладных программ автоматизации схмотехнического проектирования для персональных компьютеров	63
Семенов О. И., Бокуть В. В., Гриншпан Л. А., Малюш Я. Т., Ткачев Е. Е., Шерлин Д. Р. — Система автоматизированного проектирования мультимикропроцессорных устройств на секционированных микропроцессорах	66
Горшков Б. Л. — Особенности построения измерительных трактов для систем автоматизации на основе ЭВМ семейства «Электроника 60»	70
Глуценко К. А., Дудыкевич В. Б., Котыло О. Б. — Реализация функции измерения частоты в микропроцессорных системах	73
Хромов А. В., Покровский В. В. — Восьмиканальный микропроцессорный ультразвуковой дальномер	75
Фроштейн Р. А. — Программно-аппаратные средства расширения возможностей микроЭВМ «Электроника 60» и ДВК	77
Гершкович С. К., Крынкин В. В., Востропятюв А. Ю. — Автоматизированный учебно-лабораторный комплекс	82
Журавский Д. А., Барков С. Н. — Организация управления расширенной памятью для процессора КР580ИК80А	87
Интерфейсные БИС микропроцессорного комплекта К1801	89

Главный редактор  
А. П. ЕРШОВ

Редакционная  
коллегия:

А. Г. Алексенко  
В. М. Брябри  
А. А. Васенков  
(зам. главного редактора)  
И. В. Вельбицкий  
А. Б. Венгеров  
Г. Р. Громов  
(ответственный секретарь)  
В. П. Иванников  
М. Б. Игнатъев  
А. В. Каляев  
И. З. Карась  
В. П. Куприянов  
С. С. Лавров  
В. В. Липаев  
К. А. Меликян, И. А. Мизин  
С. М. Пеленов  
(зам. главного редактора)  
А. К. Платонов  
А. А. Попов  
Д. А. Поспелов  
Б. И. Рамеев  
О. Л. Смирнов  
А. А. Стогний  
М. К. Сулим  
Н. М. Шарунико

Редакционный совет:

И. В. Бабынин  
С. Н. Бушев  
Е. П. Велихов  
Н. Н. Говорун  
В. В. Корчагин  
В. П. Макаревич  
Ю. Е. Нестерихин  
А. Р. Назарьян  
А. Л. Нефедкин  
И. В. Прангшвили  
Л. Н. Преснухин  
В. В. Пржнялковский  
Н. Л. Прехоров, Г. Г. Рябов  
В. И. Хохлов  
Н. И. Шереметьевский  
В. В. Шильдин, А. В. Яковлев  
Э. А. Якубайтис

Номер подготовили:

Е. И. Бабич, Г. Г. Глушкова,  
В. М. Ларионова, С. С. Матвеев  
Корректор Е. М. Кучерявенко  
Технический редактор  
Г. И. Колосова

Адрес редакции журнала:

103051, Москва, Малый  
Сухаревский пер., д. 9А  
Телефоны: 208-73-23, 208-19-94  
Сдано в набор 30.05.88.

Подписано к печати 09.08.88.  
Формат 84x108<sup>1</sup>/<sub>16</sub>. Т-14675.

Высокая печать.

Усл. печ. л. 10.08

Уч.-изд. л. 14,6. Тираж 100 500

Заказ 132. Цена 1 р. 10 к.

Орган Государственного  
комитета СССР  
по вычислительной технике  
и информатике  
Московская типография № 13  
ПО «Периодика» ВО «Союзпо-  
лиграфпром» Госкомиздата  
СССР

107005, Москва, Денисовский  
пер., дом 30

## НАДЕЖНОСТЬ, НАДЕЖНОСТЬ И ЕЩЕ РАЗ НАДЕЖНОСТЬ

Вспомню годичное собрание Отделения информатики и вычислительной техники АН СССР. Один членкор, директор, напрямую спрашивает другого членкора, министра, почему не работает его ПЭВМ. Министр — опытный полемист, и он дает директору весьма достойный ответ такого рода, что и ПЭВМ надо употреблять умеючи, а, кроме того, он лично сможет оказать коллеге по Отделению необходимой содействие.

Министр слегка слукавил. Даже на уровне директорских кабинетов проблема работающих ПЭВМ решена далеко не на сто процентов. А уж рассчитывать на то, что такой ответ удовлетворит один миллион сто тысяч пользователей, более чем не приходится. Действительность, тем временем, такова, что при распаковке до сих пор направляемых в школу комплектов КУВТ-86, хорошо если удастся запустить хотя бы две трети машин, а практически любой отечественный дисконд деградирует без регулировки или ремонта в течение первого квартала работы.

Пока что проблема надежности массовой вычислительной техники «решается» путем строительства заборов. Производящие отрасли возводят баррикады входного контроля, ставят соломку заводом-изготовителем, умело торгуются при согласовании ГЗ и актов всевозможных приемок. Государство десантирует на предприятие отряды госприемки, сочиняет грозные стандарты, технические условия и налагает штрафы. Мы же, привычно критикуя систему и в то же время уповая на нее, по-индустриально-пожизненно помещая себя во «всесдаправое» положение пользователя, возлагаем реальные надежды на импортированные комплектующие и технологическое оборудование.

Мы, однако, не решим проблему, если ограничимся обречением нескладной плоти нашей бесхозяйственности и некомпетентности в золоченые одежды валютных ассигнований. Великий Менделеев сказал еще сто лет назад: «Технологии бывают только отечественные!». Проблемы надежности нельзя решить на границах производственного процесса, а можно только внутри него. Этот закон качества работы, осознанный еще в средневековых цехах ремесленников и мастеровых, получает очередное возрождение и перевоплощение в кружках качества, бригадном подряде, гибком автоматизированном производстве и других формах современной организации творческого труда.

Но это означает, что ни читателям, ни авторам нашего журнала никуда не уйти от понимания того, что надежность вычислительной техники находится в их руках и адресовать эту проблему никому.

Нам нужно тщательно перебрать работой рук и мысли все стадии проектирования и производственного процесса — от добычи сырья до изготовления упаковочной тары. Примат сборки над изготовлением, организационная, экономическая и интеллектуальная расчлененность процесса создания ЭВМ, слабость технологической инфраструктуры, хроническая нехватка контрольно-измерительной аппаратуры — все это приводит к тому, что энтропия находит бесчисленные лазейки на пути появления компьютера, сводя на нет радужные надежды конструкторов и ожидания потребителей. Меньшее внимание должно быть уделено человеческим процедурам, с тем чтобы каждый участник производственного процесса достоверно знал, как он должен действовать, чтобы не потерять качества, понимая последствия своих ошибок, фактически умел реализовать свою долю участия в работе. На всех сопряжениях и каскадах дискретной структуры собираемого устройства должны быть точно определены пределы рабочих точек всех физических процессов, реализующих функционирование устройства в соответствии с его абстрактной логической структурой.

Все это означает также, что изучение и обсуждение вопросов надежности должны стать темой номер один в вузовском образовании, на курсах повышения квалификации, на научных собраниях и в технической литературе. Тем не менее у нас показатели надежности электронно-вычислительной аппаратуры — это тайна за семью печатями.

Замалчивание проблем надежности с отечественной вычислительной техникой — такой же пример казенного лицемерия, как, скажем, бытовавшее у нас до недавних пор умалчивание проблем межнациональных отношений.

Если мы не навалимся всем миром и не займемся вплотную, тотально и гласно научно-техническими и организационно-экономическими вопросами надежности массовой вычислительной техники, единственным реальным итогом нашей широкой программы компьютеризации будут свехприбыли ремонтной службы ГКВТИ СССР.

А. П. ЕРШОВ

УДК 681.322.1+681.325.5

В. Е. Кушнир, Д. И. Панфилов, С. Г. Шаронин

## МНОГОФУНКЦИОНАЛЬНЫЙ КОМПЛЕКС ПРОГРАММНО-АППАРАТНЫХ СРЕДСТВ ДЛЯ СЕМЕЙСТВА ОДНОКРИСТАЛЬНЫХ ЭВМ СЕРИИ К1816

Специфика отладочных средств (необходимых для разработки микроконтроллеров на основе ОЭВМ) — применение либо универсальных микроЭВМ, выполняющих функции кросс-системы разработки программ, загрузчика и трассировщика (эмулирующие ОЭВМ [1] или использующих ее в составе отладочной модели [2]), либо специальных устройств, аппаратно реализующих лишь загрузку программ в отладочную модель и их трассировку [3]. Наиболее существенным недостатком первой группы устройств является ориентация на сложную вычислительную технику, требующую специальной подготовки разработчика и обслуживающего персонала; второй группы — сложность исследования содержимого внутреннего ОЗУ и регистров ОЭВМ (затрудняет, а при автономном использовании ОЭВМ делает невозможной трассировку программ).

Комплекс персональных программно-аппаратных средств экономичен, прост и доступен, модульный принцип построения позволяет перестраивать и наращивать его в зависимости от объема и сложности решаемых задач (обеспечивая поэтапный процесс изучения, разработки и отладки микропроцессорных систем).

Основу аппаратных средств одного из таких комплексов составляют базовый модуль микроЭВМ УМПК-48/ВМ и расширительные модули УМПК-48/МР1 и УМПК-48/МР2 [4]. Анализ возможностей для изучения особенностей построения комплексов на основе ОЭВМ и моделирования работы микроконтроллеров различного назначения вызвал большой интерес читателей, разработчиков микропроцессорных систем управления различными объектами. Определяющими мотивами для них по использованию комплекса в своих разработках явились:

- простота, дешевизна и мобильность;
- возможность изготовления в условиях собственного производства;
- универсальность, позволяющая решать задачи обучения, исследования программно-аппаратного обеспечения ОЭВМ, моделирования и отладки собственных систем;
- удобство изменения конфигурации системы с помощью модульного принципа построения комплекса.

Рассмотрим программное обеспечение в объеме, необходимом для понимания работы комплекса УМПК-48, воспроизводства и активного использования.

В зависимости от состава оборудования программные средства комплекса можно разделить на две группы. В первую группу входят резидентный монитор DM48 и тестовая программа TST48 совместно с модулями УМПК-48/ВМ и УМПК-48/МР1. Учебно-отладочная микроЭВМ (рис. 1, а) работает под управлением монитора DM48 и позволяет производить исследование процесса выполнения и отладку программ в машинных кодах. Вторую группу программных средств комплекса составляют кроссасемблер ASMK48 и кроссасемблер RASMK48 (рис. 1, б). Отличительные характеристики программ — возможность адаптивирования практически к любым персональным микроЭВМ на базе МП БИС КР580ВМ80 и выполнение резидентной работы,

Резидентный отладочный монитор DM48 осуществляет ввод, редактирование и отладку программ в машинных кодах при работе микроЭВМ в составе базового и расширительного модулей УМПК-48/МР1. Монитор (рис. 2) записан в ПЗУ ([4], с. 77, рис. 3, D2). В качестве системных устройств ввода-вывода используется дисплей и клавиатура базового модуля. После начальной установки микроЭВМ производится тестирование ОЗУ модуля УМПК-48/МР1 и в случае обнаружения ошибки записи-считывания выдается звуковой сигнал и сообщение 1 (рис. 3). При успешном завершении теста на дисплей выводится сообщение 2.

В процессе работы монитора выполняются восемь директив, задаваемых клавишами выбора режимов работы (см. [4], с. 77, рис. 5); соответствие их названий и обозначений по принципиальной схеме базового модуля ([4], с. 75, рис. 1) указано в описании директив. Для ввода цифр 0...F используются клавиши S17...S32.

Директива адреса памяти команд (АПК, S36) позволяет ввести адрес памяти команд пользователя. При ее вызове на дисплей выдается сообщение 3. После ввода трех цифр адреса монитор переходит к режиму отображения содержимого памяти. Поскольку для программ пользователя доступен только первый банк памяти команд, то при наборе адреса в пределах нулевого банка монитор выдает звуковой сигнал ошибки и переходит к повторному выполнению директивы. После набора корректного адреса памяти команд на дисплей выводится содержимое памяти и возможен последовательный просмотр адресов в сторону их увеличения и уменьшения, а также изменение содержимого выбранного адреса.

Запись и увеличение (ЗпУв, S39) применяются для просмотра ячеек памяти команд, памяти данных или регистров в сторону увеличения адресов или номеров регистров. Для ввода нового значения ячейки памяти или регистра необходимо набрать его на циф-

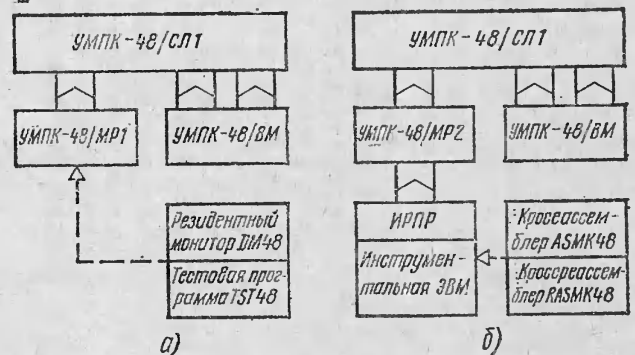


Рис. 1. Состав комплекса:

а — при разработке программ в машинных кодах;  
б — с помощью кросс-средств

БЛОК N 0      ФАЙЛ A:DM48.DMP

0000	04 09 00 84	33 00 00 00	00 26 2A 23	00 A9 F9 3A
0010	83 00 34 24	F9 03 10 A9	03 80 96 0E	23 9F 3A 88
0020	FD 34 24 23	8F 3A 83 FD	34 24 23 70	3A 23 08 88
0030	F1 90 C8 27	90 14 90 B4	9C 23 42 14	B6 14 A0 04
0040	3D 14 90 23	48 14 06 B4	7D 23 3C 10	86 B4 89 04
0050	43 23 70 3A	88 F0 80 AE	18 80 AA 14	C8 34 06 14
0060	AD D3 14 95	47 84 9E F8	03 16 C6 73	F8 D3 15 C6
0070	80 84 00 F8	17 AB 86 5B	FA 17 00 04	89 00 00 00
0080	FB 07 AB D3	FF 96 8E FA	07 53 0F 43	08 AA 04 5B
0090	88 FF 23 8F	3A 23 09 90	23 23 90 23	3C 14 06 93
00A0	14 E1 AB D3	11 96 A9 04	51 F8 03 10	96 20 24 38
00B0	F8 03 13 96	87 24 E4 F8	D3 12 96 BE	44 2F F8 93
00C0	FA 47 53 70	3A FB A9 81	AC 93 A9 23	8F 3A 88 FF
00D0	23 90 90 C8	F9 93 14 CA	BF 06 F9 03	90 19 EF DA
00E0	93 23 8F 3A	88 FF 23 40	9F 06 53 07	C6 E9 88 80
00F0	53 3F 93 14	CA BF 04 04	DA AF 47 34	01 FF 34 01

КОНТРОЛЬНАЯ СУММА БЛОКА N 0 : 7063

БЛОК N 5      ФАЙЛ A:DM48.DMP

0500	88 C8 B9 A1	80 91 18 19	80 91 B8 AF	80 62 C8 80
0510	D7 88 A1 80	AB F2 19 A4	4A D2 1D A4	4A B2 21 A4
0520	4A 92 25 A4	4A 72 29 A4	4A 52 2D A4	4A 0D 32 32
0530	A4 4A 12 3F	88 A2 80 A9	88 AD 80 88	FE A4 5E 03
0540	A2 80 A9 16	AD 80 88 FF	A4 5E B9 AD	81 A9 25 9F
0550	3A F9 90 23	70 3A C9 A2	81 A9 23 9F	3A 80 9A 00
0560	3A 6F 8A 00	F5 05 06 90	07 A8 53 FE	03 FE 96 73
0570	F8 A4 68 23	1F 3A 80 A9	23 70 3A F9	83 88 18 89
0580	50 27 62 55	F9 00 00 07	96 85 99 7F	F9 00 00 07
0590	76 80 69 80	16 98 A4 84	EE 84 65 93	8F C0 FF A3
06A0	A9 1F FF A3	AB 1F C6 AC	B4 61 A4 9E	93 0D 00 00
06B0	00 02 B4 93	53 3F 43 20	93 B8 18 B9	01 A4 21 00
06C0	39 18 22 06	01 02 22 08	26 08 22 08	26 08 39 06
06D0	C1 02 39 08	01 02 39 08	22 06 01 02	22 06 20 08
06E0	26 08 1C 10	FF 0C 14 CA	FA 47 34 01	23 40 80 FB
06F0	14 F9 C6 14	F9 93 00 00	00 00 00 00	00 00 00 00

КОНТРОЛЬНАЯ СУММА БЛОКА N 5 : 6ATD

БЛОК N 1      ФАЙЛ A:DM48.DMP

0100	93 53 0F E3	90 93 14 CA	FA 14 F9 FB	14 F9 FC 14
0110	F9 14 CA 27	90 93 A9 23	8F 3A 88 FF	23 94 90 F9
0120	C8 14 F9 93	23 FF 90 80	D3 FF 96 36	27 90 80 D3
0130	00 96 36 EB	24 93 04 A1	BA 00 23 70	3A 23 AD 6A
0140	A8 80 AC 8F	FA 04 01 E7	0F 53 FC 03	10 14 F3 F0
0150	34 16 14 A0	34 16 C6 68	F8 D3 15 C6	A1 78 53 F0
0160	96 52 FB AB	D3 BD 24 52	B6 6C 24 85	23 70 3A 23
0170	44 DA 96 7F	FC D3 08 53	FF C6 7F B4	7D 24 3A 23
0180	AD 6A AB FC	90 1A FA D3	0B 96 0F BA	18 24 3A 23
0190	D3 23 96 93	8F 43 2A 3A	FA D3 45 96	9F BA 00 24
01A0	3A CA FA 03	BF 96 AB BA	44 24 3A FA	D3 42 96 B4
01B0	8A 22 24 3A	FA D3 14 96	8B BA 0A 24	3A B6 C4 95
01C0	F8 AC 24 1A	FC 47 53 F0	68 AC 34 16	E6 CF 83 23
01D0	8F 3A B8 FF	23 65 90 C8	80 18 AF 23	95 90 C8 FF
01E0	43 8D 90 83	23 60 14 D6	14 A0 53 F0	96 E6 F8 14
01F0	CA 18 23 91	90 C8 F9 AA	34 01 14 A0	53 F9 96 FA

КОНТРОЛЬНАЯ СУММА БЛОКА N 1 : 84DF

БЛОК N 6      ФАЙЛ A:DM48.DMP

0600	27 AA AB AC	B4 E6 14 E1	C6 00 23 C8	62 55 16 12
0610	C4 0E FC 03	01 57 AC 96	29 FB 03 01	57 AB D3 60
0620	96 29 8B 00	FA 03 10 57	AA B4 E6 88	FF 23 40 90
0630	80 53 07 C6	0A C8 80 53	3F C6 00 C4	06 00 00 00
0640	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

КОНТРОЛЬНАЯ СУММА БЛОКА N 6 : 1AEF

БЛОК N 2      ФАЙЛ A:DM48.DMP

0200	F8 47 53 F0	AB F8 14 CA	18 23 92 90	C8 F9 34 01
0210	14 A0 53 F0	96 10 F8 C8	AB F8 14 CA	18 23 93 90
0220	C8 F9 34 01	FA 53 08 96	2D B4 7D 24	E4 04 78 23
0230	4E 14 06 14	AD D3 00 C6	5B FB 03 01	C6 53 FC D3
0240	0F 96 45 24	00 F8 03 DE	96 44 44 33	F9 03 0D 96
0250	33 44 33 C3	54 14 06 A5	85 44 60 23	5A 14 06 A5
0260	14 A0 53 F0	96 60 F8 14	CA 18 23 92	90 C8 F9 47
0270	53 F0 AA F9	34 01 14 A0	53 F0 96 76	FB 6A AA F8
0280	14 CA 18 23	93 90 C8 F9	34 01 76 93	FB 53 C0 C6
0290	95 B4 7D 44	2F 85 76 A0	23 70 3A 23	B0 6A 44 A4
02A0	23 8F 3A FA	AB 80 AC 14	CA 76 AF 23	08 44 B1 23
02B0	79 90 23 40	90 FA 14 F9	FC 14 F9 14	A0 D3 16 C6
02C0	0D F8 D3 15	C6 D1 F8 53	FD 96 88 FB	AD 34 B0 44
02D0	8B CA 76 98	23 FF DA 96	DB BA 3F 44	95 B6 E1 44
02E0	F2 76 EB 23	70 3A 23 80	6A 44 EF 23	8F 3A FA 88
02F0	FC 90 1A 76	FC 23 40 DA	96 FC BA 00	44 95 00 00

КОНТРОЛЬНАЯ СУММА БЛОКА N 2 : 80BE

ровой клавиатуре. Режим ввода индицируется запятой в младшем разряде дисплея. После нажатия клавиши ЗпУв записывается новое значение; адрес памяти или номер регистра автоматически увеличивается на единицу.

Уменьшение (Ум, S38) делает возможным просмотр содержимого регистров или ячеек памяти в сторону уменьшения их адреса и выключение режима записи.

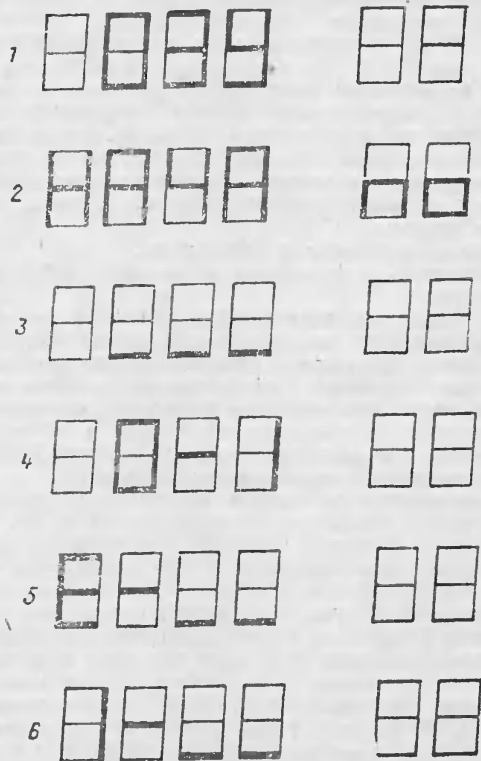
Программный счетчик (ПрСч, S34) вызывает вывод на дисплей адреса последней точки останова и переходит к режиму отображения содержимого памяти команд. После начальной установки микроЭВМ монитор присваивает программному счетчику значение, равное первому доступному адресу памяти команд (800H).

Пуск (П, S37) осуществляет запуск программы пользователя в реальном масштабе времени или в пошаговом режиме. До ввода директивы задается стартовый адрес памяти команд (выполняются АПК или

БЛОК N 3      ФАЙЛ A:DM48.DMP

0300	3F 06 5B 4F	66 6D 7D 07	7F 6F 77 7C	39 5E 79 71
0310	00 00 00 77	00 00 00 71	00 00 00 78	7C 3F 50 3F
0320	7C 3F 50 06	7C 3F 50 58	7C 3F 50 4F	7C 3F 50 66
0330	7C 3F 50 6D	7C 3F 50 7D	7C 3F 50 07	00 00 00 00
0340	00 00 76 77	66 7C 75 45	00 3F 4F 6E	00 00 00 3F
0350	40 06 00 00	79 40 08 08	00 00 06 40	08 08 00 00
0360	00 08 08 08	00 00 00 00	00 00 00 00	00 00 00 00
0370	00 00 00 00	00 00 00 00	00 00 00 00	7C 06 50 3F
0380	7C 06 50 06	7C 06 50 58	7C 06 50 4F	7C 06 50 66
0390	7C 06 50 6D	7C 06 50 7D	7C 06 50 07	00 73 39 38
03A0	00 73 39 76	00 00 00 00	00 00 00 00	00 00 00 00
03B0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
03C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
03D0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
03E0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
03F0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

КОНТРОЛЬНАЯ СУММА БЛОКА N 3 : 21C3



БЛОК N 4      ФАЙЛ A:DM48.DMP

0400	F8 AC 00 34	16 34 CF 14	A0 D3 15 C6	1E FB D3 15
0410	C6 20 F8 53	F0 96 07 FC	47 53 F0 68	94 01 04 80
0420	FA 47 53 70	3A FB A9 FC	91 81 DC 96	2F 04 73 04
0430	7D 04 5B 65	9A 00 8A 9F	90 18 90 18	90 F9 18 90
0440	18 90 18 90	23 70 3A B9	AE C7 07 91	42 19 91 B9
0450	A0 F8 91 89	EF B8 3F F0	91 C9 E8 57	F0 91 89 A0
0460	81 07 07 84	68 B8 AD 90	B8 A0 80 17	B4 58 C8 AE
0470	80 AF 92 78	8B 01 84 7A	8B C9 F9 90	C8 89 A0 81
0480	07 07 07 07	07 90 FF E7	53 0E 03 88	AS 80 89 F0
0490	91 19 18 80	53 0F 00 91	15 E5 14 90	04 51 23 4E
04A0	14 06 14 A0	D3 00 C6 B5	F8 D3 01 96	A2 23 70 3A
04B0	08 90 23 8A	90 18 23 80	90 84 C4 23	70 3A 88 90
04C0	27 90 18 90	00 00 00 00	FA 47 E7 53	ED 43 04 18
04D0	90 F8 18 90	8B AE 80 92	E7 C5 B9 EF	B8 3F E1 A0
04E0	C9 E8 DE 88	80 A4 02 D5	B7 EF 88 3F	81 A0 C9 C8
04F0	23 19 08 96	EC B9 C7 68	17 81 A0 C9	EB F9 81 A0

КОНТРОЛЬНАЯ СУММА БЛОКА N 4 : 805B

Рис. 3. Сообщения монитора

Рис. 2. Распечатка содержимого ПЗУ модуля УМК-48/МР1

ПрСч); непосредственное исполнение начинается с запроса режима запуска сообщением 4 (клавиша 0 — режим реального масштаба времени, 1 — пошаговый). Монитор загружает внутреннюю память данных и регистры и передает управление программе пользователя по стартовому адресу. В пошаговом режиме покомандное выполнение программы производится нажатием на клавишу ШК (S7).

Стоп (Ст, S5) — директива возврата, выполняется по прерыванию и возвращает в монитор из программы пользователя (содержимое регистров и внутренней памяти данных ОЭВМ сохраняется и на дисплей выводится адрес останова). По директиве адреса памяти данных (АПД, S35) просматривается и модифицируется внутренняя и внешняя память данных ОЭВМ, ее выполнение начинается с вывода сообщения 4 на дисплей. В случае выбора внешней памяти данных (клавиша 1) поступает сообщение 5, монитор ожидает ввода двух цифр адреса и переходит к режиму отображения содержимого памяти данных. Аналогично задается адрес внутренней памяти данных (клавиша 0), при этом на дисплей выводится сообщение 6 и проверяется доступность введенного адреса (для ОЭВМ КМ1816ВЕ48 от 0 до 3FH). Последовательный просмотр и изменение содержимого ячеек памяти данных производится аналогично АПК.

С помощью директивы регистров (Рг, S33) отображается и изменяется содержимое регистров А, Т, PSW, R0...R7 обоих банков и программного счетчика. Для облегчения работы на дисплей выводится содержимое регистров и мнемоническое обозначение (регистрам А, Т и PSW соответствуют обозначения А, t и F, а остальным — birk, где i — номер банка (0, 1); k — номер регистра в банке (0...7)). Значение программного счетчика при выводе разбивается на 2 байта (PCL — младший, PCH — старший). Последовательный просмотр регистров после входа в директиву и изменение содержимого выбранного регистра осуществляются аналогично АПК и АПД. После начальной установки микроЭВМ содержимое регистров и внутренней памяти данных обнуляется.

С помощью кроссассемблера ASMK48 создаются и редактируются исходные тексты программ на языке ассемблера ОЭВМ КМ1816ВЕ48, транслируется исходный текст и создается загрузочный файл.

Кроссассемблер ASMK48 предназначен для работы в среде операционной системы ОС1800, написан на языке ассемблера МП БИС КР580ВМ80, занимает объем 6 Кбайт и обладает высоким быстродействием (загрузочный файл объемом около 2 Кбайт формируется за 2 мин). Он обеспечивает выполнение директив ввода и вывода входного и выходного файлов, вставки и удаления и замены строк, редактирования строки и задания режима трансляции. Буферы входного и выходного файлов (32 Кбайт и до 4 Кбайт соответственно) размещаются в памяти, что исключает дисковые операции при трансляции. Вместе с наличием встроенного редактора это позволяет адаптировать кроссассемблер ASMK48 для работы совместно с любой персональной ЭВМ на базе МП БИС КР580ВМ80, имеющей в качестве устройства ввода-вывода терминал. Такая резидентная версия кроссассемблера ASMK48 создана для работы в составе лаборатории по изучению микропроцессорных систем «Пирамида» [6] и используется в качестве внешнего запоминающего устройства бытовой магнитофон.

Синтаксис команд кроссассемблера соответствует приведенным в [5]. В процессе трансляции поступают сообщения об ошибках и формируется листинг трансляции, который может быть выведен на печатающее устройство. Загрузочный файл размещается на диске в виде, необходимом для работы программатора ПЗУ. Кроме того, он может быть передан в модуль эмулятора ПЗУ команд УМПК-48/MP2 для последующего выполнения и отладки. При этом обмен информацией

между модулем и инструментальной ЭВМ осуществляется через ИРПП.

С помощью кроссассемблера RASMK48 восстанавливаются исходные тексты программ ОЭВМ КМ1816ВЕ48. Кроссассемблер создает таблицы внешних и перекрестных ссылок и формирует исходный текст с метками для транслирования программы после внесения необходимых изменений в ее текст. Объем входного и выходного буферов, структура и характеристики кроссассемблера идентичны кроссассемблеру ASMK48.

Тестовая программа TST48 предназначена для проверки функционирования БИС КМ1816ВЕ48 (КР1816ВЕ35, КР1816ВЕ49) и расширителя портов ввода-вывода КР580ВР43.

Тестовая программа состоит из четырнадцати блоков (все последующие блоки теста используют проверенные ранее команды). В отдельных блоках проверяются внутренняя память данных, правильность выполнения арифметических и логических операций, адресация к внешней памяти команд и внешней памяти данных, ввод-вывод через порты P1, P2, схема пошагового выполнения, таймер-счетчик событий, система прерываний, команды условных и безусловных переходов. При сбое в одном из блоков теста на дисплей выводится номер ошибки, определяющий сбой группы команд. Тестовая программа TST48 работает в составе базового модуля и расширительного модуля УМПК-48/MP1 (ПЗУ с монитором на расширительном модуле заменяется на ПЗУ с тестовой программой).

Таким образом, описанный комплекс программно-аппаратных средств УМПК-48 позволяет самостоятельно изучать основы построения микропроцессорных систем на базе ОЭВМ серии К1816 и эффективно решать широкий круг задач разработки и отладки программного и схемотехнического обеспечения. Недостаток комплекта, ограничивающий его использование в качестве прототипной системы при разработке собственных микроконтроллеров, — ограниченный объем памяти команд ОЭВМ (2 Кбайт), в которой может размещаться отлаживаемая под управлением монитора DM48 программа пользователя. В настоящее время разработан новый вариант модулей УМПК-48/ВМ и УМПК-48/MP1 (см. вкладку); монитор хранится в теневого ПЗУ и все адресное пространство ОЭВМ может быть занято программами пользователя. Реализована возможность записи программ в машинных кодах на магнитную ленту и считывание с нее (облегчается отладка программ); все входы и выходы ОЭВМ могут быть ограничены применяться для подключения собственных схем сопряжения с объектом управления. На основе учебно-отладочной микроЭВМ удобно строить модели разрабатываемых микроконтроллеров, используя для отладки их схемотехническое и программное обеспечение устройства ввода и индикации модулей, а также средства монитора DM48.

В заключение следует отметить, что в принципиальной схеме модуля УМПК-48/ВМ ([4], с. 75, рис. 1) допущена ошибка: вход элемента D10 должен быть подключен к выходу P27 микросхемы D3, а не к выходу элемента D11.4. В целях упрощения программного обеспечения схема модуля УМПК-48/MP1 ([4], с. 77, рис. 2) доработана. Сигнал INT должен поступать на вход R триггера D8.1, а не на вход инвертора D1.4. При этом выход Q0 децифратора D6 необходимо соединить со входом инвертора D1.4. Выход Q2 децифратора D6 остается свободным, сигнал с выхода Q3 должен поступать на вход S триггера D8.2 через инвертор D1.5 (на схеме не показан), а на входы S и D триггера D8.2 — «Лог. 1». Освободившийся выход элемента D4.2 необходимо соединить со входом инвертора D1.6, а его выход — со входом R триггера D8.2. Кроме того, вход выборки CS2 микросхемы D7 должен быть под-

ключен к общей линии, а не к выходу ВЗ усилителя Д3.

Адрес: 103498, Москва, К-498, МИЭТ;  
тел. 532-99-69

### ЛИТЕРАТУРА

- Иванов Е. А., Муренко Л. Л., Широков Ю. Ф. Универсальная отладочная система автоматизации проектирования микропроцессорных устройств // Микропроцессорные средства и системы.— 1984.— № 3.— С. 53—57.
- Евламинев Р. А., Галузо Е. В., Голованов В. П. Отладочная система для однокристалльной микроЭВМ // Микропроцессорные средства и системы.— 1986.— № 3.— С. 32—33.

- Лобанов В. И. Архитектура отладочных средств для микроконтроллера // Микропроцессорные средства и системы.— 1986.— № 3.— С. 30—32.
- Кушнир В. Е., Панфилов Д. И., Шаронин С. Г. Учебная микроЭВМ на основе однокристалльной ЭВМ КМ1816ВЕ48 // Микропроцессорные средства и системы.— 1986.— № 6.— С. 75—82.
- MCS-48 Family of Single Chip Microcomputers // Users Manual. Intel Corp., 1979.
- Преснухин Л. Н., Панфилов Д. И., Романелко О. А., Шаронин С. Г. Микропроцессорная лаборатория по изучению микропроцессорных комплектов с фиксированным набором команд // Микропроцессорные средства и системы.— 1985.— № 1.— С. 77—81.

Статья поступила 30 сентября 1987

## ПЕРИФЕРИЙНОЕ ОБОРУДОВАНИЕ МИКРОЭВМ И МИКРОПРОЦЕССОРОВ

УДК 681.326—181.4

Г. М. Нулупаров

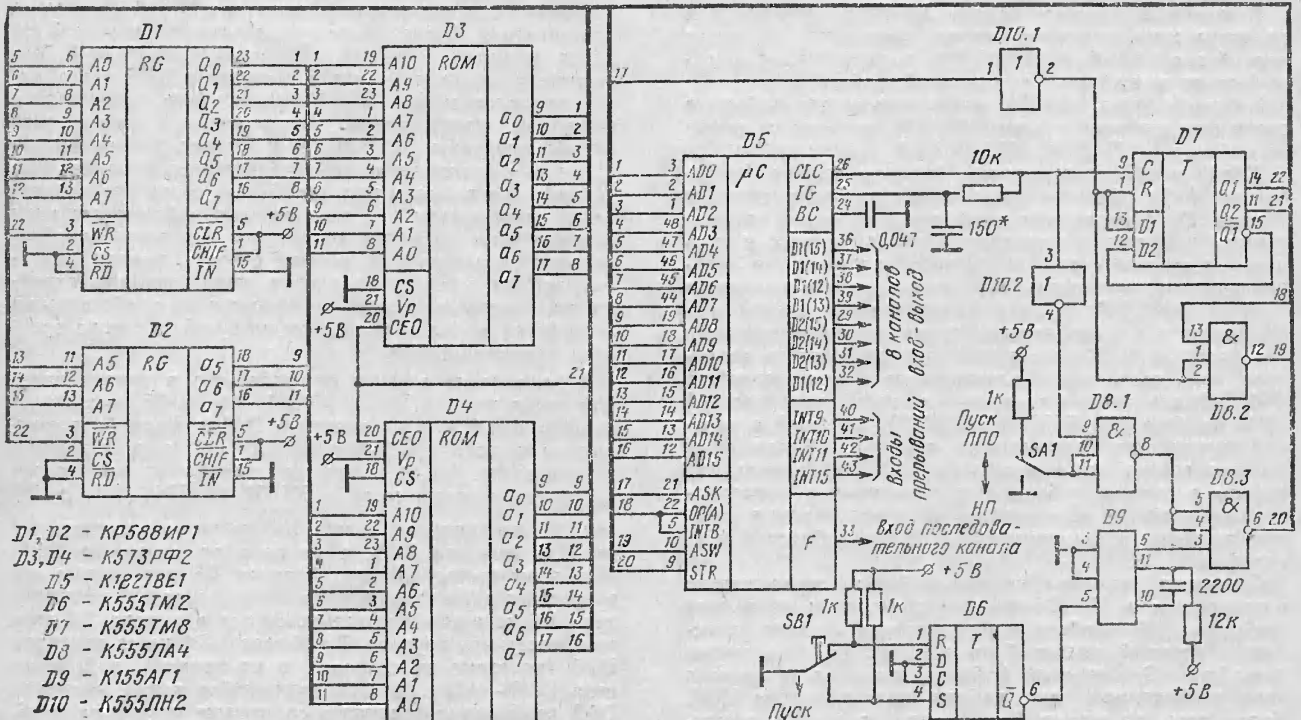
### СОПРЯЖЕНИЕ МИКРОЭВМ К1827ВЕ1 С ВНЕШНИМ ППЗУ

Однокристалльная 16-разрядная микроЭВМ К1827ВЕ1 («Электроника С5—31») предназначена для построения систем управления, обработки и передачи данных и создания компактных устройств и приборов.

МикроЭВМ К1827ВЕ1 может работать как со встроенным в процессе ее производства ПЗУ с различными целевыми программами пользователя, так и с внешним ППЗУ (часто необходимо при разработке и отладке программ и при создании уникальных образцов приборов и устройств).

Взаимодействие микроЭВМ К1827ВЕ1 с внешним ПЗУ

По команде «начальный пуск» (НП) микроЭВМ устанавливается в режим активного интерфейса и выполняет стандартную программу подготовки к работе. Для работы с внешним ППЗУ за время выполнения этой программы (около 80 мкс) необходимо подать импульс на вход младшего разряда системы прерываний. Схема управления активным интерфейсом микроЭВМ вырабатывает сигнал  $OP(A)$  — импульс сопровождения адреса (ИСА), и сигнал  $ASK$  — запрос (Зпр), а внешнее ППЗУ сигнализирует об установке информации сигналом  $ASW$  — ответ (ОтВП). Знак инверсии означает, что все сигналы, поступающие на внешние контакты из микроЭВМ и наоборот, инвертируются по отношению к зафиксированным значениям «Лог. 0» и «Лог. 1». Внешнее ППЗУ подключается к микроЭВМ через контакты двух 8-разрядных регистров ( $PgA$  и  $PgB$ ), и микроЭВМ



Принципиальная схема сопряжения микроЭВМ с ППЗУ



получает информацию по двунаправленной 16-разрядной шине адрес-данные. Схема сопряжения микроЭВМ К1827ВЕ1 с ППЗУ К573РФ2 содержит регистры адреса D1 и D2, ППЗУ — D3 и D4, формирователь сигнала обмена D7, узел формирования сигнала «Пуск» на микросхемах D6, D8 и D9. До появления сигнала  $\bar{Z}_{пр}$  на входе R формирователя D7 присутствует «Лог.0» (0В), при этом на выходах D7 уровни «Лог.0» поддерживают регистры D1 и D2 в режиме хранения информации, а ППЗУ D3 и D4 — в режиме выключенного состояния выходов. Появление сигнала  $\bar{Z}_{пр}$  подготавливает триггеры D7 к работе — по спаду тактового импульса, приходящего следом после  $\bar{Z}_{пр}$ , переключается первый триггер в D7, и адрес, выставленный микроЭВМ на шине адрес-данные, запоминается в регистрах D1 и D2. Спадом следующего тактового импульса переключается второй триггер D7, переводя ППЗУ D3 и D4 в режим выдачи информации, которая воспринимается микроЭВМ. При исчезновении сигнала  $\bar{Z}_{пр}$  триггеры D7 устанавливаются в исходное состояние.

Сигнал ОтВП формируется в схеме с помощью элемента D8.2.

Для обеспечения режимов пуска служит узел формирования сигнала «Пуск», содержащий переключатель SA1 «НП — Пуск ППО», кнопку «Пуск» SB1, и указанные выше микросхемы. В случае, если переключатель SA1 находится в положении «НП», нажатие кнопки SB1 приводит к запуску одновибратора D9, вырабатывающего одиночный импульс заданной длительности (не менее нескольких тактов). Поскольку на выходе элемента D8.1 присутствует «Лог.1» (+5 В), этот импульс инвертируется элементом D8.3 и поступает на вход STR микроЭВМ.

Если переключатель SA1 переведен в положение «Пуск ППО», описанный процесс повторяется с той лишь разницей, что на выходе элемента D8.3 при нажатии кнопки SB1 появится серия импульсов, «привязанных» к тактовым импульсам. Длительность этой серии определяется одновибратором D9. Триггер D6 служит для исключения влияния дребезга контактов кнопки SB1.

140006, г. Люберцы Моск. обл., Октябрьский проспект, д. 380Е, кв. 45. Тел. 553-57-93

Сообщение поступило 26 мая 1987

УДК 681.326—181.4

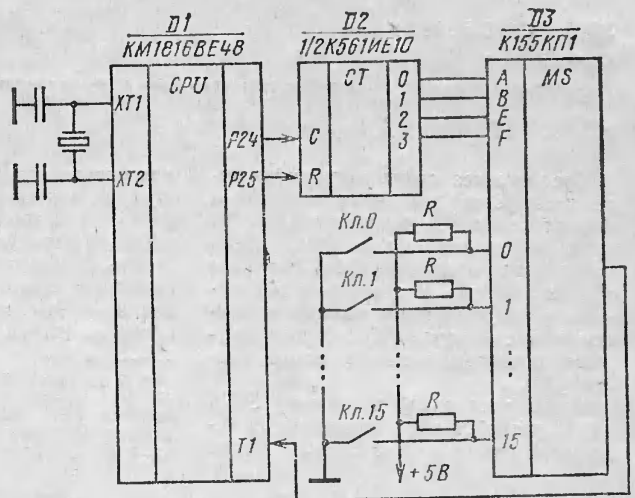
С. В. Каменский

## СПОСОБ ПОДКЛЮЧЕНИЯ 16-КОНТАКТНОЙ КЛАВИАТУРЫ К КМ1816ВЕ48

Чтобы разработать на базе однокристалльных микроЭВМ (ОМЭВМ) микроконтроллеры с минимальным привлечением дополнительных аппаратных средств, необходимо более полно использовать вычислительную мощность и линии ввода-вывода ОМЭВМ.

При проектировании микроконтроллеров часто необходимо вводить цифровую и командную информацию через небольшую клавиатуру (например, задать числовые уставки в компьютеризированных вольтметрах, дозаторах, таймерах и т. д.). Однако ввод информации с клавиатуры в подобных микроконтроллерах — не основная функция, и поэтому число линий для связи с клавиатурой должно быть минимально.

Подключить 16-контактную клавиатуру к трем линиям ввода-вывода ОМЭВМ КМ1816ВЕ48 (см. рисунок) можно с помощью двоичного счетчика D2 и селектора-мультиплексора D3. Линия P25 осуществляет сброс счетчика, а линия P24 — приращение значения показаний счетчика.



Подключение БИС КМ1816ВЕ48 к клавиатуре с минимальным использованием линий ввода-вывода

Совокупность сигналов на выходе счетчика позволяет последовательно передавать состояние каждой клавиши, подключенной ко входам селектора-мультиплексора на ОМЭВМ.

Программно в ОМЭВМ создается дополнительный внутренний счетчик, работающий синхронно с внешним счетчиком D2. Обновлению внутреннего счетчика соответствует выдача stroba сброса по линии P25, а приращению — выдача stroba по P24. Поэтому в каждый момент времени опрашивается клавиша, эквивалентная состоянию внутреннего счетчика с коэффициентом пересчета 16.

«Вписать» подобный способ сканирования клавиатуры в реальную программу можно двумя методами:

1. Опрос клавиатуры ведется в строго определенные моменты времени (например, в начальном диалоге пользователя с микроконтроллером). Такие моменты должны индцироваться системой, сложные операции в это время не выполняются. По линии P24 через определенные интервалы времени выдаются стробирующие внешний счетчик импульсы, длительность которых можно задавать любым образом. Получив информацию от клавиатуры, микроконтроллер может прекратить ее сканирование (на время).

2. Организуется непрерывная обработка прерывания по циклическому переполнению внутреннего таймера ОМЭВМ. Во время обработки прерывания выполняются следующие действия: загрузка уставки в таймер (задается частота сканирования клавиатуры), выдача stroba на приращение внешнего счетчика, приращение внутреннего счетчика, анализ линии T1. Если T1=0, нажата клавиша, номер которой равен значению счетчика; если коэффициент пересчета достигнут, то надо выставить ноль на внутреннем счетчике и подать strob по линии P25 на внешний счетчик.

Частоту сканирования клавиатуры целесообразно выбирать в интервале 50...2000 Гц равной или кратной частоте, необходимой для работы других узлов системы (индикация, тактирование работы шаговых двигателей, управление генератором пилообразного напряжения и т. д.).

В обоих случаях необходимо программно устранить дребезг клавиш. Для организации динамической памяти к выходу счетчика D2 можно подключить дешифратор-мультиплексор.

191002 Ленинград, Наб. реки Фонтанки, д. 50, кв. 30. Тел. 315-96-94

Сообщение поступило 8 июня 1987

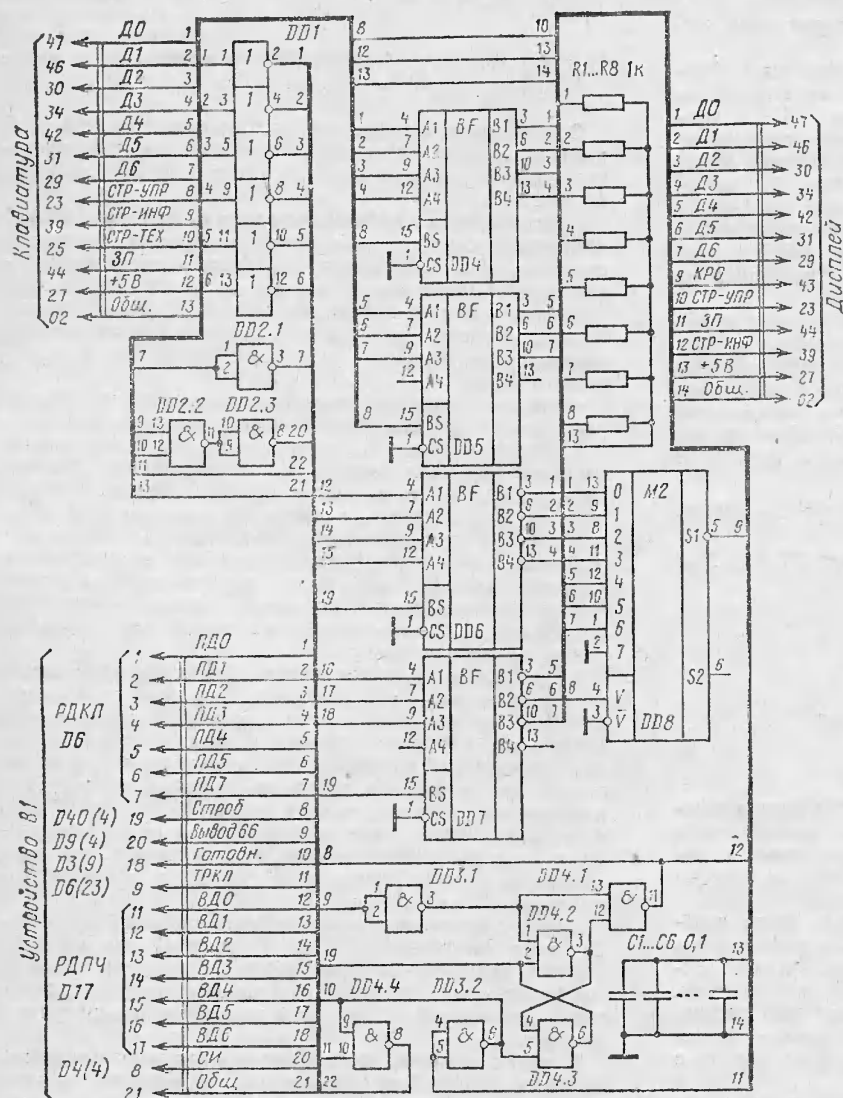
### СОПРЯЖЕНИЕ ДИСПЛЕЯ СМ 7204 С МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Предлагается схема промежуточно-го интерфейса для связи микроЭВМ «Электроника 60» и дисплея СМ 7204 через имеющееся в комплекте ЭВМ устройство управления В1. Интерфейс может работать и с другими дисплеями, у которых коды команд и символов в стандарте КОИ 7 формируются непосредственно в блоке клавиатуры.

Перед установкой в корзину ЭВМ плата устройства В1 подвергается некоторой переделке\*. Удалены ключи

для управления печатающим механизмом и микросхемы DD22, DD34, DD45; резисторы R1, R2 (при отсутствии печатающего устройства R1 omitted); конденсаторы C1, C4, C19; выводы 5, 6 микросхемы DD33 и вывод 22 — DD6 подключены к общему проводу. Вывод 4 ждущего мульти-

\* Долгих В. А., Ступенков И. В., Шинаков В. Г. Сопряжение дисплея VDT 52106 и печатающего устройства DZM-180 с микроЭВМ «Электроника 60» // ПТЭ. — 1984. — № 2. — С. 77.



DD1 - K155AH1; DD2 ... DD4 - K155ЛАЗ; DD4, DD5 - K569AP16;  
DD6, DD7 - K569AP26; DD8 - K155ИП2

Принципиальная схема устройства сопряжения дисплея СМ 7204 с микроЭВМ «Электроника 60»

вibratorа DD4 отсоединен от вывода 5 и подключен на разъем. Для подключения кабеля интерфейса на плате установлен разъем МРН22.

Принципиальная схема интерфейса изображена на рисунке. Коды символов или функций из клавиатуры по шинам D0 ... D6 поступают на шинах формирователи DD4, DD5 и инверторы DD1, DD2.1. Если это код информационной или технологической клавиши, он сопровождается импульсами СТР-ИНФ или СТР-ТЕХ и поступает по шинам ПД0 ... ПД7 в регистр РДКЛ (DD6) платы В1. Из стробов СТР-ИНФ и СТР-ТЕХ формируется импульс СИ, поступающий на одновибратор DD4 устройства В1. При занесении кода в РДКЛ вырабатывается импульс требования обслуживания клавиатуры ТРКЛ, срез которого служит для формирования сигнала ЗП, информирующего клавиатуру о том, что код принят. Информация, выводимая из ЭВМ, снимается с выходов регистра РДПЧ (DD17) платы В1 по шинах ВД0 ... ВД6, стробируется сигналом ВЫВОД 66, снимаемым с DD9, и поступает на формирователи интерфейса DD6, DD7. Коды клавиш управления сопровождаются стробом СТР-УПР, который открывает формирователи DD4, DD5 и с их выходов по шинах D0 ... D6, сопровождаемые этим же стробом, поступают в дисплей.

Выходы формирователей DD4, DD5 запараллелены с DD6, DD7, управление ими ведется по выходам BS, поэтому для нормальной работы входных цепей дисплея они нагружены резисторами R1 ... R7. Для обеспечения контроля верности передачи информации в дисплей в интерфейс введена схема контроля четности DD8, вырабатывающая контрольный разряд КРО.

Сигнал готовности дисплея ЗП поступает в плату В1 через инвертор DD3.2 и в клавиатуру через DD4.4. Элементами DD4.1 ... DD4.3 формируется сигнал СТР-ИНФ, сопровождающий код, переданный из В1 в дисплей. Этот сигнал поступает в В1.

Установив на плату В1 несколько дополнительных элементов и еще один разъем, можно подключить к ЭВМ печатающее устройство DZM-180. Такая схема позволяет организовать обмен информацией между ЭВМ и периферийными устройствами со скоростью, максимально возможной для самого медленного из них.

Интерфейс конструктивно выполнен на печатной плате с печатным разъемом, который устанавливается в разъем дисплея, предназначенный для клавиатуры. В разъем интерфейса устанавливается разъем с кабелем от блока клавиатуры.

634055, Томск, пр. Академический, 1.  
Институт оптики атмосферы СО АН СССР; тел. 1-81-11, доп. 7-39

Сообщение поступило 1 июля 1987

# А. Н. Соколов, В. Г. Бремшmidt, В. С. Зайдель

## КОНТРОЛЛЕР НСМД ЕС-5061 ДЛЯ МИКРОЭВМ ТИПА «ЭЛЕКТРОНИКА 60»

Эффективное применение микроЭВМ для широкого класса задач сдерживает возникающая в настоящий момент диспропорция между большими вычислительными возможностями микроЭВМ и ограниченными — подключаемых к ним устройств внешней памяти. Запоздывание массового производства миниатюрных «винчестерских» дисков заставляет разработчиков АСУ обращаться к техническим средствам ЕС и СМ ЭВМ.

Из большого набора внешних запоминающих устройств наиболее распространены накопители на магнитных дисках (НМД) ЕС-5061, СМ-5400 и СМ-5408. Для включения этих накопителей в комплекс на базе микроЭВМ с помощью контроллеров СМ-5407, СМ-5402 и СМ-5415 необходим дополнительный адаптер интерфейса ОШ-МПИ. Кроме того, значительные габариты и энергопотребление этих устройств снижают экономические и эксплуатационные характеристики микросистемы.

Контроллер НМД, выполненный в стандарте микроЭВМ и подключаемый к системной магистрали в качестве функционального модуля, обеспечит высокую надежность и эффективность работы микросистемы.

В Белорусском филиале Всесоюзного научно-исследовательского и проектно-технологического института статистической информационной системы (ВНИПИстатинформ) Госкомстата СССР разработан контроллер УУМД-02 для подключения НСМД ЕС-5061 к микроЭВМ типа «Электроника 60».

Контроллер выполнен в виде двух модулей (табл. 1),

Таблица 1  
Модули контроллера УУМД-02

Модуль	Тип платы по ГОСТ 10317-72	Габаритные размеры, мм	Число корпусов, ИС/ИС <sub>14</sub>	Плотность монтажа, см <sup>2</sup> /ИС <sub>14</sub>	Пл. отн. к монта-жа, выв/см <sup>2</sup>	Мощность, потребляемая от источника 5 В, Вт, не более
Процессорный (МП)	7Д3	280×240×12	64/79	8,5	2,2	17
Управления НСМД ЕС-5061	3Д3	135×240×12	35/34	9,5	2,2	8

размещаемых в корпусе микроЭВМ. В блоке сопряжения (480×180×50 мм) установлено 7 разъемов РШ7П-10ШЗТ-В для подключения четырех НСМД ЕС-5061 и один разъем СНП59—96/94×11В-23 для подключения контроллера.

Модуль процессорный (рис. 1) — универсальный 16-разрядный с развитой системой команд, позволяет реализовать стандартные функции взаимодействия между микроЭВМ типа «Электроника 60» и специализированными модулями управления внешними устройствами. Модуль процессорный состоит из схемы (на базе БИС серии 1804) микропрограммного управления; схемы связи с каналом «Электроника 60» (основной элемент — ПЛМ типа К556РТ1); ОЗУ с объемом 4К×16 бит,

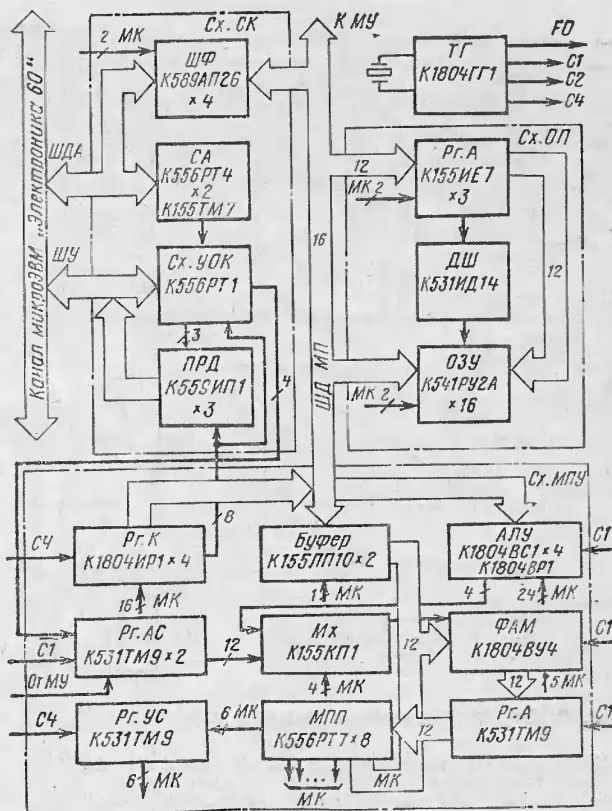


Рис. 1. Структурная схема модуля процессорного (МП)

Сх. СК — схема связи с каналом; ШФ — шинный формирователь; СА — селектор адреса; Сх. УОК — схема управления обменом; ПРД — передатчики; ТГ — тактовый генератор; Сх. ОП — схема оперативной памяти; Рг. А — регистр адреса; ДШ — дешифратор банка; Сх. МПУ — схема микропрограммного управления; Рг. К — регистр констант; Рг. АС — регистр асинхронных событий; ФАМ — формирователь адреса микрокоманд; Рг. УС — регистр управляющих сигналов; МПП — микропрограммная память

Модуль управления НСМД ЕС-5061 (рис. 2) — специализированное устройство, выполняющее следующие инструкции процессорного модуля: запись слова  $\alpha$ -пробега (1), маркера зоны заголовка (2), маркера зоны данных (3), маркера конца зоны заголовка или данных (4), слова заголовка или данных (5), поиск маркера заголовка (6) и маркера данных (7), чтение слова заголовка или данных (8).

Инструкции выполняются под управлением микропрограммного автомата (МПА); каждая — за один цикл МПА, состоящий из 32 тактов. Тактовая частота ( $f_T = 5$  МГц) формируется в режиме «Запись» из частоты  $F_0$ , а в режиме «Чтение» выделяется из последовательности синхронных импульсов и данных, считываемых с НМД. После каждого цикла модуль управления формирует сигнал «Требование прерывания» (ТПР). Для записи или чтения информации реализуются комбинации инструкций процессорного модуля.

Для повышения надежности контроллера и устранения влияния эффекта «пиковых сдвигов», возникающих в процессе магнитной записи на диск, МПА обеспечивает введение компенсационных предсказаний в последовательность синхронных импульсов и данных в зависимости от порядка их следования.

Контроллер имеет две системы команд. Пользователь, устанавливая соответствующую переключку в блоке МПП, выбирает систему команд I (для эффективной работы контроллера в стандартных ОС, используемых на микроЭВМ типа «Электроника 60» (например, РА-

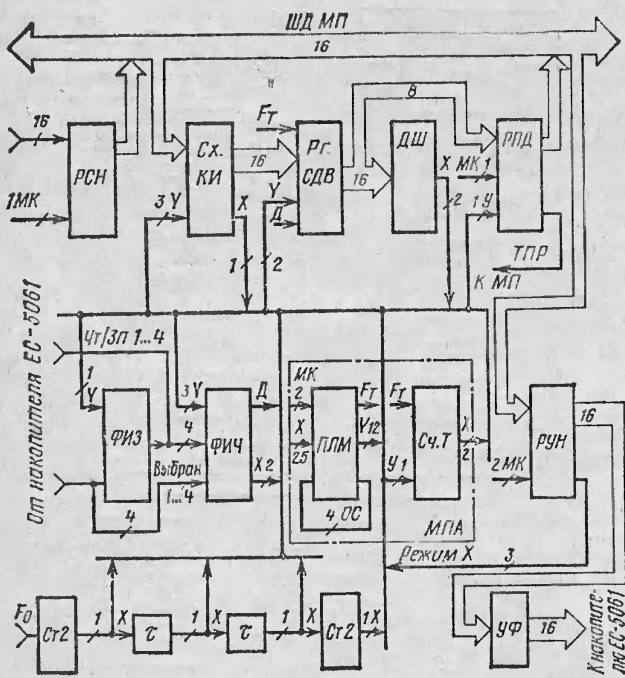


Рис. 2. Структурная схема модуля управления НСМД EC-5061

РСУ — регистр состояния накопителя (K589ИР12×2); Сх. КИ — схема кодирования (K556РТ7×2); Рг. СДВ — регистр сдвига (K155ИР13×2); ДШ — дешифратор кода маркера (K556РТ4×2); РПД — регистр принимаемых данных (K589ИР12×2); ФИЗ — формирователь импульсов записи (K531ЛА3, K170АП3×2); ФИЧ — формирователь импульсов чтения (K155ЛР3, K155АГ3, K531ТМ2×1,5); ПЛМ — программируемые логические матрицы (K556РТ1×2); Сч. Т — счетчик тактов (K155ИЕ7, K531ТМ2); Ст2 — делитель на 2 (K531ТМ2×0,5); т — задержка (K531ЛА3×0,5); РУН — регистр управления накопителем (K531ТМ8×4); УФ — усилители-формирователи (K170АП3×8); ОС — обратная связь; МПА — микропрограммный автомат; ШД МП — шина данных модуля процессорного; X — множество входных переменных; Y — множество выходных переменных

ФОС) или систему команд 2 (для программно-аппаратного комплекса «Кадры» \*).

\* Илюкович А. А. Информационно-поисковая система «Кадры» // Микропроцессорные средства и системы. — 1986. — № 6. — С. 40.

### Система команд 1.

В этой системе команд контроллер выполняет следующие команды: «холостой ход», «установка», «возврат к нулю», «запись», «чтение», «контроль записи», «контроль чтения», «форматирование дорожки», «тест самодиагностики».

Информация записывается секторами с фиксированной длиной зоны данных. Каждая дорожка содержит 12 секторов. Каждый сектор состоит из зоны заголовка и зоны данных. Длина зоны данных каждого сектора — 256 слов (512 байт). Таким образом, объем полезной информации на дорожке — 6 Кбайт, а максимально полезный объем пакета — 24360 Кбайт. Структуры дорожки и сектора показаны на рис. 3, 4.

По командам «запись», «чтение», «контроль записи», «контроль чтения» за одно обращение к контроллеру может быть обработано до 65535 слов данных. При этом переход с дорожки на дорожку, с цилиндра на цилиндр автоматический.

По команде «форматирование дорожки» на заданной дорожке записываются α-пробелы, заголовки и зоны данных секторов. Эта команда позволяет разметить «чистый» пакет дисков. Команды «запись», «чтение», «контроль записи», «контроль чтения», «форматирование дорожки» выполняются с установкой.

Программно-доступный регистр управления и состояния (РУС) одинаков для обеих систем команд (рис. 5).

Разряд 07 (готовность контроллера сигнализирует о том, что контроллер завершил выполнение текущей команды и готов к приему новой. Разряды 08, 09 (номер НМД) — номер дисковод, на котором выполнялась текущая команда. Разряд 15 (ошибка) сигнализирует о том, что текущая команда завершилась с ошибкой. Вид ошибки уточняется в регистре ошибок. Остальные разряды нулевые.

Для выполнения команд контроллеру необходимо задать таблицу параметров (табл. 2), которая записывается в ОЗУ микроЭВМ. Выполнение команды инициируется занесением адреса таблицы параметров в РУС.

Регистр команд имеет следующую структуру: Разряды 1...4 — код команды контроллера. Разряд 6 (разрешение прерывания) разрешает контроллеру вызвать прерывание при появлении «Лог. 1» в седьмом разряде РУС, т. е. после завершения выполнения команды. Разряды 8, 9 (номер НМД) — номер накопителя, на котором должна быть выполнена команда. Разряд 11 (блокировка модификации адреса ОЗУ) — запись «Лог. 1» в этот разряд запрещает контроллеру наращивать адрес массива данных при выполнении опера-

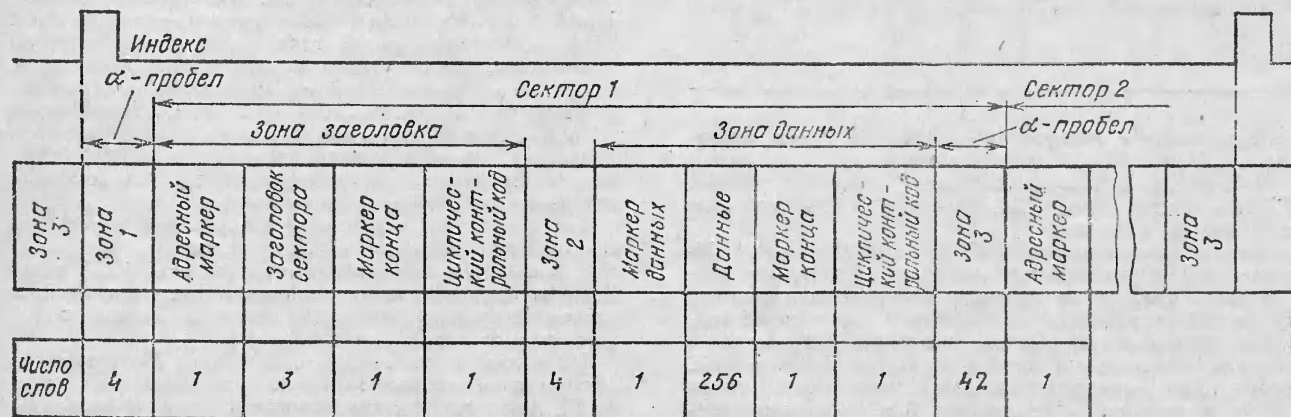


Рис. 3. Структура дорожек

Сектор											
Зона заголовка						α-пробел	Зона данных				α-пробел
Адресный маркер	Номер цилиндра	Номер дорожки	Номер сектора	Маркер конца	Циклический контрольный код	Зона 2	Маркер данных	Данные	Маркер конца	Циклический контрольный код	Зона 3
1	1	1	1	1	1	4	1	256	1	1	42

Рис. 4. Структура сектора

Таблица 2  
Таблица параметров контроллера

Номер слова	Регистр	Система команд	
		1	2
1	Команд	+	+
2	Номера цилиндра	+	+
3	Номера дорожки	+	+
4	Номера сектора (записи)	+	+
5	Счетчика слов	+	+
6	Адреса массива данных	+	+
7	Ошибок	+	+
8	Остатка на дорожке	—	+

ций «запись», «чтение», «контроль записи», «форматирование дорожки», таким образом, буфером ввода-вывода является одно слово ОЗУ микроЭВМ. Разряд 12—запись «Лог. 1» в этот разряд запрещает контроллеру продолжать выполнение команды в случае возникновения негрубой ошибки. При этом контроллер прекратит выполнение команды, а содержимое таблицы параметров будет указывать на сектор, в котором произошел сбой. Остальные разряды регистра команд—резервные.

#### Система команд 2.

В системе команд 2 контроллер выполняет следующие команды: «холостой ход»; «установка»; «возврат к нулю»; «запись заголовка и данных»; «запись»; «чтение»; «поиск по ключу на равно (однородный) и чтение»; «тест самодиагностики».

Структура физической записи в системе команд 2 отличается от структуры сектора в системе команд 1 переменной длиной зоны данных. В связи с этим в заголовке присутствует дополнительное слово—длина данных, стоящее за номером сектора. Кроме того, длина α-пробела после зоны данных (зона 3, рис. 3) не фиксирована, а зависит от длины зоны данных. Минимальная длина зоны данных—1 слово, максимальная—3670 слов (вся дорожка). Возможна запись с нулевой длиной зоны данных, состоящая только из заголовка. Максимальный объем пакета (при записях длиной в дорожку) равен 29 Мбайт. За одно обращение к конт-

роллеру может быть записана или прочтана одна физическая запись.

По команде «запись заголовка и данных» находится предыдущая запись (или индекс), пропускается ее зона данных, за всю записывается α-пробел, заголовок требуемой записи, ее зона данных, а остаток дорожки стирается. При стирании остатка дорожки рассчитывается длина записи, которая может поместиться на оставшуюся часть дорожки. С помощью команды «запись заголовка и данных» можно форматировать «чистый» пакет.

По команде «запись» корректируются данные одной физической записи.

Команда «поиск по ключу на равно (однородный) и чтение» выполняется так: от начала дорожки находится запись, у которой первые восемь байтов зоны данных совпадают с ключом, находящимся в первых восьми байтах буфера ввода. Если такая запись найдена, то ее данные вводятся в буфер ввода микроЭВМ. Эта команда может применяться для ускорения поиска метки файла в оглавлении диска, если каждая метка представляет собой отдельную физическую запись.

В отличие от системы команд 1 в таблицу параметров (табл. 2) добавлен регистр остатка на дорожке.

Регистр команд имеет следующую структуру. Разряды 1...4—код команды контроллера. Разряд 6 (разрешение прерывания) разрешает контроллеру вызвать прерывание при появлении «Лог. 1» в седьмом разряде РУС, т. е. после завершения выполнения команды. Разряды 8...11 (номер НМД) содержат номер накопителя в позиционном виде. Остальные разряды—резервные.

Команды запускаются так же, как и в системе команд 1.

Выводы. Применение ПЛМ и МП комплекта БИС серии К1804 помогло создать высокопроизводительный интеллектуальный малогабаритный контроллер для НСМД ЕС-5061. Его магистрально-модульная структура позволяет подключить другие типы НМД (например, СМ-5400), а также, адаптируя интерфейс модуля управления с объектом и перепрограммируя ПЗУ, управлять другими объектами.

Контроллер значительно расширяет область применения микроЭВМ типа «Электроника 60». Комплекс можно эффективно использовать как инструментальный для разработки ПО микроЭВМ непосредственно на этих машинах. Перспективно применение микроЭВМ в интегрированных системах сбора и обработки информации, в создании однородных локальных сетей микроЭВМ, реализующих совместный доступ к НМД.

Описываемый контроллер внедрен и успешно эксплуатируется в составе комплекса технических средств ИПС «Кадры» (см. сноску на стр. 10).

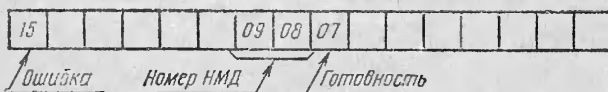


Рис. 5. Структура регистра управления и состояния

220070, Минск, пр. Партизанский, 14;  
тел. 46-05-41

Статья поступила 27 апреля 1987

# УСТРОЙСТВО СОПРЯЖЕНИЯ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ КТС ЛИУС-2 С ДИСПЛЕЕМ РИН-609

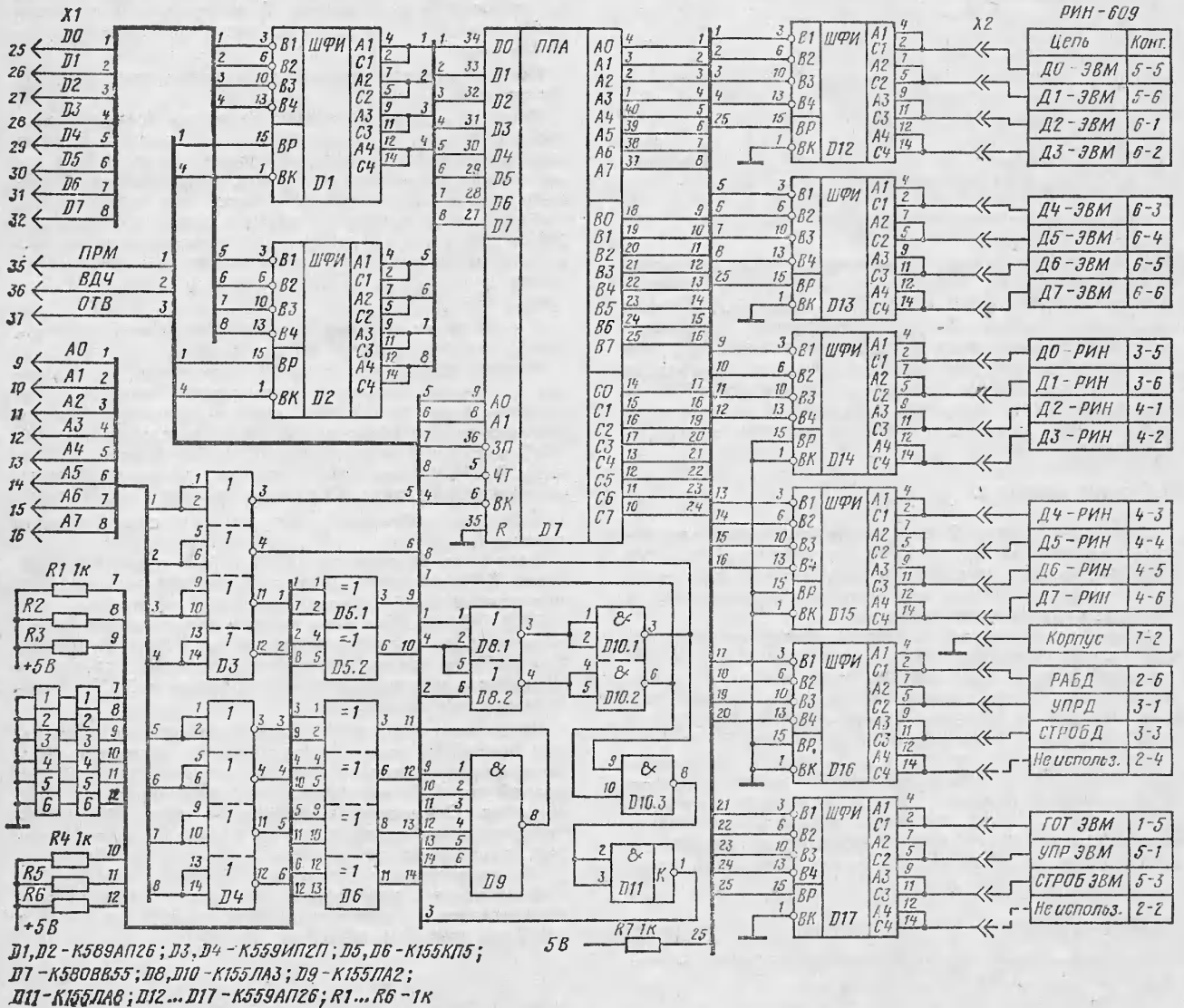
Микропроцессорный отладочный комплекс КТС ЛИУС-2 ориентирован на использование в качестве консоли клавиатуры и телевизионного индикатора. Для расширения возможностей комплекса, а также для организации многопультной работы с системой в качестве дополнительного терминала использован дисплей РИН-609. Схема электронного сопряжения дисплея с микропроцессорной системой изображена на рисунке. Сопряжение с магистральным интерфейсом системы осуществляется сигналами D0...D7, A0...A7, ПРМ, ВДЧ, ОТВ, поступающими на разъемы соединения X1, X2.

Сопряжение выполнено по схеме, ставшей классической. В ее основе — программируемый параллельный адаптер ввода-вывода D7 и шинные формирователи D1,

D2, D12...D17, которые под управлением программных драйверов настройки, ввода и вывода осуществляют обмен данными между микроЭВМ и терминалом. Драйверы полностью соответствуют описанным в работе [1].

Устройству может быть присвоен любой адрес 0...255 с помощью установки переключателей на колодке XN. В соответствии с установленными переключателями, устройству сопряжения присвоен шестнадцатеричный адрес IC, а это значит, что оно будет выбрано при установке на шинах A0...A7 инверсного значения одного из четырех адресов: IC, ID, IE, IF. При этом шина данных коммутируется с одним из четырех портов параллельного программируемого адаптера: А, В, С или регистром управляющего слова.

Сравнение поступившего адреса с присвоенным осуществляется микросхемами D5, D6, D9 после инвертирования поступившего на шину A0...A7 адреса микросхемами D3, D4, выполняющими роль магистральных приемников. Опознанию адреса соответствует низкий уровень сигнала на выходе D9. Этот сигнал разрешает работу входных шинных формирователей D1, D2, адап-



Принципиальная схема устройства сопряжения комплекса КТС ЛИУС-2 с дисплеем РИН-609

тера D7 и инициализирует работу схемы выдачи сигнала ОТВ, который вырабатывается адресованным устройством после выполнения предписанной ему функции.

Схема выработки сигнала ОТВ собрана на микросхемах D8, D10, D11. Сигнал ОТВ формируется на выходе D11 в виде импульса отрицательной полярности. Элемент D11 выполнен с открытым коллекторным выходом, что позволяет объединить сигналы ОТВ нескольких устройств по схеме «монтажное ИЛИ».

Сигнал ОТВ появляется после опознавания адреса устройством и после выдачи микропроцессором низкого (L-активного) уровня сигналов ПРМ или ВДЧ. Получив сигнал ОТВ, микропроцессор по истечении определенного времени снимает низкий уровень сигналов ПРМ или ВДЧ, на что адресованное устройство также отвечает съемом сигнала ОТВ. Выходные буферы устройства обладают некоторой избыточностью, что делает его универсальным, обеспечивающим обмен данными в любом направлении, и позволяет использовать рассматриваемое сопряжение для реализации различных протоколов обмена.

Для описанной реализации возможна замена шинных формирователей D12, D17 элементами HE серии K155 или другими подобными элементами, обеспечивающими фиксированное направление передачи данных. На схеме устройства не показаны развязывающие конденсаторы.

Разработанное сопряжение выполнено в виде сменной платы и устанавливается в каркас на одно из свободных мест, подключенных к магистрали связи микропроцессорной системы. Сопряженные существенно расширили технические и функциональные возможности системы. Ввод и вывод информации в интерпретаторе с языка БЕЙСИК и кадровый редактор [2] ориентированы на дисплей РИН-609.

#### ЛИТЕРАТУРА

1. Гуртовцев А. Л., Гурчик М. Е., Дреннов А. Н. Сопряжение микроЭВМ «Электроника К1-10» с дисплеем РИН-609 // Приборы и техника эксперимента. — 1983. — № 5.
2. Динес В. Л., Мартюхин В. А. Кадровый редактор для микропроцессорной системы КТС ЛИУС-2 с дисплеем РИН-609 // Микропроцессорные средства и системы. — 1987. — № 2.

Телефон: 5-21-71, 7-30-12, г. Рубежное, Днепропетровской области

Сообщение поступило 24 июля 1987

УДК 681.325 : 181.4

В. Л. Динес, В. А. Мартюхин

### МОДИФИКАЦИЯ ТЕЛЕВИЗИОННОГО МОНИТОРА КТС ЛИУС-2 ПОД ДИСПЛЕЙ ВТА 2000 ИЛИ РИН-609

Системный монитор микропроцессорной системы КТС ЛИУС-2 версии 2.1 настроен на работу со стандартным телевизионным приемником и символьной клавиатурой. Известно, что по качеству отображения информации на экране такой приспосабливается значительно уступает дисплеям типа ВТА 2000 или РИН-609. Если использовать способ сопряжения, описанный в работе [1], и несколько изменить содержимое ПЗУ, то можно организовать ввод и вывод информации в системе через дисплей.

При изменениях, вносимых в ПЗУ (см. рисунок), необходимо учитывать следующие требования:

- настройка адаптера начинается с адреса F080;
- подпрограмма ввода символа располагается со следующего адреса, переход на нее осуществляется с адреса FCB3;

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F080	0E	00	11	00	06	21	00	D0	CD	5A	F1	0E	00	11	00	0C
F090	21	00	E0	CD	5A	F1	21	21	E1	E5	01	63	F1	1E	07	16
F0A0	5F	0A	77	2C	03	15	C2	A1	F0	1D	E1	CA	B3	F0	24	E5
F0B0	C3	9F	F0	3E	7E	32	5F	E9	11	FC	F3	21	FE	E9	1A	B7
F0C0	CA	C9	F0	77	13	24	C3	BE	F0	21	00	D0	22	00	D6	36
F0D0	FE	AF	32	02	D6	C3	C5	D5	E5	2A	00	D6	3A	02	D6	77
F0E0	79	FE	0C	C2	F5	F0	21	00	D0	22	00	D6	11	00	06	CD
F0F0	5A	F1	C3	0C	F1	FE	0A	CA	2A	F1	FE	0D	CA	51	F1	77

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FCB0	C2	BD	FC	DB	F8	E6	80	CA	B3	FC	DB	F9	C9	FE	02	CA
FCC0	CF	FC	3E	E8	C3	F7	FC	CD	7D	FD	B7	C8	C3	E0	FF	E5

a)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F080	3E	83	D3	17	3E	06	D3	17	3E	0D	D3	17	3E	08	D3	17
F090	C9	DB	16	1F	DA	91	F0	3E	0A	D3	17	DB	16	E6	04	C2
F0A0	9B	F0	DB	15	E6	7F	F5	3E	0B	D3	17	F1	C9	F0	24	E5
F0B0	C3	9F	F0	3E	7E	32	5F	E9	11	FC	F3	21	FE	E9	1A	B7
F0C0	CA	C9	F0	77	13	24	C3	BE	F0	21	00	D0	22	00	D6	36
F0D0	FE	AF	32	02	D6	C9	DB	16	1F	DA	D6	F0	1F	DA	D6	F0
F0E0	79	A7	E2	E7	F6	F6	80	D3	14	3E	0C	D3	17	3E	0D	D3
F0F0	17	C9	C3	0C	F1	FE	0A	CA	2A	F1	FE	0D	CA	51	F1	77

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FCB0	C2	BD	FC	C3	91	F0	80	CA	B3	FC	DB	F9	C9	FE	02	CA
FCC0	CF	FC	3E	E8	C3	F7	FC	C9	7D	FD	B7	C8	C3	E0	FF	E5

b)

Содержимое ПЗУ до внесения изменений (а), после внесения изменений (б)

подпрограмма вывода символа начинается с адреса F0D6;

команда возврата записана по адресу FCC7.

Программы макроассемблер, компоновщик и редактор изменений не требуют. Программы настройки адаптера и драйверы ввода-вывода описаны в [1]. Предложенный способ не мешает использовать кадровый редактор [2], ориентированный на определенный тип дисплея. (Литературу п.п. 1, 2 см. в предыдущей статье).

Рубежанский филиал Днепропетровского химико-технологического института, г. Рубежное. Тел. 5-21-71, 7-30-12

Сообщение поступило 24 июля 1987

### Организовать централизованное техобслуживание и ремонт микроЭВМ

Есть в Ленинграде ремонтно-восстановительный завод вычислительной техники (192102, Ленинград, Волковский пр., 146, тел. директора Тихонова Э. Ю. — 269-84-24), который принимает ежедневно от организаций и предприятий не более 70 заказов на ремонт ЭКВМ и 2 — на микроЭВМ типа «Электроника 60», чего для г. Ленинграда очень мало. Причем сроки ремонта — до месяца и более.

Такое «проблемное» безобразие с организацией централизованного техобслуживания микроЭВМ приняло хронический характер и вполне годится для отражения в «Прожекторе перестройки» ЦТ.

Считаю, что на страницах журнала МП должен выступить по данному вопросу ГКВТИ СССР. Так дальше продолжаться не может.

Ю. В. Антонов, Ленинград (218-54-07)

## ИСПОЛЬЗОВАНИЕ ДИСПЛЕЯ «ГРАФИТ» В КАЧЕСТВЕ УДАЛЕННОГО ГРАФИЧЕСКОГО ТЕРМИНАЛА ИМС КАПРИ

Техническая реализация систем автоматизированного проектирования (САПР) и технологической подготовки производства (АСТПП) в интегрированной машиностроительной системе (ИМС) КАПРИ [1] основана на широком применении графических терминалов. Среди серийно выпускаемых отечественных терминалов одним из наиболее пригодных для решения задач САПР и АСТПП является терминал «Графит» АС 7060 [2]. Он предназначен для формирования, отображения и редактирования текстовой и графической информации в диалоговом режиме. Терминал может использоваться в автономном режиме и в режиме связи с мини-ЭВМ СМ3, СМ4 в составе комплексов автоматизированных рабочих мест (АРМ).

Дисплейный контроллер, сканируя ОЗУ дисплейного файла, управляет функциональными генераторами, формирующими изображение на экране индикатора (рис. 1). Дисплейный файл готовится программой в микроЭВМ «Электроника 60» (Э60) 1-го уровня, имеющей выход на магистраль дисплейного контроллера через модуль связи шин. К микроЭВМ 1-го уровня подключены также планшет для кодирования чертежей, алфавитно-цифровая и функциональная клавиатуры, модули связи с СМ4 и Э60 2-го уровня.

Программное обеспечение Э60 1-го уровня реализует графический язык для автономного режима работы или выполнение команд в режиме связи с СМ4. Э60 2-го уровня предназначена для расширения функциональных возможностей терминала в автономном режиме, например, за счет подключения накопителя на гибких

магнитных дисках (НГМД), но практически не используется.

К сожалению, терминал «Графит» имеет следующие недостатки, затрудняющие его применение в качестве графического терминала ИМС КАПРИ:

программное обеспечение Э60 1-го уровня препятствует эффективной реализации функциональных возможностей пакета базовой графики (ПБГ) [3], обеспечивающего независимость прикладных пакетов ИМС КАПРИ от конкретного графического оборудования, вычислительная мощность двух Э60 используется только на 50 %;

канал связи терминала с СМ4, выполненный на основе параллельного интерфейса, не позволяет работать на удалении свыше 15 м от ЭВМ, отличается низкой помехоустойчивостью, большой номенклатурой используемых блоков элементов;

упрощенная схема прерываний в модулях связи микроЭВМ терминала не позволяет реализовать приемлемый протокол асинхронного обмена.

Для устранения этих недостатков сделаны следующие доработки:

1. Каналы связи с СМ4 и между микроЭВМ заменены унифицированными каналами на базе стандартного устройства последовательного интерфейса DLKC-2 [4]. Такой унифицированный канал обеспечивает стандартный протокол асинхронного обмена и возможность удаления терминала до 1 км.

2. Канал связи с СМ4 подключен не к Э60 1-го уровня, а к Э60 2-го уровня, что позволяет более полно использовать вычислительную мощность обоих ЭВМ.

3. Разработано программное обеспечение, поддерживающее такую конфигурацию связей с возможностями загрузки программ в Э60 и их отладки.

4. Разработано новое программное обеспечение Э60 1-го уровня, составляющее базовый уровень графического терминала КАПРИ.

Устройство последовательного интерфейса DLKC-2 предназначено для организации дистанционной межпроцессорной связи между микроЭВМ Э60 и СМ3, СМ4. Оно используется для связи ЭВМ по коаксиальному кабелю РК-50, РК-75 на расстоянии до 1 км. Способ передачи данных по линии связи — асинхронный дуплексный с подтверждением и контролем по паритету. Разрядность передаваемого информационного слова — 16 бит. Скорость передачи в линии связи — 1 Мбит/с. Высокая помехоустойчивость канала связи обусловлена гальваническим разделением линий связи с входом и выходом устройства DLKC-2 через импульсные трансформаторы [5].

При децентрализации и широком распространении терминальных станций на рабочих местах конструкторов и технологов становится актуальной задача унификации аппаратуры связи. Сокращение номенклатуры блоков в результате унификации позволяет снизить затраты на приобретение и эксплуатацию аппаратуры связи, повысить коэффициент готовности всей системы. Естественной основой унификации может служить единый интерфейс магистралей всех ЭВМ, входящих в систему.

Для унификации технических средств каналов связи между мини- и микроЭВМ разработано устройство сопряжения магистральных интерфейсов (УСМИ), обеспечивающее системотехническое, электрическое и конструктивное согласования интерфейса «Общая шина» СМ ЭВМ и интерфейса канала микроЭВМ Э60 на аппаратном уровне.

В отличие от известного адаптера магистралей [6], позволяющего соединить микроЭВМ с контроллерами интерфейсных устройств СМ ЭВМ, УСМИ осуществляет двустороннее преобразование интерфейсов: «Общая шина» — канал микроЭВМ при управлении сопряженными магистральными арбитрами ОШ или канал микроЭВМ — «Общая шина» при управлении арбитрами канала.

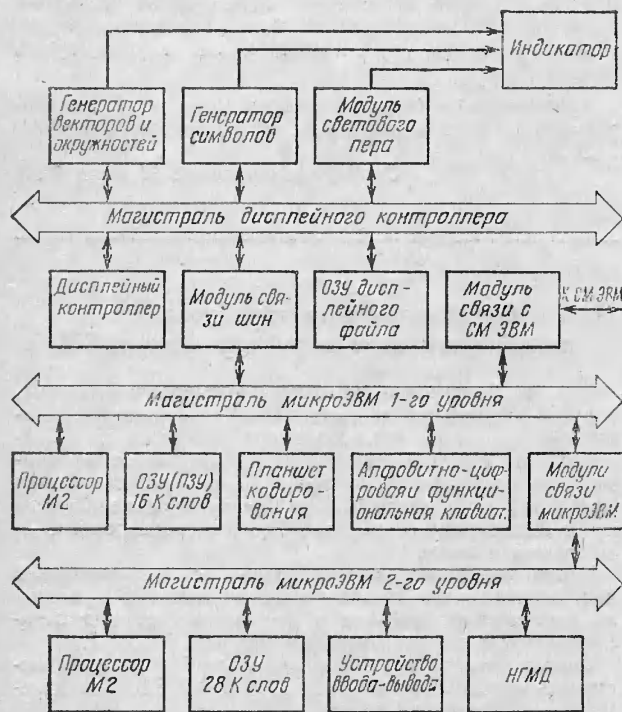


Рис. 1. Структурная схема терминала «Графит»



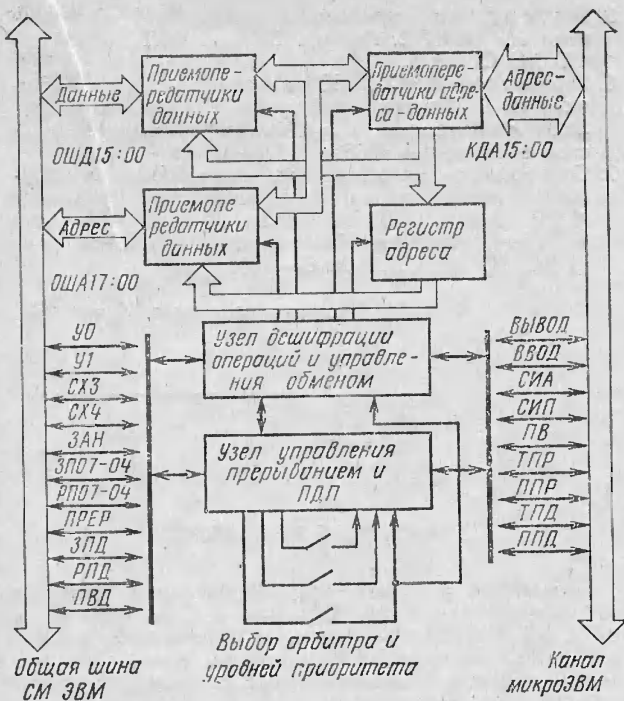


Рис. 2. Функциональная схема устройства сопряжения магистральных интерфейсов УСМИ

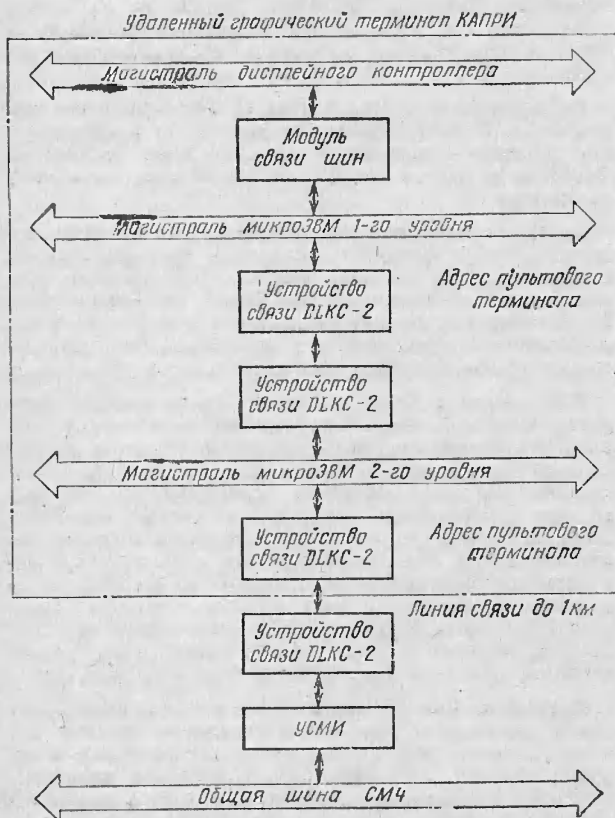


Рис. 3. Структурная схема связей удаленного графического терминала КАПРИ

Функционально УСМИ (рис. 2) представляет собой автомат со схемной реализацией алгоритмов управления операциями переключения линий адресов и данных, синхронизации протоколов обмена по прерыванию и прямому доступу к памяти. УСМИ выполнено на печатной плате размерами 240×280 мм, устанавливаемой в корпус микроЭВМ, на которой имеется разъем для подключения стандартного кабеля интерфейса «Общая шина». Ток потребления от источника питания 5 В — не более 2,5 А. Задержка сигналов интерфейса, вносимая УСМИ, — не более 250 нс. Выбор типа преобразования интерфейса и уровней прерывания ОШ задается установкой перемычек на плате УСМИ.

Подключение устройств с интерфейсом канала микроЭВМ к «Общей шине» через УСМИ (рис. 3) позволяет унифицировать аппаратуру связи: сократить в 3-4 раза номенклатуру используемых блоков элементов и упростить обслуживание системы. В то же время сохранена возможность адаптера магистралей [6] подключать к микроЭВМ весь набор контроллеров периферийных устройств СМ ЭВМ.

Структура связей терминала позволяет разместить в Э60 1-го уровня базовое программное обеспечение, а в Э60 2-го уровня — прикладное. Имеется достаточно развитое программное обеспечение для канала связи на базе DLKC-2 [7], предназначенное для обслуживания удаленной Э60 с подключенным к ней пультовым терминалом. Э60 работает под управлением операционной системы RSX11-S, занимающей значительные ресурсы памяти. Отладка программ ведется с пультового терминала средствами отладчика RSX11-S.

Такая схема работы, вполне пригодная для запуска задач на удаленной ЭВМ, не совсем подходит для графического терминала КАПРИ по следующим причинам:

программное обеспечение графического терминала не помещается в универсальных средствах RSX11-S, в то время как графическая информация может занимать значительное пространство оперативной памяти, чему препятствовало бы размещение программ операционной системы. Кроме того, программа отладчика компонуется вместе с отлаживаемой задачей, что еще больше уменьшает остающиеся ресурсы памяти;

для отладки программ в Э60 необходимо было бы к каждой из них подключить пультовой терминал, который сложно использовать для других задач, а из-за отсутствия в Э60 пользовательского режима выполнение команды останова привело бы к выходу на пультовую программу, вести отладку в терминалах которой при наличии операционной системы крайне затруднительно.

Поэтому при использовании канала связи на DLKC-2 был предпринят следующий подход:

не размещать в Э60 операционную систему, а ограничиться несложным монитором параллельных процессов\*, предназначенным для обслуживания прерываний; подключить линию связи от СМ4 к пультовому каналу (т. е. к устройству DLKC-2 с адресом пультового терминала) Э60 2-го уровня, а линию связи от Э60 2-го уровня — к пультовому каналу Э60 1-го уровня, что позволяет совместить обмен и отладку в одном физическом канале и избавиться от пультовых терминалов (рис. 3).

Драйвер канала связи на базе DLKC-2, интерфейс которого вполне удовлетворяет требованиям графического терминала КАПРИ, не рассчитан на диалог с новым партнером (пультовой программой), появляющимся при таком подключении. Модифицировать его оказалось затруднительно, и поэтому был разработан новый драйвер. Подключение линии связи к пультовому каналу позволяет вести начальную загрузку в терминалах команд пультового терминала; при этом загрузчик, запущенный в ПЗУ, не нужен.

\* Монитор разработан сотрудником ИАЭ им. И. В. Курчатова В. В. Муравьевым.

Итак, программное обеспечение, поддерживающее подключение графического терминала КАПРИ к СМ4, состоит из следующих основных частей:

простой системы загрузки небольших программ обеих Э60 командами пульта терминала. Для загрузки больших программ используется промежуточный загрузчик. Загружаемая программа компонуется обычным порядком;

драйвера, обеспечивающего содержательный обмен параллельно с отладкой программы в любой ЭВМ по командам пульта терминала, причем отладка автоматически синхронизируется с обменом;

отладчика, с помощью которого с любого терминала СМ4 можно отлаживать программы в любой Э60 графического терминала. При этом отлаживаемую программу не нужно компоновать вместе с какой-либо программой-отладчиком. Команды отладчика такие же, как и у отладчика RSX11;

программ в Э60, обеспечивающих обмен с СМ4 и отладку Э60 1-го уровня. Их объем (вместе с монитором параллельных процессов) в Э60 2-го уровня — 2К слов, а в Э60 1-го уровня — 1,2К слов, что существенно меньше, чем при размещении RSX11-S (14К слов).

Программное обеспечение базового уровня, размещенное в Э60 1-го уровня, обеспечивает следующие возможности;

сегментацию дисплейного файла, т. е. создание, удаление и преобразование сегментов без остановки регенерации изображения с полной гарантией правильности структуры (доступ к дисплейному файлу помимо сегментов закрыт);

работу с физическими устройствами ввода (клавиатурами и планшетом);

работу с виртуальными устройствами ввода (локатором, световыми и планшетными клавиатурами, датчиком, циркулем, линейкой и т. д.) в режимах запроса и прерывания с разнообразными видами эхосопровождения.

Таких возможностей базового уровня вполне достаточно для эффективной реализации пакета ПБГ. Программное обеспечение, реализующее возможности пакета ПБГ, размещается в Э60 2-го уровня.

Описанная в работе структура связи двух микроЭВМ Э60 с СМ4, поддерживающие эту связь программы и программы базового уровня графического терминала КАПРИ могут быть с успехом использованы при разработке специализированного программного обеспечения (например, при разработке автоматизированных рабочих мест технолога или конструктора). Особенно полезно, по мнению разработчиков, достигнутые на базе УСМИ унификация аппаратуры связи и автоматическая синхронизация обмена и отладки, ведущиеся по одному физическому каналу.

Телефон: 972-40-64, Москва, Хухлаев Е. В.

## ЛИТЕРАТУРА

1. Адамов Е. О., Дукарский С. М., Корягин Д. А. Интегрированная машиностроительная система КАПРИ // Микропроцессорные средства и системы. — 1985. — № 4. — С. 53—56.
2. Графический интеллектуальный терминал «ГРАФИТ» АС-7060. — Техническое описание ОМ2.043. 700.ТО.
3. Волков Д. В., Однорядцева Е. В., Хухлаев Е. В. Инструментальные пакеты хранения данных и базовой графики. — Препринт. — М., 1986. — (ИПМ им. М. В. Келдыша, № 163).
4. Елинер Э. И., Клименко А. Д., Костылев Д. А. Локальная сеть на базе сетевого программного обеспечения АЛИСА // Микропроцессорные средства и системы. — 1986. — № 5. — С. 42—43.

5. Интерфейс последовательный к ЭВМ «Электроника 60» DLKC-2. Паспорт 5 P.036.211ПС. СКБ НПСО АН СССР.

6. Лопатин В. С., Юрочкин А. Г., Баранов Н. Д. Адаптер магистралей СМ ЭВМ и микроЭВМ «Электроника 60» // Микропроцессорные средства и системы. — 1985. — № 1. — С. 11—12.

7. Системное матобеспечение многомашинного комплекса автоматизации физического эксперимента / М. В. Бейлин, О. В. Вьюшин, А. Д. Клименко и др. — Препринт. — Новосибирск, 1982. — 23 с. — (ИЯФ СО АН СССР № 82—72).

Статья поступила 15 мая 1987.

## УВАЖАЕМАЯ РЕДАКЦИЯ!

Прочитал я в Вашем журнале письмо в редакцию от программиста Ю. В. Слуцкого («МП», 1987, № 5, с. 74, 85). Он поднимает очень важные вопросы о внедрении АСУ на средних предприятиях. Невозможно автоматизировать беспорядок в оперативном управлении и учете. А мы пытаемся.

«Принцип главного руководителя АСУ» практически нигде не применяется. Мало учат руководителей предприятий современным методам управления. Проблемы хозрасчета, подряда пытаются решать на средствах вычислительной техники при отсутствии стандартных пакетов, при наличии нескольких средних программистов.

Информация о работе других организаций у нас минимальна. Попытки пополнить знания по современным программным средствам и вычислительной технике на ВДНХ и в Центре информатики в Москве ни к чему не привели.

Центр информатики обслуживает школьников, студентов, аспирантов, предоставляя им персональные компьютеры для расчетов или игр. Консультации профессиональным пользователям Центр не может дать. Недостаток информации приводит к тому, что вычислительная техника, которую заказываем, не удовлетворяет требованиям той или иной области применения.

Необходимо создать в стране Центр информатики, который должен иметь всю серийно выпускаемую технику. На базе этого центра проводить обучение пользователей (показывая в работе все средства вычислительной техники), обзорные семинары по выбору средств ВТ (начиная с микро- и кончая большими ЭВМ, без личных симпатий разработчиков к своему семейству ЭВМ), по работе системного и стандартного математического обеспечения. Необходимо издавать журнал о всех средствах вычислительной техники: больших ЭВМ (типа ЕС), мини-ЭВМ, аппаратуре передачи данных, модемах и т. д. Там же можно было бы почерпнуть и знания по созданию АСУ предприятиями.

Я полагаю, что журналом «МП» начат большой серьезный разговор о проблемах внедрения средств ВТ, чтобы сложная дорога техника не превращалась в игрушку. Кстати, на ВДНХ на персональных компьютерах играют в компьютерные игры и только сиротливо стоящие таблички напоминают о разработанных пакетах.

С уважением  
Серединский Ю. С.

УДК 681.322.1

Е. А. Иванов

## СОВМЕСТИМОСТЬ ПЭВМ С ПОМОЩЬЮ АППАРАТНО-ПРОГРАММНЫХ СОПРОЦЕССОРОВ

В настоящее время в стране развиваются несколько семейств программно несовместимых ПЭВМ:

16-разрядные семейства «Электроника» («Электроника 85» и ДВК);

16-разрядные семейства ЕС, «Искра» и «Нейрон» (ЕС 1840, 1841, «Искра 1030», «Нейрон И9.66» и др.);

8-разрядные («Корвет», «Ириша», «Океан 240.2» и др.).

Для массового эффективного применения ПЭВМ каждого из семейств в различных областях народного хозяйства необходимы доступные пакеты прикладных программ общего и конкретного назначения. Средняя трудоемкость разработки одного пакета для ПЭВМ составляет 2—4 человеко-года, поэтому весьма важно обеспечить возможность «перекрестного» совместного использования уже готового программного продукта.

Поскольку строгого определения термина «совместимость» пока не существует, дадим определения некоторых используемых терминов.

**Программная совместимость** — возможность эксплуатации прикладной программы, разработанной для некоторой базовой модели (или семейства) ПЭВМ, на ПЭВМ другой модели (или семейства) — назовем ее «чужой» ПЭВМ.

**Программная совместимость на уровне операционной системы** — прикладная программа выполняется в среде одной и той же операционной системы, поставленной на разнородных ПЭВМ.

**Полная совместимость** — предполагает кроме программной совместимости на уровне операционной системы еще и совместимость на уровне контроллеров и устройств ввода-вывода.

**Совместимость «снизу вверх»** — программное обеспечение младшей модели переносимо на старшие модели ПЭВМ.

Один из способов обеспечения программной совместимости ПЭВМ различных семейств, следствием которой является резкое снижение затрат на разработку прикладных программ, является использование в составе базовой ПЭВМ аппаратно-программных сопроцессоров. Данный способ по сравнению с другими известными способами (программная эмуляция с помощью кросс-систем, полная физическая модель чужой ПЭВМ, использование «мобильных» операционных систем типа Юникс) позволяет получить для системы «ПЭВМ + сопроцессор» программную совместимость при относительно небольших дополнительных затратах.

**Сопроцессором (СП)** называется вспомогательный процессор, служащий для установки на базовой ПЭВМ операционной системы чужой ПЭВМ. Конструктивно сопроцессор выполняется в виде одно- или двухплатного модуля, устанавливаемого в расширительный отсек базовой ПЭВМ.

Для того чтобы прикладная программа чужой ПЭВМ выполнялась в системе «базовая ПЭВМ + сопроцессор», необходимо предоставить ей эквивалентную вычислительную среду (рис. 1), т. е. те аппаратные и про-

граммные средства, которые «окружают» прикладную программу (ПП) и используются при ее выполнении — операционная система (ОС), драйверы устройств ввода-вывода (ДУВВ), контроллеры (КУВВ) и, наконец, сами устройства ввода-вывода (УВВ). На программном уровне эквивалентности вычислительной среды достигается путем установки с помощью сопроцессора ОС того же типа, который используется в чужой ПЭВМ (адаптируемая ОС). Поскольку характеристики УВВ и особенно контроллеров базовой и чужой ПЭВМ, как правило, резко отличаются, то на аппаратном уровне возникает проблема эмуляции УВВ чужой ПЭВМ.

Эта проблема может решаться несколькими способами, каждый из которых зависит от выбранной архитектуры аппаратно-программных средств СП.

В состав программных средств СП включаются: программа связи двух ОС (базовой и адаптируемой), выполняющая функции эмуляции УВВ чужой ПЭВМ. При эмуляции используются драйверы УВВ базовой ОС;

контроллеры УВВ чужой ПЭВМ, а в состав программных средств — драйверы УВВ чужой ПЭВМ.

Перерабатываются драйверы УВВ адаптируемой ОС для использования УВВ базовой ПЭВМ. Базовая или адаптируемая ОС эксплуатируется в составе ПЭВМ только поочередно.

Каждый из этих способов эмуляции УВВ характеризуется определенным объемом переработок аппаратных и программных средств, а также достигаемым уровнем потребительских характеристик ПП (время ее выполнения в системе «ПЭВМ + сопроцессор»), однако общая закономерность следующая: при возрастании объема аппаратных средств сопроцессора уменьшается объем перерабатываемых программных средств базовой или адаптируемой ОС, время выполнения ПП приближается к времени ее выполнения на «своей» ПЭВМ.

Большинство пакетов ПП, имеющихся на рынке, взаимодействует с УВВ через ОС. Для них может быть обеспечена программная совместимость на уровне ОС с помощью СП. Однако часть прикладных программ

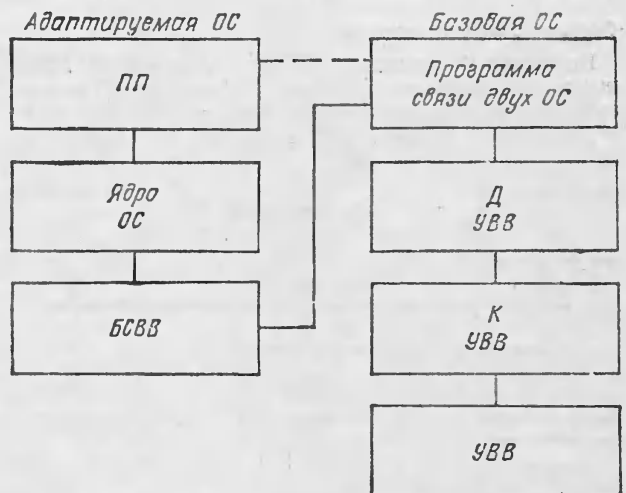


Рис. 1. Взаимодействие прикладной программы с УВВ базовой ПЭВМ: БСВВ — базовая система ввода-вывода

взаимодействует с УВВ минуя ОС. Дело в том, что взаимодействие через ОС увеличивает общее время выполнения программы, что нежелательно для некоторых специальных применений (например, высококачественная графика и др.). Наконец, некоторые интегрированные пакеты (Lotus 1—2—3, Symphony и др.) также не используют стандартную ОС для взаимодействия с УВВ, а содержат специальную среду внутри пакета. Все эти пакеты ПП не будут выполняться в системе «ПЭВМ + сопроцессор», поэтому необходима их доработка. Однако в ряде случаев (особенно если имеется программная документация) стоимость доработки может быть во много раз ниже, чем разработки для базовой ПЭВМ «с нуля» пакета ПП с такими же характеристиками.

Рассмотрим общие требования, предъявляемые к архитектуре аппаратных средств сопроцессора со стороны ПП.

Очевидно, что в сопроцессоре должен использоваться микропроцессор того же типа, что и в чужой ПЭВМ, либо полностью совместимый с ним по архитектуре и системе команд.

Если в сопроцессоре имеется собственная память, то ее объем и организация должны быть по возможности такими же, как и в чужой ПЭВМ (имеется в виду максимальный объем памяти чужой ПЭВМ). Если собственная память в сопроцессоре отсутствует, а в качестве ее используется память базовой ПЭВМ, захватываемая в режиме прямого доступа, то объем этой памяти должен быть достаточным для размещения как адаптируемой ОС, так и наибольшего из имеющихся по объему пакетов ПП. Например, для установки ОС типа MS DOS объем памяти ПЭВМ должен быть не менее 512 Кбайт, так как только один интегрированный пакет Symphony требует для своего размещения 392 Кбайт памяти.

Чтобы максимально уменьшить время выполнения ПП, необходимо для связи сопроцессора с базовой ПЭВМ использовать параллельную системную магистраль (СМ) с высокой пропускной способностью, например «Общая шина». Обмен информацией между сопроцессором и центральным процессором базовой ПЭВМ должен происходить как в режиме прерывания, так и по готовности. Для ускорения передачи больших массивов информации (например, содержимого памяти видеоданных) возможен обмен в режиме прямого доступа к памяти.

Рассмотрим некоторые из известных архитектур систем «сопроцессор + ПЭВМ».

#### Система с общей памятью

Центральный процессор (ЦП) и сопроцессор (СП) имеют общую память ОЗУ (рис. 2), которую СП может захватить в режиме прямого доступа. ЦП при этом переключается в пассивное состояние. При включении

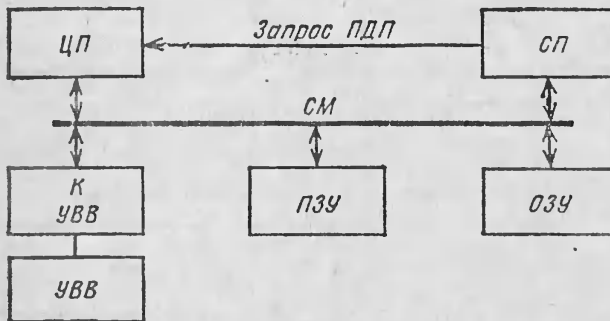


Рис. 2. Структура системы с общей памятью

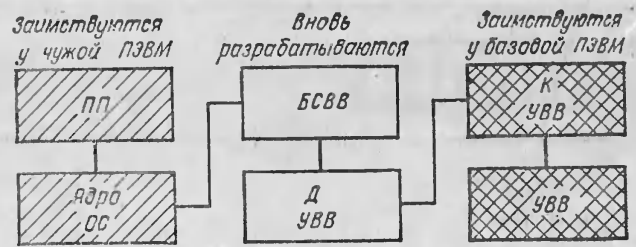


Рис. 3. Взаимодействие прикладной программы с УВВ в системе с общей памятью

питания системы всегда загружается базовая ОС, а затем анализируется, в какой ОС должна выполняться ПП, если в адаптируемой ОС, то происходит перезагрузка ОС (загружается с помощью НГМД адаптируемая ОС) и захват памяти СП в режиме прямого доступа.

Основным преимуществом данной архитектуры является минимальный объем аппаратных средств. Однако система имеет и ряд существенных недостатков:

невозможность одновременной эксплуатации двух ОС и в связи с этим необходимость полной переработки БСВВ и драйверов УВВ, так как нельзя воспользоваться драйверами УВВ базовой ОС (рис. 3);

частая перезагрузка винчестерского НМД при переходе от базовой к адаптируемой ОС и обратно увеличивает время готовности до 1 ч и снижает надежность работы системы;

невозможность организации параллельной работы ЦП и СП при выполнении ПП снижает общее быстродействие системы;

объем памяти базовой ПЭВМ в некоторых случаях может быть недостаточным для установки сложных ОС и пакетов ПП.

#### Система «компьютер в компьютере»

Сопроцессор имеет собственную память (ОЗУ и ПЗУ), что несколько увеличивает объем аппаратных средств системы (рис. 4), но позволяет устранить недостатки системы с общей памятью. Для взаимодействия ПП с УВВ необходима доработка БСВВ, а также разработка программы-эмулятора УВВ, драйверы УВВ могут быть использованы от базовой ОС (рис. 5). БСВВ дорабатывается таким образом, что со стороны ядра ОС она остается неизменной, а изменяются подпрограммы обслуживания УВВ.

При выполнении ПП, загружаемой в память СП, последний берет на себя функции ЦП, а ЦП — функции процессора ввода-вывода.

Недостатками системы «компьютер в компьютере» являются:

невозможность эксплуатации без доработок ПП, взаимодействующих с УВВ минуя ОС;

увеличение времени выполнения ПП по сравнению с ее выполнением на своей ПЭВМ.

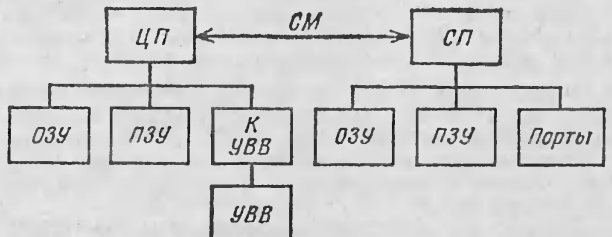


Рис. 4. Структура системы «компьютер в компьютере»

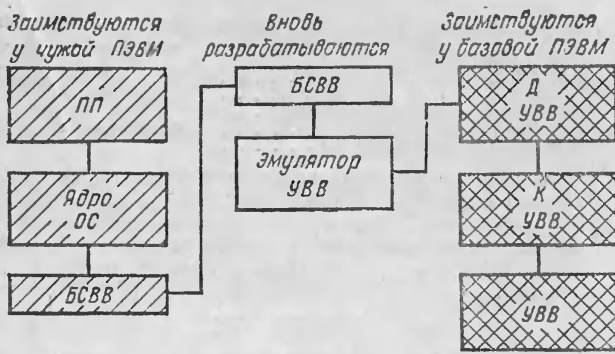


Рис. 5. Взаимодействие прикладной программы с УВВ для системы «компьютер в компьютере»

**Система «компьютер в компьютере с контроллерами УВВ»**

Кроме собственной памяти в состав СП включены контроллеры УВВ, эмулирующие аппаратными средствами контроллеры УВВ чужой ПЭВМ (рис. 6). В системе, таким образом, имеются два комплекса контроллеров:  $K_1...K_n$  — базовой ПЭВМ и  $K'_1...K'_n$  — чужой ПЭВМ, работающих на одни и те же УВВ. При работе ПП с контроллерами  $K'_1...K'_n$  контроллеры  $K_1...K_n$  электрически блокируются. Такая архитектура системы возможна в том случае, если ПЭВМ различаются характеристиками контроллеров УВВ, а характеристики самих УВВ — близки или совпадают. Это справедливо, например, для 16-разрядных профессиональных ПЭВМ. При установке ОС на базовую ПЭВМ ядро ОС, БСВВ и драйверы УВВ полностью заимствуются у чужой ПЭВМ и достигается полная программная совместимость.

Возможны также промежуточные варианты данной архитектуры СП, когда наиболее критичные с точки зрения быстродействия ПП контроллеры УВВ (например, видеоконтроллер) реализуются аппаратно, а остальные контроллеры УВВ отсутствуют. В этом случае драйверы эмулируемых аппаратно контроллеров УВВ заимствуются у ОС, а остальные драйверы — у базовой ОС ПЭВМ (рис. 7). Тогда необходима доработка БСВВ и разработка программы-эмулятора некоторых УВВ.

Таким образом, данный вариант архитектуры СП отличается наибольшим объемом аппаратных средств и наименьшим объемом переработок программных средств.

Некоторые сравнительные характеристики модели сопроцессоров, позволяющих реализовать рассмотренные архитектуры систем, приведены в таблице. Архи-

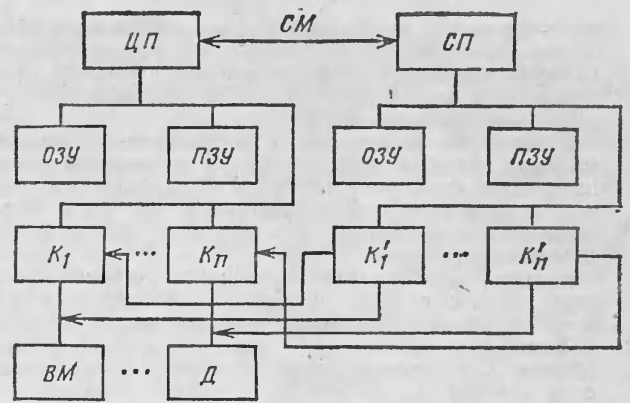


Рис. 6. Структура системы «компьютер в компьютере с контроллерами УВВ»

тектура с общей памятью может быть рекомендована для построения систем с минимальным объемом аппаратных средств, в особенности, если базовая ПЭВМ

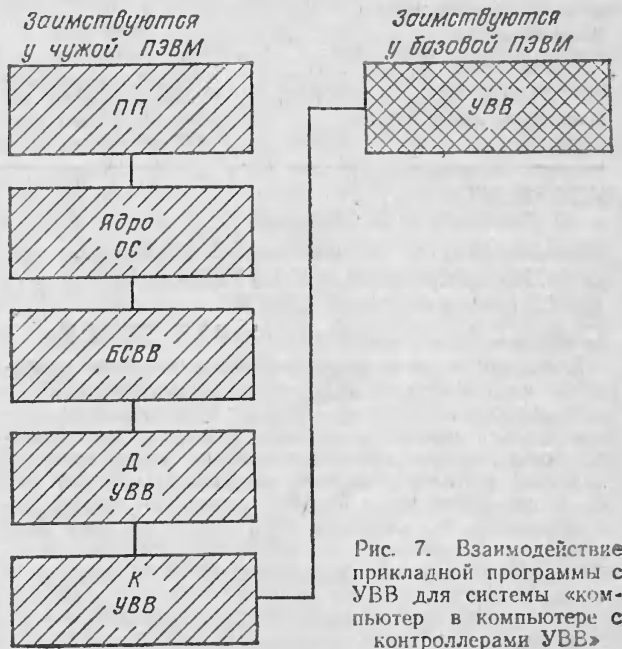


Рис. 7. Взаимодействие прикладной программы с УВВ для системы «компьютер в компьютере с контроллерами УВВ»

Архитектура систем	Модель СП	Тип базовой ПЭВМ	Типы адаптируемых ОС	Число микросхем в СП	Число и габариты печатных плат, мм	Примечание
С общей памятью Компьютер в компьютере	MC1686	ДВК2М «Электроника 85»	CP/M86	40	1 плата, 140×240	
	MC1701		CP/M86 MOC86	90		
Компьютер в компьютере с контроллерами УВВ	PC-Bridge	PC350	MS DOS	135	1 многослойная плата, 104×300	Аппаратно реализован видеоконтроллер IBM PC
	AT-Bridge	Микро-Vax	MS DOS		6 плат, отдельный конструктив с источником питания	Аппаратно реализованы все контроллеры УВВ IBM PC AT

разрабатывается заново, так как для оптимальной реализации системы могут потребоваться дополнительные магистрали связи с общей памятью базовой ПЭВМ. Эта архитектура характеризуется наибольшим объемом доработок программных средств.

Архитектура «компьютер в компьютере» позволяет получить при среднем уровне затрат на разработку аппаратных и программных средств СП приемлемый уровень программной совместимости и может быть рекомендована как основная для систем на базе серийно выпускаемых моделей ПЭВМ.

Наконец, использование архитектуры «компьютер в компьютере» с полным набором контроллеров УВВ вряд ли является целесообразным для систем на базе 16-разрядных ПЭВМ, так как объем аппаратных средств СП приближается к объему аппаратных средств самой базовой ПЭВМ.

Для 16-разрядных ПЭВМ возможны промежуточные варианты этой архитектуры (PC-Bridge), в которых аппаратно реализован только видеоконтроллер, являющийся наиболее критичным с точки зрения быстродействия.

Примером архитектуры «компьютер в компьютере» с аппаратно-программной реализацией видеоконтроллера служит сопроцессор БА 86М. Ниже приведены его основные характеристики.

Микропроцессор . . . . .	K1810BM86
Арифметический сопроцессор . . . . .	K1810BM87
Тактовая частота, МГц . . . . .	5
Типы адаптируемых ОС . . . . .	MS DOS, Альфа ДОС
Магистраль связи с ПЭВМ . . . . .	CP/M 86, МОС 86
	Модифицированная «Общая шина»

Режимы обмена информацией с ПЭВМ	По готовности, по прерыванию, ПДП
Объем, Кбайт:	
ОЗУ . . . . .	512
ПЗУ . . . . .	8...16
буферной памяти видеоданных . . . . .	16
Конструктивное исполнение . . . . .	Одноплатное
Потребляемая мощность, Вт . . . . .	18

Данный сопроцессор позволяет установить на ПЭВМ «Электроника 85» ОС типа MS DOS (Альфа ДОС) и CP/M 86 (МОС 86) и обеспечить программную совместимость на уровне ОС с ПЭВМ семейств ЕС, «Искра» и «Нейрон» при достаточно высоком быстродействии ПП.

#### ЛИТЕРАТУРА

1. Генюк Б. Л., Муренко Л. Л., Иванов Е. А., Красовский С. Я. Аппаратно-программные адаптеры операционных систем персональных ЭВМ // Микропроцессорные средства и системы.— 1986.— № 4.— С. 43.
2. Барышнев Е. Е., Хохлов М. М., Шабалин А. И. Совмещение операционных систем персональных компьютеров с различной системой команд в одном изделии // Микропроцессорные средства и системы.— 1986.— № 4.— С. 37.
3. Черемных С. В., Сенни Г. В. О совместимости персональных ЭВМ // Прикладная информатика: Сб. статей / Под ред. В. М. Савинкова.— М.: Финансы и статистика.— 1987.— Вып. 2 (13).
4. Philip A. Naecker. The AT-Bridge // Professional — 1987.— N 4.

Статья поступила 3 марта 1988

УДК 681.327.13

А. Ф. Манкевич, С. Д. Шайдорov

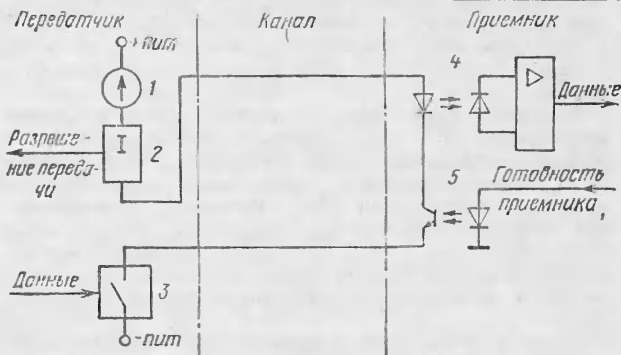
### РЕАЛИЗАЦИЯ ОТВЕТНОГО КАНАЛА В АСИНХРОННЫХ АДАПТЕРАХ ДИСТАНЦИОННОЙ СВЯЗИ С ИНТЕРФЕЙСОМ «ТОКОВАЯ ПЕТЛЯ»

Довольно широкое распространение получает применение последовательной передачи данных между ЭВМ и низкоскоростным периферийным оборудованием. Обмен ведется через асинхронные адаптеры дистанционной связи, реализующие дуплексную связь по стандартному протоколу битовой передачи: стартовый бит, 5...8 информационных бит, бит паритета (может отсутствовать), 1-2 стоповых бита. По стартовому биту определяется начало послышки данных. Частота следования битов в сообщении фиксирована и известна в обоих адаптерах (как со стороны ЭВМ, так и со стороны периферийного устройства, например терминала). Для передачи в линию применяется интерфейс RS232C или «токовая петля».

Ряд периферийных устройств не обеспечивают буферизацию данных, принятых от ЭВМ, поэтому возможна потеря данных — следует передача следующего байта, когда предыдущий еще не обработан (не отображен) и устройство не готово к приему информации. Появляется необходимость введения ответного канала, отвечающего за готовность адаптера принять следующий байт (символ).

Для интерфейса «токовая петля» роль ответного канала выполняет передающий канал адаптера. В адаптере приемный канал снабжается «разрывателем токовой петли» (применена оптопара), а передающий канал — датчиком «токовой петли» (обычный преобразователь тока в напряжение) (см. рисунок)

Когда нет передачи, линия находится в состоянии «активной токовой петли», в ней протекает ток 20 мА (60 мА). Условие готовности приемного канала адаптера — ток в линии улавливается датчиком «петли» в передающем канале другого адаптера. При наличии «токовой петли» формируется готовность передающего



Структурная схема канала передачи данных с подтверждением приема:

1 — источник (стабилизатор) тока; 2 — датчик тока; 3 — ключ (передающий разрыватель); 4 — оптический приемник; 5 — оптический ключ (разрыватель)

канала этого адаптера. Приемный канал второго адаптера (в середине приема стоповой последовательности) «разрывателем» выводит линию из состояния «активной токовой послышки» до считывания принятого байта устройством или ЭВМ. Приемный канал одного адаптера оповещает передающий канал другого адаптера, разрешая передачу данных.

Такой вариант ответного канала не требует дополнительной линии. Отключение датчика «петли» и «разрывателя» (при помощи перемычек на плате) позволяет стыковать адаптер напрямую — без ответного канала.

Были разработаны адаптеры с ответным каналом (в конструктиве СМ, и микроЭВМ «Электроника 60»), реализующим связь между компьютерами и полностью исключаящим потерю данных.

Телефон для справок: 77-01-93, Новосибирск

Статья поступила 25 мая 1987

\* Шевкопляс Б. В. Микропроцессорные структуры. Инженерные решения. — М.: Радио и связь, 1986.

УДК 681.3.06

А. П. Кулаичев

## ГРАФИЧЕСКИЙ БЕЙСИК ДЛЯ ДВК-3

Популярные вычислительные комплексы ДВК-3 содержат контроллер графического дисплея, позволяющий выдавать графическую информацию в растре 286×400 точек на экран черно-белого монитора. Для графического вывода на монитор и копирования экрана на принтер в программном обеспечении необходимы эффективные и простые в пользовании базовые средства. Однако существующие графические пакеты не вполне удовлетворяют этим требованиям. Поэтому произведено расширение языка БЕЙСИК-11 включением в интерпретатор подпрограмм графического вывода, вызов которых из БЕЙСИК-программы может производиться по оператору CALL или указанием имени подпрограммы с параметрами:

**MODE (x)** — установка экранной моды в зависимости от параметра  $x$  ( $x=0$  — только текстовый вывод;  $x=1$  — запрет любого вывода на экран;  $x=2$  — графический и текстовый вывод;  $x=3$  — только графический вывод);

**GCOLOR (x)** — установка текущего графического цвета ( $x=0$  — черный;  $x=1$  — белый) или цвета фона ( $x=128$  — черный;  $x=129$  — белый);

**WINDOW (x1, y1, x2, y2)** — установка графического окна, определенного координатами левого нижнего ( $x1, y1$ ) и правого верхнего ( $x2, y2$ ) углов. После установки окна графическая информация выдается только в пределах рамки окна, а начало координат устанавливается на левый нижний угол окна;

**CLG** — очистка графического экрана в рамках установленного окна принятым цветом фона;

**CLS** — очистка текстового экрана; **DUMP** — получение на принтере копии с графического экрана;

**MOVE (x, y)** — перемещение графического курсора в точку с координатами  $x, y$ . При установленном графическом окне координаты будут считываться от левого нижнего угла окна;

**DRAW (x, y)** — рисование линии от текущего положения графического курсора до точки с координатами  $x, y$  с перемещением графического курсора в эту точку. При установленном графическом окне координаты будут считываться от левого нижнего угла окна;

**CIRCLE (x, y, a, da, r)** — рисование дуги окружности, заданной координатами центра  $x, y$ , начальным углом  $a$ , дуговым углом  $da$  и радиусом  $r$ ;

```

10 DIM X(1,100),M1(1),M2(1),D(1) \ CLS \ PRINT"NUMBER OF POINTS"; \ INPUT NX
20 FOR I=0 TO NX \ X(0,I)=1 \ X(1,I)=SIN(I/5) \ NEXT I \ GOSUB 70
30 G=2 \ FOR I=0 TO NX \ X(1,I)=0-X(1,I) \ NEXT I \ GOSUB 190
40 G=1 \ FOR I=0 TO NX \ X(1,I)=X(1,I)/3*2 \ NEXT I \ GOSUB 190
50 G=0 \ FOR I=0 TO NX \ X(1,I)=X(1,I)/2 \ NEXT I \ GOSUB 190 \ DUMP \ STOP
70 GOSUB 140 \ IF M1(0)=M2(0) THEN RETURN
80 IF M1(1)=M2(1) THEN RETURN
90 MODE (2%) \ CLG \ IF G<3 THEN K%=50 \ GO TO 110
100 K%=10 \ M2(0)=X(0,1)-X(0,0)+M2(0) \ M2(1)=M2(1)*1.05
110 M1(0)=M1(0)+(M1(0)-M2(0))/K% \ M1(1)=M1(1)+(M1(1)-M2(1))/K%
120 J%=1 \ GOSUB 360 \ J%=0 \ GOSUB 360 \ GOSUB 190 \ RETURN
130 REM *****MAX-MIN
140 FOR J=0 TO 1 \ M1(J%)=X(J,0) \ M2(J%)=M1(J%) \ GOSUB 150 \ NEXT J \ RETURN
150 FOR I%=0 TO N% \ IF M1(J%)>X(J,I%) THEN M1(J%)=X(J,I%)
160 IF M2(KJ)<X(J,I%) THEN M2(J%)=X(J,I%)
170 NEXT I% \ RETURN
180 REM ***** POINT TO CHART
190 D1%=(0-M1(1))*D(1) \ W=(X(0,1)-X(0,0))/4*D(0) \ D%=W**1/3 \ E=D% \ D=E*3/4
200 FOR I%=0 TO N% \ P0%=X(0,I%)-M1(0))*D(0)
210 P1%=(X(1,I%)-M1(1))*D(1) \ ON G+1 GO TO 220,230,260,280
220 MOVE(P0%,P1%) \ DRAW(P0%,P1%) \ GO TO 340
230 A%=P0%+1 \ B%=P1%+1 \ MOVE(A%,B%)
240 B%=P1%-1 \ DRAW(A%,B%) \ A%=P0%-1 \ DRAW(A%,B%)
250 B%=P1%+1 \ DRAW(A%,B%) \ A%=P0%+1 \ DRAW(A%,B%) \ GO TO 340
260 IF I%=0 THEN MOVE(P0%,P1%) \ GO TO 340
270 DRAW(P0%,P1%) \ GO TO 340
280 IF P0%=0 THEN 340 \ P0%=P0%+D% \ A%=P0%-D% \ MOVE(A%,D1%)
290 A%=P0%-D%+W \ B%=D1%+D \ DRAW(A%,B%)
300 MOVE(P0%,D1%) \ DRAW(P0%,P1%) \ A%=P0%+W \ DRAW(A%,P1%) \ DRAW(A%,D1%)
310 DRAW(P0%,D1%) \ MOVE(P0%,P1%) \ A%=P0%+E \ B%=P1%+D \ DRAW(A%,B%)
320 A%=P0%+W+E \ DRAW(A%,B%) \ B%=D1%+D \ DRAW(A%,B%) \ A%=P0%+W
330 DRAW(A%,D1%) \ MOVE(A%,P1%) \ A%=P0%+W+E \ B%=P1%+D \ DRAW(A%,B%)
340 NEXT I% \ RETURN
350 REM ***** DRAW AXES
360 M0=M2(J%)-M1(J%) \ K0=1 \ L=265 \ IF J%=0 THEN L=380
370 IF M0>10 THEN K0=K0*10 \ M0=M0/10 \ GO TO 370
380 IF M0<1 THEN K0=K0/10 \ M0=M0*10 \ GO TO 370
390 M0=INT(M0) \ N0=1 \ IF M0<2 THEN N0=2 \ K0=K0/10 \ GO TO 420
400 IF M0<3 THEN N0=5 \ K0=K0/10 \ GO TO 420
410 IF M0>7 THEN N0=2
420 MOVE(0,0) \ DRAW(0,285%) \ DRAW(399%,285%) \ DRAW(399%,0) \ DRAW(0,0)
430 M9=INT((M1(J%)+N0*K0)/(N0*K0))*(N0*K0) \ D(J%)=L/(M2(J%)-M1(J%))
440 MOVE(0,0) \ IO%=M9-M1(J%))*D(J%) \ N=M9 \ I1=INT(INT(N0*K0*D(J%))/5)
450 I1=IO%-INT(IO%/I1)*I1 \ IF I1<2 THEN I1=INT(N0*K0*D(J%))/5
460 GOSUB 550 \ IF J%<>0 THEN 490
470 MOVE(IO%,285%) \ DRAW(IO%,275%) \ MOVE(IO%,0) \ DRAW(IO%,10%)
480 A%=IO%-3% \ MOVE(A%,16%) \ GO TO 510
490 MOVE(399%,IO%) \ DRAW(389%,IO%) \ MOVE(0,IO%) \ DRAW(10%,IO%)
500 A%=IO%+3% \ MOVE(16%,A%)
510 IF IO%>M1(J%) THEN A$=STR$(N) \ PRINT(A$)
520 I1=IO%+INT(N0*K0*D(J%))/5 \ N=N+N0*K0 \ IO%=IO%+INT(N0*K0*D(J%))
530 IF IO%<=L THEN 460 \ GOSUB 550 \ RETURN
540 REM ***** DRAW SCALE
550 FOR K%=1 TO 4 \ A%=I1 \ IF J%<>0 THEN 570
560 MOVE(A%,0) \ DRAW(A%,5%) \ MOVE(A%,285%) \ DRAW(A%,280%) \ GO TO 580
570 MOVE(0,A%) \ DRAW(5%,A%) \ MOVE(399%,A%) \ DRAW(394%,A%)
580 I1=I1+INT(N0*K0*D(J%))/5 \ NEXT K% \ RETURN

```

Программа 1

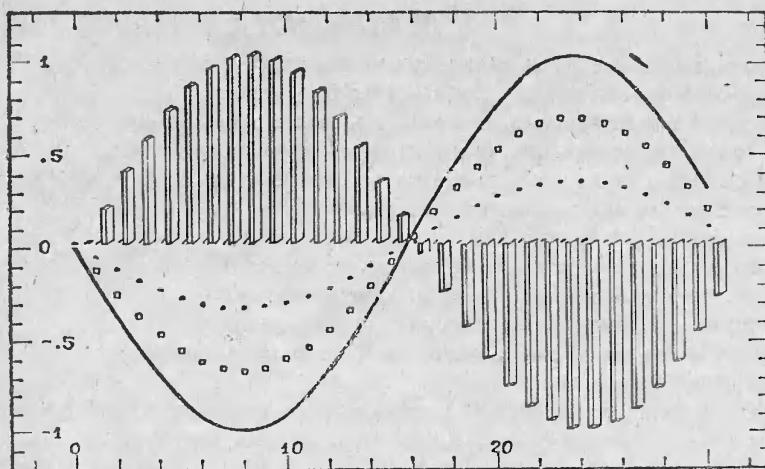


Рис. 1. Результат работы программы 1

ELLIPS ( $x, y, a, da, gx, gy$ ) — рисование дуги эллипса, заданной координатами центра  $x, y$ , начальным углом  $a$ , дуговым углом  $da$  и полуосями  $gx, gy$ ;

PRING ( $a\%$ ) — выдача на графический экран текста, представленного символьной переменной или символьной константой  $a\%$ . При установленном графическом окне и выходе тек-

```

10 CALL MODE(2%) \ CLG
20 DATA 0, 285, 399, 285, 399, 0, 0, 0, 399, 285, 399, 142, 0, 142, 200, 285
30 DATA 399, 142, 200, 0, 0, 142, 0, 285, 399, 0, 200, 0, 200, 285, 400, 0
40 READ A%, B% \ IF A% < 400% THEN DRAW(A%, B%) \ GO TO 40
50 DATA 50, 213, 17, 255, 0, 150, 213, 17, 255, 1, 250, 213, 17, 255, -1, 350, 213, 255, 17, 0
60 DATA 50, 71, 255, 255, 0, 150, 71, 63, 17, 0, 250, 71, 51, 85, 1, 350, 71, 17, 17, 0
70 DATA 100, 177, 51, 255, -1, 300, 177, 15, 255, 0, 300, 107, 51, 255, 0, 100, 107, 45, 15, 0
80 DATA 400, 0
90 READ A%, B% \ IF A% = 400 THEN 110
100 MOVE(A%, B%) \ READ A%, B%, C% \ FILL(A%, B%, C%) \ GO TO 90
110 END

```

Программа 2

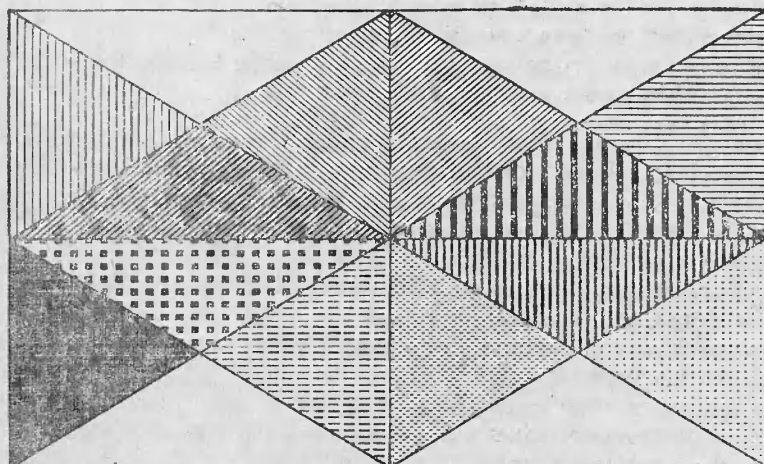


Рис. 2. Результат работы программы 2

ста за рамку окна остаток текста переносится на следующую строку;

FILL ( $a, b, c$ ) — заполнение штриховкой замкнутого контура, внутри

которого находится графический курсор, по заданным горизонтальному ( $a=1 \dots 255$ ) и вертикальному ( $b=1 \dots 255$ ) шаблонам. Они интерпре-

тируются как 8-разрядные двоичные коды, определяющие расположение черных и белых точек по горизонтали и заполненных и пустых строк по вертикали. Параметр  $c$  (область значений от  $-2$  до  $+2$ ) определяет значение сдвига горизонтального шаблона (влево, вправо) для очередной по вертикали строки штриховки.

Рассмотренные подпрограммы оформлены также в виде самостоятельной объектной библиотеки и могут вызываться из программ, написанных на языках компилирующего типа, таких как Фортран, Паскаль и др. Аналогичные графические средства разработаны и для цветного телевизионного контроллера типа АЭ.410.10.

В качестве иллюстрации применения рассмотренных графических средств приведены тексты двух БЕИ-СИК-программ с примерами выдачи результатов их работы на принтер по обращению к подпрограмме DUMP (рис. 1 и 2). Первая программа обеспечивает построение графиков таблиц по осям, выбором и маркировкой шкал. Вторая программа иллюстрирует возможности штриховки.

Телефон: 939-50-05, Москва  
Сообщение поступило 25 ноября 1987

### На книжной полке издательства «Наука»

В 1988—89 гг. ожидается выпуск книг:

по основам информатики — С. А. Абрамов и др. «Задачи по программированию»; В. К. Власов, Л. Н. Королев, А. Н. Сотников «Элементы информатики»; Н. И. Вьюкова, В. А. Галатенко, А. Б. Ходулев «Систематическое введение в программирование»; С. А. Абрамов, Е. В. Зима «Основы информатики»; Ж. Арсак «Программирование игр и головоломок (перевод с франц.)»;

по языкам программирования и их применению для решения задач — Ю. М. Баяковский, В. С. Штаркман, Н. И. Вьюкова «Программирование на Фортране-77»; А. А. Корнейчук, Г. И. Макаренко, А. В. Ракитский «Численные решения задач и язык Фортран»; В. М. Пентковский «Автокод Эльбрус», изд. 2-е; В. О. Сафонов «Языки и методы программирования в системе Эльбрус»; Г. Л. Семашко, А. И. Салтыков «Программирование на языке Паскаль».



## Уважаемая редакция!

Получившие уже сегодня колоссальное распространение компьютерные игры дают возможность резко ускорить развитие интеллекта детей еще больше, чем «развивающие игры» Никитиных, дети которых почти вдвое опережают в своем развитии среднестатистических.

Но, учитывая, что человечество уже не раз обжигалось как на отдельных индивидуумах, так и на целых сообществах, у которых интеллектуальное развитие опережало нравственное, предлагаю Вашему вниманию следующую идею:

вести в обучающие и развивающие компьютерные игры моральные нормы и нравственный выбор.

В реальной жизни наших, ограждаемых от сложностей, детей проблемы нравственного выбора встречаются редко. И если интеллектуальные упражнения способствуют развитию интеллекта, который все же есть функция наследственная, то становление нравственной личности вообще невозможно без душевной работы, «тренировки нравственности».

Как простейший пример идеи сценария могу предложить ситуационно-целевую игру (типа, например, «Пэкмен», «Колобок»), в которой по ходу движения к цели возникают морально-нравственные ветвления, отдаляющие ее достижение. При этом, конечно, наказание за «аморальность» и «безнравственность» должно

быть опосредованным, близким к тому, что мы имеем в жизни.

В игры для самых маленьких следует закладывать понятия о добре и зле, правде и лжи, любви и ненависти, радости и горе. Причем так, чтобы эти понятия вместе с соответствующими оценками усваивались и на подсознательном уровне.

Для работы в этом направлении, учитывая появление у нас первых мультимедийных игр, надо, не дожидаясь выпуска отечественных компьютеров с достаточным объемом памяти, создать группу специалистов высшей квалификации, включающую: сценаристов, режиссеров и художников, психологов и педагогов, программистов. В работе должны принять участие мастера мультипликации (кстати, задача в техническом плане близка к задаче компьютерной мультипликации).

Общезвестен отрицательный психофизиологический эффект компьютерных игр, не учитывающих особенности психологии и физиологии человека, особенно ребенка. Думаю, что и социальный эффект в связи с большим распространением компьютерных игр вскоре даст о себе знать.

Призываю Вас, пока в нашей стране компьютеризация в зародыше, поспособствовать претворению в жизнь этой пионерской идеи.

С уважением,

В. Корецкий, Москва; тел. 413-22-56 (д.)

УДК 681.03

Б. И. Гехман

## Диалоговая система на микроЭВМ «Электроника ДЗ-28» для подготовки графических иллюстраций

В рамках лабораторной информационной системы [1] разработана диалоговая система для подготовки цветных и черно-белых графических иллюстраций (рис. 1). МикроЭВМ «Электроника ДЗ-28» (с объемом ОЗУ 128 Кбайт) имеет пульт (клавиатура и цифровой дисплей) и встроенный кассетный накопитель на магнитной ленте (КНМЛ), на котором хранится системный загрузчик.

Программная реализация системы выполнена на языке высокого уровня МИКРОИНФ [4, 5] и занимает шесть 8-Кбайт сегментов [6]. Ядро системы — графический монитор (рис. 2), управляющий работой всех остальных программных модулей (кроме системного загрузчика, являющегося независимым модулем). После запуска системы графический монитор запрашивает инициатор о наличии

стандартных внешних устройств (ДАЦ и ПАЦ) — готовность их аппаратного интерфейса к обмену с микроЭВМ. При наличии ДАЦ обмен информацией с пользователем осуществляет монитор ДАЦ; монитор пульта в этом случае служит для сообщений о системных ошибках [1]. При отсутствии ДАЦ обмен информацией обеспечивает монитор пульта. Полный протокол работы системы, включая редактирование под управлением графического редактора, документирование через монитор ПАЦ. По указанию графического монитора тестовое изображение формируется иллюстратором. Вывод изображений через драйвер ЦВКУ на экран осуществляют драйверы обычной графики и гистограмм. Эти драйверы обрабатываются к модулям, обеспечивающим нанесение осей координат, их разметку, оцифровку, а также нанесение надписей (драйвер линейных осей и драйвер осей гистограммы). Знакогенератор формирует два вида символов (матрицы 7x5 и 8x6) в за-

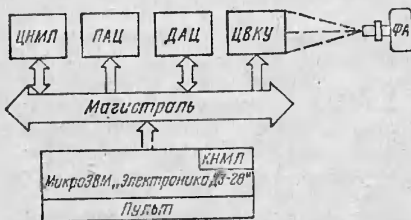


Рис. 1. Аппаратный состав системы подготовки графических иллюстраций.

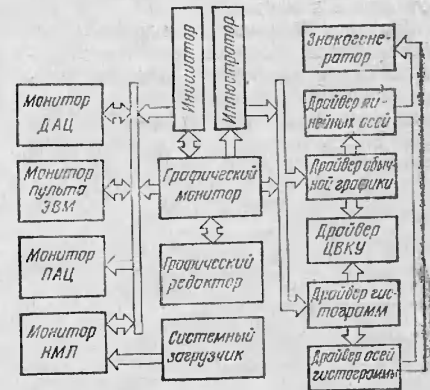


Рис. 2. Структура программного состава системы подготовки графических иллюстраций

данной ориентации в соответствии с кодировкой КОИТ-Н2.

Для «сжатия» графической информации, исходя из целей иллюстрации и логической общности экспериментального материала, на экране ЦВКУ можно выделять до четырех неперекрывающихся окон (полей). При этом в памяти микроЭВМ хранится необходимая информация только об одном (текущем) поле и только эта информация может редактироваться.

ЦНМЛ — цифровой накопитель на магнитной ленте (АП-5080) для хранения программной части системы; ПАЦ — алфавитно-цифровое печатающее устройство (ДЗМ-180) для документирования процесса подготовки иллюстраций; ДАЦ — алфавитно-цифровой дисплей (1БНЭ-00—013) для ввода необходимой информации (при отсутствии ДАЦ или его несправности эта информация может быть введена с пульта микроЭВМ); ЦВКУ — цифровое видеоконтрольное устройство (ВК51 Ц61) для отображения графической информации (256x256 точек). Сопряжение ЦВКУ с микроЭВМ обеспечивает привод [2], модернизированный для работы с эталонным телевизионным приемником типа ВК51 Ц61, обеспечивающим высококачественное цветное и черно-белое изображение [3]; ФА — фотоаппарат с встроенным фотозащитным экраном («Зенит 111» с объективом «Гелиос-44М»), жестко закрепленный относительно экрана ЦВКУ; Пульт — клавиатура и цифровой дисплей микроЭВМ.

Отредактированное изображение хранится в памяти привода ЦВКУ. При формировании иллюстрации на каждом поле обеспечиваются следующие функциональные возможности (некоторые из них — на вкладке):

изображение в виде графиков либо в виде гистограмм (до четырех на одном поле);

нанесение координатных осей и их разметка в линейном масштабе. По указанию любая ось может не наноситься, что позволяет представлять графический материал в произвольном (нелинейном) масштабе;

оцифровка координатных осей с учетом масштаба изображения;

нанесение заголовков (размерности) для каждой координатной оси. Заголовок оси ординат наносится вдоль оси (поворот шрифта на 90°);

представление графиков, гистограмм произвольной длины и статистических параметров;

задание цветов (градаций яркости для черно-белого изображения) каждого графика, координатных осей и фона;

автоматический выбор пиксела при изображении графиков в зависимости от длины и масштаба.

На рис. 3 приведен алгоритм работы системы подготовки иллюстраций. После загрузки и запуска система осуществляет инициацию стандартных внешних устройств — ДАЦ и ПАЦ. Вся информация в микроЭВМ вводится с экрана ДАЦ, а при его

отсутствии — с клавиатуры микроЭВМ с цифровой маркировкой каждого ввода на ее цифровом дисплее. Интерфейс пользователя, обеспечивающий ввод необходимой информации, реализован в виде четырех шаблонов, которые заполняются цифровой информацией (кроме шаблона для ввода заголовков по осям с экрана ДАЦ). Это обеспечивает независимость работоспособности системы от укомплектованности микроЭВМ стандартной периферией и упрощает контроль вводимой информации (система осуществляет контроль на основании формата чисел, заложенных в шаблон).

Данные для графического отображения представляются в виде матрицы, число строк которой определяет число графиков (гистограмм) на данном поле, а число столбцов — их длину.

Первый шаблон включает дату, класс, шифр и номер кадра на фотопленке (в соответствии с принятым стандартом [1]), а также размерность матрицы данных и запрос на формирование тестового изображения. По желанию пользователь может заказать такое изображение для заданной размерности матрицы. Для этого в шаблоне необходимо указать, на какое число полей дать пример. В качестве примера выдается детектированная синусоидальная функция, представляемая как в виде графика, так и в виде гистограммы.

При вводе матрицы данных с экрана ДАЦ проверяется соответствие введенной и заданной матрицы по длине (при этом используются все редакторские возможности ДАЦ). При ошибке ввода выдается сообщение и требование на повторный ввод. Реализация изображений графиков (гистограмм) разной длины на одном поле обеспечена за счет ввода значений, выходящих за предельные по оси ординат, которые задаются пользователем в шаблоне параметров. При выводе на экран сформированного изображения такие значения не отображаются.

В шаблон параметров входят: вид изображения (график или гистограмма), минимальные и максимальные значения по осям абсцисс и ординат для их оцифровки (при равенстве минимального и максимального значений разметка соответствующей оси не производится), максимальное число полей на экране ЦВКУ, текущий номер поля, цвет (градации яркости для черно-белого изображения) фона, осей и каждого графика (гистограммы). Указанием системы, что строка матрицы данных — статистический параметр для предыдущего графика, является задание ее цвета путем прибавления к его значению константы.

Заголовки по осям координат с клавиатуры микроЭВМ вводятся в виде десятичных значений кодов соот-

ветствующих символов в кодировке КОИТ-Н2.

После вывода изображения на указательное поле экрана ЦВКУ пользователь оценивает его и принимает решение: редактировать изображение, формировать следующее или осуществлять съемку на фотопленку. При редактировании (изменение заголовков, набора цветов, разметки осей и т. д.) может возникнуть потребность стереть изображение на каком-либо поле и после редактирования вывести его на экран. Возможности редактирования отображены на рис. 3. С ДАЦ редактирование упрощено: пользователь осуществляет изменения в шаблонах, хранящихся в памяти ДАЦ, и их ввод в микроЭВМ.

Практическая эксплуатация системы для подготовки иллюстраций результатов экспериментальных исследований по физиологии слуха показала ее надежность и эффективность при наличии широких функциональных возможностей. Высокая скорость освоения и простота работы с системой неподготовленных пользователей (физиологов, психологов, инженеров), гарантированное качество получаемых иллюстраций подтверждают эффективность принятых технических и программных решений.

Телефон: 218-43-59, Ленинград

#### ЛИТЕРАТУРА

1. Гехман Б. И. Двухуровневая информационная система для экспериментальных исследований // Электронная техника. Сер. «Экономика и системы управления». — 1987. — Вып. 1 (62). — С. 59—62.
2. Байдаков Е. Л., Яковлев А. В. Графический цветной дисплей и пакет графических программ для микроЭВМ «Электроника ДЗ-28» / В сб.: Алгоритмы и математическое обеспечение для физических задач. — Л.: ФТИ АН СССР, 1987.
3. Устройства видеоконтрольные ВК51 Ц61, ВК42 Ц61: Техническое описание и инструкция по эксплуатации.
4. Гехман Б. И., Равкин И. А., Темов В. Л. Кросс-система для микроЭВМ «Электроника ДЗ-28» // Электронная техника. Сер. «Экономика и системы управления». — 1984. — Вып. 3 (52). — С. 43—45.
5. Темов В. Л., Гехман Б. И., Равкин И. А. Технологический комплекс производства программ для микроЭВМ «Электроника ДЗ-28» // Автоматизация производства систем программирования: Тез. докл. III Всесоюзной конференции. — Таллин, 1986. — С. 177—179.
6. Устройство специализированное управляющее вычислительное «Электроника ДЗ-28». Справочник программиста.

Статья поступила 12 августа 1987

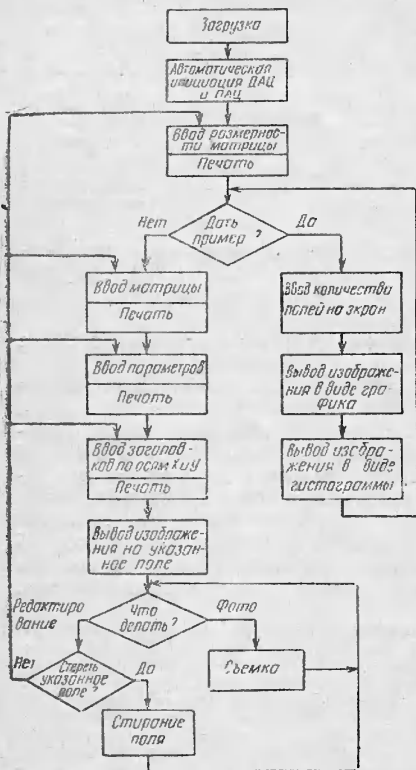


Рис. 3. Алгоритм работы системы подготовки графических иллюстраций

## ГРАФИЧЕСКИЙ АДАПТЕР ДЛЯ ППЭВМ ЕС 1840

Профессиональная персональная ЭВМ (ППЭВМ) ЕС 1840 — универсальная микрокомпьютерная система для решения широкого круга научно-технических, экономических и административных задач [1]. В базовый состав ЕС 1840 включен только монохромный алфавитно-цифровой дисплей. Для организации рабочей станции широкого назначения необходимы дополнительные аппаратные средства для представления графической информации на растре с размерами не менее  $640 \times 480$  точек (отображаемых с частотой регенерации 60 Гц) и числом цветов не менее 16, выбираемых из возможной палитры в 64 цвета и более (рис. 1).

Широко используемый в настоящее время принцип растрового сканирования основывается на преобразовании графических примитивов в совокупность элементов отображения (ЭО), которые записываются и хранятся в ЗУ большой емкости [2]. Такое ЗУ (кадровый буфер, память регенерации изображения или видеопамять) образуется из нескольких растровых плоскостей, размеры которых равны или превышают формат изображения на экране. «Глубина» ЭО, определяемая количеством плоскостей видеопамати, задает число одновременно отображаемых цветовых оттенков.

При генерации ЭО могут использоваться универсальные процессорные БИС (упрощается аппаратная реализация) и специальные графические процессоры (увеличивается скорость преобразования примитивов в растровую форму [3]).

Системы с большим числом растровых плоскостей дороги и громоздки. Часто необходимо иметь несколько цветных оттенков в одном рисунке, чтобы изменять цветную палитру от картинке к картинке без трансформации записанной в видеопамати графической информации. Для этого используется видеопроцессор, обрабатывающий и преобразующий коды ЭО в сигналы управления яркостью по трем цветовым составляющим с помощью видеотаблицы цветности и цифроаналоговых преобразователей. Время доступа при считывании на монитор не должно превышать времени вывода одного ЭО, поэтому видеотаблица — быстродействующее

четырьмя одновременно отображаемыми цветами; EGA — форматом  $640 \times 350$  точек с 16 цветами, выбираемыми из палитры в 64 цвета. Формирование ЭО в CGA и EGA осуществляется программно с помощью базовой системы ввода-вывода (BIOS).

Наибольшим быстродействием обладает разработанный для семейства персональных ЭВМ Personal System 2 профессиональный графический адаптер PGA, обеспечивающий формат раstra  $640 \times 480$  с числом одновременно отображаемых цветов 256 из программируемой палитры в 262144 цвета. Высокие технические характеристики и большие функциональные возможности адаптера достигнуты благодаря использованию заказных СБИС [5].

Разработанный для ППЭВМ ЕС 1840 графический адаптер (рис. 2) формирует на экране цветного монитора  $768 \times 512$  точек с прогрессивной разверткой и частотой регенерации 60 Гц. Картинка имеет 16 основных цветов, выбираемых из программируемой видеотаблицы с 4096 цветами.

Преобразование графических примитивов (отрезок, дуга, символ, область заполнения) в растровую форму выполняется программными средствами на центральном процессоре (ЦП) ППЭВМ с помощью последовательной цепочки операций чтение-модификация-запись над данными видеопамати [4]. В разработанном адаптере аппаратными средствами реализуется изменение ЭО в одном цикле видеопамати (увеличивается скорость построения изображения; среднее время вывода ЭО при построении отрезка — 80 мкс); режим записи восьми ЭО, расположенных рядом в строке, и стирание (заливка фоном) экрана в течение одного кадра; обеспечивается возможность выборочной записи индекса цвета ЭО в отдельные растровые плоскости видеопамати и просмотр плоскостей на экране.

Графический адаптер состоит из двух электронных модулей: видеопамати и видеопроцессора, связанных графической шиной, использующей свободные сигнальные линии системной шины ППЭВМ.

Функциональные узлы модуля видеопамати: буфер изображения (БИ), адресные счетчики и формирователи, сдвиговые регистры, синхронизация обмена и управления.

Буфер изображения (256 Кбайт) хранит изображение в четырех плоскостях форматом  $1024 \times 512$  (каждая плоскость на восьми БИС ОЗУ динамического типа K565PY5).

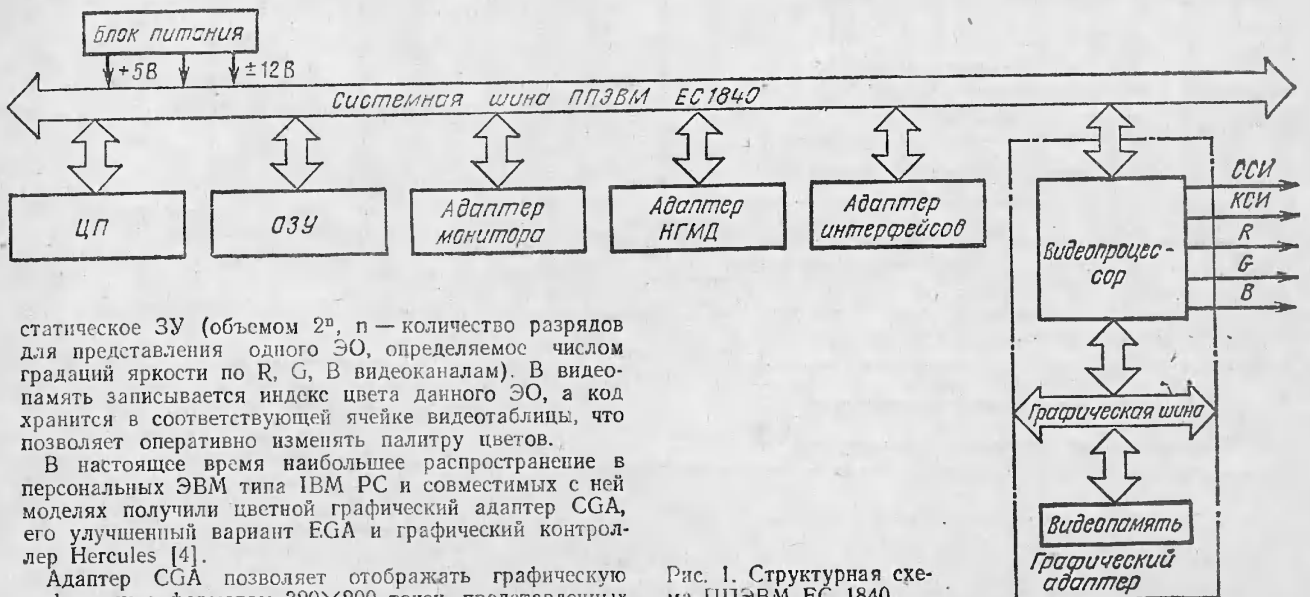


Рис. 1. Структурная схема ППЭВМ ЕС 1840

статическое ЗУ (объемом  $2^n$ ,  $n$  — количество разрядов для представления одного ЭО, определяемое числом градаций яркости по R, G, B видеоканалам). В видеопамати записывается индекс цвета данного ЭО, а код хранится в соответствующей ячейке видеотаблицы, что позволяет оперативно изменять палитру цветов.

В настоящее время наибольшее распространение в персональных ЭВМ типа IBM PC и совместимых с ней моделях получили цветной графический адаптер CGA, его улучшенный вариант EGA и графический контроллер Hercules [4].

Адаптер CGA позволяет отображать графическую информацию форматом  $320 \times 200$  точек, представленных

Один порт двухпортового ЗУ модифицирует память и связан с ЦП, второй читает информацию. Обмен данными с ЦП ЗУ происходит в обычном режиме чтение-запись, асинхронно — во время «свободных» циклов БИ (во время обратного хода по строке и кадру, за исключением интервала времени строчной синхронизации, в период которого происходит регенерация динамического ОЗУ). Страничный режим считывания на мониторе повышает быстродействие памяти.

Адресные счетчики (АС) циклически просматривают БИ с частотой регенерации изображения и выработывают адреса регенерации, необходимые для восстановления информации, содержащейся в ОЗУ динамического типа. Адреса памяти генерируются синхронно с координатами раstra по сигналам строчной (ССИ), кадровой (КСИ) и гасящей (ГСИ) синхронизаций, поступающим от контроллера ЭЛТ. Счетчики, построенные на микросхемах К555ИЕ7, функционально разделены на горизонтальный и вертикальный (текущий адрес восьми ЭО). Счетчик регенерации поддерживает работоспособность ОЗУ динамического типа.

Адресные формирователи (АФ), реализованные на ИС К555КП11, выбирают источник адреса БИ и преобразуют его в мультиплексированный адресный код, необходимый для правильного функционирования БИС ОЗУ К565РУ5.

Сдвиговые регистры (СР), построенные на четырех микросхемах К531ИР24, передают видеoinформацию (ДВ0...ДВ3) по сигнальным линиям графической шины на модуль видеопроцессора. На синхронизирующий вход поступают тактовые импульсы (ТИ).

Узел управления (УУ) координирует работу всех узлов модуля видеопамати, вырабатывая сигналы управления БИ (RAS, CAS) адресными счетчиками (ПЕР), адресными формирователями (МУЛ), сдвиговыми регистрами (ЗАГР-СДВ), узлом синхронизации обмена (ЦП, WE).

Узел синхронизации обмена (УСО) функционально можно разделить на блоки записи и чтения данных. При обмене с БИ процессор устанавливает

адрес памяти, режим работы и вырабатывает запрос на обращение (ЗЦП) к БИ; при записи — индекс цвета ЭО и маску, предназначенную для выборки в плоскости БИ; при чтении считывает данные по сигналу завершения цикла чтения (СЦП).

Функциональные узлы модуля видеопроцессора: селектор плоскостей БИ, видеотаблица, цифроаналоговые преобразователи, контроллер ЭЛТ, модуль сопряжения с ЦП.

Селектор плоскостей (СП) имеет регистр маски отображения, программируемый от ЦП, и коммутатор видеoinформации.

Видеотаблица (ВТ) — двухпортовая память: один порт программирует палитру цветов и подключается к ЦП, второй получает видеoinформацию с выхода селектора плоскостей, которая программируется процессором во время обратного хода по кадру.

Данные с выхода цветности (по трем 4-разрядным каналам) поступают на вход цифроаналогового преобразователя (КМ1118ПА1), формирующего уровни яркости R, G, B видеоканалов.

Контроллер ЭЛТ (КЭЛТ) вырабатывает сигналы строчной, кадровой и гасящей синхронизаций для монитора и управления адресными счетчиками. Примененный программируемый КЭЛТ (БИС СМ607) повышает гибкость адаптера.

Узел сопряжений (УС) осуществляет взаимодействие между ЦП и функциональными узлами. Все используемые в адаптере порты (регистры) расположены в адресной области IE0...IEF.

Программное обеспечение графического адаптера состоит из двух частей: программного монитора и библиотеки подпрограмм для языка Паскаль. Программный монитор — резидентная и реентерабельная программа, расширяющая функции базовой системы ввода-вывода ЕС 1840. Обращение возможно непосредственно из программ, написанных на ассемблере, с помощью команды INT 10h. Код функции больше 15 записывается в регистр ah ЦП, КМ1810ВМ86 (регистры al, sh, dx передают параметры).

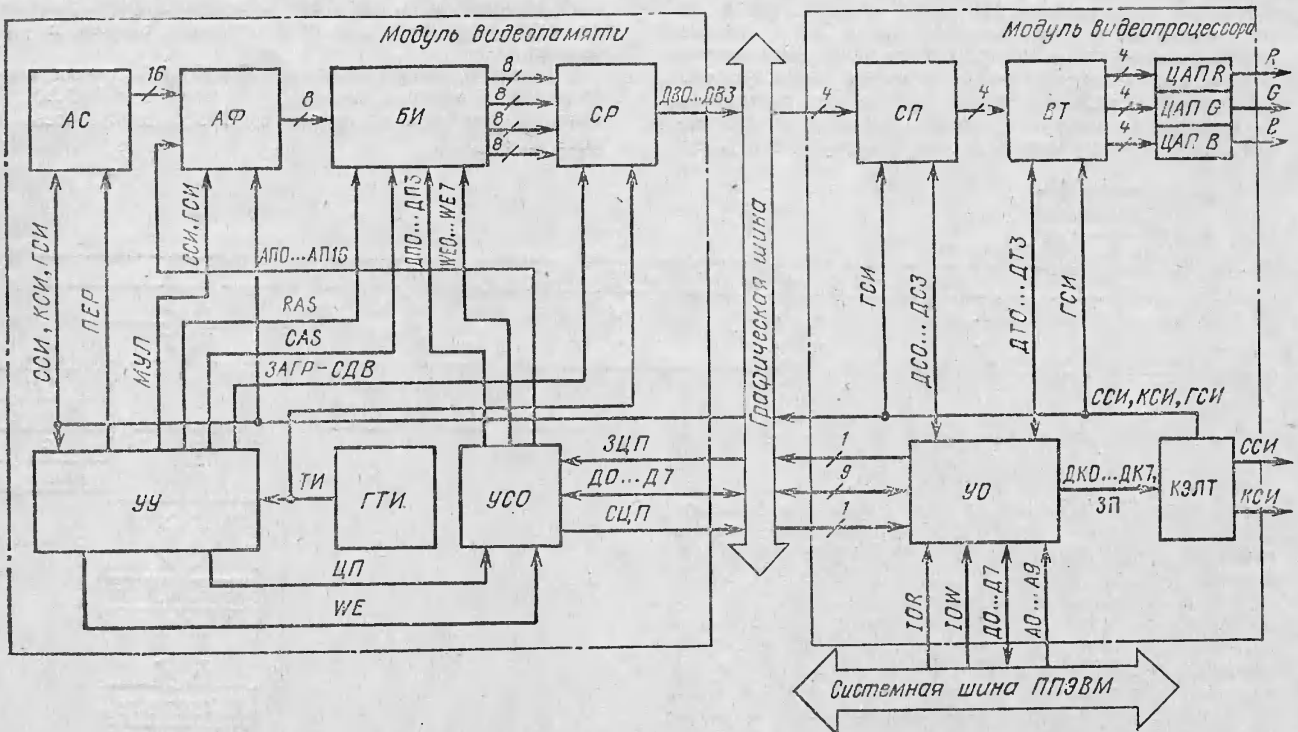


Рис. 2. Функциональная схема графического адаптера

Существующая версия ПО реализована для ОС MS DOS и MS PASCAL.

Графический адаптер выполнен на двух платах площадью 200×240 мм, имеющих 135 контактных разъемов СМП 34-135Р-В для подключения к системной шине.

Питание модулей осуществляется от источника базового блока ППЭВМ; для функционирования преобразователей уровня и ЦАП используется дополнительный источник питания — 5,2 В мощностью 5 Вт.

Адрес: 220605, Минск-72, ул. Сурганова, 6, ИТК АН БССР; тел. 39-59-85

### ЛИТЕРАТУРА

1. Пыхтин В. Я. ЕС 1840 — базовая персональная ЭВМ Единой системы // Микропроцессорные средства и системы. — 1986. — № 4. — С. 15—16.
2. Фоли Дж., Дэи А. Основы интерактивной машинной графики. — М.: Мир, 1985. — Т. 2. — С. 159—162.
3. Палташев Т. Т., Стародубцев Э. В. Управление формированием изображения в растровых графических терминалах. — М.: Зарубежная радиоэлектроника. — 1985. — № 10. — С. 74—75.
4. Williams T. Graphics controller boards bring speed and resolutions to bus architecture // Computer Design. — March. — 1986. — Vol. 25. — N 5. — P. 40—41.
5. Graphics boards // Computer Design. April. — 1987. — Vol. 26. — N 8. — P. 100—110.
6. Безобразов В. С., Димов В. А. Контроллер графического дисплея // Микропроцессорные средства и системы. — 1986. — № 3. — С. 50—52.

Статья поступила 17 июня 1987,

УДК 681.32

Н. Г. Големин, А. А. Званцев

## РЕГЕНЕРАТИВНЫЙ ГРАФИЧЕСКИЙ ДИСПЛЕЙ

Недорогие вычислительно-измерительные системы (ВИС), имеющие табло, цифровые дисплеи и транспаранты, не позволяют представить данные графиче-

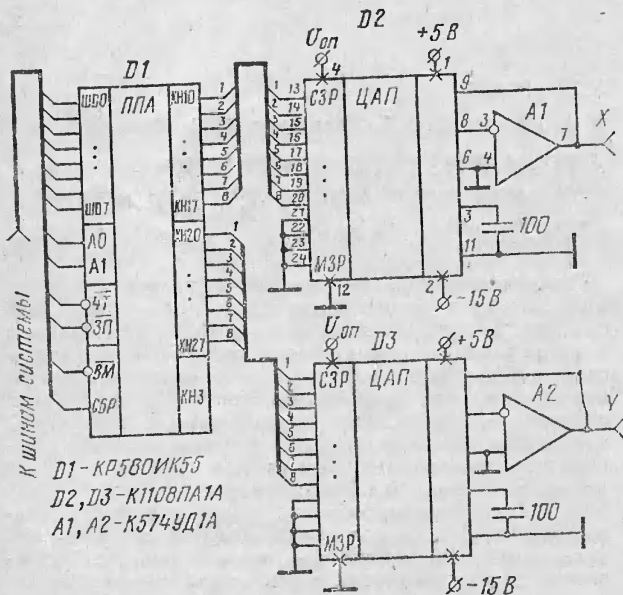


Рис. 1. Структурная схема регенеративного графического дисплея

ски, т. е. наиболее информативным среди всех способов оценки результатов эксперимента.

В простейшем случае в систему вводят осциллограф и два цифроаналоговых преобразователя (ЦАП), подключенных к портам вывода. Такой графический дисплей (рис. 1) называется регенеративным\*. Электронный луч, управляемый напряжением с ЦАП, способен вычерчивать на экране любые графические символы вплоть до алфавитно-цифровых. Использовать такое средство графического представления данных в ВИС возможно для систем с ОЗУ емкостью больше 1К байта (так как графический файл расположен в памяти системы) и высоким быстродействием (300·10<sup>3</sup>... 500·10<sup>3</sup> операций/с типа регистр-регистр). Отметим гибкость такого дисплея: например, многостраничный режим реализуется без увеличения аппаратных затрат лишь программными средствами.

Развитие этой идеи приводит к осциллографическому дисплею, использующему буферную память для хранения отображаемой картины\*, как в обычных растровых ТВ-дисплеях. Преимуществом осциллографического дисплея (рис. 2): простота аппаратного и программного сопряжения при высоких качественных показателях, отсутствие длительных промежутков обратного хода по строке и кадру, доступ к любой зоне в произвольный момент времени.

Сопряжение с системой осуществляется через три порта ввода-вывода, определяющих координаты точки и режим работы. Точка записывается или удаляется четырьмя циклами вывода данных: в течение двух первых — запись координат X, Y; в третьем — разрешение записи; в четвертом — отмена записи, перевод дисплея в режим воспроизведения.

В схеме дисплея (см. рис. 3) программируемый параллельный адаптер КР5801К55 осуществляет связь с системой, есть два ЦАП К1108ПА1А; два контроллера КР580ВТ42; четыре микросхемы ОЗУ К565РУ1А (4К). Остальные микросхемы выполняют функции генератора тактовых импульсов К155ЛН1, регистра хранения управляющих сигналов К155ТМ2, устройства тактового питания микросхем ОЗУ К155ЛЛ2. Циклы МП-системы и дисплея фазированы перезаписью состояния управляющих сигналов шины канала С в регистр хранения на двух D-триггерах.

Принцип работы дисплея прост. Яркостью луча в осциллографах управляют подачей импульса по входу Z. Желательно использовать подсвечивающий импульс,

\* Хоровид П., Хилл У. Искусство схемотехники. — М.: Мир, 1986, т. 2, с. 126.

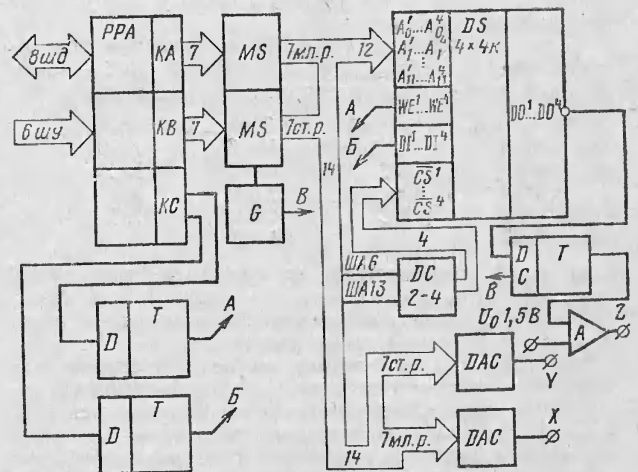


Рис. 2. Структурная схема дисплея с буферным ОЗУ

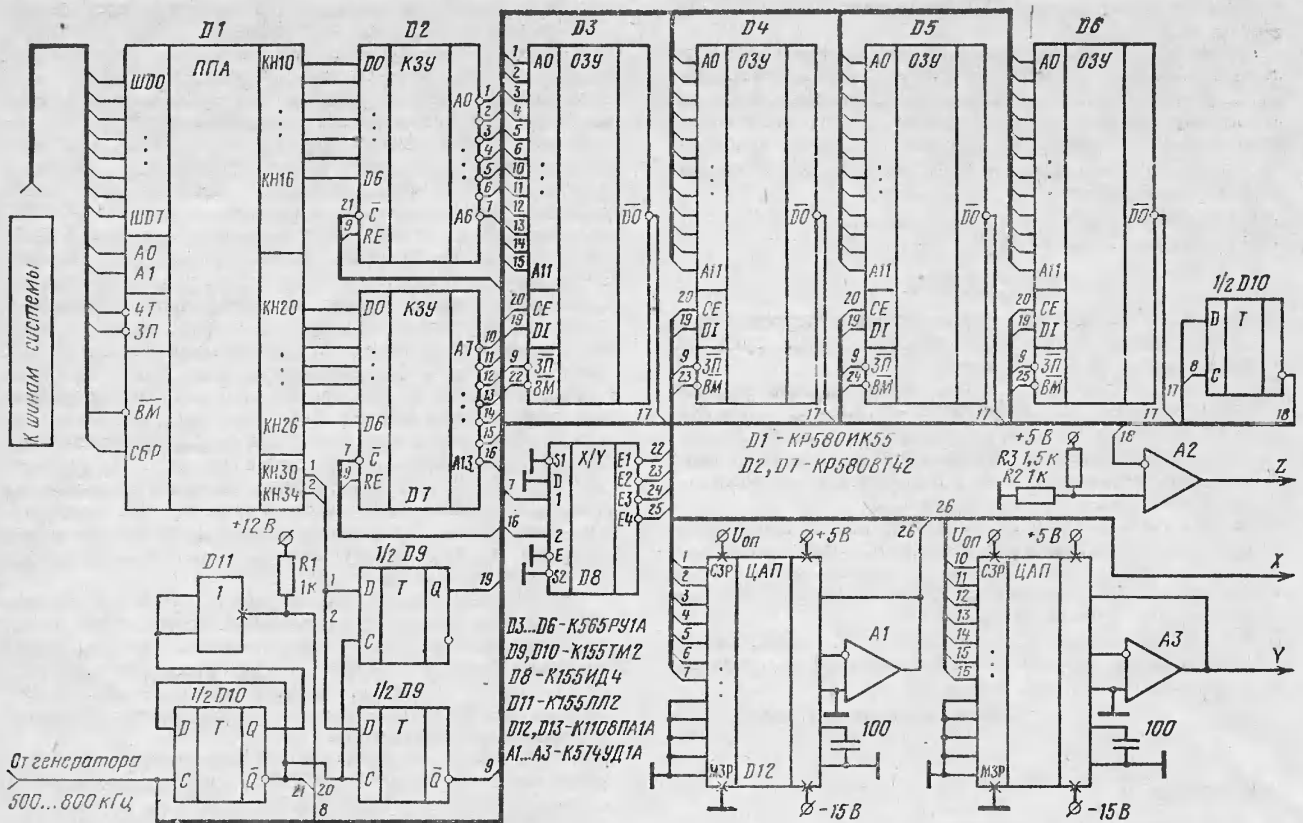


Рис. 3. Принципиальная схема дисплея

генерируемый всякий раз, когда встречается единица среди массива данных в ЗУ. Точки раstra находятсЯ во взаимно однозначном соответствии с координатами ячеек памяти. Поэтому любой графический символ синтезируется посредством записи единиц в ячейки ЗУ. Координаты ячеек обеспечивают возникновение именно этого символа. Растр дисплея содержит 16364 точек (128 строк  $\times$  128 точек в строке).

Это определяется емкостью ЗУ дисплея, которую увеличивать не имеет смысла из-за ограниченных полос пропускания каналов X и особенно Z ( $\leq 1$  МГц). Это не позволяет увеличить тактовую частоту для устранения мерцаний изображения.

Точка записывается, стирается следующими подпрограммами (ассемблер МП КР580ВМ80, NС — номер порта, в который выводится сигнал управления).

Запись 1 (точки)		Запись 0 (стирание)	
MVI	A.11	MVI	A.10
OUT	NC	OUT	NC
MVI	A.10	XRA	A
OUT	NC	OUT	NC
RET		RET	

При таких подпрограммах не требуется специальной генерации сигнала СЕ при записи. Разумеется, в порты NA и NB предварительно выведены координаты точки X и Y, определяемые семью разрядами.

Экран дисплея может быть очищен программно или нажатием на кнопку «сброс». Алфавитно-цифровая информация при необходимости легко синтезируется программными либо аппаратными средствами.

Работа с микропроцессорными ВИС показывает, что подобный способ представления данных расширяет их возможности и не увеличивает издержки на экспериментальные исследования, так как основной элемент

системы — осциллограф — органическая часть любой экспериментальной установки.

Телефон для справок: 324-34-81, Москва  
Статья поступила 10 октября 1987.

УДК 681.3.06

В. Г. Долгушев, В. А. Коваленко, А. В. Левшин

## АВТОМАТИЗАЦИЯ ЧЕРТЕЖНО-ГРАФИЧЕСКИХ РАБОТ НА МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Система автоматизации предназначена для генерации, отображения и модификации двумерных графических объектов в интерактивном режиме. Она обеспечивает: формирование и корректировку чертежно-конструкторской документации; выполнение геометрических операций (нахождение касательных, центров сопрягающих окружностей, вычисление длин, углов, пересечений, кратчайших расстояний); ввод и редактирование данных геометрического характера для пакетов и систем автоматизированного проектирования.

Новые графические объекты разрабатываются с использованием ранее созданных объектов или генерируются из простых элементов — отрезков прямых, окружностей, дуг окружностей, строк текста и областей заполнения. Для включения элемента в конструируемый объект из «меню» выбирается соответствующая команда и с помощью графического курсора указывается место его расположения на чертеже. При необходимо-

сти система запрашивает дополнительные параметры, такие как радиус окружности, угол наклона прямой, высота символов строки текста. После прорисовки заданного элемента на экране графического дисплея можно вводить следующий элемент и т. д. Неверно вычерченный элемент легко удаляется.

Имеется возможность изменять масштаб выбранного участка изображения для проработки крупных и мелких деталей на чертежах или рисунках. Подготовленный рисунок записывается на гибкий магнитный диск (ГМД) и в любой момент может быть отображен на экране дисплея или вычерчен на графопостроителе.

#### Технические средства и программное обеспечение

Данная система работает на микроЭВМ «Электроника 60», «Электроника МС1201.01» или на любой другой совместимой с ними по системе команд микроЭВМ.

Минимальная конфигурация технических средств для обеспечения работы системы включает в себя: процессор с объемом оперативной памяти 56 Кбайт; запоминающее устройство на ГМД емкостью не менее 200 Кбайт на одну дискету; символьно-графический видеотерминал «Электроника МС7401»; планшетный графопостроитель ЭМ7042АМ.

Программное обеспечение полностью написано на языке Паскаль и работает под управлением ОС ДВК.

Для ускорения работы описание рисунка хранится в оперативной памяти в виде связанных списков. Из-за ограниченной емкости ОЗУ максимальное число графических элементов (прямых, окружностей, дуг) в списках составляет 800. Сложный чертеж может не поместиться в секции, отведенной для хранения данных. В этом случае он подготавливается по частям, которые при выводе на графопостроитель объединяются в одно целое.

#### Графический пульт системы

В качестве пульта используется символьно-графический видеотерминал «Электроника МС7401». В алфавитно-цифровом режиме дисплей обеспечивает формирование 24×80 изображений прописных и строчных букв русского и латинского алфавитов, цифр и специальных символов в позитивном или негативном изображении. Параметры отображения графической информации: информационная емкость экрана — 560×400 точек; вывод векторов, окружностей и дуг — в виде семи типов линий.

При работе системы экран дисплея разделен на области, каждая из которых имеет определенное функциональное назначение. В области графического диалога отображаются все операции по построению и преобразованию рисунка. Размер области фиксирован, и те части рисунка, которые выходят за ее пределы, отсекаются.

В области локатора постоянно индицируются параметры, отражающие текущее состояние системы координат и положение графического курсора.

С помощью меню выбираются необходимые команды и устанавливается тип линии, которым будет прорисован элемент.

Область символьного диалога предназначена для вывода подсказок, сообщений об ошибках, ввода числовой и символьной информации. Для ввода чисел, текстовых строк, управления положением графического курсора, выбора команд и типов линий используется клавиатура дисплея.

#### Краткий перечень команд

ПРЯМАЯ	Нарисовать отрезок прямой
ОКРУЖНОСТЬ	Нарисовать окружность заданного радиуса
ДУГА	Нарисовать дугу окружности
ТЕКСТ	Нарисовать строку алфавитно-цифровых символов
ТОЧКА	Нарисовать точку
ПАРАЛЛЕЛЬ	Провести прямую, параллельную заданной прямой

НАКЛОННАЯ

КАСАНИЕ

СОПРЯЖЕНИЕ

ЗАПОЛНЕНИЕ

СТИРАНИЕ

УДАЛЕНИЕ

СИММЕТРИЯ

ПЕРЕНОС

СЕКМЕНТ

КОПИЯ

ОКНО

СДВИГ

ПРОРИСОВКА

ОБНОВЛЕНИЕ

ЗАПИСЬ

ЧТЕНИЕ

СБРОС

ВЫХОД

РЕЖИМ

Провести прямую под углом к заданной прямой

Провести прямую, касательную к двум заданным окружностям или дугам окружностей

Провести из заданной точки прямую, касательную к данной окружности или к дуге окружности

Нарисовать окружность данного радиуса, касающуюся двух заданных элементов — прямых, окружностей или дуг окружностей

Задать контур области заполнения и заштриховать ее

Удалить элемент из рисунка

Удалить выбранную группу элементов рисунка. Удалить область заполнения

Симметрично отобразить выбранную группу элементов рисунка относительно заданной прямой

Выполнить над группой элементов заданные преобразования сдвига, поворота и масштабирования

Запомнить группу элементов рисунка под заданным именем

Вставить копию именованной группы элементов в указанное место на рисунке с заданным масштабом и углом поворота

Отобразить на экране графического дисплея заданную часть рисунка с необходимым увеличением

Передвинуть текущее «окно» в заданном направлении

Прорисовать на экране дисплея файл с заданным именем без занесения в список

Очистить экран и перерисовать текущее изображение с удалением всех вспомогательных элементов

Записать текущий рисунок в файл с заданным именем

Прочитать рисунок из файла и нарисовать его на экране дисплея

Очистить экран и удалить текущий рисунок из списков

Закончить сеанс работы с системой

Установка режимов работы (включенное/выключенное «резиновой инти», расчета точек пересечений, штриховки областей заполнения при обновлении экрана).

#### Ввод команд

Список команд постоянно отображается в области меню команд на экране дисплея. Из-за небольших размеров этой области меню разбито на две страницы. Смена страниц меню происходит при нажатии на клавишу (↵).

Текущая команда изображается на экране негативными символами — темными буквами на светлом фоне. С помощью клавиш (↑) и (↓) можно «бегать» по меню вверх и вниз соответственно.

Одновременно со сменой команд в области символьного диалога изменяются связанные с командами подсказки. Сразу после выбора необходимой команды можно вводить дополнительные параметры. Если команда запрашивает группу параметров, после ввода каждого из них изменяется подсказка, показывая, какие еще параметры осталось доопределить.

До начала ввода параметров для следующей команды можно отменить все результаты, которые были получены во время выполнения предыдущей команды.

#### Установка типов линий

Используется пять типов линий: основная, вспомогательная, дополнительная, пунктирная и осевая (штрихпунктирная). Необходимый тип линии выбирается аналогично команде. Отличие лишь в том, что для «скольжения» по строке меню типов линий используются клавиши (←) и (→).

#### Управление графическим курсором

Максимальные размеры чертежа, который может целиком поместиться на экране дисплея (размеры рабочего поля), составляют 940×760 мм. После того как система загружена в память ЭВМ, текущие координаты устанавливаются равными координатам центра экрана. Для визуального контроля за положением точки, соответствующей текущим координатам, используется

графический курсор. На экране он изображается в виде мерцающего перекрестия.

Устройством управления положением графического курсора является дополнительная цифровая клавиатура. Каждое нажатие на клавиши цифр <1>...<4> и <6>...<9> вызывает перемещение курсора на один шаг в следующих направлениях:

- |                    |                     |
|--------------------|---------------------|
| <1> — влево вниз;  | <6> — вправо;       |
| <2> — вниз;        | <7> — влево вверх;  |
| <3> — вправо вниз; | <8> — вверх;        |
| <4> — влево;       | <9> — вправо вверх. |

Шаг перемещения курсора задается клавишами <ПФ1>...<ПФ4>. Если стандартные значения шагов по каким-либо причинам не подходят, их можно переопределить.

При запуске в системе устанавливаются абсолютные координаты. Точке с координатами ( $X=0,0$ ;  $Y=0,0$ ) соответствует левый нижний угол области графического диалога. В относительной системе центром координат может быть выбрана любая точка на экране дисплея. Для точной установки графического курсора в опорные точки элементов (начало и конец отрезка прямой, центр окружности, начало строки текста и т. д.) и в точки пересечений элементов служит операция «захват» — автоматическое перемещение курсора в ближайшую к нему точку.

#### Ввод числовых параметров

Числовые параметры вводятся в целом, или вещественном виде, а также в виде простых арифметических выражений, которые могут включать в себя операции сложения, вычитания, умножения или деления и символические обозначения текущих координат, шага перемещения курсора, расстояния до начала координат и угла. Фактические значения символов постоянно высвечиваются на экране дисплея. Использование символических обозначений и арифметических выражений удобно при выполнении геометрических построений.

#### Вычерчивание элементов

**ПРЯМАЯ.** Для всех типов линии, кроме вспомогательного, вычерчивается отрезок прямой, соединяющей две заданные точки. Положение точек указывается с помощью графического курсора.

Если линия вспомогательного типа, то изображается прямая, проходящая через две заданные точки и пересекающая весь экран.

Размерная прямая может заканчиваться по выбору стрелкой, обращенной к начальной и (или) конечной точкам отрезка.

**ОКРУЖНОСТЬ и ДУГА.** Окружность строится по ее центру и радиусу. Для дуги необходимо задание трех точек: начальной, дополнительной и конечной. Координаты центра и точек задаются графическим курсором, значение радиуса вводится с клавиатуры.

Для линии вспомогательного типа центр и радиус дуги окружности рассчитываются автоматически: для линий других типов необходимо указать уже существующий элемент чертежа (окружность или дугу), на основе которого будет построена новая дуга окружности.

Дуга может заканчиваться стрелками.

**ТОЧКА.** При вычерчивании отрезков прямых, окружностей и дуг одновременно рассчитываются все точки пересечений с уже существующими элементами рисунка и помечаются на экране специальными маркерами. Координаты маркеров хранятся до момента подачи команды **ОБНОВИТЬ ЭКРАН**. Точки пересечений, существенно необходимые для дальнейших построений, фиксируются командой **ТОЧКА**.

Элементы, создаваемые по этой команде, выполняют вспомогательную роль и на графопостроителе не выводятся. Изображение точки на бумажном носителе может быть получено с помощью окружности малого радиуса.

**ТЕКСТ.** Допустимая высота алфавитно-цифровых символов лежит в пределах 1,0...76,0 мм. Значение угла,

под которым располагаются строки текста по отношению к оси X, от 0 до 360°. Начало строки задается с помощью графического курсора.

Текст может включать в себя прописные и строчные буквы латинского и русского алфавитов, цифры и знаки. Так как размеры символов на экране дисплея и готовом чертеже различаются (ширина символов у дисплея МС7401 равна их высоте, у графопостроителя ЭМ7042АМ в среднем составляет 0,6 от высоты символа), одновременно со строкой на экран дисплея выводится габаритная рамка.

**ШТРИХОВКА.** В системе обеспечена возможность штриховки произвольных областей рисунка, ограниченных замкнутыми контурами, состоящими из отрезков прямых, дуг и окружностей. Расстояние между соседними штрихами — 0,1...100,0 мм, наклон штрихов — 0...360,0°.

В одной команде задается произвольное число областей заполнения. Отдельные области могут касаться друг друга, иметь общие границы, одна или несколько областей могут быть охвачены другой областью и т. д.

Граничные элементы каждой области последовательно указываются с помощью курсора и заносятся в отдельные списки, поэтому после задания области их можно удалять, корректировать и т. д.

При многократной штриховке одной и той же области с различными шагами, углами наклона и типами линий можно получать разнообразные виды заполнения.

#### Геометрические построения

Система обеспечивает выполнение следующих геометрических построений: нахождение касательных, центров сопрягающих окружностей, построение прямых, параллельных или под заданным углом к данной прямой.

Для касательных, сопрягающих окружностей и прямых, генерируемых по командам данной группы, рассчитываются все точки пересечений этих элементов с уже существующими элементами рисунка и помечаются на дисплее маркерами.

Базовые элементы (т. е. элементы, по отношению к которым строятся параллели, наклонные, касательные и сопрягающие окружности) указываются с помощью графического курсора. Базовым элементом для параллелей и наклонных может быть задана любая прямая на экране дисплея, для касательных — любая точка, окружность или дуга окружности, для сопряжений — прямая, окружность или дуга окружности.

Задавая базовые элементы при построении касательных и сопрягающих окружностей, положение курсора одновременно определяет приблизительное расположение точек касания. После расчета точных координат точек касания вычерчивается та касательная или сопрягающая окружность, у которой сумма расстояний между приближенными и действительными значениями координат точек касания минимальна.

#### Управление экраном

**ОКНО и СДВИГ.** Дисплей «Электроника МС7401» по своим характеристикам относится к растровым графическим устройствам со средней разрешающей способностью, и при работе с большими рисунками появляются проблемы, связанные с грубосковой мелкими деталями. С помощью операции **ОКНО** можно получить изображение любой части рисунка с заданным увеличением, выслынить необходимые действия и либо вернуться к исходному изображению, либо перейти к другой части рисунка в удобном для работы с ней масштабе.

Центр нового окна задается графическим курсором. Значение масштаба может быть целым или вещественным числом в диапазоне 1,0...50,0. Если требуемый масштаб больше текущего, то новое окно после выполнения команды будет содержать лишь часть изображения старого окна, но с большим увеличением. Если задать масштаб меньший, чем текущий, то новое окно «охватит» старое и на экране станет видимой



большая часть рисунка с меньшим увеличением. При значении масштаба, равном единице, происходит возврат к исходному окну, которое охватывает целиком весь рисунок.

Произвольные перемещения текущего окна над рисунком обеспечиваются командой СДВИГ. Смещение окна по горизонтали и вертикали определяется числовыми параметрами «смещение X» и «смещение Y» в миллиметрах.

**ОБНОВЛЕНИЕ ЭКРАНА.** При выполнении рисунков с повышенными требованиями по точности, необходимо проведение большого числа геометрических построений, и вспомогательные элементы довольно быстро загромождают экран. Избавиться от них можно с помощью обновления экрана, когда старое изображение удаляется, а затем прорисовывается вновь без вспомогательных построений.

Кроме того, после команд УДАЛЕНИЕ и СТИРАНИЕ в оставшихся элементах рисунка могут появиться разрывы. Когда их слишком много, зрительное восприятие рисунка на экране дисплея ухудшается. После перерисовки изображения разрывы исчезают.

#### Редактирование рисунков

**УДАЛЕНИЕ.** Удаляемый элемент указывается курсором. Чтобы удалить группу элементов, расположение которых на рисунке локализовано, используется команда УДАЛЕНИЕ. По этой команде удаляются все элементы, находящиеся внутри заданной прямоугольной области. Прямоугольная область строится по двум противоположным вершинам, указанным курсором. Если прямоугольник с нулевой длиной сторон будет задан внутри заштрихованной области, то штриховка будет удалена.

**КОПИЯ.** Данная команда позволяет получить копию именованного объекта, который предварительно был задан командой СЕГМЕНТ. При создании копии объект можно сдвигать, поворачивать, растягивать или сжимать.

**ПЕРЕНОС.** Выполняются операции сдвига, поворота и масштабирования над группой элементов рисунка, лежащих внутри заданной прямоугольной области.

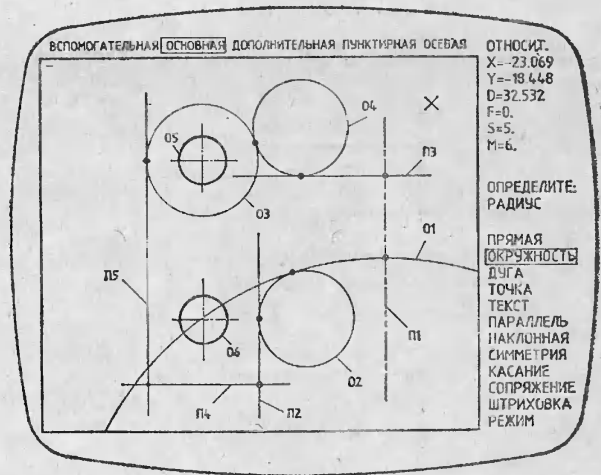
**СИММЕТРИЯ.** Обеспечивается симметричное отображение группы элементов рисунка относительно заданной прямой. Как и в командах СТИРАНИЕ и ПЕРЕНОС, симметрично отображаются все элементы, целиком лежащие внутри заданного прямоугольника.

#### Работа с файлами

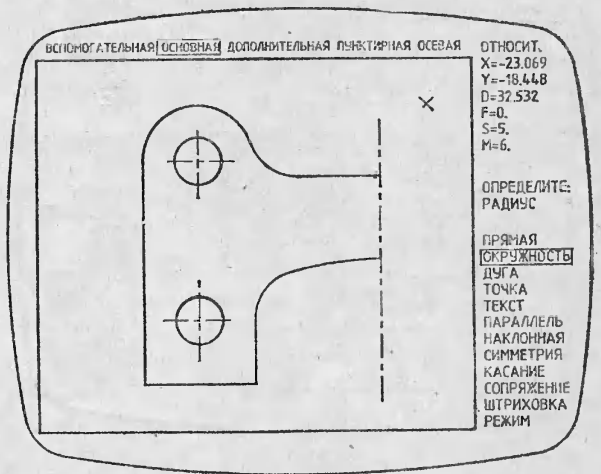
Команды этой группы обеспечивают запись и хранение рисунков в файлах на ГМД. Записи файлов имеют специальный формат, обеспечивающий их компактность. Возможно создание файлов в коде ASCII, из которых графические данные легко распечатать или передать для обработки другим программам.

**ЗАПИСЬ.** Имя файла должно соответствовать соглашениям операционной системы. Стандартное расширение файла по умолчанию RIS. При вводе имени его можно опускать. При расширении ASC данные будут записаны в коде ASCII.

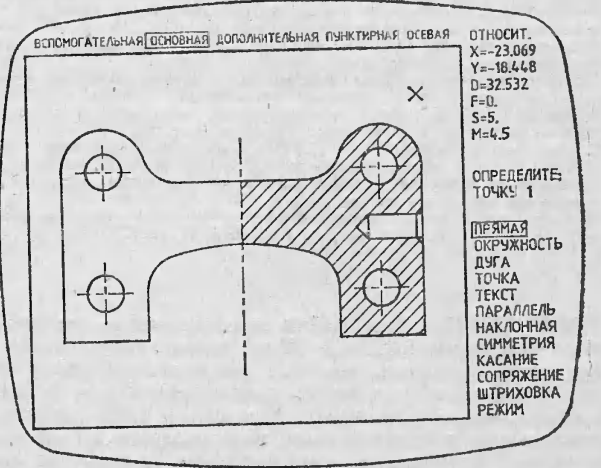
**СЕГМЕНТ.** Создается сегмент-файл, содержимое которого — группа элементов текущего рисунка, ограниченная заданным прямоугольником. Элементы рисунка, лежащие внутри прямоугольной области, записываются в файл, а элементы, находящиеся за пределами области или пересекающие ее границы, в файл не записываются. Размеры ограничивающего прямоугольника определяются двумя его противоположными вершинами. Дополнительным параметром команды является также точка базы — центр локальной системы координат сегмента, относительно которой будут проводиться необходимые преобразования сдвига, поворота и масштабирования при создании копии. Имя сегмента совпадает с именем файла. Возможно создание безымянного (временного) сегмента.



а)



б)



в)

## Вывод рисунков на графопостроитель

Качество рисунков, подготовленных с помощью системы и затем перенесенных на бумагу, определяется только разрешающей способностью используемого графопостроителя, так как графические данные внутри системы хранятся с точностью в семь значащих цифр. Система обеспечивает получение чертежей на устройствах типа ЭМ7042АМ, DIGIGRAF-1712 и АП-7251. Для вывода подготовленных рисунков на графопостроитель служит специальная программа, назначение которой — формирование файла, содержащего необходимый рисунок в форматах команд графопостроителя заданного типа, и передача его на устройство. При выводе рисунка (см. рисунок) разделяется на слои (для каждого слоя можно задать цвет и толщину линий, которыми он будет вычерчен): первый слой — элементы рисунка, выполненные основным типом линий; второй — элементы рисунка, выполненные дополнительным, штрих-пунктирным и пунктирным типами линий; третий — размерные линии и надписи к ним, специальные знаки и их текстовые атрибуты; четвертый — текстовые строки; пятый — штриховка областей заполнения. Число слоев задается перед началом работы программы.

Пример подготовки чертежа

На рис. а—д показаны этапы изготовления чертежа простой детали.

Телефон: 42-15-40, Фрунзе  
Статья поступила 30 ноября 1987

## КРАТКОЕ СООБЩЕНИЕ

УДК 681.327

Е. Л. Лизунова, М. Я. Фенстер

## КОНТРОЛЛЕР КАСЕТНОГО НАКОПИТЕЛЯ НА МАГНИТНОЙ ЛЕНТЕ РК-1 ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

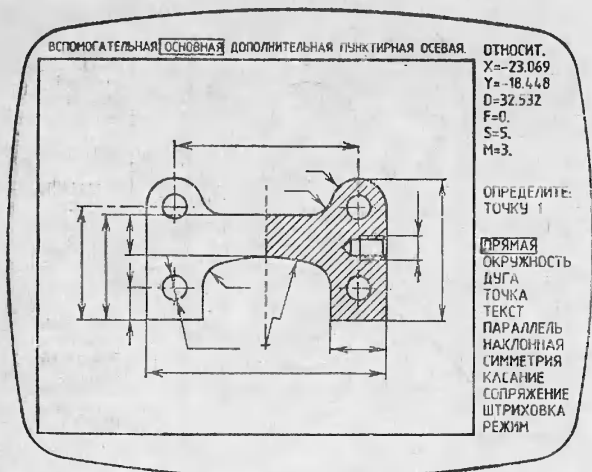
В настоящее время в вычислительных комплексах на базе микроЭВМ типа «Электроника 60» широко применяются «электронные диски». Для загрузки «электронного диска» можно использовать не только накопители на ГМД, но и другие аппаратные средства, в том числе и кассетные накопители на магнитной ленте (КНМЛ).

Предлагаемый контроллер совместим на уровне формата записи на магнитную ленту со стандартным накопителем мини-ЭВМ СМ-4. Это позволяет информацию, записанную в ходе эксперимента на кассету накопителя «РК-1» (входящего в состав лабораторной системы автоматизации на базе микроЭВМ «Электроника 60»), обрабатывать на мини-ЭВМ СМ-4, имеющей более широкие возможности.

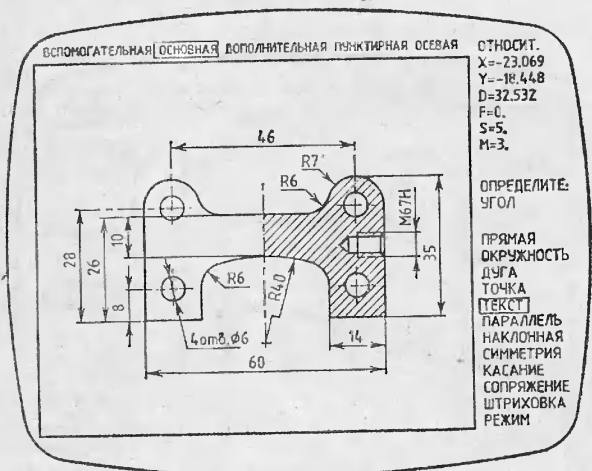
Оператор микроЭВМ «Электроника 60» может использовать для работы операционную систему РАФОС, поскольку рассматриваемый контроллер программно совместим с ней.

Контроллер выполен на микросхемах серий К580, К555, К589, К559. Его габаритные размеры с блоком питания КНМЛ «РК-1» не превышают размеров самого КНМЛ «РК-1». Предусмотрено также раздельное использование контроллера и блока питания. Контроллер подключается к микроЭВМ «Электроника 60» при помощи переходной платы, соответствующей одинарной плате «Электроника 60».

310103, Харьков, ул. Деревянка, 18, кв. 15.  
Фенстеру М. Я. Тел. 30-03-73  
Сообщение поступило 13 октября 1987



2)



а)

Работа начинается с геометрических построений. В произвольном месте на экране дисплея вычерчиваются осевая линия П1 и прямая П3. Далее на заданных расстояниях от П1 строятся отрезки П2 и П5, а от П3 вычерчивается отрезок П4. Затем строятся окружности 05, 06, 01 и 03 и с помощью команды СОПРЯЖЕНИЕ — окружности 04 и 02. Окружности 06 и 05 вычерчиваются основным типом линий, остальные элементы — вспомогательным. Обозначения элементов (П1...П5, 01...06) на рисунке условные, так как реально на экране дисплея они не отображаются (рис. а).

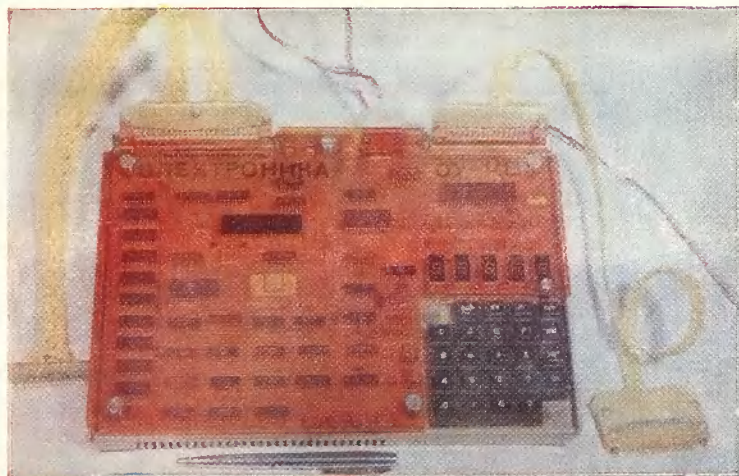
Производится «фиксация» необходимых дуг и отрезков прямых по рассчитанным точкам пересечений. После команды ОБНОВЛЕНИЕ ЭКРАНА все элементы, построенные вспомогательным типом линии, будут удалены (рис. б).

С помощью команды СИММЕТРИЯ полученный фрагмент чертежа зеркально отображается относительно оси П1. Затем вычерчивается отверстие под резьбу и производится штриховка требуемой области (рис. в).

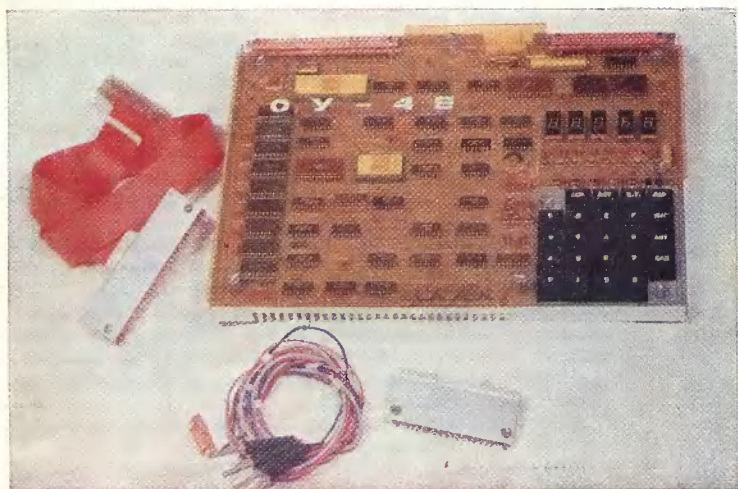
В последнюю очередь проставляются размерные линии и надписи с использованием команд ПРЯМАЯ И ТЕКСТ (рис. г, д)

**ЧТЕНИЕ.** Читаются файлы, предварительно записанные командами ЗАПИСЬ. Имя файла, определяющее конкретный рисунок, является единственным параметром команды. После ввода имени содержимое файла копируется в соответствующие списки, и часть рисунка, попадающая в текущее окно, вычерчивается на экране дисплея. Элементы рисунка, целиком выходящие за пределы формата текущего рисунка, в списки не заносятся.

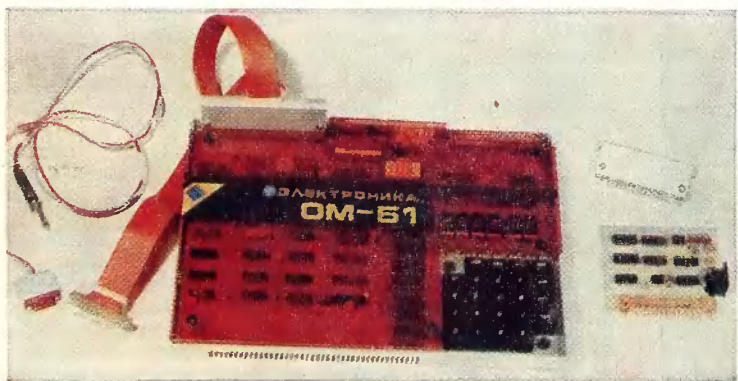
# АВТОНОМНЫЕ ОБУЧАЮЩИЕ УСТРОЙСТВА для ОМЭВМ серии К1816



«Электроника ОУ-48» — учебно-отладочное устройство для ОМЭВМ КМ1816ВЕ48



«Электроника ОУ-49» — отладочное устройство для разработки, отладки и тестирования аппаратуры на базе ОМЭВМ КМ1816ВЕ39 (48 и 49) с последующей записью отлаженных программ во внутреннее СППЗУ ОМЭВМ или в СППЗУ серий К573



«Электроника ОМ-51» — автономное учебно-отладочное средство для микропроцессорных устройств с системой команд КМ1816ВЕ51

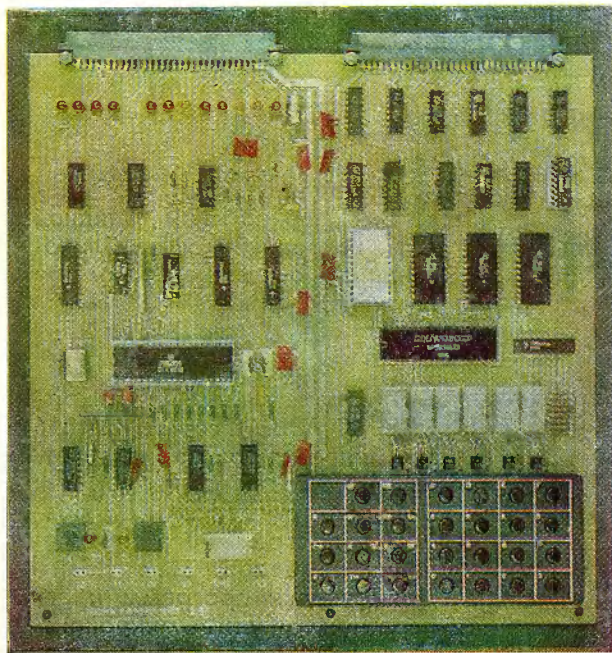
(К ст. Жданова В. И., Бобылева В. Н., Гринь Н. Ф., Уткиной Т. Г., которая будет опубликована в очередном номере журнала)

Однокристалльные микроЭВМ (ОМЭВМ) серии КМ1816 все более широко применяются при создании недорогих микроконтроллеров с объемом программного обеспечения не более 2...4 Кбайт. Для отладки программ и аппаратной части микроконтроллеров разработаны простые автономные отладочные устройства, называемые часто «оценочными модулями» (ОМ). Применение этих модулей весьма эффективно на начальных этапах освоения ОМЭВМ, при проверке работы микроконтроллеров в цеховых условиях и т. п. ОМ позволяют детально изучить работу и взаимодействие всех составных частей ОМЭВМ, разработать и отладить в реальном масштабе времени узлы РЭА и программное обеспечение для них.

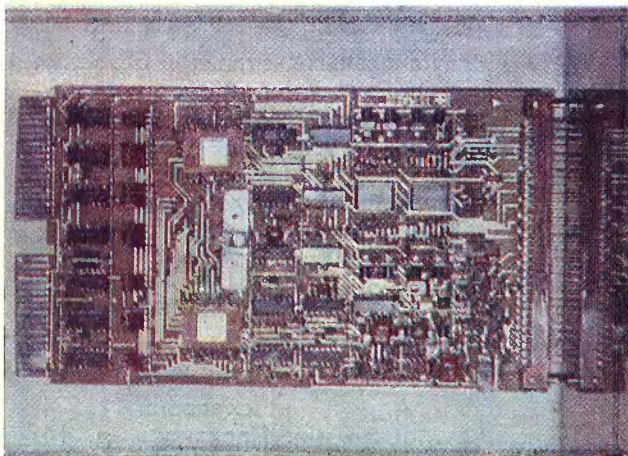
Адрес для справок: 111524, Москва, ул. Плеханова, д. 14, кор. 3, кв. 49, Бобылеву В. Н. Тел.: 306-26-57 (д.)

**МНОГОФУНКЦИОНАЛЬНЫЙ КОМПЛЕКС  
ПРОГРАММНО-АППАРАТНЫХ СРЕДСТВ  
ДЛЯ РАЗРАБОТКИ КОНТРОЛЛЕРОВ  
НА ОСНОВЕ ОДНОКРИСТАЛЛЬНЫХ ЭВМ  
СЕРИИ K1816**

(К ст. Кушнира В. Е., Панфилова Д. И., Шаронина С. Г.)



Модуль одноплатной учебно-отладочной микроЭВМ УМПК-48/ВМ



Модуль расширителя ввода-вывода УМПК-48/МР1

Многофункциональный комплекс предназначен для изучения программирования и функционирования ОЭВМ серии K1816, исследования особенностей построения микроконтроллеров на их основе, разработки программного обеспечения, а также физического моделирования и отладки управляющих систем (как автономно, так и во взаимодействии с реальным объектом управления).

Модуль одноплатной учебно-отладочной микроЭВМ УМПК-48/ВМ — ядро аппаратной части комплекса, при построении микроконтроллеров реализует все возможности, заложенные в ОЭВМ; в его состав входят:

однокристалльная ЭВМ типа КМ1816ВЕ48 (с возможностью применения внутренней памяти команд);

клавиатура (24 клавиши) и дисплей (6 восьмисегментных светодиодных индикаторов) на основе БИС КР580ВВ79;

внешняя память команд объемом 4 Кбайта;

внешняя память данных объемом 256 байт.

Кроме того:

все линии ОЭВМ выведены на разъемы модуля (сигналы управления, МД и сформированная МА усилены); линии ввода-вывода ОЭВМ могут задействоваться (P10 ... P17, P24 ... P27, T0, T1, INT — без ограничений, а DB0 ... DB7, P20 ... P23 — с учетом их применения для подключения средств расширения);

линии P10 ... P17, T0, T1 снабжены переключателями для имитации внешних сигналов и индикаторами;

сигнал запроса внешнего прерывания на вход INT ОЭВМ может подаваться с разъема, от клавиши или БИС контроллера клавиатуры и дисплея модуля;

векторы внешнего прерывания и от таймера-счетчика событий ОЭВМ свободны и могут использоваться;

имеется схема выдачи звуковых сигналов;

резидентное программное обеспечение — монитор-отладчик DM48 (дополнительно введены директивы работы с магнитофоном);

возможны ввод во внешнюю память команд информации с клавиатуры или магнитофона, а также ее запись на магнитную ленту (скорость обмена — 1,5 Кбод, модуляция — фазовая, сформированный файл содержит в памяти одnobайтовое имя, адреса начала и окончания, данные, контрольные суммы по адресам и данным);

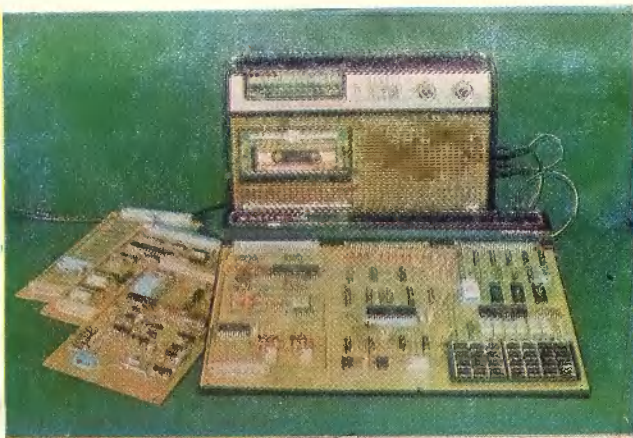
объем отлаживаемых программ — 4 Кбайта (полное адресное пространство ОЭВМ);

программы могут выполняться в реальном масштабе времени или покомандно.

Модуль УМПК-48/МР1 (расширитель ввода-вывода) предназначен для изучения программирования и функционирования БИС КР580ВР43 и макетирования устройств сопряжения с объектом управления. При исследовании режимов работы БИС расширителя портов ввода-вывода ОЭВМ используются переключатели имитации сигналов и индикаторы на ее линиях P40 ... P43 и P50 ... P53. Макетное поле (100×160 мм) позволяет устанавливать ИМС с любым типом корпуса.

Программные средства помимо монитора-отладчика DM48 и тестовой программы TST48 (также функционирующей с модулями УМПК-48/ВМ и УМПК-48/МР1) включает кроссассемблер ASMK48 и кроссассемблер RASMK48, которые могут эксплуатироваться на любой ПЭВМ с МП БИС КР580ВМ80.

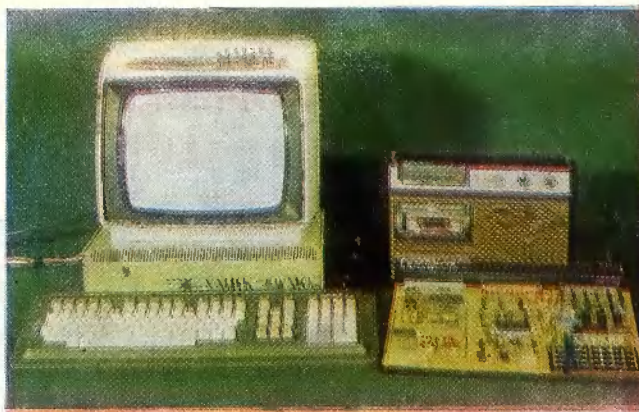
Многофункциональность комплекса, наряду с простотой в эксплуатации, облегчает его освоение и обеспечивает возможность широкого внедрения в процесс разработки микроконтроллеров. Наибольший выигрыш от применения достигается в случаях, когда необходимо подготовить квалифицированных специалистов для проведения таких разработок. На этом этапе он может использоваться при выполнении исследований: структуры ОЭВМ и функционирования ее элементов (линий ввода-вывода, таймера-счетчика событий, системы прерываний и т. п.); системы команд и ориентированных на нее алгоритмов обработки информации; временных диаграмм работы; организации различных интерфейсных функций (параллельного и последовательного ввода-вывода данных, временных интервалов и т. п.);



Рабочее место для исследования ОЭВМ серии К1816 и микроконтроллеров на ее основе

расширения возможностей (за счет использования интерфейсных БИС серии КР580, внешних ЗУ и т. п.); организации отладочных режимов работы (пошаговое выполнение программ, эмуляция памяти команд); структуры и программного обеспечения микроконтроллеров конкретных объектов и методов отыскания неисправностей в них. Материалы, необходимые для проведения исследований, и порядок их выполнения изложены в методическом обеспечении комплекса.

Модульная конструкция и многофункциональность комплекса УМПК-48 позволяют легко переходить от его использования как средства обучения к применению в качестве инструментального средства при разработке микроконтроллеров или их моделировании для оценки выбранного варианта структуры. В этих случаях кроме перечисленных модулей пользователь может применять модули комплекса УМПК-80 (например, последовательный интерфейс УМПК-80/МИ1, ЦАП УМПК-80/МР4, АЦП УМПК-80/МР5), подключая их через согласующее устройство, или самостоятельно разработанные модули для реализации необходимых функций (преобразования сигналов от датчика, управления шаговыми двигателя-



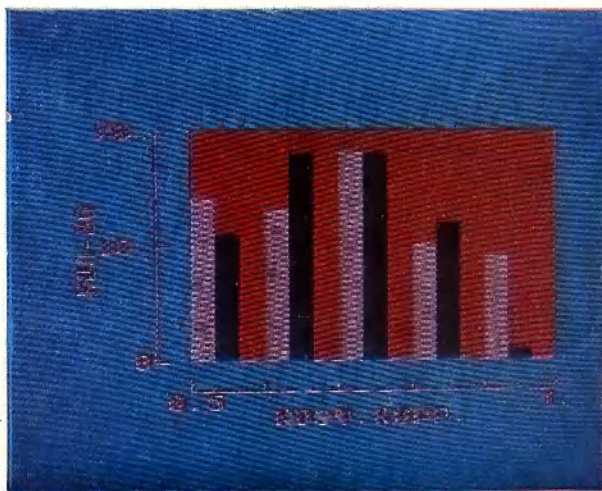
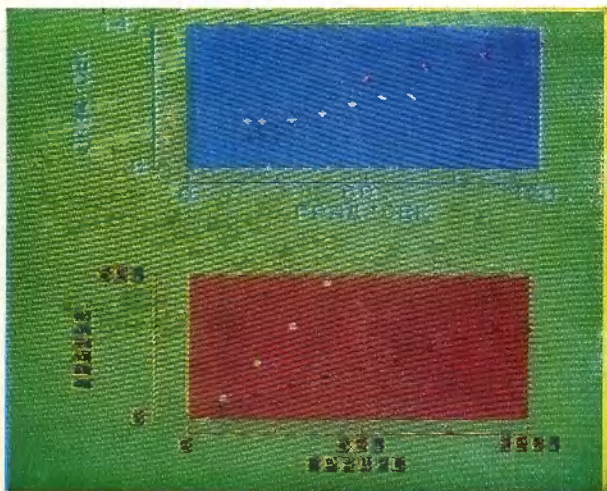
Пример организации персонального рабочего места разработчика микропроцессорных управляющих систем

ми и т. п.). При моделировании микроконтроллеров ОЭВМ модуля УМПК-48/ВМ может использоваться автономно, вместе с БИС КР580ВВ79 и(или) внешней памятью. Для упрощения работы на начальных этапах моделирования и отладки программ сигналы, поступающие от устройств сопряжения с объектом, имитируются. На линиях ввода-вывода ОЭВМ их получают с помощью переключателей модуля, а регистры внешних УВВ (адресуемых как ячейки внешней памяти данных) эмулируются записью необходимой информации.

Персональное рабочее место, построенное на основе комплексов УМПК-48 и УМПК-80 с учетом стоящих перед разработчиком задач, может использоваться на всех этапах создания микропроцессорных управляющих систем. При этом отладочная микроЭВМ УМПК-80/МО, сопряженная с помощью модулей набора УМПК-80/МР с необходимыми периферийными устройствами (НГМД, НМЛ, АЦПУ и др.), применяется для работы с кросс-ассемблером АСМК-48, кроссассемблером RASMK48 и другим программным обеспечением (в том числе совместимым с ОС СР/М), а также для загрузки подготовленных программ в память УМПК-48/ВМ.

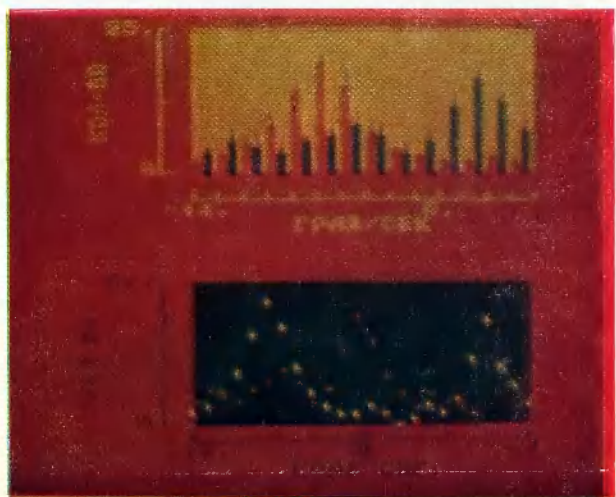
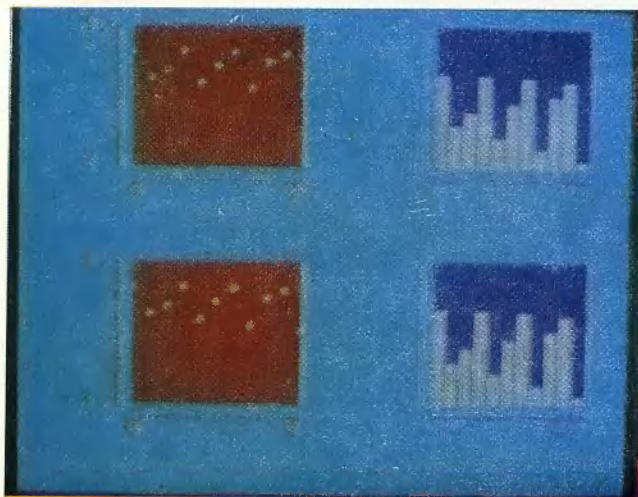
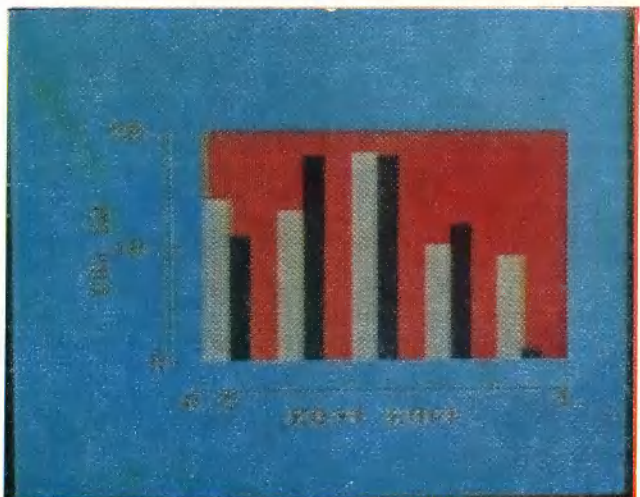
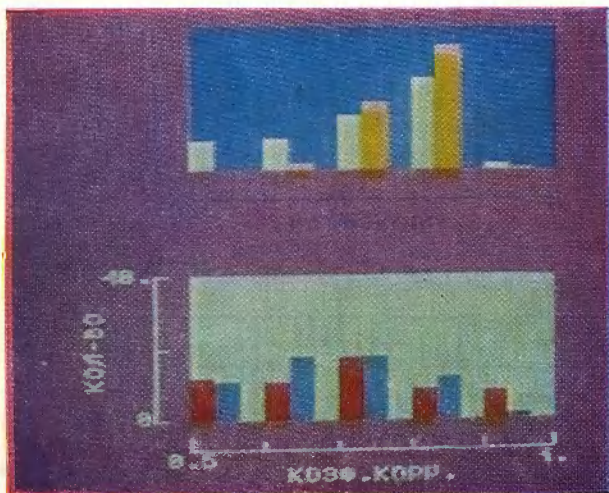
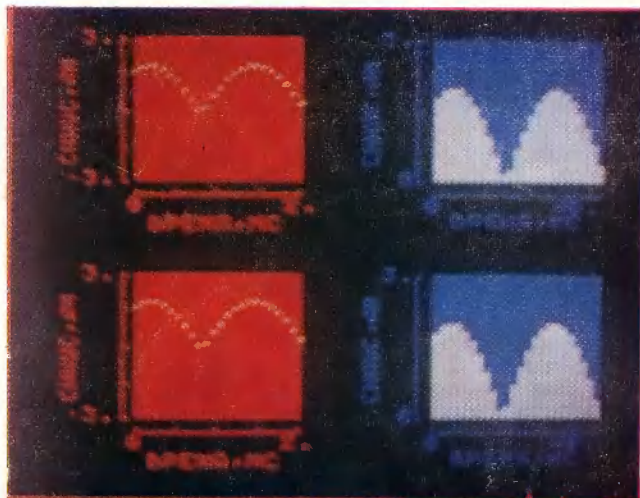
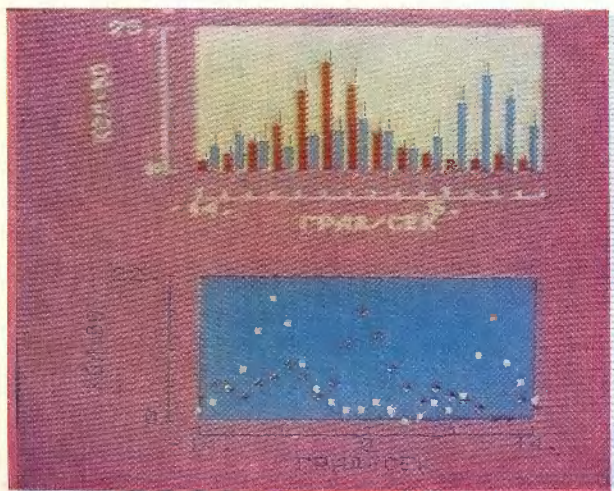
## ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ДИАЛОВОЙ СИСТЕМЫ ПОДГОТОВКИ ГРАФИЧЕСКИХ ИЛЛЮСТРАЦИЙ

(К ст. Гехмана Б. И.)



# ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ДИАЛОВОЙ СИСТЕМЫ ПОДГОТОВКИ ГРАФИЧЕСКИХ ИЛЛЮСТРАЦИЙ

(К ст. Гехмана Б. И.)



## ОПЕРАЦИОННЫЕ СИСТЕМЫ МИКРОЭВМ

УДК 681.142.2

В. Н. Глухов, Л. А. Долбилов, Е. Е. Дудников, Я. Лоозе

### МИКРОДОС — АДАПТИВНАЯ СИСТЕМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ 8-РАЗРЯДНЫХ МИКРОЭВМ

В 1984 г. в Международном научно-исследовательском институте проблем управления (МНИИПУ) для широкого класса 8-разрядных микроЭВМ была начата разработка мобильной операционной системы, получившей название ОС МикроДОС [1, 2], а затем в сотрудничестве с рядом организаций — создание разнообразного программного окружения для этой ОС, т. е. системы программного обеспечения на базе ОС МикроДОС. С 1985 г. Международный центр научной и технической информации (МЦНТИ) приступил совместно с МНИИПУ к публикации справочной документации по системе МикроДОС в виде серии выпусков с общим заглавием «Библиотека МикроДОС». Система МикроДОС прошла длительную опытную проверку в 50 производственных и научных организациях СССР и ГДР. С 1988 года МЦНТИ начал ее коммерческое распространение на договорных началах.

Для работы системы МикроДОС необходима микроЭВМ с аппаратными ресурсами:

центральный процессор (ЦП) — микропроцессор типа K580 или программно совместимый с ним (например, U880 производства ГДР);

исполняемая память (ИП) объемом не менее 32 Кбайт; устройства ввода-вывода (УВВ) — клавиатура, генерирующая коды КОИ7 (ASCII); алфавитно-цифровой дисплей и принтер, воспроизводящие коды КОИ7;

внешняя память (ВП) на дисках со съемными или постоянными носителями (чаще всего используются дисководы для гибких дисков диаметром 203 или 133 мм).

МикроДОС обеспечивает индивидуальное использование микроЭВМ: все аппаратные ресурсы ЭВМ поступают в полное распоряжение пользователя. ОС МикроДОС предоставляет гибкий язык команд, выполняет загрузку вызываемых программ в память, по запросам этих программ осуществляет управление устройствами ввода-вывода и информационным ресурсом во внешней памяти.

При разработке ОС МикроДОС отправной точкой послужила ОС CP/M2.2 — наиболее распространенная за рубежом ОС для 8-разрядных микроЭВМ. Сравнение МикроДОС с CP/M2.2 и CP/M3.1 проведено в [1]. Совпадая с CP/M по файловой системе, МикроДОС оператору более удобна, программисту заметно облегчает разработку и кодирование программ на языке ассемблера. Кроме того, МикроДОС существенно улучшает использование ИП и ВП, увеличивает число обслуживаемых устройств символьного ввода-вывода, позволяет вести обмен с консолью не только в кодах КОИ7, но и в кодах КОИ8 и т. д.

Совместимость МикроДОС сверху с CP/M2.2 гарантирует функционирование всех корректных программных пакетов, разработанных для CP/M2.2, и системных утилит CP/M3.1.

Прикладные программы МикроДОС

Обработка данных ведется с помощью следующих программных пакетов:

**ТЕКСТ** — позволяет создавать, корректировать, форматировать и печатать макеты разнообразных текстовых материалов [6];

**ИТ-7920** — функционально превращает микроЭВМ в интеллектуальный терминал ЕС-7920 для работы с ЕС ЭВМ, СМ ЭВМ или другой микроЭВМ [8];

**ДИАЛОГ-М** — обеспечивает создание, обновление и выполнение поиска в простых фактографических и смешанных базах данных [9].

К распространению готовятся также пакеты СУБД для ведения фактографических информационных систем; экранной обработки электронных таблиц; обменов файлами между микроЭВМ и ЕС ЭВМ, СМ ЭВМ или другой микроЭВМ.

Обслуживающие программы общего назначения включают [2, ч. 2; 4]:

1) программы обслуживания файловой системы:  
**ATTR** — устанавливает атрибут доступа группе файлов;

**ERAQ** — удаляет группу файлов в режиме диалога;  
**NDIR** — показывает на экране оглавление диска (или его части), имена файлов упорядочены по алфавиту;

**RENA** — переименовывает группу файлов в режиме диалога;

**UNERA** — восстанавливает ошибочно удаленный файл, если записывавшиеся им области диска еще не использованы для хранения другого файла;

2) программы пересылки данных между УВВ:

**MEDIT** — редактор, позволяющий с помощью клавиатуры микроЭВМ создавать и изменять текстовые дисковые файлы в экранном (визуальном) или командном режиме; используется для подготовки исходных текстов программ (на любом языке программирования) и текстовых файлов исходных данных для прикладных программ;

**TRANSF** — копирует дисковые файлы на другой диск;  
**TYPE** — показывает на экране дисплея текстовый дисковый файл; вывод текста производится непрерывно или постранично, сопровождается фильтрацией (пропуском) ряда управляющих (непечатных) кодов; доступна избирательная параллельная выдача показываемого текста на устройство печати;

3) программы простейшей обработки файлов:

**CHECK** — вычисляет циклические контрольные суммы файлов для их идентификации;

**NDIR** — сообщает размеры файлов в килобайтах и объем свободного места на диске;

**MCOMP** — выявляет несоответствия двух файлов при их побайтовом сравнении; текстовые файлы можно сравнивать и построчно;

**TRANSF** — создает новый дисковый файл, последовательно меняя содержимое исходного файла выполнением цепочки простейших преобразований (из фиксированного множества).

## Система программирования МикроДОС

Ряд пакетов обеспечивает программирование на следующих языках высокого уровня:

БЕЙСИК — два интерпретатора, отличающихся выбором подмножеств реализованных функций общепринятого языка программирования и требованиями к объему необходимой памяти [7];

BASIC/F — интерпретатор и компилятор существенно расширенной версии языка, позволяющие создавать структурированные программы, собираемые из нескольких модулей [3];

Си — транслятор, редактор связей, стандартные библиотеки и служебные программы [5].

К распространению готовятся:

Паскаль-T80 — интегрированный пакет, объединяющий редактор исходных текстов, транслятор и компоновщик программ; может использоваться только в микроЭВМ, построенных на базе микропроцессора U880;

ПЛ/1 — транслятор, библиотеки стандартных программ, служебные программы.

Программирование на языке ассемблера для микропроцессора K580 поддерживается следующими пакетами [2, ч. 2; 4]:

MAS — макроассемблер, по исходному тексту программы и дополнительным файлам макроопределений создает перемещаемый объектный модуль, файл символьных меток с их относительными адресами, файл распечатки с генерированными машинными кодами и диагностикой обнаруженных ошибок;

MLIB — библиотекарь, объединяет группу перемещаемых объектных модулей в библиотеку, показывает содержимое библиотеки, модифицирует библиотеку удалением одних модулей и добавлением других;

MLINK — компоновщик, собирает исполняемые файлы программ и их сегментов перекрестий из перемещаемых объектных модулей и библиотек таких модулей, дает информацию о структуре адресного пространства созданного файла;

DEBUG — символьный отладчик, предоставляет широкие возможности для анализа работы программы от неконтролируемого выполнения отдельных ее блоков до трассировки (контроля за выполнением каждой отдельной машинной инструкции), обеспечивает установку регистрируемых точек прохода и точек прерывания с формированием разнообразных условий прерывания; наряду с абсолютными адресами аргументов исполняемых инструкций показывает и отвечающие им символьные метки, что значительно облегчает разработчику процесс отладки его программы.

### Интерфейс оператора с МикроДОС

Оператор обращается к МикроДОС с помощью команд, которые обрабатываются специальной компонентой ОС — интерпретатором команд. Для вызова любой прикладной программы обычно достаточно набрать на клавиатуре название файла ее исполняемых машинных кодов.

Некоторых пользователей системный программист может избавить от всякого общения с ОС, сформировав на их загрузочном диске специальный файл INITIAL.SUB, уже содержащий нужную последовательность команд. Сразу после загрузки МикроДОС без вмешательства пользователя выполнит автоматически все необходимые команды, а пользователю останется лишь взаимодействие с нужной ему прикладной программой, уже вызванной и (частично) подготовленной к работе. Файл INITIAL.SUB облегчает, таким образом, создание на базе микроЭВМ различных проблемно-ориентированных систем (так называемых «рабочих мест»), не требующих от своих пользователей никаких знаний об ОС МикроДОС.

Текст команды запоминается в специальной области ИП — в буфере консоли. Признаком завершения набора команды служит нажатие клавиши возврата каретки <CR>, после этого ОС приступает к исполнению коман-

ды. До нажатия клавиши <CR> можно изменять уже набранный текст: удалять последний символ, последнее слово, добавлять новые символы, удалить весь набранный текст целиком, повторно вызывать на экран последнее сформированное содержимое буфера консоли.

В начале команды указывается имя вызываемой программы, а затем (если необходимо) — предназначенные для передачи программе в качестве параметров символьные строки, отделяемые друг от друга пробелами. Например, страничный показ на экране файла распечатки может быть вызван командой TYPE<SPACE>PROG.PRN<SPACE><SPACE>F1.P, а переименование группы файлов в режиме диалога начнется по команде RENAME<SPACE>\*.ASM<SPACE>\*.MAC<SPACE>Q.

Команда, для исполнения которой требуется загрузка в ИП файла машинных кодов нужной программы, называется транзитной; команда, исполняемая процедурами ОС МикроДОС без обращения к транзитным программам, — резидентной.

Резидентные команды МикроДОС выполняют такие операции управления ресурсами микроЭВМ, необходимые в которых возникает чаще всего. Для удобства вызова имени резидентных команд сокращены до одной литеры (буквы, цифры, специального знака или управляющего кода).

Команды управления работой ВП на дисках:

CTL-C — выполняет «горячий старт» ОС МикроДОС;

D — показывает на экране оглавление диска или его часть (имена файлов следуют в порядке их появления в оглавлении), а также защищающие атрибуты доступа к файлам;

E — удаляет группу файлов. Во избежание возможных ошибок пользователю задается дополнительный вопрос: «Удалить?». И только при получении утвердительного ответа на него производится фактическое удаление: делаются соответствующие отметки в оглавлении диска без изменения содержимого области данных на диске;

O — показывает на экране существующее или устанавливает новое число системных дорожек на диске; используется при доступе к файлам на дисках, сформированных в другой ОС (например, в CP/M2.2);

S — задает имя диска, объявляемого системным;

U — назначает код области пользователя на дисках.

Команды управления работой консоли и принтера:

CTL-F — в состоянии приостановки отменяет сделанные ранее переназначения консоли к дисковому файлу;

CTL-P — в состоянии приостановки переключает (включает или выключает) режим дублирования консольного вывода печатающим устройством; это удобная возможность для избирательного документирования результатов работы программы;

CTL-Q — возобновляет приостановленный ранее вывод данных на консоль;

CTL-S — приостанавливает вывод данных на консоль; состояние приостановки возникает в момент обращения транзитной программы к МикроДОС за выводом на консоль символа (строки символов) либо за вводом с консоли символа (строки символов);

< — переназначает консоль ввода к указанному дисковому файлу; данные начнут считываться не с клавиатуры, а из указанного файла; ввод с клавиатуры возобновится при отмене переназначения консоли либо при достижении конца указанного файла;

> — переназначает консоль вывода к указанному дисковому файлу; выводимая на экран информация начнет записываться и в указанный дисковый файл; вывод в указанный файл прекратится при отмене переназначения консоли либо при «горячем старте» ОС;

A — начинает вести обмена с консолью в коде ASCII (КОИ7, набор 0);

K — начинает вести обмена с консолью в коде КОИ8: значения старших битов передаваемых байтов форми-



руются на основании встречаемых в потоке данных управляющих кодов STL-N и STL-O.

В МикроДОС предусмотрено 10 резидентных команд с цифровыми именами от 0 до 9 для учета специфики УВВ микроЭВМ. Назначения этих команд не стандартизируются: они выбираются системным программистом при генерации ОС и на некоторых микроЭВМ какие-то цифровые команды могут вообще не выполнять никаких действий.

**Фоновый режим МикроДОС.** Существенной особенностью МикроДОС является то, что резидентные команды можно исполнять не только на командном уровне общения с ОС, но и во время выполнения транзитной программы (в фоновом режиме), не мешая дальнейшей работе этой программы. Когда выполнение программы приостановлено на вводе, управляющим кодом STL-D можно вызвать интерпретатор команд и исполнить любую из указанных выше резидентных команд МикроДОС (например, можно просмотреть оглавление диска, удалить какую-нибудь группу файлов и т. п.). Затем можно продолжить нормальное выполнение приостановленной программы, введя ожидаемые ею данные.

**Командные файлы.** Переназначения консоли облегчают организацию поточной обработки данных группой транзитных программ-фильтров (получающих все исходные данные с консоли и выводящих все результаты работы на консоль): выдачу программы переназначают к дисковому файлу, к нему же переназначают и консоль ввода для следующей программы.

Другое важное применение переназначений консоли связано с командными файлами. На практике часто возникает необходимость в многократном исполнении одной и той же группы команд. Например, в процессе отладки программы это могут быть команды трансляции модулей, сборки из них программы и запуска программы на выполнение. Удобно создать дисковый файл, содержащий необходимые командные строки (его называют командным файлом), и затем простым переназначением консоли ввода к этому файлу заставить ОС автоматически продвигать всю рутинную работу. Примером командного файла служит уже упоминавшийся файл INITIAL.SUB.

С помощью входящей в состав МикроДОС программы JINN можно использовать и более сложные параметризованные командные файлы, включающие упоминания до 10 фиктивных параметров, фактические значения которых задаются в списке параметров команды вызова программы JINN на основе позиционного соответствия.

Для документирования командных файлов в МикроДОС предусмотрен префикс команды «;». Обнаружив его, интерпретатор команд считает всю командную строку комментарием, не подлежащим обработке, и сразу переходит к вводу следующей команды.

Необходимость в исполнении очередной команды часто зависит от успешности исполнения предыдущей. Так, собирать программу целесообразно только, если все составляющие ее модули были оттранслированы без ошибок. Для условно исполняемых команд в МикроДОС предусмотрен командный префикс «>».

Транзитная программа перед своим окончанием может сформировать значение специальной переменной ОС — кода возврата. Встретив условно исполняемую команду, интерпретатор команд считывает текущее значение кода возврата, классифицирует его как признак успешного или неудачного завершения предыдущей программы и в соответствии с результатом распознавания исполняет либо пропускает условную команду.

Механизм условного исполнения МикроДОС позволяет создавать командные файлы с гибкими процедурами, координирующими действия нескольких транзитных программ в многоступенчатой обработке данных.

**Сообщения МикроДОС об ошибках.** При невозмож-

ности выполнять запрошенное действие ОС выдает сообщение об ошибке. Чаще всего ошибки возникают при работе с дисками — это может быть неисправная ошибка чтения-записи, попытка удаления файла, защищенного атрибутом доступа, отсутствие места в области данных или в оглавлении при записи файла и т. д. Так как тексты сообщений об ошибках формируются системным программистом при генерации ОС, то обычно они вполне достаточны для уяснения причины ошибки и принятия мер по ее устранению.

МикроДОС предусматривает три различных уровня вмешательства ОС в процесс обработки ошибок; выбор нужного уровня делается вызовом соответствующей функции МикроДОС. В частности, транзитная программа может взять на себя обработку всех ошибочных ситуаций, полностью исключив вмешательство ОС в эту деятельность.

## Интерфейс программ с МикроДОС

**Виртуальная МикроДОС-ЭВМ.** МикроДОС предоставляет программам пользователя следующие ресурсы:

ИП — образуется из базовой страницы памяти (первые 256 байт оперативной памяти) и из так называемой области транзитных программ (ОТП) — сегмента оперативной памяти, начинающегося за базовой страницей и оканчивающегося перед областью, занимаемой ОС;

УВВ — МикроДОС позволяет работать с 12 различными устройствами ввода-вывода, которым присвоены логические номера от 0 до 11. Операции ввода-вывода выполняются с помощью широкого набора стандартных вызовов МикроДОС. Специально выделены три наиболее часто используемых виртуальных устройства: 0 — консоль ввода-вывода (клавиатура для ввода и дисплей для вывода); 1 — устройство печати (только вывод); 2 — дополнительная консоль (ввод и вывод);

ВП — МикроДОС поддерживает до 16 логических дисков с именами от А: до Р:, на каждом из которых могут раздельно хранить свои данные 16 различных пользователей с кодами от 0 до 15 (этим обеспечивается защита пользователей друг от друга). Один диск выделен в качестве системного диска, а один пользователь (с кодом 0) — в качестве системного пользователя.

Каждый диск подразделяется на системные дорожки (первые дорожки не содержат данных) и на область данных, начинающуюся с оглавления диска и подразделяемую на блоки (группы секторов, являющиеся распределяемыми ОС единицами хранения информации в файлах).

Обмены с дисками ОС осуществляются по соответствующим вызовам из программ; каждый обмен затрагивает не отдельный байт, а группу секторов (число байтов в ней кратно 128). Можно рассматривать диск как простой набор секторов, упорядоченный номерами дорожек и номерами секторов на дорожках, и осуществлять обмены с диском на физическом уровне. Для большей части задач, однако, удобнее использовать структуризацию дискового пространства, порождаемую файловой системой ОС МикроДОС, и выполнять обмены с диском на логическом уровне. При работе на логическом уровне доступ к записям файла может быть как последовательным, так и произвольным.

**Загрузчик транзитных программ.** Установив, что команда не является резидентной, интерпретатор команд вызывает загрузчик МикроДОС. Если файл исполняемых кодов соответствующей программы имеет тип .COM, то он загружается в память с начала OTP; если тип .SPR, то — в конец OTP. Последовательность строковых параметров программы, образующая остаток командной строки, специальным образом записывается в базовую страницу памяти — в начале своего исполнения программа может извлечь всю переданную ей информацию. После формирования содержимого информационной области в базовой странице памяти за-

грузчик передаст управление на начало исполняемых кодов.

**Вызовы МикроДОС из программ.** Функции МикроДОС предоставляют программисту стандартные средства для управления устройствами ввода-вывода, работы с дисковой внешней памятью на физическом и логическом (файловом) уровнях, организации нужного способа возврата из программы в ОС, получения и установок значений параметров, управляющих функционированием ОС.

Представление о доступных программе функциях ОС можно получить, выполнив входящую в состав МикроДОС программу MTEST, предназначенную для тестирования функций ОС и работающую в режиме диалога. Подробно сам интерфейс и структуры данных, которыми обмениваются транзитные программы и ОС МикроДОС, приведены в [2, ч. 3].

### Генерация ОС МикроДОС

**Адаптивность ОС МикроДОС.** Как уже отмечалось ранее, требования ОС МикроДОС к аппаратным ресурсам микроЭВМ достаточно общи. В результате эта ОС может использоваться на очень широком классе существующих 8-разрядных микроЭВМ. Этому в значительной мере способствует легкость генерации ОС МикроДОС на различных микроЭВМ и четкая структуризация процедур доступа к различным аппаратным ресурсам микроЭВМ.

**Структура ОС МикроДОС.** Функционально ОС МикроДОС подразделена на два модуля: базовой дисковой ОС (БДОС) и базовой системы ввода-вывода МикроДОС (МБСВВ). МБСВВ содержит процедуры, выполняющие простейшие (элементарные) операции обмена с устройствами конкретной микроЭВМ и имеющие стандартный интерфейс с вызывающими их процедурами из БДОС. БДОС образована процедурами, которые не зависят от особенностей аппаратуры микроЭВМ и выполняют операции управления ресурсами на качественно более высоком уровне, освобождая программиста от необходимости решения многих сложных и часто возникающих проблем.

Именно выделение модуля МБСВВ обеспечивает адаптивность МикроДОС, в частности, облегчает приспособление ОС к изменению конфигурации исходной микроЭВМ вследствие добавления (изменения) состава и возможностей аппаратурных компонентов.

**Средства и способы генерации МикроДОС.** МикроДОС поставляется в двух версиях, различающихся степенью отражения специфики конкретной аппаратуры микроЭВМ. Конкретные версии, уже сгенерированные для стандартной конфигурации микроЭВМ, подготовлены для микроЭВМ СМ 1803, Роботрон А5120, Роботрон 1715 и др. Открытая версия, содержащая дополнительные программы и документацию (облегчающие системному программисту самостоятельную генерацию ОС), поставляется для остальных микроЭВМ.

Для генерации ОС МикроДОС на конкретной микроЭВМ необходимо создать процедуры выполнения всех простейших операций обмена с устройствами этой микроЭВМ в рамках стандартного интерфейса обращений к этим процедурам из БДОС. Убедившись в правильности реализации процедур, системный программист объединяет модули МБСВВ и БДОС в сегмент исполняемых кодов ОС МикроДОС.

Обычно ОС располагается в области оперативной памяти с наибольшими доступными адресами; однако часть (неизменяемых) кодов ОС можно разместить в ПЗУ, а в оперативной памяти расположить только модифицируемые данные ОС. В последнем случае при запуске микроЭВМ ОС МикроДОС сразу будет начинать функционирование, и генерация системы на этом заканчивается. В первом случае необходимо еще разработать процедуру (называемую начальным загрузчиком), которая при запуске микроЭВМ будет считана с диска в память, получит управление и перенесет с диска в нужное место памяти весь исполняемый сегмент ОС МикроДОС. Генерация ОС МикроДОС завершается в этом случае созданием загрузочного диска ОС, на нескольких первых дорожках которого записываются начальный загрузчик и исполняемый сегмент ОС.

Учет изменений конфигурации микроЭВМ путем модификации уже существующей ОС возможен не всегда: например, нельзя добавлять новые процедуры или области для данных, но можно изменять некоторые параметры в таблицах или вносить изменения в процедуры, сопровождающиеся уменьшением их объема. Для некоторых микроЭВМ разработан набор служебных программ, выполняющих допустимые реконфигурации ОС МикроДОС.

Генерация ОС решает одновременно и вопрос об адаптации к конкретной микроЭВМ большинства связанных с ней программных пакетов. Однако для тех пакетов, которые вынуждены максимально использовать все функциональные возможности аппаратуры, может потребоваться дополнительная адаптация. Например, экранные редакторы нуждаются в настройке их на конкретный дисплей, а пакеты подготовки текста к печати — на конкретный принтер.

**Заключение**  
По результатам пробной эксплуатации программного окружения ОС МикроДОС в СССР и ГДР сделано общее заключение о большей комфортности прикладных пакетов МикроДОС и выражено пожелание расширить набор прикладных программ.

В настоящее время в составе системы МикроДОС отсутствует ряд важных прикладных пакетов: статистическая обработка данных, кроссовые средства программирования встраиваемых процессоров, инженерное проектирование, графика, настройка многофункциональных устройств ввода-вывода (в частности, высококачественных принтеров) на определенный режим работы, тесты оперативной памяти и других устройств, печать текстов в фоновом режиме, перепорядочение записей файла в порядке возрастания (убывания) содержимого тех или иных полей в записях и т. д.

В некоторых случаях, когда требуется составление программ на языке ассемблера (например, формирование библиотек стандартных функций для языков программирования высокого уровня), целесообразно создать макробibliothek, облегчающую работу программистов. В частности, учитывая известную сложность интерфейса программ с ОС, желательно иметь удобную макробibliothek вызовов МикроДОС. Пока МикроДОС не содержит никаких специализированных макробibliothек.

Авторы заинтересованы в организации совместных работ по указанным выше пакетам и другим возможным программным продуктам в рамках системы МикроДОС. Повышение комфортности и эффективности существующих программ, а также нахождение новых областей применения 8-разрядных микроЭВМ — проблемы очень важные и будут сохранять свою актуальность в ближайшие годы.

Телефон: 135-52-95, Москва

### ЛИТЕРАТУРА

1. Дудников Е. Е., Лоозе Й., Чуклов В. В., Долбилов Л. А. Переносимая дисковая операционная система МикроДОС // В кн.: Применение микропроцессорной техники, вып. 3. — М.: МЦНТИ / МНИИПУ, 1984. — С. 19—24.
2. МикроДОС: мобильная операционная система для микроЭВМ / Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)». — вып. 40 в 3-х частях. — М.: МЦНТИ / МНИИПУ, 1985.

3. Язык программирования БЕИСИК/Ф/Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)», вып. 44 в 3-х частях.— М.: МЦНТИ/МНИИПУ, 1986.
4. Системные программы МикроДОС/Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)», вып. 45.— М.: МЦНТИ/МНИИПУ, 1986.
5. Язык программирования Си/Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)», вып. 46.— М.: МЦНТИ/НИИСчетМаш, 1986.
6. Система подготовки документации ТЕКСТ/Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)», вып. 42.— М.: МЦНТИ/МНИИПУ, 1987.
7. Язык программирования БЕИСИК/Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)», вып. 41.— М.: МЦНТИ/НИИСчетМаш, 1987.
8. Пакеты программ телеобработки данных на микроЭВМ/Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)», вып. 47.— М.: МЦНТИ, 1987.
9. Д И А Л О Г — М.: Информационно-поисковая система/Сер. «Методические материалы и документация по пакетам прикладных программ (Библиотека МикроДОС)», вып. 50.— М.: МЦНТИ, 1987.

Статья поступила 20 февраля 1987

УДК 681.3.06

А. А. Анисимов, Г. А. Анисимов

## РЕЖИМ РАЗДЕЛЕНИЯ ВРЕМЕНИ В СИСТЕМАХ С ОДНОЗАДАЧНЫМ МОНИТОРОМ

Применение встроенных микропроцессорных систем (МС) (например, СМ 1600, СМ 1800, контроллеры на базе однокристалльной микроЭВМ К1816ВЕ48 [1] и однокристалльного микропроцессора К1801ВМ1 [2]) практически лишает возможности использования готовой операционной системы для организации режима ее работы. Управляющая программа работы такой МС обычно «зашита» в ПЗУ, а роль самой МС часто сводится к управлению одним автономным объектом, например роботом, станком с ЧПУ и т. д.

Учитывая, что большинство МС способно работать с адресным пространством в 64 Кбайт, а сами программы управления объектами не превышают 1...2 Кбайт, можно возложить на МС задачу управления другими объектами или процессами, организовав режим разделения времени. Это позволяет эффективнее использовать МС. Например, в автоматической линии по производству изолированной телефонной жилы такие параметры, как ток отжига жилы, температура массы в экструдере, линейная скорость изолирования и другие, контролируются и управляются отдельными локальными регуляторами. Оформив алгоритмы управления параметрами в виде отдельных программ и организовав режим разделения времени работы этих программ, управлять всей линией или подобным процессом можно с помощью одного контроллера, например на базе микропроцессора К1801ВМ1 [3]. Существенное достоинство К1801ВМ1 — собственная микропрограмма обращения к прерыванию по вектору 100 при подаче сигналов на 32-й (IRQ2) вход этой микросхемы.

Алгоритм организации режима разделения времени можно представить следующим образом:

1) определить вектор прерывания. Обозначим его через ВТИМ. В частности, ВТИМ=100. Такое же значе-

ние он имеет и в ЭВМ серии СМ-4, «Электроника 60»;

2) организовать индивидуальный стек для каждой программы, предусмотрев для него буфер в 50...100 слов;

3) задать квант времени, по истечении которого текущая программа приостановит свою работу и передаст управление следующей. Квант времени можно менять для различных программ;

4) написать собственную программу обработки прерывания по таймеру (диспетчер программ), предусмотрев в ней возможность сохранения необходимых параметров текущей программы, например внутренних регистров общего назначения, адреса указателя стека и восстановления ранее сохраненных параметров той программы, которой передается управление.

Используя операционную систему РАФОС с SJ-монитором, процесс организации многопрограммного режима осуществляется одним из следующих способов. Пишется управляющая программа на языке MACRO-11 (рис. 1). С помощью этой программы оператор с пультового терминала указывает те программы, которые он хочет запустить в работу (до 16 программ). Для увеличения числа программ до 32 необходимо задействовать второе слово оперативной памяти и т. д. По окончании данного диалога начинается последовательное обращение к командам монитора (выполнение операторов с меток А1 до А2). Суть этих команд состоит в том, что идет процесс последовательного парирования текстовых файлов РЈ.PAS (рис. 2) и МЈ.MAC (рис. 3) соответственно к файлам ВЕГ.PAS (рис. 4) и ВЕГ.MAC (рис. 5).

В задачу файлов МЈ.MAC входит организация начального запуска данной программы, сохранения регистров и вершины стека прерванной программы, возобновление работы данной программы с ранее прерванного

места. Для окончательного образования файлов Р00.PAS М00.MAC с ними объединяются файлы END.PAS (рис. 6) и END.MAC (рис. 7) соответственно. Только таким образом получаем файлы Р00.PAS и М00.MAC, которые в дальнейшем подвергаются компиляции, компоновке и образованию окончательной задачи Z00.SAV с помощью командного файла CMF.COM (рис. 8). Итоговая задача Z00 загружается автоматически. Чередование работы программы осуществляется диспетчером.

Приведенный процесс формирования программ носит общий характер. Его достоинство в том, что те программы, которые не подлежат выполнению, в оперативную память не загружаются. Запустить их в работу можно, обратившись вновь к формирователю программ.

При использовании встроенных систем практически известно число программ заранее, причем в основном они пишутся на ассемблере. В связи с этим отпадает необходимость в программе FOR.MAC, а используются только, как единое целое, объединенные файлы ВЕГ.MAC+МЈ.MAC+END.MAC. Единственным требованием в любом случае является, чтобы J-я программа начиналась с метки STJ (J=0116). Однако большинство операторов языка Паскаль и, в частности операторы ввода-вывода, работают со стеком и «портят» его [4]. Следовательно, при «выходе из кванта времени» необходимо выполнить команду типа MOV#STJ, \$RESR6 (где \$RESR6 — ячейка памяти в библиотеке Паскаля, хранящая начало буфера стека J-й программы).

Так как обычно прерывания наступают с частотой питающей сети, т. е. F=50 Гц, то минимальный квант времени не может быть меньше 20 мс. Следовательно, при необходимости уменьшения кванта времени следует повысить частоту прерывания, реализовав аппаратно генератор, и его выход подключить на вход прерывания микроЭВМ, в частности на 32-й вход микросхемы К1801ВМ1.

Заключение. Предложен алгоритм организации группового управления

## УВАЖАЕМЫЕ ЧИТАТЕЛИ!

Если вас интересуют статьи или подборки статей из отечественных и зарубежных журналов по микропроцессорной технике и персональным компьютерам, обращайтесь в службу «ИНФОРМТЕХНИКА» кооператива «Техника».

Наша служба также переводит на русский язык указанные вами или присланные нам материалы по микропроцессорной технике.

Ксерокопии статей при всех видах информационных услуг выставляются заказчику наложенным платежом.

Работы, связанные с просмотром журналов, т. е. составление тематических подборок, оперативная высылка ксерокопий статей по мере их опубликования, оплачиваются заказчиком предварительно согласно упомянутому ниже прейскуранту.

Прейскурант на наши услуги и бланки для оформления заказов на информационное обслуживание и переводы вы можете получить по почте, направив в наш адрес почтовую карточку с запросом этих материалов.

Наш адрес: 117334, Москва, Проспект 60-летия Октября, д. 5, корп. 3, «ИНФОРМТЕХНИКА»

```

. TITLE FOR
. ENABL LC
JSW =44
. MCALL .EXIT, .PRINT, .WRITE
. MCALL .READW, .TTINR, .ENTER
. MCALL .FETCH, .CLOSE, .LOOKUP
;.....
; открытие и чтение файла P.DAT
START: . FETCH =DRVR, =FTRP
      . BCC 10$
      . PRINT =S2
      . EXIT
10$: . LOOKUP =Y, #1, =FTRP
     . BCC 20$
     . PRINT =S3
     . EXIT
20$: . MOV RO, R3
     . ENTER =Y, #2, =FTRP, =R3
     . BCC 30$
     . PRINT =S4
     . EXIT
30$: . READW =Y, #1, =BF, =256, =0
     . BCC 40$
     . PRINT =S5
     . EXIT
;.....
40$: . TSTB BF ;конец ввода?
     . BEQ 60$ ;нет
     . CMQB =P, BF
     . BNE 50$
     . JSR PC, OFB
     . JSR PC, WR
; обращение к SJ-монитору
     . MOV =A3, Y
     . MOV =S2, Y+2
     . JMP T5
50$: . JMP T1
60$: . PRINT =S 70
     . MOV =Y, R1
; ввод номеров запускаемых программ
70$: . TTINR
     . BCS 70$
     . MOVB RO, R1
     . CMQB =12, (R1)
     . BNE 70$
; удаление запятых из буфера ввода
100$: . MOV =BF+4, R1
     . MOV =Y, RO
110$: . CMQB =, @RO
     . BNE 150$
     . TSTB (R0)
     . BR 110$
120$: . MOVB (R0)+, (R1)+
     . BR 110$
130$: . PRINT =S7
     . JMP 60$
140$: . ASL R3
     . BR 200$
150$: . CMQB =K, @RO
     . BNE 120$
; определение номеров запускаемых программ
; и установка соответствующих битов в слове
; BF+100, старший - крайний слева бит
; соответствует 16-ой программе
     . MOVB (R0)+, (R1)+
     . CLR R1
     . CLR R4
     . CLR BF+100
     . MOV =BF+4, R
.. 160$: . MOVB (R0)+, R1
     . SUB = 60, R1
     . MUL =12, R1 DAM1M

```

```

MOVB (R0)+, R2
SUB =60, R2
ADD R2, R1
MOV =1, R3
INC R4
CMP =17, R4
BEQ 130$
CMP R4, R1
BNE 140$
BIS R3, BF+100
CLR R1
CLR R4
CMQB =K, @RO
BNE 160$
INC BF
MOV BF+2, R2
CMQB =K, @RO
BNE 160$
INC BF
MOV BF, BF+100
MOVBF =P, BF
JSR PC, WR
MOV =A2, Y
MOV =A3, Y+2
JMP T5
CMQB =1, BF
BNE T3
MOV =BF+4, R2
BR T4
T3: . MOVB =P, M1+11
     . MOV =00, M1+12
     . MOVB =M, M2+11
     . MOV =00, M2+12
T4: . MOV @R2, M1+26
     . MOV (R2)+, M2+26
     . MOV R2, BF+2
     . JSR PC, WR
; обращение к SJ-монитору
     . MOV =A1, Y
     . MOV =A2, Y+2
T5: . JSR PC, PER
     . EXIT
;.....
; последовательность команд монитору
;.....
A1: . WORD A2-A1
M1: . ASCIZ /COPY DX1:BEQ.PAS+DX1:PJ.PAS DX1:P00.PAS/
     . EVEN
M2: . ASCIZ /COPY DX1:BEQ.MAC+DX1:MJ.MAC DX1:M00.MAC/
     . EVEN
     . ASCIZ /RU DX1:FOR/
     . EVEN
A2: . WORD A3-A2
     . ASCIZ /COPY DX1:P00.PAS+DX1:END.PAS DX1:P00.PAS/
     . EVEN
     . ASCIZ /COPY DX1:M00.MAC+DX1:END.MAC DX1:M00.MAC/
     . EVEN
     . ASCIZ /RU DX1:FOR/
     . EVEN

```

```

A32: .WORD      $2-A3
      .ASCIZ    /DX1:CMF/
      .EVEN
S2:   .ASCIZ    /ошибка FETCH/
S3:   .ASCIZ    /ошибка LOOKUP/
S4:   .ASCIZ    /ошибка ENTER/
S5:   .ASCIZ    /ошибка READW/
S6:   .ASCIZ    /ошибка WRITW/
S7:   .ASCIZ    /ошибка номера/
S10:  .ASCIZ    /укажите через запятую двузначные/ <12><15>
      .ASCIZ    /номера запускаемых программ/ <12><15>
      .ASCIZ    /заканчивайте ввод символом "R"/
      .EVEN
PTRP: .RAD50   /DX1P   DAT/
BF:   .BLKW    256..
Y:    .BLKW    100
; .....
; PER-п/п формирования команд
; * передачи управления SJ-монитору
; .....
PER:  MOV      #510,R0
      MOV      Y,R1
1$:   MOV      (R1)+,(R0)+
      CMP      R1,Y+2
      BLO     1$
      BIS      #4000,JSW
      CLR      R0
      RTS     PC
OBF:  MOV      #BF,R0
      MOV      #BF+76,R1
10$:  CLR      (R0)+
      CMP      R0,R1
      BNE     10$
      RTS     PC
WR:   .WRITW   #Y,#2,#BF,#256.,*0
      BCC     10$
      .PRINT  #S6
      .EXIT
10$:  .CLOSE   #1
      .CLOSE   #2
      RTS     PC
      DRVR    =.
      .END    START
      FOR.   MAC

```

Рис. 1. Формирователь программ

```

PROGRAM BEG (OUTPUT);
BEGIN

```

Рис. 4. Файл BEG.PAS

## РЕКЛАМНОЕ СООБЩЕНИЕ

Организация в интересах сторонних предприятий проводит:

технические консультации, работы по проектированию, программированию, тестированию устройств на основе программируемых логических интегральных схем (ПЛИС) типа K556PT1, K556PT2, K556PT3, K1556 ХЛ8, K1556 ХП4, K1556 ХП6, K1556 ХП8. Тел. 468-81-75, Москва

```

(*$ C
.GLOBAL STJ
.BLKW ,100
; .....
; *
; (* начало программы J*)
WRITELN ("запускается программа J");
(*$ C
; *
WRITELN ("выполняется программа J");
WRITELN (SQRT (J));
; *$ C
JMP MJ *

```

Рис. 2. Файл PJ.PAS

```

; .....
; .. организация адресов запуска
; .. J-ой программы J=01-16
; .....
ZJ:   .IRPC    X,012345
      MOV      R^X,-(SP)
      .ENDM
      MOV      SP,@Y+6
      MOV      #STJ,SP ; организация стека J-ой программы
      ;STJ -адрес запуска J-ой программы
      CLR      -(SP)
      MOV      #STJ,-(SP)
      MOV      #STJ,$RESR6
      JMP      @BF+106
; сохранение регистров
; предыдущей выполняемой программы
PJ:   .IRPC    X,012345
      MOV      R^X,-(SP)
      .ENDM
      MOV      SP,@Y+6; сохранение текущего
; адреса указателя стека J-1 -ой программы
      ADD     #2,Y+10
      MOV      @Y+10,SP ; восстановление адреса
; указателя стека J-ой программы
; восстановление регистров J-ой программы
      .IRPC    X,543210
      MOV      (SP)+,R^X
      .ENDM
      MOV      #STJ,$RESR6
      JMP      @BF+106

```

Рис. 3. Файл MJ.MAC

```

;.....
;   ассемблер MACRO V04.00
;.....
; пример составления работы программ
; с разделением времени
; с использованием SJ-монитора
;.....
.TITLE      BEG
.ENABL      LC
.MCALL      .EXIT, .PRINT, .WRITW, .READW, .TTINR
.MCALL      .FETCH, .CLOSE, .LOOKUP, .FNTER
.GLOBL      $RESR6
.GLOBL      Z01, Z02, Z03, Z04, Z05, Z06, Z07, Z08
.GLOBL      Z09, Z10, Z11, Z12, Z13, Z14, Z15, Z16
.GLOBL      P01, P02, P03, P04, P05, P06, P07, P08
.GLOBL      P09, P10, P11, P12, P13, P14, P15, P16
BTIM        =100 ; вектор прерывания таймера
T:          0 ; буфер оставшейся длительности
           ; времени работы программы
Y:          .BLKW 6
DTI:        5 ; задание длительности
           ; кванта времени 100 мс
BF:         .BLKW 256 ; номер единицы в слове
; BF+100 справа налево указывает запрос
; на запуск программы
;.....
; открытие и чтение файла P.DAT
;.....
START:      ; использовать рис. 1
;
;
;
;.....
40$        MOV     =VSTZ-4, Y+6
           MOV     =ADDR-2, Y ; организация начального
           ; адреса запуска программ
           MOV     =1, Y+4
           MOV     =1, BF+104
           MOV     BTIM, BF+106 ; сохранение
; системного адреса программы обработки
; прерывания по таймеру
           MOV     =OT, BTIM
           MOV     =1, T
50$:       TST     T ; ожидание прерывания от таймера
           BEQ     50$
;.....
; обработка прерываний по таймеру
; (диспетчер программ)
;.....
OT:         DEC     T ; конец кванта?

```

```

BEG        10$ ; да
JMP        @ BF+106 ; уход на системную
; программу обработки для
; отсчета календарного времени
10$:      ADD     =2, Y ; переход
; к адресу следующей программы
20$:      BIT     Y+4, BF+100; программу запускать?
           BEQ     40$ ; нет
           DEC     BF+104
           ASL     Y+4 ; переход
; к следующей программе
           ADD     =2, Y+6 ; переход к адресу указателя
; стека следующей программы
           CMP     =VSTZ, Y+10; начало
; программ продолжения?
           BNE     30$ ; нет
           MOV     =VSTZ, Y+6 ; переход
; на продолжение первой программы
30$:      MOV     @ Y, Y+2 ; формирование адреса перехода
           MOV     DTI, T ; задание кванта времени
           JMP     @ Y+2 ; переход на выполнение
; следующей программы
40$:      ASL     Y+4 ; все программы запущены?
           BNE     10$ ; нет
           TST     BF+104 ; одна программа?
           BNE     50$ ; нет
           MOV     BF+106, BTIM; восстановление
; системного адреса
           RTI
50$:      MOV     =1, BF+104
           MOV     =ADDR-2, Y; организация
; начального адреса для продолжения программ
           MOV     =VSTZ-2, Y+10 ; организация
; начального адреса стека для продолжения программ
           MOV     =1, Y+4
           BR      10$
;.....
; массив адресов запускаемых программ
;.....
ADRZ:      Z01, Z02, Z03, Z04, Z05, Z06, Z07, Z08
10$:      Z09, Z10, Z11, Z12, Z13, Z14, Z15, Z16
;.....
; массив адресов продолжения программ
;.....
ADRP:      P01, P02, P03, P04, P05, P06, P07, P08
10$:      P09, P10, P11, P12, P13, P14, P15, P16
;.....
VSTZ:      .WORD 0
           .BLKW 16 ; буфер адресов
; хранения указателей для стека программ

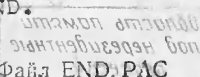
```

Рис. 5. Файл BEG, MAC

```

DEL/NOQ   DX1:Z00.SAV
R PASCAL
DX1:P00=DX1:P00
MACRO DX1:P00/OBJ;DX1:
MACRO DX1:M00/OBJ;DX1:
LINK PASCAL,DX1:P00,DX1:M00/EXE:DX1:Z00
DEL/NOQ DX1:P00.*,DX1:M00.*
RU DX1: Z00

```

END. 

DRVР = .  
.END START

Рис. 7. Файл END, MAC

Рис. 8. Командный файл CMF, COM

автономными объектами с помощью одной управляющей МС в режиме разделения времени. Алгоритм успешно внедрен в производство на одном из предприятий электротехнической промышленности.

Телефон: 361-35-50, Москва

### ЛИТЕРАТУРА

1. Акуней Ю. А., Антонов Б. В. и др. Универсальный микроконт-

роллер «Электроника МК-48» на основе однокристалльной микро-ЭВМ K1816BE48 // Микропроцессорные средства и системы.— 1986.— № 3.— С. 39.

2. Ку克林 В. В., Калинин Н. М., Бобров Ю. А. Отладочный модуль на базе однокристалльного микропроцессора K1801BM1 // Микропроцессорные средства и системы.— 1986.— № 2.— С. 33—37.

3. Дшхунян В. Л., Борщенко

Ю. И. и др. Однокристалльные микропроцессоры комплекта БИС серии K1801 // Микропроцессорные средства и системы.— 1984.— № 4. С. 12—18.

4. Бутото И. Д., Самочадин А. В., Усанова Д. В. Программирование на алгоритмическом языке Паскаль для микро-ЭВМ. Учебное пособие.— Л.: Изд. ленинградского университета, 1986.

Статья поступила 6 июня 1987

УДК 681.3.066+681.31.31:324

В. В. Корнеев, М. С. Тарков

## ОПЕРАЦИОННАЯ СИСТЕМА МИКРОМАШИНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ МИКРОС

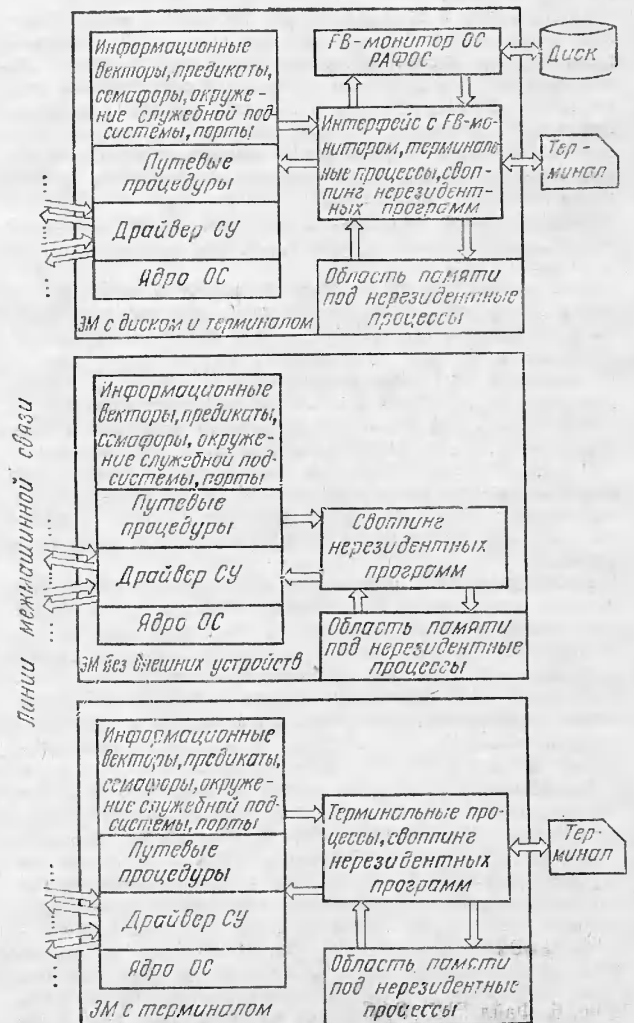
Вычислительная система (ВС) с программируемой структурой — это совокупность элементарных машин (ЭМ), объединенных сетью линий связи [1, 2], программно управляемой из этих машин. В ВС МИКРОС [3] в качестве ЭМ выступает микроЭВМ, совместимая с «Электроникой 60» и оснащенная системным устройством. Системное устройство подключается к общей шине микроЭВМ, аналогично контроллеру внешнего устройства (ВУ), и обеспечивает подключение к микроЭВМ от одной до шести дуплексных линий межмашинной связи. Конфигурация входящих в состав ВС машин различна: от процессора с памятью до микроЭВМ с полным комплектом ВУ. Граф межмашинных связей ВС — произвольный, степени его вершин не превышают шести. Выбор графа определяется областью применения ВС. При отсутствии сосредоточенного коммутатора, объединяющего микроЭВМ, и соединении ЭМ линиями по схеме «точка — точка» функционирование ВС при отказе части машин и линий связи продолжается.

В основу построения операционной системы (ОС) ВС МИКРОС положены следующие посылы: ОС должна быть способна функционировать на любом подмножестве исправных машин, связанных исправными линиями; число терминалов и внешних устройств-носителей ОС произвольно, и в ВС существуют машины, к которым не подключены устройства обоих вышеуказанных типов; ВС должна обеспечивать реализацию функций сетевой ЭВМ по управлению удаленными файлами и заданиями, а также режим высокопараллельных вычислений.

Удовлетворить эти требования можно, создав ОС ЭМ, которые при слиянии образуют ОС ВС в целом. Разработанная для ВС МИКРОС ОС КОСМОС (коллектив однородных самостоятельных модульных операционных систем) имеет двухуровневую организацию. Первый уровень — резидентная ОС (РОС) со средствами порождения, уничтожения и организации взаимодействия систем процессов, распределенных по машинам ВС. Второй уровень — совокупность систем совместно протекающих процессов, каждая из которых реализует либо прикладную программу пользователя, либо некоторую программу ОС. РОС загружается в каждую ЭМ, и поэтому должна иметь минимальный объем. Дублирование одного и того же программного кода в оперативной памяти каждой ЭМ служит платой за распределенность ОС и организацию живучести.

Структура ОС КОСМОС приведена на рисунке. Состав РОС варьируется в зависимости от комплектации ЭМ внешними устройствами. В ЭМ без ВУ в РОС входят: 1) ядро ОС [4...6]; 2) драйвер системного устройства (СУ) [7]; 3) окружение служебной подсистемы [2]

с универсальными путевыми процедурами и портами программ ОС [8, 9]; 4) совокупности программ, осуществляющих свопинг нерезидентных процессов, включая чтение текста программы с удаленного ВУ или из оперативной памяти другой ЭМ, загрузку нерезидентного процесса в память и запуск на исполнение; 5) область данных РОС, включающая: а) информационные векторы порожденных в ЭМ процессов; б) структуры данных, задающие предикаты порожденных процессов [2, 4]; в) семафоры и порты программ пользователей, в которых сохраняются состояния этих программ в случае, если они затираются при загрузке на их место в памяти другой нерезидентной программы; г) стеки резидентных программ; д) свободную область памяти,



из которой выбираются буферы для временного хранения сообщений.

При комплектации ЭМ терминалом в РОС добавляются процессы обслуживания терминала. Если к ЭМ подключено ВУ, являющееся носителем ОС РАФОС, то в РОС включаются процессы, реализующие интерфейсы с FB-монитором ОС РАФОС [9, 10], и сам FB-монитор со своими структурами данных.

В машинах с подключенными ВУ-носителями ОС РАФОС резидентная ОС размещается в оперативном разделе. Фоновый раздел предназначается для нерезидентных программ, загружаемых посредством свопинга. Разделы переключаются процессом РОС, реализующим интерфейс с FB-монитором.

### Функционирование ОС КОСМОС

Резидентные процессы обслуживания терминала обеспечивают диалог с пользователями ВС. Число терминалов в ВС произвольно, и все терминалы равноправны. Пользователь посредством терминала ведет диалог с системными программами ОС КОСМОС (монитор ОС РАФОС, системные программы ОС РАФОС, системные программы ОС КОСМОС для управления файлами и заданиями, а также поддержки параллельных вычислений). При вводе с терминала полученное сообщение помещается в соответствующий семафор (порт), приспосабливаемый системой программ, которой предназначено сообщение. Все семафоры, играющие роль портов системных и пользовательских программ, находятся в области памяти РОС.

Поступление в семафор (порт) сообщения — это условие активизации ядром ОС процесса, обрабатывающего это сообщение. Если процесс нерезидентный, то посредством резидентного процесса, осуществляющего свопинг, он загружается в область памяти для нерезидентных процессов и пускается на исполнение. Результатом будет сообщение в соответствующем порте для выдачи на терминал или для активизации другой системной или прикладной программы. Сообщение, предназначенное для другой машины, будет помещено в порт, приспосабливаемый процессу, осуществляющему эту процедуру.

При необходимости обращение к ВУ в ходе исполнения программ выполняется через процесс интерфейса с монитором ОС РАФОС. Этот процесс формирует команды монитора и передает их на исполнение в FB-монитор. При этом стандартная ОС РАФОС играет роль как бы драйвера для ВУ, подключенных к ЭМ.

Работа ВС [9] инициируется любой машиной с терминалом и ВУ-носителем ОС. При этом ручные операции выполняются только при начальной загрузке одной машины (загружается резидентная ОС и пускается на выполнение нерезидентная программа INIT). Все остальные ЭМ запускаются далее посредством этой программы.

Программа INIT определяет состояние соседних ЭМ и, если среди них есть неиницированные, то включает их питание и пересылает содержимое всей памяти ЭМ (в которой она исполняется) в память этих ЭМ. Далее управление в этих ЭМ передается на программу INIT. По завершении инициации соседних ЭМ формируется слово конфигурации машины, биты которого отмечают работоспособные линии связи и внешние устройства, подключенные к ЭМ. В соответствии со словом конфигурации ЭМ формируется состав резидентной ОС. Например, если к ЭМ подключен терминал, то порождаются процессы его обслуживания. Инициация ВС завершается генерированием сообщения, активизирующего построение служебной подсистемы с универсальной путевой процедурой. По окончании построения подсистемы на всех терминалах высветится приглашение к работе.

Резидентная ОС служит фундаментом построения всех системных программ ОС, реализующих функции сети ЭВМ и организующих параллельные вычисления. Функции сети ЭВМ реализует интерпретатор языка уп-

равления заданиями (ИЯУЗ), параллельными вычислениями управляет интерпретатор языка управления параллельными вычислениями (ИЯУПВ) [9]. Оба интерпретатора реализуются нерезидентными параллельными программами ОС КОСМОС и имеют в каждой ЭМ порт, в который в виде сообщений помещаются команды языков управления параллельными вычислениями и заданиями. Команда каждого из языков может поступить либо с терминала, либо из любой программы, протекающей в ЭМ. Интерпретаторы исполняют команду и передают результаты в специфицированные процессы.

Язык управления заданиями — это язык команд монитора РАФОС, расширенный до уровня управления всей ВС [11]. Язык команд расширяется введением в спецификацию файлов адресов машин (к которым подключены ВУ-носители файлов) специальной интерпретацией команд монитора, допускающей обращение к объектам, распределенным по ВС. Задание в спецификации файла адреса машины не обязательно, но сокращает время выполнения команды (исключается поиск файла по имени по всем машинам ВС). Это дает возможность пользователю работать с ВС, как с одной машиной, оснащенной всеми ВС.

Язык управления параллельными вычислениями включает в себя команды для выполнения следующих действий:

- порождения подсистем из виртуальных машин, связанных виртуальными линиями и обладающих заданными ресурсами (числом машин, объемом оперативной памяти и т. д.), а также заданным типом графа межмашинных связей подсистемы (дерево, кольцо, линейка, решетка, произвольный связный граф и т. д.);
- загрузки параллельных программ в подсистемы и инициации их работы;
- уничтожения подсистем;
- организации отказоустойчивых вычислений.

Организация параллельных вычислений в ОС КОСМОС предусматривает предварительную настройку ресурсов на исполняемую параллельную программу. После формирования подсистемы и загрузки в ее машинные ветви параллельной программы подсистема функционирует подобно специализированной систолической или потоковой ВС. Блоки ОС КОСМОС при этом не работают. Все ресурсы отдаются параллельной программе. Для ее эффективного исполнения используется «замораживание» в ЭМ всех процессов, кроме тех, что исполняют параллельную программу [5, 9]. При замораживании процесса (т. е. удалении его из расписания ОС на время существования подсистемы) межмашинные обмены выполняются специализированными драйверами. Они обеспечивают максимальную скорость передачи данных по линиям связи подсистемы. В результате организуется монопрограммный режим работы машин подсистемы над одной параллельной программой. По окончании параллельной программы возобновляется исполнение всех загруженных в машины процессов. Программы «замораживания» и «размораживания» процессов (около 200 слов оперативной памяти) — компоненты РОС.

Работа программ в ЭМ согласуется ядром ОС ЭМ [2, 4, 6], которое базируется на расширенной концепции семафоров. Ядро ОС интерпретирует примитивы ОС, предназначенные для динамического распределения памяти ЭМ, а также для порождения, уничтожения вычислительных процессов и организации их взаимодействия (синхронизации и обмена данными) в пределах одной ЭМ. В совокупности с драйвером системного устройства ядро ОС позволяет порождать и уничтожать процессы в соседней машине и выполнять взаимодействия процессов, протекающих в этих машинах. Ядро ОС ЭМ состоит из подпрограмм, реализующих операции над семафором. В зависимости от области применения ВС пользователь может из этих подпрограмм сгенерировать требуемое ему ядро ОС (от 1,5 до 3,5К слов).



Драйвер системного устройства (2,5К слов) организует обмена данными и командами между непосредственно соединенными линиями связи машинами, а также для включения питания в соседней ЭМ пересылкой информации в нес из данной ЭМ [7]. В ВС МИКРОС применена байториентированная процедура передачи сообщений на основе протокола BSC. Адаптация ОС КОСМОС для использования с другими средствами межмашинной связи, например в ВС с общей шиной, требует лишь написания другого драйвера с той же функцией, что и у драйвера системного устройства.

Набор универсальных путевых процедур [8, 9] предназначен для реализации схем межмашинных обменов в пределах подсистем и всей системы в целом. Фактически путевые процедуры распространяют функции ядра ОС на всю систему. Объем памяти, отводимый в РОС под путевые процедуры, включая необходимые для их функционирования структуры данных, — 3К слов. При необходимости объем можно сократить, используя специализированные путевые процедуры вместо универсальных.

Загрузка (своппинг) нерезидентной программы Q в фоновый раздел оперативной памяти ЭМ и запуск этой программы на исполнение инициируются программой ИСПОЛНИТЕЛЬ при выполнении условий ее готовности, определяемой ядром ОС ЭМ. Очередность готовых процессов на исполнение устанавливается в соответствии с их приоритетами. Одно из условий готовности — наличие разрешения на загрузку, которое дает находящаяся в оперативной памяти нерезидентная программа P. Другое условие готовности программы Q к исполнению — наличие сообщения во входном порту этой программы. Порт нерезидентной программы располагается в оперативном задании. При загрузке очередной программы Q программа P в оперативной памяти ЭМ стирается. При новых условиях готовности программы P вновь происходит ее загрузка и запуск на исполнение, а программа, находящаяся при загрузке в оперативной памяти, стирается. Таким образом, делится память фоновой раздела между нерезидентными программами ОС.

В зависимости от места расположения файла нерезидентной программы своппинг ее можно инициировать различными способами. В момент инициации загрузки файл нерезидентной программы может находиться: 1) на ВУ ЭМ<sub>а</sub>, в которую программа загружается; 2) на ВУ ЭМ<sub>а</sub>, адрес которой известен в ЭМ<sub>з</sub>; 3) в оперативной памяти ЭМ<sub>с</sub>, соседствующей с ЭМ<sub>а</sub>.

В первом случае загрузка и исполнение нерезидентной программы инициируются передачей в монитор из оперативного задания команды RUN NAME, где NAME — имя загружаемой программы.

Во втором случае из ЭМ<sub>а</sub> в ЭМ<sub>а</sub> передается запрос процессу ЧИТАТЕЛЬ на считывание файла NAME с ВУ. Для ограничения объема памяти, занимаемой РОС, процесс ЧИТАТЕЛЬ представлен фоновым заданием. Оно инициируется в ЭМ программой ИСПОЛНИТЕЛЬ, т. е. передачей в монитор команды RUN RW, где RW, SAV — файл, содержащий тело процесса ЧИТАТЕЛЬ. Блоки файла NAME передаются из ЭМ<sub>а</sub> в ЭМ<sub>а</sub>, где загружаются в фоновый раздел процессом ЗАГРУЗЧИК, после чего инициируется ее исполнение.

В третьем случае из ЭМ<sub>с</sub> в ЭМ<sub>с</sub> посылается запрос на передачу (копирование) из ЭМ<sub>с</sub> в ЭМ<sub>а</sub> программы фоновой раздела. Процесс ПЕРЕДАТЧИК поблочно передает программы фоновой раздела из ЭМ<sub>с</sub> в ЭМ<sub>а</sub>. Процесс ЗАГРУЗЧИК в ЭМ<sub>а</sub> принимает блоки программ из ЭМ<sub>с</sub> и записывает их в фоновый раздел.

Отметим, что программа, передаваемая из фоновой раздела одной ЭМ в фоновый раздел другой ЭМ, должна либо быть реентерабельной, либо содержать в себе подпрограмму инициации структур данных этой программы. При этом программа, посылая сообщение в соседнюю машину, тем самым инициирует свою пере-

сылку в эту машину. Иначе говоря, программа как бы «протягивает» себя по машинам ВС. Третий способ загрузки программы исключает необходимость посылать запрос на считывание файла программы в ту ЭМ, где она хранится на ВУ. Тем самым удается снизить накладные расходы на загрузку программы. Объем средств для загрузки нерезидентных программ — около 1К слов.

Терминальный процесс ВХОД<sub>тт</sub> принимает командные строки с клавиатуры терминала. Работа пользователя состоит в подаче команды и ожидании ответа о ее исполнении. Только после завершения исполнения данной команды возможна подача следующей. Для аварийного ввода следующей команды до завершения данной предусмотрена подача с клавиатуры специального разрешающего ввода символа с кодом CTRL/O. Пользователь может приостанавливать и возобновлять вывод на терминал подачей кодов CTRL/S и CTRL/Q соответственно.

В терминальном процессе ВХОД<sub>тт</sub> на экран терминала данной ЭМ выдаются строки информации, которая может быть выработана в произвольной ЭМ системе в ответ на команду, данную с клавиатуры данной ЭМ. Объем памяти, выделенной под терминальные процессы, — 250 слов.

Взаимодействие РОС с монитором ОС РАФОС реализует процесс ПЕРЕКЛЮЧАТЕЛЬ [9, 10], который инициирует исполнение в данной ЭМ команд монитора РАФОС, переданных из любой ЭМ системы. Исполнение команды инициируется записью ее в кольцевой буфер ввода терминала с последующим выполнением системной макрокоманды SPND. Строки информации, выдаваемой на экран терминала при исполнении системной программы, которая интерпретирует команду РАФОС, перехватываются процессом ПЕРЕКЛЮЧАТЕЛЬ. Затем они передаются в виде сообщений в удаленную ЭМ, из которой поступила команда монитора. Перехват основан на модификации программы обработки прерываний по выводу на экран терминала, входящей в состав резидентного монитора системы РАФОС. Строка символов записывается этой программой в буфер, после чего фоновое задание блокируется по выводу на терминал. Управление передается оперативному заданию, где на основе буфера со строкой формируются сообщение, которое далее передается в удаленную ЭМ. Общий объем программ работы с терминалом и средств сопряжения РОС с монитором — около 750 слов.

Использование в составе ОС КОСМОС других ОС (ОСРВ, RSX-11, UNIX) требует лишь создания новых программ, реализующих интерфейс с этими ОС (занесение команд в буфер ввода с терминала и перехват информации, выдаваемой на терминал), а также программ преобразования форматов файлов.

Итак, из модулей РОС можно скомпоновать однородные способные самостоятельно функционировать ОС машин ВС. При их слиянии в каждой ЭМ могут коллективно использоваться общие нерезидентные программы ОС. ОС КОСМОС проходит опытную эксплуатацию в составе макета ВС МИКРОС. Ее можно использовать в вычислительных комплексах различного назначения, а также в качестве основы для выработки технических требований и технического задания на аппаратурно-программные средства ВС в микроэлектронном исполнении со встроенной (кремниевой) ОС.

Телефон: 35-23-71, Новосибирск

## ЛИТЕРАТУРА

1. Евреинов Э. В., Хорошевский В. Г. Однородные вычислительные системы. — Новосибирск: Наука, 1978. — 318 с.
2. Корнеев В. В. Архитектура вычислительных систем с программируемой структурой. — Новосибирск: Наука, 1985. — 166 с.

3. Димитриев Ю. К., Корнеев В. В., Хорошевский В. Г. Вычислительная система с программируемой структурой МИКРОС.— В кн.: Вычислительные системы с программируемой структурой (Вычислительные системы; Вып. 94).— Новосибирск, 1982, с. 3—15.
4. Корнеев В. В., Монахов О. Г., Тарков М. С. Ядро операционной системы ЭМ вычислительной системы с программируемой структурой.— В кн.: Однородные вычислительные системы (Вычислительные системы; вып. 90).— Новосибирск, 1981, с. 22—42.
5. Задорожный А. Ф., Корнеев В. В., Тарков М. С. Об организации коммуникаций между процессами в вычислительной системе МИКРОС.— В кн.: Распределенная обработка информации (Вычислительные системы; Вып. 105).— Новосибирск, 1984, с. 70—84.
6. Исследование архитектуры вычислительных систем с программируемой структурой. Разработка принципов построения операционной системы. Ядро операционной системы / Отчет ИТПМ СО АН СССР; Руководитель темы: В. Г. Хорошевский. Исполнители: В. В. Корнеев, М. С. Тарков, А. Ф. Задорожный. № Г.Р. 81054452, инв. № 0286.0064597.
7. Димитриев Ю. К., Задорожный А. Ф., Корнеев В. В. Элементарная машина вычислительной системы с программируемой структурой МИКРОС.— В кн.: Вычислительные системы с программируемой структурой (Вычислительные системы; Вып. 94).— Новосибирск, 1982, с. 16—32.
8. Монахов О. Г., Монахова Э. А. Организация межмашинных взаимодействий в вычислительной системе с программируемой структурой МИКРОС.— В кн.: Распределенная обработка информации (Вычислительные системы; Вып. 105).— Новосибирск, 1984, с. 85—104.
9. Исследование архитектуры вычислительных систем с программируемой структурой. Разработка принципов построения операционной системы. Базовая операционная система / Отчет ИТПМ СО АН СССР; Руководитель темы: В. Г. Хорошевский. Исполнители: В. В. Корнеев, О. Г. Монахов, Э. А. Монахова, М. С. Тарков, А. Ф. Задорожный, Ю. Н. Потапова.— № Г.Р. 81054552, инв. № 0286.0058560.
10. Тарков М. С. Организация удаленного исполнения команд монитора РАФОС в вычислительной системе МИКРОС // Микропроцессорные средства и системы.— 1986.— № 6.— С. 34—36.
11. Монахова Э. А., Корнеев В. В., Монахов О. Г. Организация распределенного управления заданиями в вычислительной системе МИКРОС // УСнМ.— 1987.— № 5.— С. 31—33.

Статья поступила 15 ноября 1987.

## ОБМЕН ОПЫТОМ

УДК 681.3.069 : 681.327.23

М. А. Гайворонский

### ПРОСТОЙ СПОСОБ АДАПТАЦИИ РАЗВИТЫХ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ РАБОТЫ СО СТАНДАРТНЫМ ДИСПЛЕЕМ МИКРОСИСТЕМЫ

В настоящее время в серийно выпускаемых и специализированных микросистемах широко применяются микроЭВМ, использующие операционные системы типа CP/M. Существуют сотни типов различных CP/M-машин, совместимых по системе команд и файловой системе, но различающихся характеристиками внешних

устройств, в том числе дисплеев. В связи с этим возникает проблема эффективного использования в конкретной CP/M-машине значительных объемов программных средств (ПС), разработанных к настоящему времени для эксплуатации совместно с ОС CP/M.

В частности, это относится к прикладным ПС, используемым режим «экранного редактирования», например: редакторам текста, табличным калькуляторам, системам управления базами данных (СУБД).

Пакеты прикладных ПС обычно содержат в своем составе программу специализации на конкретный тип устройства отображения (INSTALL-программу), однако настройку ПС затрудняют специфические требования к управлению отображением в конкретном дисплее, предъявляемые INSTALL-программой.

Пример требования — распознавание в дисплее короткой последовательности, непосредственно указывающей координаты позиции, в которую должен переместиться курсор на экране. Соответствующая функция не реализована в дисплее «Символ» [1] и терминале TC-7063 [2]. Возможен обратный случай, когда предполагается возможность установки в дисплее курсора при помощи одиночных кодов управления. Такую INSTALL-программу неудобно использовать совместно с дисплеем ВТА-2000 [3].

Указанные трудности могут быть преодолены в CP/M-машинных с помощью реализации в драйвере дисплея недостающих функций управления отображением. Таким способом можно осуществлять эмуляцию развитого в функциональном отношении дисплея, используя достаточно простое устройство отображения.

В операционной системе CP/M-машин управление от прикладной программы драйвером внешних устройств производится с помощью унифицированного программного интерфейса ввода-вывода, использующего точки входа, содержащиеся в модуле базовой системы управления вводом-выводом (BIOS) [4], поэтому одним из способов замены драйвера в таких машинах является перекодирование BIOS, ее трансляция и подстановка в ОС в соответствии с [5].

Значительно более эффективной оказывается подстановка эмулирующего драйвера без перекодирования BIOS. Сам драйвер дисплея, как правило, несложен и может быть написан разработчиком аппаратных средств, а для его встраивания в ОС достаточно знать адрес точки входа в драйвер вывода на системную консоль, а также адрес начала свободной зоны ОЗУ, расположенной после области загрузки BIOS и обозначаемой далее меткой PRBASE. В указанную зону будет загружаться эмулирующий драйвер.

Преимущества способа подстановки: не требует использования исходного текста BIOS и позволяет производить оперативную настройку драйвера дисплея CP/M-машин на требуемый режим.

На рисунке приведен исходный текст драйвера, предназначенного для работы дисплея «Символ» в экранном режиме совместно с СУБД dBASE-II [6] в составе систем СО-04, СО-05 [7]. Для использования драйвера необходимо с помощью программы INSTALL этой СУБД произвести ее настройку, указав дисплей ADM-3A [8], частичную эмуляцию которого обеспечивает приведенный драйвер. В процессе его работы осуществляется позиционирование курсора в соответствии с ESCAPE-последовательностью, применяемой в этом дисплее, а также ряд вспомогательных функций.

Драйвер располагается на том же гибком диске, что и СУБД, в формате исполняемого файла и вызывается перед загрузкой СУБД. В начале программы находится вспомогательный фрагмент, пересылающий содержательную часть в свободную область памяти. Этот же фрагмент осуществляет замену адреса в точке входа (соответствующей выводу на системную консоль), и управление передается подпрограмме «горячего старта» ОС. В результате при выводе на консоль каждый

; АССЕМБЛЕР RMAC/OC 1800

\*\*\*\*\*

; ЭМУЛЯТОР ДИСПЛЕЯ АДМ-ЗА ДЛЯ РАБОТЫ

; СО-05 С СУБД В ЭКРАННОМ РЕЖИМЕ

\*\*\*\*\*

; АДРЕСА:

CO EQU 0F807H ; ИТАТНОГО ДРАЙВЕРА;
CONOUT EQU 0E200H ; ТОЧКИ ВХОДА В BIOS'E;
WBOOT EQU 0E203H ; 'ГОРЯЧЕГО СТАРТА';
PRBASE EQU 0E800H ; ЗАГРУЗКИ ЭМУЛЯТОРА.

; УПРАВЛЯЮЩИЕ КОДЫ ДИСПЛЕЯ "СИМВОЛ":

CDOWN EQU 1AH ; КУРСОР ВНИЗ;
CUP EQU 19H ; -- ВВЕРХ;
CLEFT EQU 08H ; -- ВЛЕВО;
CRIGH EQU 18H ; -- ВПРАВО;
HOME EQU 0CH ; В НАЧАЛО ЭКРАНА;
ERASE EQU 1FH ; ОЧИСТИТЬ ЭКРАН.

CSEG ; ПЕРЕСЛАТЬ ЭМУЛЯТОР
LXI H, 100H;MODEY-PRBASE; ИЗ ЗАГРУЗОЧНОГО
LXI D, MODEY ; В РАБОЧИЙ АДРЕС
LXI B, MODND ; ДЛИНА ПРОГРАММЫ
MVE10: MOV A,M
STAX D
INX H
INX D
DCX B ; ПЕРЕСЫЛКА ЗАВЕРШЕНА ?
MOV A,B
ORA C
JNZ MVE10-PRBASE+100H; НЕТ -- ПРОДОЛЖИТЬ;
; ПОДСТАВИТЬ

MODEL: LXI H, MODEY ; АДРЕС ЭМУЛЯТОРА ВМЕСТО
SHLD CONOUT+1; АДРЕСА ИТАТНОГО ДРАЙВЕРА;
MVI C,HOME ; ОЧИСТИТЬ ЭКРАН
CALL CO
MVI C,ERASE
CALL CO
JMP WBOOT ; ЭМУЛЯТОР ГОТОВ К РАБОТЕ

MUND: ;СОБСТВЕННО ЭМУЛЯТОР ДИСПЛЕЯ

MODEY: PUSH PSM ; СОХРАНИТЬ РЕГИСТРЫ
PUSH H ; НА ВРЕМЯ РАБОТЫ
PUSH B ; ЭМУЛЯТОРА;
JMP ONSOL ; АНАЛИЗ КОДА СИМВОЛА:
ONSOL: MOV A,C ; ЕСЛИ
CPI 20H ; КОД НЕ УПРАВЛЯЮЩИЙ,
JNC DIRECT ; ТО ПРОСТОЙ ВЫВОД;
; ИНАЧЕ
CPI 08H ; ПРИ ПЕРЕВОДЕ СТРОКИ
JZ FEED ; ОТРАБОТАТЬ 'BC',
CPI 18H ; ПРИ КОДЕ АВТОРЕГИСТРА -
JZ VCCNT ; ESCAPE-СЕРИЮ,
CPI 1AH ; ПРИ КОДЕ ОЧИСТКИ АДМ-ЗА -
JZ HOERS ; 'HOME'+ 'ERASE';
CPI 1FH ; ПРИ КОДАХ СЕРОСА "СИМВОЛА",
JZ DIRECT
CPI 0DH ; 'ВОЗВРАТ КАРЕТКИ',
JZ DIRECT
CPI 08H ; 'КУРСОР ВЛЕВО',
JZ DIRECT
CPI 18H ; 'КУРСОР ВПРАВО' И
JZ DIRECT
CPI 0CH ; 'HOME' - ВМВОД СИМВОЛА
JZ DIRECT ; БЕЗ ИЗМЕНЕНИЯ;
JMP FINIR ; ОСТАЛЬНЫЕ КОДЫ ИГНОРИРОВАТЬ

; УСТАНОВИТЬ КУРСОР В ТРЕБУЕМУЮ КОЛОНКУ
; ВХОД: C = НОМЕР КОЛОНКИ

COLNO: MOV A,C ; ЕСЛИ НОМЕР КОЛОНКИ БОЛЬШЕ
CPI 08H ; ДОПУСТИМОГО ЧИСЛА КОЛОНОК,
JNC FINIR ; ТО ИГНОРИРОВАТЬ ЕГО;
MOV B,C ; ИНАЧЕ -- УСТАНОВИТЬ КУРСОР,
MVI C,CRIGH ; ПЕРЕДВИГАЯ ЕГО ВПРАВО
COLM10: DCR B ; ДО ТРЕБУЕМОЙ КОЛОНКИ.
JM FINIR ;

CALL CO
JMP CLM10 ; ПЕРЕВОД СТРОКИ:
; 'КУРСОР ВНИЗ' --
FEED: MVI C,CDOWN ; ВЫДАТЬ НА ЭКРАН
CALL CO ; ОЧИСТКА ЭКРАНА:
JMP FINIR ;
HOERS: MVI C,HOME
CALL CO
MVI C,ERASE
CALL CO
MVI C,HOME ; ВЫВОД СИМВОЛА БЕЗ ИЗМЕНЕНИЯ:
DIRECT: CALL CO
JMP FINIS

; УСТАНОВКА КУРСОРА ПО КОДАМ ESCAPE-СЕРИИ,
; СОДЕРЖАЩЕЙ НОМЕР СТРОКИ В ТРЕТЬЕЙ ПОЗИЦИИ
; СЕРИИ, НОМЕР КОЛОНКИ - В ЧЕТВЕРТОЙ

LINO: MOV A,C ; ЕСЛИ НОМЕР СТРОКИ БОЛЬШЕ
CPI 24 ; ДОПУСТИМОГО ЧИСЛА СТРОК,
JNC FINIR ; ТО ИГНОРИРОВАТЬ ЕГО;
MVI C,HOME ; ИНАЧЕ -- КУРСОР В НАЧАЛО
CALL CO ; ЭКРАНА,
POP B ; ВОССТАНОВИТЬ
PUSH B
MOV A,C ; НОМЕР СТРОКИ;
MVI C,CDOWN ; ПЕРЕДВИГАТЬ КУРСОР ВНИЗ
DCR A
MOV B,A
JZ LNO20
LNO10: CALL CO
DCR B ; ДО ТРЕБУЕМОЙ СТРОКИ;
JNZ LNO10
LNO20: LXI H, COLNO ; СЛЕДУЮЩИЙ КОД СЕРИИ ОБРА-
SHLD ONSOL-2 ; БАТЫВАТЬ КАК НОМЕР КОЛОНКИ.
JMP FINIS

; ОБРАБОТКА КОДА АВТОРЕГИСТРА

VCCNT: LXI H, VCAHE ; ПЕРЕАДРЕСОВАТЬ СЛЕДУЮЩИЙ СИМ-
SHLD ONSOL-2 ; БОЛ НА АНАЛИЗ ESCAPE-СЕРИИ
JMP FINIS
VCAHE: MOV A,C ; ЕСЛИ ВЫВОДИМЫЙ СИМВОЛ --
CPI 3DH ; ЗАГЛОВОК ПОЗИЦИИ,
JZ VXES ; ТО ОБРАБОТАТЬ ЕГО;
JMP FINIR ; ИНАЧЕ -- ВЫЙТИ ИЗ ОБРАБОТКИ AP2.
VXES: LXI H, LINO ; ПЕРЕАДРЕСОВАТЬ СЛЕДУЮЩИЙ СИМВОЛ
SHLD ONSOL-2 ; НА ОБРАБОТКУ НОМЕРА СТРОКИ,
JMP FINIS
FINIR: LXI H, ONSOL ; УСТАНОВИТЬ АДРЕС ОБРАБОТКИ СИМ-
SHLD ONSOL-2 ; ВОЛА В НАЧАЛЬНОЕ СОСТОЯНИЕ,
FINIS: POP B ; ВОССТАНОВИТЬ
POP H ; РЕГИСТРЫ.
POP PSM
RET
MODND: END

Пример адаптируемого драйвера

раз будет осуществляться переход на драйвер дисплея,
загруженный с диска. Символ, подлежащий обработке,
будет при этом находиться в регистре С.

Для правильного вычисления адресов, используемых
описанным фрагментом программы, при получении за-
грузочного модуля редактору связей в ключе старто-
вого адреса должно быть указано значение метки
PRBASE.

В пачале собственно драйвера расположена коман-
да перехода JMP ONSOL. непосредственно предше-
ствующая метке ONSOL. При обработке обычного симво-
ла переход осуществляется на метку с этим именем;
если на входе драйвера появляется код авторегистра
(1BH), то производится замена адреса в команде пе-
рехода. Последующие символы будут интерпретировать-

ся как принадлежащие ESCAPE-последовательности и управление будет передаваться другим подпрограммам драйвера, ответственным за обработку соответствующего кода. Каждая из этих подпрограмм в свою очередь записывает в команде перехода адрес подпрограммы обработки следующего символа. При обработке последнего символа ESCAPE-последовательности в команду перехода вновь заносится адрес ONSOL. Такой алгоритм позволяет добиться повышенной скорости драйвера по сравнению с табличными методами вычисления адресов перехода.

Приведенный драйвер может быть легко модифицирован для оперативной настройки дисплеев различных CP/M-машин для взаимодействия с программами, использующими работу с системной консолью в режиме «экранного редактирования».

Телефон: 272-96-79, Киев

## ЛИТЕРАТУРА

1. Изделия промышленности средств связи: Каталог серии «Аппаратура передачи данных». — М.: ЦООНТИ «ЭКОС», 1983.
2. Средства ввода, вывода информации электронных вычислительных машин. — М.: НИИЭИР, 1985.
3. Средства вычислительной техники: Номенклатурный каталог. — М.: ЦНИИ ТЭИ приборостроения, 1985.
4. Дамке М. Операционные системы микроЭВМ. — М.: Финансы и статистика, 1985.
5. CP/M 2.2 System Alteration Guide // Pacific Grove, CA, Digital Research, 1979. — 212 p.
6. dBASE-II Assembly Language Relational Database Management System: User Manual. Ashton-Tate, 1981.
7. Малиновский Б. М., Палагин А. В., Погорелый С. Д., Сигалов В. И., Слободянюк А. И., Яковлев Ю. С. Комплекс средств микропроцессорной техники // УСПМ. — 1982. — № 6. — С. 12—14.
8. Machover C., Ed. Computer Terminals Review. — GML Corp. 1980. — 316 p.

Статья поступила 23 марта 1987.

УДК 681.3

А. Ю. Сидристый

## ТРЕХУРОВНЕВЫЙ ИНФОРМАЦИОННЫЙ ИНТЕРПРЕТАТОР КОМАНДНЫХ СТРОК SHURSH V3.6

Система SHURSH предназначена для интерактивного диалога с малоподготовленным пользователем на уровне директив операционной системы ОС РВ версии 3.0 (RSX11M V4.0/4.1). Предполагаемой средой использования системы являются машины серий CM-4, CM 1420 или любые другие вычислительные машины с совместимой архитектурой.

Интерпретатор SHURSH V3.6 — базовый вариант для последующих версий системы, который учитывает возможность его модификации и может быть применен для ОС РВ (версии 3.0/3.1 и выше) без каких-либо доработок. В отличие от стандартного интерпретатора CLI система может вести диалог на русском языке, проводить обучение пользователя, классифицировать свои возможности. Помимо этого она характеризуется простым способом получения справочной информации и автоматическим созданием командных файлов.

Основой системы является трехуровневый режим меню. Это позволяет пользователю за конечный интервал времени выполнить необходимую ему команду или серию команд. В меню обеспечивается доступ ко всем командам монитора MCR и некоторым наиболее часто используемым программам обслуживания операционной системы (BRU, BCD, PIP, SRD, FMT, EDI, FOR, MAC, ТКВ, ...).

Для пояснения непонятных пользователю терминов существует словарь интерпретатора, в который входит глоссарий из документации ОС РВ и HELP файлы.

При диалоге используются информационные файлы, поясняющие суть выполняемой работы, основные команды редактора и т. д.

Для реализации системы выбран кнопочный вариант, как наиболее простой для восприятия пользователя. Он предполагает, что за каждой клавишей терминала закреплена определенная операция или функция. При необходимости можно подписать прямо на клавише ее назначение.

В зависимости от характера задач пользователя в системе SHURSH предусмотрены три уровня детализации существующего сервиса. Все операции объединены в логически близкие группы. В каждой группе выделена одна наиболее часто выполняемая операция. Она является операцией первого уровня и используется при классификации клавиатуры дисплея по назначению. Если данная операция не удовлетворяет пользователя, он может перейти на следующий уровень детализации.

На втором уровне в режиме меню предлагается выбрать операцию из группы близких операций. Таким образом, второй уровень расширяет первый по функциональной полноте системы.

На третьем уровне пользователь может сформировать в выходной строке некоторые из характеристик выбранной им операции. При установке определенной характеристики в меню происходит отметка о том, что она установлена. После всех установок строка передается на обработку монитору MCR. Предлагаемая трехуровневая схема проста для понимания и легко усваивается.

Данный вариант программы обеспечивает реинтерпретацию работы системы для нескольких терминалов. Число одновременно обслуживаемых терминалов устанавливается при генерации системы (по умолчанию оно равно десяти). Система диспетчеризации и квантования времени (супервизор интерпретатора) обеспечивает последовательную работу всех подключенных к интерпретатору терминалов с соблюдением текущих режимов и характеристик терминалов.

Подсистема сопровождения предназначена для изменения некоторых характеристик интерпретатора, касающихся каждого отдельно взятого терминала:

установить фиктивное выполнение команд интерпретатора. Фиктивность выполнения заключается в том, что формируемая для задачи MCR... командная строка не передается интерпретатором на обработку. Характеристика полезна при ознакомлении с системой;

установить высвечивание командной строки. В ходе меню интерпретатором формируется определенная командная строка. При установке данной характеристики командная строка перед передачей на обработку задаче MCR... будет распечатана на терминале пользователя. Данная возможность полезна при обучении пользователя работе с ОС;

установить расширенное згакопредставление для терминала. В системе становится возможным обслуживание терминалов, допускающих прописные и строчные латинские или русские буквы;

дополнить набор сервисных команд вплоть до полного переопределения клавиатуры терминала.

Технические характеристики системы: процессор типа CM-4, число одновременно обслуживаемых терминалов — 10, объем оперативной памяти — 5 Кслов (при этом, если задана неактивна, она выгружается в файл выгрузки, исключаясь из конкуренции за процессорное время и оперативную память), объем используемой справочной информации (при загрузке в информационном режиме) — 1500 блоков на внешнем ЗУ, операционная система RSX11M V4.0/4.1.

Телефон: 43-42-97, Новосибирск

Сообщение поступило 2 марта 1987

## ПРОГРАММНЫЙ ЭМУЛЯТОР КОМАНД РАСШИРЕННОЙ АРИФМЕТИКИ

В процессоре микроЭВМ ДВК-2 отсутствуют команды расширенной арифметики (MUL, DIV, ASH, ASHC, FMUL, FDIV, FADD, FSUB). Это в общем случае исключает программную совместимость ДВК-2 с мини-ЭВМ CM-4 и микроЭВМ «Электроника 60». Однако во многих случаях необходимо обеспечить перенос программ, содержащих команды расширенной арифметики, на ДВК-2.

Для решения задач переноса был выбран подход, предложенный в [1]. Известно, что если процессор «не понимает» какую-либо команду, то возникает прерывание по вектору 10. Разработан и встроены в резидентную часть SJ-монитора ОС РАФОС эмулятор, который перехватывает прерывание по вектору 10 и проверяет, какой командой вызвано прерывание.

В случае, если прерывание вызвано одной из эмулируемых команд (MUL, DIV, ASH, ASHC, FMUL, FDIV, FADD, FSUB), то эмулятор выполняет требуемую операцию в полном соответствии с [2] (с учетом методов адресации и с установкой необходимых признаков результата в слове состояния процессора), а затем возвращает управление команде, следующей после эмулируемой. Если прерывание вызвано какой-либо командой, отличной от эмулируемой, то управление передается на ту часть ОС РАФОС, которая отвечает за обработку прерываний по вектору 10.

Таким образом, ни программа пользователя, ни (даже) операционная система РАФОС «не знают», что работают через эмулятор.

Самый удобный способ встроить эмулятор в ОС РАФОС — это поместить исходный текст эмулятора (см. рисунок) непосредственно внутри исходного текста резидентной части монитора (в файл RMONSJ.MAC), сразу после меток

TRAP4::

TRAP10::

по до первой команды, помеченной этими метками в исходном варианте текста RMONSJ.MAC, а затем сгенерировать SJ-монитор обычным образом.

Однако данный способ нельзя реализовать, если в распоряжении имеются только накопители на гибких магнитных дисках (НГМД), так как для генерации ОС РАФОС требуется около двух тысяч блоков дисковой памяти.

Для пользователей, имеющих в распоряжении только НГМД, можно предложить другой, менее удобный и экономичный метод, который мы рассмотрим на примере ОС ФОДОС-2 (RT-11.SJ V05.00).

При загрузке монитора ФОДОС-2 размер памяти, имеющейся в аппаратной конфигурации, определяется системным загрузчиком автоматически. Но можно сделать так, чтобы SJ-монитор использовал только нижние 27 Кслов из 28 Кслов. В верхние же 1 Кслов можно разместить эмулятор. Для этого необходимо:

- 1) поместить при помощи программы SIPP в ячейки 1030, 1032 и 1034 файла монитора FMONSJ.SYS значения 000240, 012704 и 154000 соответственно;
- 2) записать начальный загрузчик модифицированного монитора по команде COPY/BOOT:

```

      .DSABL   LSB
      BCS     EMLST
      JMP     EMLEND

EMLST: .WORD 162706,000002,010616,010546,010446,010346
        .WORD 010246,010146,010046,002766,000006,000014
        .WORD 011600,000016,162700,000002,011001,042701
        .WORD 000007,022701,075000,001003,004767,001540
        .WORD 000410,022701,075010,021062,004767,001524
        .WORD 062767,000100,001412,160405,001412,100005
        .WORD 005405,006202,006003,075003,000404,006054
        .WORD 006200,006001,077503,032767,000010,001354
        .WORD 001405,005100,005101,062701,000001,003500
        .WORD 032767,000100,001332,001403,005102,005103
        .WORD 062703,000001,005502,042767,000001,001310
        .WORD 000301,005500,060200,100106,005267,001274
        .WORD 005100,005101,062701,000001,005500,000476
        .WORD 022701,075020,001024,004767,001352,005305
        .WORD 010446,010546,012704,000031,012746,000000
        .WORD 005005,006216,006003,006200,006001,103003
        .WORD 060305,005516,060216,077411,000443,022701
        .WORD 075030,001161,004767,001274,005702,001004
        .WORD 052777,000013,001154,000037,005305,005405
        .WORD 010446,010546,012704,000031,012746,000000
        .WORD 005005,006305,006116,160301,005600,160200
        .WORD 100004,000301,005500,060200,000402,052705
        .WORD 000001,006301,006100,077417,012600,010501
        .WORD 012605,012604,005700,001006,005701,001004
        .WORD 052777,000004,001044,000462,032700,001000
        .WORD 001403,006200,006001,005204,032700,000400
        .WORD 001004,006301,006100,005304,000771,062701
        .WORD 000001,005500,032700,001000,001360,006200
        .WORD 006001,042700,000200,000405,062705,000200
        .WORD 100004,052777,000012,000746,000434,022705
        .WORD 000377,100004,052777,000002,000730,000425
        .WORD 000305,006005,050000,032767,000001,000714
        .WORD 001405,052700,100000,052777,000010,000676
        .WORD 016704,000676,062714,000004,011404,010024
        .WORD 010124,000167,000616,012600,012601,012602
        .WORD 012603,012604,012605,013716,000244,000257
        .WORD 000207,042701,000777,022701,072000,001024
        .WORD 004767,001000,004767,001010,001607,100004
        .WORD 005405,006203,077502,000402,006303,077502
        .WORD 004767,001162,010311,001002,052716,000004
        .WORD 000167,000510,022701,073000,001031,004767
        .WORD 001002,004767,000732,001411,100005,005405
        .WORD 006203,006004,077503,000403,006304,006103
        .WORD 077503,004767,001100,010311,001004,010412
        .WORD 001003,052710,000034,010412,000167,000420

```

```

      .WORD 022701,070000,001063,004767,000712,012746
      .WORD 000000,011102,100002,005405,005116,005705
      .WORD 100002,005405,005116,012703,000020,005004
      .WORD 000241,006004,006005,105001,060204,077306
      .WORD 016703,000372,005704,001006,005705,001006
      .WORD 005213,000004,005726,000415,052713,000001
      .WORD 006004,006003,005726,001407,052713,000010
      .WORD 005104,005103,062705,000001,005504,032710
      .WORD 000100,001001,015421,010511,000167,000246
      .WORD 022701,071000,001105,000476,000536,012746
      .WORD 000000,032710,000100,001627,012102,010103
      .WORD 005705,003006,001003,032714,000001,000417
      .WORD 005216,005405,005702,100007,062716,000011
      .WORD 005102,005103,062705,000001,005502,063030
      .WORD 006102,020502,105005,052714,000002,005726
      .WORD 000167,000130,012704,000017,012700,000001
      .WORD 006303,006102,169502,003003,060502,077406
      .WORD 000402,000003,077411,016704,000130,005703
      .WORD 001003,052714,000004,000406,032716,000001
      .WORD 001403,005403,052714,000010,032726,000010
      .WORD 001401,005402,016241,010341,000167,000024
      .WORD 012600,012601,012602,012603,012604,012605
      .WORD 005726,000261

```

```

JMPRM: JMP EMLEND
      .WORD 012600,012601,012602,012603,012604,012605
      .WORD 000024,012605,016645,000002,011645,010566
      .WORD 016705,000006,000002,000000,000000,000000
      .WORD 012100,011161,010004,006104,000304,062704
      .WORD 177400,001402,005700,000200,162704,000200
      .WORD 006301,006100,042700,170000,000207,010500
      .WORD 062704,000024,042714,000017,010467,177712
      .WORD 000207,006301,042701,177764,006001,062701
      .WORD 000004,000207,005007,177670,004767,177732
      .WORD 011001,004767,177744,010167,177654,011161
      .WORD 010102,004767,177646,003003,062767,000101
      .WORD 177632,010103,010201,010002,010403,062701
      .WORD 000004,004767,177615,010003,062767,000001
      .WORD 177602,000207,010102,032710,000100,000101
      .WORD 005722,011103,011204,016700,177554,042705
      .WORD 177600,032705,000400,001402,052705,177600
      .WORD 005705,000207,011001,004767,177614,011005
      .WORD 042705,177717,022705,000000,001412,011105
      .WORD 000040,001411,022705,000000,001412,162711
      .WORD 000417,011105,000416,000002,000000,000001
      .WORD 000002,011105,000406,017605,000020,000001
      .WORD 000002,000020,001120,000101,000101,000001
      .WORD 001401,011505,011001,000101,000101,000301
      .WORD 004767,177476,004767,177452,000207,103002
      .WORD 005210,000001,010305,100002,052710,000010
      .WORD 074311,100002,052710,000002,000207,000000
      .WORD 000000,000000,003000
      .ENABL   LSB

```

EMLEND: Эмулятор команд расширенной арифметики

Исходный текст эмулятора команд расширенной арифметики

3) разместить на системной дискете под именем GETEML загрузочный модуль следующей программы:

```

CALL .EXIT
START: MOV #137, JMPRMN
      MOV @#10, JMPRMN+2
      MOV #154000, R0
      MOV R0, @#10
      MOV #EMLST, R1
LOOP:  MOV (R1)+, (R0)+
      CMP R1, #EMLEND
      BNE LOOP
      .EXIT

```

```

;*****
;* СЮДА СЛЕДУЕТ ПОМЕСТИТЬ ИСХОДНЫЙ ;
;* ТЕКСТ ЭМУЛЯТОРА *
;*****
.END START

```

4) в стартовый командный файл поместить команду R GETEML.

После этого при загрузке системы загрузчик будет оставлять свободными верхние 1 Кслов ОЗУ, программа GETEML, запускаемая из стартового командного файла, будет размещать там эмулятор и переадресовывать на него вектор прерывания 10.

В заключение отметим, что эмулятор может быть встроен не только в ОС РАФОС, но и в любую другую ОС или отдельную программу, предназначенную для эксплуатации на ДВК-2.

Адрес: 644011, Омск, Харьковская, 15, кв. 42

#### ЛИТЕРАТУРА

1. Сянггер М. Мини-ЭВМ PDP-11: программирование на языке ассемблера и организация машины.— М.: Мир, 1984.— С. 148—152.
2. Вигдорчик Г. В., Воробьев А. Ю., Праченко В. Д. Основы программирования на ассемблере для СМ-ЭВМ.— М.: Финансы и статистика, 1983.— С. 59—65.

Статья поступила 6 июля 1987

УДК 681.3.06

Н. А. Кривенко

### ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ ВУМС-001-01

Написана ОС для вычислительной управляющей микросистемы ВУМС-001-01, состоящей из микроЭВМ «Электроника ДЗ-28» с объемом памяти 32 Кбайт на-

копителя на гибких магнитных дисках «Электроника ГМД-70» или «Электроника ГМД-7012», таблично-знакового индикатора РИН-609, термопечатающего устройства ТПУ15ВВП80-002 и блока сопряжения БС-01. Система реализована на компакт-кассете (программа компилятор) и системном диске

Компилятор загружается в ЭВМ со встроенного накопителя. В начальном состоянии ОС занимает верхние 6 Кбайт памяти и системную зону машины с адреса 32000. Около 26 Кбайт выделяются под данные. Распределение памяти динамическое, при необходимости можно изменять и топологию памяти. В функцию компилятора входят загрузка с системного диска и запуск программных модулей ОС и прикладных программ. В состав компилятора включены наиболее часто используемые системные подпрограммы.

На диске записаны утилиты дисковых операций, Макроассемблер, модули управления данными, блоки поддержки слабосвязанной кросспроцессорной системы «Электроника ДЗ-28» — «Электроника К1-10», прикладные программы. Реализованы макропроцессоры, обеспечивающие последовательное прохождение прикладных программ по заранее составленной инструкции в режиме программного свопинга.

Система позволяет эффективно использовать вычислительные возможности микроЭВМ «Электроника ДЗ-28».

340098, Донецк, пр. Ильича, 16, Мединститут,  
Центральная научно-исследовательская лаборатория

Сообщение поступило 2 июля 1987

УДК 681.3

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

### БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ НА МИКРОПРОЦЕССОРЕ КР580ИК80А

При быстром преобразовании Фурье (БПФ) на микропроцессоре КР580ИК80А основное время затрачивается на выполнение умножения чисел. В [1] соответствующая процедура оформлена в виде базовой операции БПФ, причем на ее выполнение затрачивается около 90% от времени работы программы. С учетом современных программ умножения [2] и при эффективном использовании системы команд упомянутого микропроцессора удается значительно ускорить выполнение БПФ без дополнительных аппаратных затрат.

Ниже приводится текст базовой операции БПФ, полностью идентичный по входным данным и результатам работы программе, описанной в [1], но выполняемой в 2,5 раза быстрее (при тактовой частоте 2,5 МГц менее чем за 1800 мкс против 4430 в [1]), что позволяет более чем в 2 раза сократить общее время работы всей программы БПФ из [1].

При выполнении программы используется подпрограмма ML0 — перемножения двух 16-разрядных чисел с последующим округлением результата, в основе которой программа ML2 из [2].

Телефон: 3-63-31, Сыктывкар

#### ЛИТЕРАТУРА

1. Цифровые фильтры и устройства обработки сигналов на интегральных микросхемах / Ф. Б. Высоцкий, В. И. Алексеев, В. Н. Пачин, М. Н. Ушкар, В. М. Федоров; Под ред. Ф. Б. Высоцкого.— М.: Радио и связь, 1984.— 216 с.
2. Глазов А. Б., Костарев С. А., Суханова Е. В. Эффективные программы быстрого умножения для микропроцессора КР580ИК80А // Микропроцессорные средства и системы.— 1986.— № 5.— С. 43—44.

LHLD	1, 2	ЗАПИСАТЬ В RG HL ЧИСЛО RE В ИЗ ЯЧЕЕК 1, 2
MOV	B, H	RE В → RG BC
MOV	C, L	
LHLD	30, 31	ЗАПИСАТЬ В RG HL ЧИСЛО RE W ИЗ ЯЧЕЕК 30, 31
CALL	ML0	ВЫЧИСЛИТЬ RE В * RE W И ПЕРЕСЛАТЬ РЕЗУЛЬТАТ В RG DE
PUSH	D	RE В * RE W → В СТЕК
LHLD	32, 33	ЗАПИСАТЬ В RG HL ЧИСЛО IM W ИЗ ЯЧЕЕК 32, 33
CALL	ML0	ВЫЧИСЛИТЬ RE В * IM W И ПЕРЕСЛАТЬ РЕЗУЛЬТАТ В RG DE
PUSH	D	RE В * IMW → В СТЕК
LHLD	28, 29	ЗАПИСАТЬ IM B В RG HL ИЗ ЯЧЕЕК 28, 29
MOV	B, H	IM B → RG BC
MOV	C, L	
LHLD	30, 31	ЗАПИСАТЬ В RG HL ЧИСЛО RE W ИЗ ЯЧЕЕК 30, 31
CALL	ML0	ВЫЧИСЛИТЬ IM B * RE W И ПЕРЕСЛАТЬ РЕЗУЛЬТАТ В RG DE
POP	H	RE В * IM W → RG HL
DAD	D	RE В * IM W + IM B * RE W → RG HL
SHLD	1, 2	ЗАПOMНИТЬ RG HL В ЯЧЕЙКАХ 1, 2
LHLD	32, 33	ЗАПИСАТЬ В RG HL ЧИСЛО IM W ИЗ ЯЧЕЕК 32, 33
CALL	ML0	ВЫЧИСЛИТЬ IM W * IM B И ПЕРЕСЛАТЬ РЕЗУЛЬТАТ В RG DE
POP	B	RE В * RE W → RG BC
MOV	A, C	RE В * RE W — IM W * IM B → RG BC
SUB	E	
MOV	C, A	
MOV	A, B	
SBB	D	
MOV	B, A	
LHLD	24, 25	ЗАПИСАТЬ В RG HL ЧИСЛО RE A
DAD	B	RE A + RE B * RE W — IM W * IM B → RG HL
XCHG		RG HL ↔ RG DE
LHLD	6, 7	ЗАПИСАТЬ В RG HL АДРЕС ЧИСЛА RE A' ИЗ ЯЧЕЕК 6, 7
MOV	M, E	ЗАПOMНИТЬ МЛАДШИЕ РАЗРЯДЫ ЧИСЛА RE A'
INX	H	ВЫЧИСЛИТЬ АДРЕС СТАРШЕГО БАЙТА ЧИСЛА RE A'
MOV	M, D	ЗАПOMНИТЬ СТАРШИЙ БАЙТ ЧИСЛА RE A'
PUSH	H	ЗАСЛАТЬ В СТЕК АДРЕС СТАРШЕГО БАЙТА ЧИСЛА RE A'
LHLD	1, 2	ЗАПИСАТЬ В RG HL ЧИСЛО RE В * IM W + IM B * RE W ИЗ ЯЧЕЕК 1, 2
XCHG		RG HL ↔ RG DE
LHLD	26, 27	ЗАПИСАТЬ В RG HL ЧИСЛО IM A ИЗ ЯЧЕЕК 26, 27
DAD	D	IM A + RE B * IM W + IM B * RE W → RG HL
XCHG		RG HL ↔ RG DE
POP	H	АДРЕС СТАРШЕГО БАЙТА ЧИСЛА RE A' → RG HL
INX	H	ВЫЧИСЛЕНИЕ АДРЕСА МЛАДШЕГО БАЙТА ЧИСЛА IM A'
MOV	M, E	ЗАПOMНИТЬ МЛАДШИЙ БАЙТ ЧИСЛА IM A'
INX	H	ВЫЧИСЛИТЬ АДРЕС СТАРШЕГО БАЙТА ЧИСЛА IM A'
MOV	M, D	ЗАПOMНИТЬ СТАРШИЙ БАЙТ ЧИСЛА IM A'
LHLD	24, 25	ЗАПИСАТЬ В RG HL ЧИСЛО RE A
MOV	A, L	ВЫЧИСЛИТЬ RE A — (RE B * RE W — IM W * IM B) И ЗАПИСАТЬ В RG BC
SUB	C	
MOV	C, A	
MOV	A, H	
SBB	B	
MOV	B, A	
LHLD	8, 9	ЗАПИСАТЬ В RG HL АДРЕС ЧИСЛА RE B' ИЗ ЯЧЕЕК 8, 9
MOV	M, C	ЗАПOMНИТЬ МЛАДШИЕ РАЗРЯДЫ ЧИСЛА RE B'
INX	H	ВЫЧИСЛИТЬ АДРЕС СТАРШИХ РАЗРЯДОВ ЧИСЛА RE B'
MOV	M, B	ЗАПOMНИТЬ СТАРШИЕ РАЗРЯДЫ ЧИСЛА RE B'
INX	H	ВЫЧИСЛИТЬ АДРЕС ЧИСЛА IM B'
XCHG		RG HL ↔ RG DE
LHLD	1, 2	ЗАПИСАТЬ В RG HL ЧИСЛО RE В * IM W + IM B * RE W ИЗ ЯЧЕЕК 1, 2
MOV	B, H	И ПЕРЕСЛАТЬ ЭТО ЧИСЛО В RG BC
MOV	C, L	
LHLD	26, 27	ЗАПИСАТЬ В RG HL ЧИСЛО IM A ИЗ ЯЧЕЕК 26, 27
MOV	A, L	ВЫЧИСЛИТЬ МЛАДШИЙ БАЙТ ЧИСЛА IM A — (RE B * IM W + IM B * RE W) И ЗАПOMНИТЬ ЕГО ПО СООТВЕТСТВУЮЩЕМУ АДРЕСУ
SUB	C	ВЫЧИСЛИТЬ СТАРШИЙ БАЙТ ЧИСЛА IM A — (RE B * IM W + IM B * RE W) И ЗАПOMНИТЬ ЕГО
STAX	D	
INX	D	
MOV	A, H	
SBB	B	
STAX	D	
SBB	B	
MOV	B, A	
XCHG		
RET		
ML0: XCHG		ПОДПРОГРАММА УМНОЖЕНИЯ ДВУХ 16 РАЗРЯДНЫХ ЧИСЕЛ С ОКРУГЛЕНИЕМ
CALL	ML2	
DAD	H	
RNC		
INX	D	
RET		

Текст программы в кодах микропроцессора:  
 ML0 — A90: EB CD 00 03 29 D0 13 C9  
 БОП — AA6: 2A C0 03 44 4D 2A DD 03 CD 90  
 ABO: 0A D5 2A DF 08 CD 90 0A D5 2A DB 08 44 4D 2A DD  
 ACO: 08 CD 90 0A E1 19 22 C0 08 2A DF 08 CD 90 0A C1  
 ADO: 79 53 4F 78 9A 47 2A D7 08 09 EB 2A C5 08 73 23  
 AEO: 72 E5 2A C0 08 EB 2A D9 08 19 EB E1 23 73 23 72  
 AFO: 2A D7 08 7D 91 4F 7C 98 47 2A C7 08 71 23 70 23  
 B00: EB 2A C0 03 44 4D 2A D9 08 7D 91 12 13 7C 98 12  
 B10: EB C9  
 ML2 — 800: см. [2, с. 44]  
 8C0: таблица размещения информации в ОЗУ (см. [1, с. 179])

Сообщение поступило  
6 июля 1987

## ОБЪЯВЛЕНИЕ — РЕКЛАМА

Научно-производственное объединение «Пищепромавтоматика» предлагает

### Систему обработки форм документов СОФОРД —

эффективное программное средство при обработке табличной информации на ПЭВМ «Искра 226».

ППП СОФОРД значительно ускоряет работу программиста; язык программирования — БЕЙСИК.

ППП СОФОРД выполняет расчеты в условиях, наиболее приближенных к естественным.

ППП СОФОРД обеспечивает высокую скорость исполнения директив программиста, максимально прост в использовании.

Основные характеристики: Время чтения формы документов с МД — 5...10 с.

Время перелистывания страниц формы на экране — 1 с.

Оперативная память, занимаемая системными модулями — до 12 Кбайт.

Объем памяти, занимаемой пакетом на МД — около 20 Кбайт.

Наибольшая ширина формы на экране — 252 символа.

Опыт применения ППП СОФОРД в течение трех лет показал высокие эксплуатационные качества пакета.

STEP — компактный программный модуль системного назначения, облегчающий работу программиста на ЭВМ «Искра 226».

Реализует функции: работа с сектором МД (чтение, корректировка содержимого, запись, распечатывание);

работа с индексом каталога МД (копирование, восстановление по распечатке, сортировке, определение сектора индекса по имени файла, сдвиг зоны каталога, «сжатие» МД);

работа с файлами. STEP занимает всего 60 секторов МД, предельно прост и удобен в эксплуатации; язык программирования — БЕЙСИК 02.

Заявки направлять по адресу: 270059, Одесса, ул. Краснова, 6. Институт «Пищепромавтоматика»

УДК 681.3.06

С. П. Брылев

## ПРОГРАММИРОВАНИЕ ВСТРОЕННЫХ МИКРОПРОЦЕССОРОВ С ПРИМЕНЕНИЕМ КОМПИЛЯТОРА ПАСКАЛЬ/МТ+

В связи с выбором компилятора для программирования встроенных систем были рассмотрены реализации, позволяющие получать кодовые сегменты программ МП КР580ВМ80 для записи в ПЗУ — ПЛ/М-80, Фортран-80, БЕЙСИК-80 (интерпретатор + компилятор), Паскаль/МТ+. Предпочтение отдано компилятору Паскаль/МТ+ версии 5.5, применение которого облегчает разработку хорошо структурированных программ (в отличие от Фортрана и БЕЙСИКа) и поддерживает вычисления с вещественными числами (отсутствуют в ПЛ/М-80). Эффективность кода, генерируемого Паскалем/МТ+, достаточно высока, хотя и ниже, чем в ассемблере, в 1.5...3 раза. Кроме того, в компилятор включены средства повышения эффективности кода, такие как дизассемблер и встроенный мини-ассемблер, которые помогают при необходимости вручную оптимизировать участки программы.

Для работы пакета программ Паскаль/МТ+ необходима инструментальная система, аналогичная СМ 1800, или «Электроника К1-30», снабженная накопителем на 203-мм гибких дисках и ОЗУ 64 Кбайт, а также ОС СР/М (ОС 1800). Объем всех файлов пакета Паскаль/МТ+ и необходимых вспомогательных программ (экранный редактор текста, перемещающий макроассемблер) составляет около 400 Кбайт.

Нестандартные средства Паскаля/МТ+. Кроме средств стандартного Паскаля данная реализация включает в себя встроенные подпрограммы манипуляции с битами, команды работы с портами ввода-вывода, встроенный мини-ассемблер и ряд дополнительных процедур и функций. Функция TSTBIT (переменная, номер\_бита) возвращает значение «истина», если разряд переменной с номером «номер\_бита» имеет значение «Лог. 1», в противном случае возвращается значение «ложь». Процедуры

SETBIT (переменная, номер\_бита) и  
CLRBIT (переменная, номер\_бита)

соответственно устанавливают и сбрасывают разряд «переменной» с номером «номер\_бита». Функции  
SHL (переменная, число) и  
SHR (переменная, число)

производят сдвиг «переменной» соответственно вправо и влево на «число» разрядов, устанавливая освобождающиеся разряды в «0». Функции

H1 (переменная) и  
L0 (переменная)

возвращают значения соответственно старших и младших восьми разрядов 16-разрядной переменной, а процедура

SWAP (переменная)

заменяет местами старший и младший байты переменной. Функция

ADDR (переменная)

возвращает значение адреса переменной, доступной на данном уровне вложения. В качестве переменной в данном случае может выступать также имя процедуры или функции или индексированная переменная. Например, если в программе описан массив А [1..20] целых чисел, то в результате выполнения оператора

START := ADDR(A[1])

переменная START получит значение адреса начала массива. Процедура

WAIT (номер\_порта, маска, полярность)  
генерирует последовательность

МЕТКА:	IN	номер порта	
	ANI	маска	
	IZ	метка	; полярность = fals
		порта	; или INZ, если
			; полярность = true

Паскаль/МТ+ позволяет обращаться к описанным по умолчанию массивам INP [0..255] и OUT [0..255], причем в массив OUT можно только записывать, а из массива INP — только читать. При этом генерируются коды команд МП КР580ВМ80 INP и OUT с соответствующими значениями адресов портов. Если адрес порта ввода-вывода является не константой, а переменной, он дополнительно должен быть заключен в круглые скобки:

ВВОД := INP [(порт\_ввода)]

В Паскале/МТ+ имеется средство для ввода данных в тело процедуры или функции. Данные можно вводить в виде чисел или мнемонических имен, принятых в ассемблере:

INLINE(\$ 2A/\$ 4C/\$ 22);

В средстве INLINE применяется встроенный мини-ассемблер. При этом мнемоническому имени команды МП должны предшествовать кавычки:

INLINE(«SHLD/\$ 4C/\$ 22»);

Компилятором игнорируются пробелы и запятые в мнемонических именах, причем последние должны содержать не более пяти символов:

вместо PUSH PSW → PUSHP  
POP PSW → POPPS

Необходимо отметить, что компилятор не проверяет адекватность вводимых с помощью средства INLINE кодов для данной программы. Ответственность за работоспособность таких фрагментов лежит на программисте.

Структура скомпилированных программ. Оптимизация. Процесс разработки программ для встроенных систем с использованием «структурного» языка высокого уровня детально рассмотрен в [2]. Для ввода текста программы в дисковый файл и ее последующих модификаций желательно применять экранный редактор текста, например УОРДСТАР. В Паскаль-программе встроенной системы должна быть задана вершина стека универсального микропроцессорного контроллера (УМК). По умолчанию компилятор назначает вершину стека, установленную в инструментальной системе. Модификация указателя стека производится внутренним ключом компилятора Z. Например, для занесения в указатель стека числа 4000H необходимо до тела основной Паскаль-программы поместить строку

(\* \$ Z \$ 4000 \*)

Пробелы в ключевой строке недопустимы. В структуре скомпилированной программы вслед за установкой указателя стека следует вызов процедуры @INI, осуществляющей программный интерфейс со средствами ввода-вывода ОС. Стандартная @INI занимает АССН ≈ 3 Кбайт и практически не нужна в программах встроенных УМК. Для сокращения объема скомпилированной программы следует заменить подпрограмму @INI, написав ее на перемещающем макроассемблере (RMAC, M80) подобно записи, приведенной на рис. 1. Такая подпрограмма должна быть оттранслирована и скомпонована с основной программой до библиотек Паскаля/МТ+.

При использовании в программах встроенных УМК операций над вещественными числами следует учесть,



```

CSEG
PUBLIC @INI
@INI: MUI A, 80H;   ПРОГРАММИРОВАНИЕ ИНТЕР-
      OUT 0D3H;   ФЕЙСОВ СИСТЕМЫ
      ;
      ;
      RET

```

Рис. 1. Подпрограмма @INI встроенной системы

что компилятор включает в объектный файл подпрограмму @ERR, осуществляющую контроль и диагностику ошибок периода выполнения. Объем @ERR — 1 Кбайт. Эта подпрограмма использует средства ОС, обычно отсутствующие в УМК. При запуске @ERR описанным выше способом необходимо учесть, что основная программа перед вызовом передает ей два параметра, помещая их в стек. Перед выходом из @ERR состояние указателя стека должно быть восстановлено (рис. 2). Последней процедурой, помещаемой трансля-

```

CSEG
PUBLIC @ERR
@ERR: SHLD HL;   СОХРАНИТЬ СОДЕРЖИМОЕ R: N, L
      POP  H;   ВОССТАНОВИТЬ SP С СОХРАНЕ-
      XTHL;   НИЕМ АДРЕСА ВОЗВРАТА
      POP  H
      XTHL
      LHL  HL;  ВОССТАНОВЛЕНИЕ СОДЕРЖИМОГО
      RET;   РЕГИСТРОВ N, L
;
DSEG
HL:   DS 2

```

Рис. 2. Подпрограмма @ERR встроенной системы

тором в объектный файл, является @HLT, осуществляющая функцию вызова «горячего» старта ОС CP/M. Если программа встроенной системы представляет собой бесконечный цикл, то замены @HLT производить не нужно. Иначе, эта процедура должна быть соответствующим образом изменена.

Обмен данными с ассемблерными модулями. Существуют два основных способа передачи параметров между программами на Паскале/MT<sub>+</sub> и ассемблере. Использование общих глобальных переменных представлено на рис. 3. В этом случае, при вызове в Паскаль-программе процедуры АЦП, описанной как EXTERNAL, управление передается по адресу метки АЦП, в ассемблере описанной как PUBLIC. Программа-драйвер АЦП должна произвести его запуск, ввести результат преобразования и поместить его в ячейку ОЗУ АЦПВВ. Содержимым этой ячейки определяется значение одноименной переменной в Паскаль-программе. Аналогично, из программы на ассемблере можно обратиться к глобальным переменным Паскаль-программы, предварительно описав их в ассемблерной программе как

(\* ОПИСАТЕЛЬНАЯ ЧАСТЬ ОСНОВНОЙ ПАСКАЛЬ-ПРОГРАММЫ

```

АЦПВВ ~ ВНЕШНЯЯ ПЕРЕМЕННАЯ, СОДЕРЖАЩАЯ
        РЕЗУЛЬТАТ А/Ц ПРЕОБРАЗОВАНИЯ
АЦП ~ ВНЕШНЯЯ ПРОЦЕДУРА, ДРАЙВЕР АЦП *)

```

```

VAR АЦПВВ : EXTERNAL BYTE;
EXTERNAL PROCEDURE АЦП;

```

(\* ТЕПЕРЬ МОЖНО ОБРАЩАТЬСЯ К ПРОЦЕДУРЕ АЦП И ПОСЛЕ ЭТОГО ИСПОЛЬЗОВАТЬ ПЕРЕМЕННУЮ АЦПВВ, СОДЕРЖАЩУЮ РЕЗУЛЬТАТ А/Ц ПРЕОБРАЗОВАНИЯ \*)

; ФРАГМЕНТ ОПИСАНИЯ ДРАЙВЕРА АЦП НА АССЕМБЛЕРЕ

```

PUBLIC АЦП, АЦПВВ
CSEG
АЦП:   TEXT ДРАЙВЕРА
      ;
      RET
DSEG
АЦПВВ: DS 1

```

Рис. 3. Обращение к драйверу АЦП из Паскаля

EXTRN. Возможна также передача параметров между программами через стек [3—5]. При обращении к подпрограмме в вершине стека содержится адрес врата управления. Под ним в стеке находятся параметры в порядке, обратном описанию. Например, если

A, B : INTEGER; C : CHAR

то в стеке будет C над B и над A. Для передачи каждого из параметров требуется минимум 16-бит слово области стека.

Внутреннее представление вещественных чисел в программе на Паскале/MT<sub>+</sub> может быть выбрано различным. Эти числа либо двоично-десятичные с 18 цифрами и четырьмя фиксированными разрядами после запятой, либо в представлении с плавающей запятой. Перед трансляцией программы при выборе чисел с двоично-десятичным представлением в командную строку компилятора необходимо ввести ключ V. Обслуживание прерываний записывается как процедура (рис. 4).

```

PROCEDURE INTERRUPT [3] СТРОБ_ВВОД; (* В КВАДРАТНЫХ
СКОБКАХ - НОМЕР ПРИОРИТЕТА ПРЕРЫВАНИЯ *)
BEGIN
  АВАРИЯ := TRUE;
END; (* СТРОБ_ВВОД *)

```

Рис. 4. Пример процедуры прерывания

В область ПЗУ, в которую передается управление при прерывании, следует поместить вектор безусловных переходов на подпрограммы прерываний, абсолютные адреса которых можно найти в карте, выдаваемой редактором связей. Транслятор в начале каждой процедуры прерывания засылает содержимое всех регистров в стек, а в конце процедуры — восстанавливает содержимое регистров и разрешает прерывание.

Пакет Паскаль/MT<sub>+</sub> включает в себя дизассемблер перемещаемого объектного файла, который генерирует листинг исходных предложений на Паскале, перемежаемый мнемоничной командой, полученных при трансляции этого текста. По этому листингу легко определить, как транслятор преобразовал исходный текст и какие подпрограммы должны вызываться при выполнении определенного оператора. При замене подпрограмм периода выполнения с помощью дизассемблера можно определить требования транслятора к замещаемой подпрограмме.

Компоновка. Редактор связей, входящий в пакет Паскаль/MT<sub>+</sub>, предназначен для формирования загрузочного файла из перемещаемых объектных файлов и библиотек. Перемещаемые файлы, компокуемые редактором, должны иметь расширение .ERL. Поэтому после трансляции макроссемблерами M80 или RMAC ОС CP/M полученные файлы до компоновки должны быть переименованы средствами ОС. В командной строке редактора связей (рис. 5) может быть задан ряд ключей:

```

LINKMT B:МОЯПАСР, B:МОЯАСМР, A:TRANEND, A:FPREALS/S,
A:PA$LIB/S/M/P:0/D:4000/H:1000

```

Рис. 5. Пример командной строки редактора связей

S — в строке перед ним расположена библиотека, допускающая поиск отдельных модулей для компоновки. Такими библиотеками в пакете Паскаль/MT<sub>+</sub> являются библиотека подпрограмм периода выполнения PASLIB и вычислений с плавающей запятой FPREALS. Остальные библиотеки поиска не допускают и при компоновке подключаются к загрузочному файлу целиком;

P: AAAA — задает начальный адрес кодового сегмента (ПЗУ УМК) в 16-ричном коде AAAA<sub>H</sub>;

D: DDDD — указывает, что сегмент данных в ОЗУ УМК начинается с адреса DDDD<sub>H</sub>;

Н: PRRP — вызывает формирование загрузочного файла в HEX-формате с начальным адресом загрузки PRRPH, возможно смещенным относительно абсолютного начального адреса. Такая загрузка полезна, например, для работы программатора РПЗУ.

**Отладка программ.** Наибольшие трудности при отладке обычно вызывает моделирование обмена с внешними устройствами, имеющими, как правило, различные адресацию и назначение у встроенной и инструментальной систем. Возможное решение в данном случае — выделение при проектировании программы двух подпрограмм: ввода байта из порта и вывода байта в порт (рис. 6). Все операции с портами ввода-вывода в программе встроенной системы должны сводиться

```

(* ВЫВОД БАЙТА В ПОРТ *)
PROCEDURE ВЫВОД_БАЙТА (ПОРТ,БАЙТ : BYTE);
BEGIN
  OUT [(ПОРТ)] := БАЙТ
END; (* ВЫВОД_БАЙТА *)

(* ВВОД БАЙТА ИЗ ПОРТА *)
FUNCTION ВВОД_БАЙТА (ПОРТ : BYTE) : BYTE;
BEGIN
  ВВОД_БАЙТА := INP [(ПОРТ)]
END; (* ВВОД_БАЙТА *)

```

Рис. 6. Подпрограммы ввода и вывода байта

```

PROCEDURE ВЫВОД_БАЙТА (ПОРТ,БАЙТ : BYTE);
BEGIN
  WRITENEX (OUTPUT,ПОРТ,1);
  WRITE (' -> ');
  WRITENEX (OUTPUT,БАЙТ,1);
  WRITELN
END; (* ВЫВОД_БАЙТА *)

FUNCTION ВВОД_БАЙТА (ПОРТ : BYTE) : BYTE;
VAR SWAP : BYTE;
BEGIN
  WRITENEX (OUTPUT,ПОРТ,1);
  WRITE (' -> ');
  READNEX (INPUT,SWAP,1);
  WRITELN;
  ВВОД_БАЙТА := SWAP
END; (* ВВОД_БАЙТА *)

```

Рис. 7. Ввод-вывод байта при отладке

к обращению к этим подпрограммам. На время отладки на инструментальной системе производится замена подпрограмм ввода-вывода (рис. 7). При этом во время вывода байта в порт на пульт инструментальной системы выдается сообщение в виде

номер порта → выводимый байт

При выполнении ввода байта выводится

номер порта →

и система ждет ввода 16-ричного числа с пульта, соответствующего реально поступающей во время работы встроенной системы информации. Значительное число встроенных систем оснащается специализированными знаковыми индикаторами для выдачи оперативной технологической информации. Заменой подпрограмм периода выполнения можно обеспечить вывод сообщений на индикатор с помощью процедуры WRITE. При компиляции Паскаль/MT<sub>+</sub> преобразует процедуру WRITE (строка) в последовательность команд (рис. 8), где @SFB, @DWD, @WRS — подпрограммы периода выполнения, извлекаемые при компоновке из библиотеки. Заменяв эти подпрограммы, пользователь может применять процедуру WRITE для вывода на индикатор встроенной системы сообщений (рис. 9). При отладке программы на инструментальной системе подпрограммы временно заменяются библиотечными исключением при компоновке замещающего модуля.

Используя описанные приемы совместно с отладчиком Паскаля/MT<sub>+</sub>, можно построчно трассировать Пас-

каль-программу, имитируя обмен с внешними устройствами с пульта или устанавливая точки разрыва, и выполнять блоки программы встроенной системы с просмотром в местах останова значений переменных. В отлаженной программе восстанавливаются все изменения для удобства отладки фрагменты и производится ее компиляция. В результате получается файл типа COM, который может быть записан в РПЗУ. Для этого желательно пользоваться программатором, производящим запись из дискового файла непосредственно в РПЗУ.

Применение пакета Паскаль/MT<sub>+</sub> в разработке ряда программ встроенных систем показало, что производительность программиста при этом увеличивается при-

; ДИЗАССЕМБЛИРОВАНИЕ ВЫРАЖЕНИЯ "WRITE ('СТРОКА')"

```

LXI H,OUTPUT;      АДРЕС ПЕРЕХОДА НА
PUSH H;            СИСТЕМНЫЙ ВЫВОД
CALL @SFB;         АДРЕС НАЧАЛА ТЕКСТА - 1) - В СТЕК
CALL МЕТКА;       ДЛИНА СТРОКИ ( 1 БАЙТ )
DB N;              СОДЕРЖИМОЕ 'СТРОКИ'
DB 'СТРОКА';
CALL @DWD;
CALL @WRS

```

Рис. 8. Процедура WRITE после дизассемблирования

; ЗАМЕНА ПОДПРОГРАММ ПЕРИОДА ВЫПОЛНЕНИЯ ДЛЯ ВЫВОДА  
; НА ИНДИКАТОР ПИУ-1 16-СИМВОЛЬНЫХ СТРОК И КОНСТАНТ

```

PUBLIC @SFB,@DWD,@WRS
CSEG
>

@SFB:   POP H
        XTHL
@WRS:   RET
@DWD:   POP H;   ЗАПОМНИТЬ АДРЕС ВОЗВРАТА
        SHLD ВОЗВРАТ
        POP H;   АДРЕС НАЧАЛА ТЕКСТА - 1) - В Р. H,L
; СЮДА ПОМЕСТИТЬ ТЕКСТ ДРАЙВЕРА ПИУ-1,
; ВЫВОДЯЩЕГО НА ЭКРАН 16 СИМВОЛОВ
        LHL ВОЗВРАТ
        PCHL;   ВОЗВРАТ В ОСНОВНУЮ ПРОГРАММУ

DSEG
ВОЗВРАТ: DS 2

```

Рис. 9. Процедура WRITE для встроенной системы

мерно на порядок по сравнению с программированием на ассемблере. Наиболее существенно повышается производительность при проектировании, кодировании и отладке программ со сложными проверками условий или содержащих значительное количество вычислений, особенно с вещественными числами. С использованием описанных средств разработаны следующие программы, успешно эксплуатирующиеся в настоящее время: автоматического управления пакет-прессом по переработке лома черных металлов (объем 5 Кбайт, 600 строк на Паскале, 130 строк на ассемблере); автоматического устройства взвешивания на мостовом крае (объем 8 Кбайт, 400 строк на Паскале, 200 строк на ассемблере, вычисления с вещественными числами и трансцендентными функциями с использованием всех библиотек Паскаля/MT<sub>+</sub>); мультипроцессорной системы обслуживания автоматизированной проходной (объем 3 Кбайт, 585 строк на Паскале, 100 строк на ассемблере для каждого процессора).

Телефон: 45-85-61, Днепропетровск

#### ЛИТЕРАТУРА

1. Фаулджер Р. Программирование встроенных микропроцессоров.— М.: Мир, 1985.— 275 с.
2. Фридмен М., Ивенс Л. Проектирование систем с микропроцессорами.— М.: Мир, 1986.— 495 с.

3. Паскаль/MT+, версия 5.5. Руководство пользователя/ВЦП-84/50640 — М., 1984. — 234 с. (Перевод материала фирмы: Digital Research Corp. 1982. — (США.)
4. Справочное руководство для пользователей языка программирования Паскаль/MT+ /ВЦП-Е-76336 (84/40111). — М., 1984. — 367 с. (Перевод материала фирмы: Digital Research Corp. Pascal/MT+TM Language. Reference Manual. User's guide. Release 5. — 1982. — 249 p. — США.)
5. Паскаль/MT+. Версия 5./ВЦП-Д-57131 (83/1010). — М., 1982. — 230 с. (Перевод материала фирмы MT MicroSYSTEMS Pascal/MT+, release 5. — 1980. — 195 p. — США.)

(Статья поступила 11 декабря 1986)

УДК 681.3

Р. В. Вершигора

## СОПРОГРАММНАЯ СВЯЗЬ ФУНКЦИЙ В ЯЗЫКЕ СИ

При разработке алгоритмов выделение процедур или функций требует определения прав доступа и обмена данными в каждой из них. Развитие способа процедурной связи по данным приводит к возникновению «информационно прочных» модулей, в которых устойчивость данных обеспечивается сохранением контекста между точками вызова процедур. При этом обработка информационного потока предстает в виде процесса, где механизм взаимодействия процедур служит для их синхронизации и изменения состояний. Сопрограмма как процесс рассматривалась уже в 1963 г. М. Конвсем [1], а детальное описание механизма взаимодействия приведено Д. Кнутом в [2]. В операционной системе UNIX, например, синхронизация взаимодействия процессов через системную фазу соответствует механизму сопрограмм.

Сопрограммная связь между несколькими процедурами или функциями удачно дополняет стандартную конструкцию вызова и является более простым средством, чем создание дополнительных точек входа, расширений циклов или ветвлений. Под простотой здесь понимаются удобство в отладке программ, упрощение логических связей в алгоритме и низкие системные затраты.

В абстрактном смысле точка взаимодействия — инициализации, возобновления или возврата — описывается парой: «данные» и «состояние». «Данные» содержат локальный контекст и доступ к глобальным переменным, обусловленный семантикой языка. «Состояние» характеризует сегмент кода, где произошло прерывание в смысле сопрограммного механизма.

Моделируя новые средства какого-либо языка на самом этом языке, мы достигаем независимости от транслятора, но получаем более тяжеловесный ассемблерный код таких конструкций. Вставка новых операций через вызов ассемблерных функций часто требует подстройки к коду, генерируемому транслятором, и ограничивает применение схемы. Например, компромиссное решение у П. Байлса [3] состоит во вставке новых конструкций в язык Си с помощью средств макрогенерации. Генерируемый текст обеспечивает эффективное управление записями активаций сопрограмм, которые создаются динамически при инициализации. Заполняются эти записи с помощью ассемблерных функций, включенных в библиотеку языка. Применение условной конструкции для возобновления сопрограммы позволяет использовать стандартный механизм передачи параметров через стек без нарушения целостности локальных областей данных, расположенных на стеке. Таким образом, макрогенерация обеспечивает трансля-

```

? ЭВМ СИ-4   ОС РВ 3.0   ВЦ АУ                               ВЕРШИГОРА Р.В.
? ФУНКЦИЯ СОПРОГРАММНОГО ВЗАИМОДЕЙСТВИЯ PLACE,RESUME,DECLINE
? ОБЪЕКТНЫЙ ЯЗЫК СИ CORSTN
?
; TITLE CORSTN
; _PSECT _PROG.
; _EVEN
; _BLKW 3 ; ЗАПИСЬ АКТИВАЦИИ
; MAINSV: _BLKW 2 ; НЕСОПРОГРАММНОГО КОРНЯ
; TESTSV: _WORD 0 ; ИНДИКАТОР КОРНЕВОЙ ЗАВИСИ

; ПАРАМЕТР: АДРЕС УКАЗАТЕЛЯ СОБСТВЕННОЙ ЗАПИСИ АКТИВАЦИИ
; _EVEN
PLACE:: MOV (R5),R1 ;
TST TESTSV ; СОХРАНИТЬ РЕГИСТРЫ КОРНЯ ?
BNE NEXT ; НЕУ -> NEXT
MOV #MAINSV,R1 ; АД: ИСПОЛЬЗОВАТЬ ОБЛАСТЬ
MOV R5,TESTSV ; MAINSV
MOV (R5),MAINSV ;
NEXT: MOV R5,#2(SP) ; ЗАГРУЗКА БАЗЫ ПО ПАРАМЕТРУ
MOV R5,R0 ;
MOV 2(R0),2(R1) ; СОХРАНЕНИЕ РЕГИСТРОВ
MOV -(R0),-(R1) ; ИНИЦИАЛИЗАТОРА И
MOV -(R0),-(R1) ; ТОЧКИ ВОЗВРАТА
MOV -(R0),-(R1) ;
RTS PC ; ВОЗВРАТ В ТОЧКУ ВЫЗОВА

; ПАРАМЕТР: УКАЗАТЕЛЬ ЗАВИСИ АКТИВАЦИИ ФУНКЦИИ ВОЗВРАТА И
; ПЕРЕДАВАЕМОЕ СЛОВО-СОСТОЯНИЕ
; _EVEN
DECLINE:: TST (SP)+ ; ЕСЛИ УКАЗАТЕЛЬ = 0, ТО
CMP (R5)+,(R5)+ ; ВОЗВРАТ В КОРЕНЬ (ЗАВИСИ
MOV (SP)+,R1 ; АКТИВАЦИИ КОРНЯ MAINSV )
MOV (SP),R0 ;
MOV R5,SP ; АДРЕС СТЕКА
MOV R1,R0 ; НОВЫЙ УКАЗАТЕЛЬ БАЗЫ
TST R1 ;
BNE M0R00T ;
MOV #MAINSV,R1 ;
MOV (R1),R0 ;
CLR TESTSV ;
M0R00T: MOV -(R1),R4 ;
MOV -(R1),R3 ; ВОССТАНОВЛЕНИЕ РЕГИСТРОВ
MOV -(R1),R2 ; R2-R4-ФУНКЦИИ ВОЗВРАТА
JMP @+10(R1) ;

; ПАРАМЕТР: УКАЗАТЕЛЬ ЗАВИСИ АКТИВАЦИИ ФУНКЦИИ ВОЗОБНОВЛЕНИЯ И
; ПЕРЕДАВАЕМОЕ СЛОВО-СОСТОЯНИЕ
; _EVEN
RESUME:: MOV (SP)+,2(R5) ;
MOV R4+,(R3) ; СОХРАНЕНИЕ РЕГИСТРОВ В
MOV R3+,(R3) ; СЕБЯ ЗАВИСИ
MOV R2+,(R5) ;
MOV (SP),R5 ;
MOV 2(SP),R0 ; ПЕРЕДАЧА СОСТОЯНИЯ И
MOV R0,R1 ; ВОССТАНОВЛЕНИЕ РЕГИСТРОВ
MOV -(R1),R4 ; ИЗ ЗАВИСИ АКТИВАЦИИ
MOV -(R1),R5 ; ( АДРЕСУЕТСЯ ЧЕРЕЗ
MOV -(R1),R2 ; ПАРАМЕТР )
JMP @+2(R5) ;
END

```

Рис. 1. Текст ассемблерных функций сопрограммного механизма

торно-независимый код, а ассемблерные функции рассчитаны на применение ЭВМ типа PDP 11. Простая семантика языка Си почти не накладывает ограничений на употребление сопрограммной схемы в языке.

В подходе Б. А. Новикова [4] к реализации сопрограммного механизма для языка ПЛ/1 все функции управления и сохранения обстановки в точке вызова переданы ассемблерным процедурам, а их правильное употребление может контролировать транслятор языка.

Любое новое средство в языке должно иметь ответственность в стиле языка и базироваться на его стандартных возможностях. В предлагаемом варианте атрибут «вызов функции» расширяется до сопрограммного через выполнение ассемблерных функций (рис. 1).

Инициализация сопрограммы осуществляется через обычную конструкцию вызова функции в языке Си. Активируемая функция при этом должна содержать вызов PLACE до первого оператора возобновления RESUME или активации сопрограммы следующего уровня. Функция PLACE преобразует линейный список записей активации, создаваемый на стеке вызовом Си-функции, в древовидный. Размещение записи активации во фрейме вызова функции позволяет ускорить взаимодействие сопрограмм и избежать трудностей, возникающих при динамическом выделении памяти для записи. PLACE освобождает собственный фрейм, перемещая регистры и адрес возврата во фрейм ини-

СТРУКТУРА ЗАПИСИ АКТИВАЦИИ НА СТЕКЕ
N - ЛОКАЛЬНАЯ ПЕРЕМЕННАЯ
.....
1 - ЛОКАЛЬНАЯ ПЕРЕМЕННАЯ
R2 РЕГИСТРЫ R3 ВЫЗЫВАЕМОЙ R4 ФУНКЦИИ R5
АДРЕС ВОЗВРАТА
1 - АРГУМЕНТ
.....
N - АРГУМЕНТ
НАПРАВЛЕНИЕ ВОЗРАСТАНИЯ
1 АДРЕСОВ У ПАМЯТИ

SP->	RET MAIN
	ADR SAVE R5
	.....
	NEXT MAIN
	OWN MAIN
	.....
	R2 MCR
	R3 MCR
	R4 MCR
R5->	R5 MCR
	RET MCR

SP->	RET MERG1
	TEST MERG1
	NEXT MERG1
	.....
	FP MERG1
	STATUS MERG1
	TEST MERG1
	OWN MERG1
	.....
	R2 MERG1
	R3 MERG1
	R4 MERG1
	R5 MAIN
	RET MERG1
	.....
	FNAME MERG1
	NEXT MERG1
	.....
	NEXT MAIN
	OWN MAIN
	.....
	R2 MAIN
	R3 MAIN
	R4 MAIN
	R5 MCR
R5->	RET MAIN

а)

б)

в)

Рис. 2. Структура данных на стеке:

а — запись активации; б — до выполнения операторов из PLACE в MAIN; в — после выполнения первого вызова RESUME в MERGER

диализатора. Первый вызов PLACE в программе использует для этого специальную область MAINSV — область сохранения несопрограммного корня или монитора операционной системы. Собственный же фрейм функция PLACE адресует через указатель, адрес которого передается ей в качестве параметра. Структура записи активации (рис. 2, а), возникающая на стеке после входа в функцию, определяется соглашением о связях функций в языке Си. Она зависит от транслятора языка и типа используемой ЭВМ.

В данном случае запись активации полностью определяет контекст вызова: сохранение регистров, положение стека и доступ к локальным данным. Возврат из PLACE производится в точку ее вызова. Инициализация последующих поколений сопрограмм приводит к росту стека, при этом инициализатор следует рассматривать как вершину более высокого уровня дерева. Единственный параметр функции PLACE типа UNSIGNED \* используется затем для обращения к сопрограмме и сохранения состояния при возобновлении.

Операция возобновления, которая реализуется через вызов ассемблерной функции RESUME, должна сохранять собственное состояние в точке вызова и восстанавливать состояние возобновляемой функции. Собственный фрейм адресуется через текущий указатель базы — регистр R5, а возобновляемый определяется с помощью параметра.

Первым параметром функции RESUME является указатель фрейма возобновляемой функции, второй параметр — передаваемое целочисленное значение. Возврат значения позволяет удобно использовать вызов RESUME в ряде операторов языка типа WHILE, условном или присваивания. Таким образом, возможна не только индикация состояния процесса, как будет ниже в примере трактоваться сопрограмма, но и передача данных с помощью указателя, т. е. возможно создание канала связи между сопрограммами. В приводимой схеме сопрограммы следует рассматривать как альтернативу обычной связи функций, так как невозможно возобновление какой-либо сопрограммы из обычной функции. Вызываемые из сопрограммы функции являются листьями сопрограммного дерева, которые возникают и «падают» с вершины стека.

Вызов функции DECLINE приводит к разрыву сопрограммной связи, при этом управление передается сопрограмме не меньшего уровня дерева. Младшие поколения уничтожаются простым сбросом стека, и их активация в дальнейшем может привести к ошибке. Активируемая сопрограмма, как и в RESUME, адресуется через указатель ее фрейма вызова — первый параметр DECLINE. Если этот указатель NULL, то происходит возврат в несопрограммный корень. Если нет необходимости экономить стековое пространство, то такой вызов достаточно сделать один раз. Вторым параметр DECLINE — передаваемое целое значение.

Приведем краткий синтаксис ассемблерных функций: PLACE (ADRPTR) UNSIGNED \* ADRPTR; RESUME (PTR, VALUE) UNSIGNED \* PTR; INT VALUE; DECLINE (PTR, VALUE) UNSIGNED \* PTR; INT VALUE.

Для иллюстрации техники использования сопрограмм и работы ассемблерных функций приведем пример программы на языке Си. Программа на рис. 3 сливает

```

/*
ПРОГРАММА СЛИЯНИЯ ДВУХ НЕПУСТЫХ ПОСЛЕДОВАТЕЛЬНЫХ
ФАЙЛОВ ЦЕЛЫХ ЧИСЕЛ "A.DAT" И "B.DAT", КАЖДЫЙ ИЗ
КОТОРЫХ УПОРЯДОЧЕН ПО ВОЗРАСТАНИЮ. ФАЙЛ РЕЗУЛЬТАТА —
СТАНДАРТНЫЙ ФАЙЛ ВЫВОДА.
ИСПОЛЬЗОВАНИЕ УКАЗАТЕЛЕЙ И ПЕРЕМЕННЫХ В СОПРОГРАММАХ:
OWN — УКАЗАТЕЛЬ СОБСТВЕННОЙ ЗАПИСИ АКТИВАЦИИ
NEXT — УКАЗАТЕЛЬ ПРЕЕМНИКА
CURRENT — УКАЗАТЕЛЬ ТЕКУЩЕЙ СОПРОГРАММЫ
TEST — ДЛЯ ЧТЕНИЯ ИЗ СВОЕГО ФАЙЛА
STATUS — ДЛЯ ПЕРЕДАЧИ СОСТОЯНИЯ ПРОЦЕССА ЗАПИСИ
*/
#include <stdio.h>
#define MAXIMUM 0077777
unsigned * current;
main ()
/*
/* ИНИЦИАЛИЗАЦИЯ СОПРОГРАММ */
merger ("A.DAT", own);
merger ("B.DAT", current);
/*
/* ПРОЦЕСС УПРАВЛЕНИЯ ФАЙЛОМ
*/
merger (fname, next)
char * fname;
unsigned * next;
/*
/* СОПРОГРАММА
*/
int test, status;
file *fp, *fopen();
place (&own);
fp=fopen (fname, "r");
fscanf (fp, "%d", &test);
current=own;
do {
status=resume (next, test);
next=current;
current=own;
/* ПРЯЖКА
ТЕКУЩЕГО ФАЙЛА
*/
while (test != status)
/* PRINTF ("%d\n", test);
if (fscanf (fp, "%d", &test) != 0)
/* CURRENT=NULL;
test=maximum;
fclose (fp);
break;
};
};
while (next != null);
decline (null, null);
}

```

Рис. 3. Слияние упорядоченных файлов

два целочисленных, упорядоченных по возрастанию файла в один файл стандартного вывода. Файлы ADAT и B.DAT предполагаются пустыми.

Однотипность используемых файлов приводит к двум экземплярам одной и той же подпрограммы, которые симметрично вызывают друг друга. Заметим, что инициализация вносит ассиметричность в схему взаимодействия подпрограмм. Ассиметрия здесь спрятана в списке входных параметров. Передаваемое значение служит в качестве канала связи или его можно трактовать как состояние процесса записи в выходной файл. Техника использования текущей ссылки на процесс — CURRENT — оказывается полезной при активации рефлексивных подпрограмм, которые возвращают управление текущему процессу, и неявной замене одного процесса другим. Здесь в цикле она несколько тяжеловесна. Сохранение текущего значения TEST между точками возобновления RESUME позволяет обойтись тем же циклом записи в случае окончания одного из файлов. Таким образом, при использовании нескольких областей данных с одним сегментом кода необходимо определить эквивалентный набор указателей на подпрограммы для их возобновления.

На рис. 2 приведены состояния стека для двух точек данной программы. Сравнение рис. 2, б и в позволяет проследить работу PLACE и результат выполнения RESUME. DECLINE просто выбирает данные из области MAINSV (см. рис. 1).

Ограничением данной версии является то, что список параметров каждой подпрограммы должен состоять из двух целочисленных параметров. Так как число параметров в вызываемой функции при данном соглашении о связях определить нельзя, то для сохранения баланса на стеке при записи и сбросе параметров был выбран такой компромиссный вариант.

В предлагаемом варианте реализации существенно использование фрейма вызова для сохранения регистров ЭВМ серии PDP-11 и соглашения о связях функций в компиляторе — использование R5 как базового, R0 и R1 для возврата значений из функции и передачи параметров через стек. Соответственно, компьютеры других серий и различные варианты компиляторов требуют адаптации ассемблерных функций, хотя для машины с аппаратным стеком легче сохранить приводимую логику взаимодействия.

Эффективность данной версии заключается в минимальном использовании дополнительной памяти на организацию сопроволнительной связи (только для указателей на запись активации) и возможности управления пространством локальных областей на стеке. Время возобновления подпрограммы — длины кода и набора команд — эквивалентно времени затрат на вызов функции в языке Си, которое, по мнению авторов языка, весьма мало. Мобильность механизма состоит в возможности построения дерева произвольной структуры с возобновлением на любом уровне, конечно, если активированная программа еще существует.

Примеры задач, где применяются подпрограммы, можно найти в работах [5—8], а описание механизма для языков программирования — в [3, 4]. Это, прежде всего, задачи с двойной буферизацией обмена, замены многопроходного алгоритма сопроволнительной связью, поиска и сортировки на деревьях, любого сочетания процессов «поставщик» — «потребитель». Анализ применения сопроволнительных классов алгоритмов требует более глубоких исследований, которые могли бы определить относительную ценность сопроволнительных схем среди других возможных вариантов.

Телефон: 257-75-69, Ленинград

#### ЛИТЕРАТУРА

1. Conway M. E. The Design of a Separable Transition-Diagram Compiler // Comm. ACM.— 1963.— N 6.— P. 396—408.

2. Кнут Д. Е. Искусство программирования для ЭВМ. Т. 1. Основные алгоритмы.— М.: Мир, 1976.
3. Bailes P. A. A low-cost implementation of coroutines for C // Software-practice and experience. April, 1985.— Vol. 15(4).— P. 379—395.
4. Новиков Б. А. Снова подпрограммы // Программирование.— 1986.— № 4.— С. 34—37.
5. Кейлингер П. Элементы операционных систем.— М.: Мир, 1985.— С. 150—152.
6. Вирт Н. Алгоритмы+структуры данных=программы.— М.: Мир, 1985.— С. 144—147.
7. Экхауз Р., Моррис Л. МиниЭВМ: организация и программирование.— М.: Финансы и статистика, 1983.— С. 119—120.
8. Новиков Б. А., Романовский И. В. Подпрограммы в ОС ЕС // Кибернетика.— 1985.— № 1.— С. 34—37.

Статья поступила 15 июня 1987

УДК 681.3.06

А. Л. Филин

### ОРГАНИЗАЦИЯ ПРОГРАММ ДИНАМИЧЕСКОЙ СТРУКТУРЫ ПРИ РАЗРАБОТКЕ ИНТЕГРИРОВАННЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ВЫСОКОГО УРОВНЯ

— Предлагается метод организации комплекса программ диалоговой интегрированной системы, осуществляющий динамическую загрузку модулей и обеспечивающий связь отдельных компонентов по данным и управлению.

Наиболее распространены модули с простой структурой — целиком загружаются в оперативную память и не требуют дозагрузки фрагментов в процессе своей работы. Связь между подпрограммами происходит с помощью передачи управления с возвратом (операторы вызова процедур в языках высокого уровня, команда CALL в ассемблере). Недостаток программ простой структуры — в памяти находятся фрагменты, не нужные в данный момент (например, используется интегрированная система в качестве текстового редактора).

Модули с перекрытием — программы оверлейной структуры (наиболее простой вариант, реализуемый стандартным редактором связей, MS DOS [1]). Такая программа создается в виде единственного EXE-файла и состоит из группы подпрограмм, постоянно находящихся в ОП (корневого сегмента), и нескольких других групп подпрограмм (оверлейных сегментов). При загрузке такой программы в ОП помещается корневой сегмент и резервируется место по размеру самого большого оверлейного сегмента (область перекрытия). В дальнейшем на это место помещается один из оверлейных сегментов и при необходимости перекрывается другим.

Более развитые редакторы связей создают многоуровневые оверлейные структуры и поддерживают несколько областей перекрытия [2]. Программа с перекрытиями использует средства операционной системы и дополнительные подпрограммы, которые включаются в модуль редактором связей.

Использование оверлейных структур не связано с изменениями в программе. Возможно обращение из одного загружаемого сегмента к подпрограммам другого, однако это приводит к загрузке одного сегмента на место другого и требует времени на обращения к диску. Планируя структуру программы с перекрытием, необходимо учитывать эти факторы [2].

Структура программ с перекрытием задается при редактировании связей и не может быть изменена во время работы, поэтому нельзя управлять загрузкой

отдельных сегментов. Память, занятую для размещения оверлейных сегментов, нельзя использовать для хранения данных. Разделение компонентов между корневым и оверлейными сегментами, эффективное при всех режимах функционирования многоцелевой диалоговой системы, выбрать невозможно.

Другой недостаток — при модификации одного из модулей приходится заново проводить сборку и редактирование связей всей программы.

Программа динамической структуры состоит из головного загрузочного модуля и нескольких динамически вызываемых. В простейшем случае вызываемый модуль — загрузочный модуль, работающий непосредственно под управлением операционной системы. При каждом вызове (с помощью функции 4B00h MS DOS, функции `spawn` XX библиотеки языка Си либо через вызов командного процессора — функции `system` из библиотеки Си [1, 3]) он загружается в оперативную память, освобождая ее перед возвратом в главную программу.

Такой подход достаточно прост и расширяет возможности системы с помощью подключения внешних модулей, однако, вызывая одну программу несколько раз подряд, приходится каждый раз загружать ее с внешнего устройства.

Другим существенным недостатком является дублирование кодов стандартных подпрограмм головной программы в каждой динамически загружаемой подпрограмме; начинают дублироваться и служебная информация и области памяти. Например, в случае использования языка Си [1] загружаемая подпрограмма создает собственный стек, собственную кучу (память, доступная для динамического распределения подпрограммам головной программы, становится недоступной для загруженной подпрограммы).

Это приводит к расходу оперативной памяти и невозможности совместного использования данных на внешних устройствах. Работа нескольких программ с одним файлом может вызвать нарушение его целостности при модификации из-за несогласованной перезаписи буферов в каждой из них при использовании программной буферизации (даже при однозадачном режиме работы программ). В оверлейных структурах такой проблемы нет: и корневой и оверлейные сегменты используют подпрограммы библиотеки, собранные в корневом сегменте, и общие области памяти для переменных, не перекрываемые другими данными.

В операционных оболочках типа WINDOWS [5] можно вместо подчиненного организовать псевдопараллельное исполнение нескольких программ, однако недостатки, связанные с дублированием кодов модулей в разных программах, остаются (как в программах с простейшей динамической структурой) и проявляются при объединении программ, написанных на языке высокого уровня.

В операционных системах с произвольным взаимодействием модулей (типа JANUS [6]) требуются специальные программы-задачи для обработки запросов от других программ.

Организация программ динамической структуры с помощью специальных модулей загрузки и связи подпрограмм

Разработка ориентирована на язык Си [4] (компилятор MSC4.0 [1]) и легко может быть перенесена на другие языки и компиляторы.

Произвольная загрузка и вызов позволяют загруженным программам обращаться к функциям, входящим в состав головной программы. Головная программа — обычный загрузочный модуль.

Для динамической загрузки подпрограмм используется функция `ovrload`. Она получает дескриптор открытого файла с подпрограммой и возвращает указатель на загруженную программу. Функция `ovrcall` вызывает загруженную программу, получая указатель и адресный параметр. После завершения работы

происходит возврат в головную программу. При необходимости загруженная программа может быть вызвана повторно без обращения к диску. Программа удаляется из памяти по функции `ovrfree`.

Загружаемая подпрограмма — загрузочный модуль, подготовленный редактором связей (выясняет особые соглашения о начале и завершении работы, не совпадающие с принятыми в программах, предназначенных для исполнения под управлением MS DOS). Головная функция программы на языке Си называется `ovrmain` (вместо `main`), а редактору связей доступен файл `ovrslart.obj` (вместо `crt0.obj`) [1]. Используемые соглашения близки к применяемым при передаче управления обычным процедурам и функциям, поэтому загружаемая подпрограмма не создает целиком собственную среду, а наследует созданную головной программой. Модифицированные программы стандартного ввода-вывода обеспечивают передачу управления на пужные подпрограммы, локализованные в ядре (головном модуле). Объем загружаемой программы значительно сокращается.

В загружаемой подпрограмме стек и данные разделены, поэтому непосредственное обращение возможно только к таким же функциям. При использовании языка Си необходима компиляция в режиме `-Awlf` [1]. Подпрограммы на языке ассемблера [7] не должны использовать совпадение адресов в регистрах SS и DS (в большой модели памяти это обстоятельство не является существенным и большинство стандартных функций могут правильно работать с разделенными сегментами стека и данных).

Для обращения к функциям головного модуля применяется `ircall` (единственная команда `-INT`). В загруженной программе текста кодов функций нет (существенно сокращается объем). При вызове `ircall` первым параметром указывается номер функции головного модуля, затем остальные параметры (их количество `ircall` не ограничивает).

Возможность такого использования обеспечивает функция `intact`. Она вызывается в головной программе перед обращениями к динамическим программам и получает массив адресов точек входа, доступный загруженным программам. При вызове происходит активизация обработчика прерывания. Функция `intcall` обращается к нужной процедуре по номеру в массиве адресов (первый параметр). Остальные параметры передаются вызываемой процедуре. Можно подготовить соответствующие операторы `#define`, которые позволят преобразовать обычные вызовы процедур в вызовы через `intcall` без изменения исходного текста программы.

При обращении загруженной подпрограммы к функциям головной программы состояние загруженной подпрограммы сохраняется. На время вызова восстанавливаются регистры ES и DS по состоянию на момент вызова `intact`, а после завершения вызванной функции — состояние загруженной программы (происходит возврат в нее).

При таком использовании стандартных функций распределения памяти для загружаемой подпрограммы доступна динамическая память головной программы (куча) и ей не требуется резервирование новых областей. Общие процедуры обработки и хранения информации на внешнем носителе позволяют организовать совместное использование данных несколькими загружаемыми подпрограммами (рис. 1, 2).

Данный подход особенно удобен при разработке интегрированных прикладных систем [5, 8] на языке высокого уровня. Система может состоять из небольшого ядра, реализующего функции работы с экраном дисплея и БД, и специализированных загружаемых программ, ведущих работу с данными с помощью «точек зрения» на информацию в БД [8].

Программы получают только адреса соответствующих областей с рабочими и управляющими данными и помещают результаты работы в эти области (если

```

/* Головная программа */
#include "overlay.h"
#include "io.h"
#include "fcntl.h"
/* массив адресов точек входа в функции */
#define ENTRCNT 2
static func ENTRYARR [ENTRCNT]
{
    printf,
    puts,
};

main(CNT, VALS)
int CNT;
char **VALS;
{
    int COL;
    int HND;
    ovrhead *PTR;
    int RT;
    /*-----*/
    printf("начало работы\n");
    RT = intcall(ENTRCNT, ENTRYARR);
    printf("обработчик прерывания активизирован, RT = %d\n", RT);

    printf("файл %s \n", VALS[1]);
    HND = open(VALS[1], O_RDONLY);
    printf("открыт, HND=%d, загрузка: \n", HND);
    if (HND < 0)
        return -1;
    RT = ovrload(HND, &PTR); /* загрузить программу */
    close(HND);

    printf("выполнена, RT=%d, вызов: \n", RT);
    RT = ovrcall(PTR, VALS[2]);
    printf("выполнен, RT=%d \n", RT);
    ovrfree(PTR);
    return 0;
}

```

Рис. 1. Головная программа

коды программ работы с данными и соответствующие управляющие блоки находятся в ядре). При определенных соглашениях о режиме обработки данных и вы-

```

/* Загружаемая программа */
ovrmain(VALS)
char *VALS;
{
    int RT;
    /*-----*/
    RT = intcall(1, "вызов puts");
    intcall(0, "ovrmain: RT=%d, VALS=%s\n",
        RT, VALS); /* обращение к printf */
    return 4521;
}

```

Рис. 2. Загружаемая программа

воде информации на экран возможна совместная работа нескольких программ-процессоров с общими данными. Переключные режимы по команде пользователя приводит к тому, что одна из подпрограмм завершает свою работу, а головная программа в соответствии с кодом возврата подпрограммы производит запуск другой подпрограммы с передачей адресов управляющих блоков.

Другим способом организации взаимодействия подпрограмм может быть резервирование клавиши под вызов «главного меню». При ее нажатии завершается работа процессора, происходит возврат в ядро и вы-

вод на экран меню, в котором выбирается новый обрабатывающий процессор. Это позволяет продолжить обработку данных в другом режиме. В зависимости от наличия или отсутствия ОП ядро может перекрывать программы или одновременно держать их в памяти.

Возможна независимая разработка на языке Си специализированных программ-процессоров, ориентированных на нужды конкретных пользователей.

Рассмотренный подход предлагается для развития интегрированной системы «Спектр» [8], разрабатываемой в ВЦ АН СССР. Ее основные компоненты: администратор окон, база данных и администратор фреймов, процессор трафаретов, текстовый редактор, подсистема деловой графики, вычислительный процессор, программы печати и архивирования информации из БД.

В основном используется простая структура программ. Через главное меню выбирается нужная точка зрения, из таблицы возможен вызов трафарета. Для редактирования динамически вызывается мини-редактор — программа, предназначенная для работы в среде MS DOS. Аналогично реализован процессор деловой графики, однако информация к этим компонентам поступает через специальные текстовые файлы (использование ОП не очень эффективно). Программы преобразования и копирования информации работают только автономно, и их динамический вызов из главной программы недопустим из-за особенностей буферизации данных в используемой БД нижнего уровня.

Следующие версии системы состоят из ядра и специализированных динамически загружаемых программ-процессоров. В ядро включаются программы доступа к фреймам, БД и программы стандартной библиотеки (распределение памяти, форматные преобразования и т. п.). Администратор окон остается резидентной системой или включается в ядро [9]. Оставшиеся компоненты разбиваются по следующим программам-процессорам: администратор фреймов (создание и корректировка описателей фреймов), процессор выбора и активизации/деактивизации точки зрения, администратор таблиц (создание и корректировка описателей таблиц), табличный редактор (корректировка данных через таблицы), администратор трафаретов, трафаретный редактор (корректировка отдельных записей БД), редактор фильтров, редактор текстов, (возможно дальнейшее расширение списка: графика, вычисления и т. п.).

Каждая программа-процессор представляет собой динамически загружаемый модуль. Такой модуль рассчитан на получение адресного параметра с указателем на блок управления задачами (TCB). В нем указаны текущий открытый фрейм, свойство, активные точки зрения и др.

Ядро обеспечивает переключение процессоров (меняются точки зрения на объект), работу с вложенными объектами (типа таблица в таблице и т. п.). При поступлении команды перехода на подобъект завершается текущий процесс и создается новый блок управления задачами с сохранением старого в «стеке». По окончании работы с данным объектом происходит возврат к предыдущему или выход на меню выбора.

Телефон: 135-13-40, Москва

#### ЛИТЕРАТУРА

1. Microsoft C. Compiler for the MS DOS Operating System. Users Guide. Microsoft Corporation, 1986.
2. Лебедев В. Н., Соколов А. П. Введение в систему программирования ОС ЕС.— М.: Статистика, 1978.
3. Disc Operating System Technical Reference. Microsoft Corporation, 1983.
4. Керниган Б., Ритчи Д. Язык программирования Си.— М.: Финансы и статистика, 1985.
5. Гнездилова Г. Г. К созданию интегрированной операционной среды персонального компьютера.— М.: ВЦ АН СССР, 1985.

(Окончание см. на стр. 66)

И. В. Гуржуенко, В. Ю. Лапидус,  
Л. А. Летник

## КРОССАССЕМБЛЕР ДЛЯ МП КР580 НА БЕЙСИКЕ

Предлагаемый транслятор ориентирован на общепринятую версию ассемблера и написан на языке БЕЙСИК, используемом в СМ ЭВМ и микроЭВМ типа ДВК (версия BASIC/RAFOS V02-030). Транслятор построен по двухпроходной схеме: собирается таблица символов, включающая имена меток и констант; затем осуществляется синтаксический анализ, генерируются диагностические сообщения, выдается листинг и распечатывается таблица символов.

Директивы транслятора: ORG — задание начального адреса, DS — определение памяти, DB — определение байтов, DW — определение слова, EQ — присваивание имени константе, END — окончание ассемблирования. Его программа структурирована и все логические модули выделены в распечатке. Ограничения, накладываемые на язык ассемблера при использовании данного транслятора, вызваны желанием сократить его объем (отсутствуют модули, создающие 16-ричный файл и возможность выполнения операции в поле операндов при трансляции, недопускается наличие разделителей в поле операндов, запрещается символьное предоставление констант).

Для ассемблирования необходимо запустить программу соответствующей командой на языке БЕЙСИК, указав имя транслируемого файла. Невыполнение команды запуска, вызванное ограничением памяти, устраняется использованием команды монитора SET USR SWAP и повторным запуском.

Пример работы ассемблера приведен на рисунке. В конце второго прохода формируются и выводятся листинг, таблица символов и статистические сообщения.

Первый столбец листинга трансляции — порядковый номер строки, второй — адрес, далее 16-ричное представление байтов команды и мнемоническое, затем комментарий. Таблица символов представлена в две колонки: слева — символическое имя, справа — 16-ричный адрес или константа. В статистическом сообщении выдается число обнаруженных ошибок, байтов и строк программы.

Работа ассемблера заканчивается выводом на печать загрузочного модуля, представляющего собой таблицу, в строках которой распечатывается через пробел содержимое ячеек памяти, объединенных в группы по 16. Каждая строка начинается с 16-ричного адреса ячейки (адрес второй строки увеличен на 16 и т. д.).

```

100 B-TTYSET(255,255)
110 REM *****
120 REM * КРОСС - АССЕМБЛЕР ДЛЯ МИКРОПРОЦЕССОРА КР580ИК80А / МИГ (1988) *
130 REM *****
135 REM
140 REM |----- ЗАПУСК И ОТКРЫТИЕ ФАЙЛОВ -----|
150 PRINT \ PRINT "КРОСС - АССЕМБЛЕР КР580" \ PRINT
160 PRINT "ВХОДНОЙ ФАЙЛ ";CHR$(27); \ INPUT I$ \ IF I$="" THEN STOP
170 PRINT "ФАЙЛ ЛИСТИНГА ";CHR$(27); \ INPUT L$ \ PRINT
180 IF POS(I$,".",1)=0 THEN I$=I$+".ASH"
190 OPEN I$ FOR INPUT AS FILE $I
200 IF L$="" THEN L$="TT:"
210 IF POS(L$,".",1)=0 THEN L$=L$+".LST"
220 OPEN L$ FOR OUTPUT AS FILE $2
1000 REM
1001 REM |----- ИНИЦИАЛИЗАЦИЯ МАССИВОВ И ВЕРЕМЕННЫХ -----|
1002 DIM M$(100),A(100),C$(255),CX(255),H$(15),E0(50)
1005 E0$="????" \ E1$="65535 D" \ E2$="255 B" \ T$=CHR$(9) \ O$=CHR$(15)
1007 E3$="ПЕРЕПОЛНЕНИЕ, ОПЕРАНД" \ B$=CHR$(13) \ G$=CHR$(7)
1008 E4(0)=E0$E3$E2$E1$ \ E5(1)=E0$E3$E1$E1$
1020 E4(2)=E0$E3$HEADQ$СТИМЫЙ СИМВОЛ В ОПЕРАНДЕ"
1030 E4(3)=E0$E3$ПОВТОРНОЕ ОПРЕДЕЛЕНИЕ МЕТКИ"
1040 E4(4)=E0$E3$НЕОПРЕДЕЛЕННЫЙ АРГУМЕНТ"
1050 E4(5)=E0$E3$НЕОПЗНАННАЯ КОМАНДА"
1060 E4(6)=E0$E3$НЕОПРЕДЕЛЕННАЯ МЕТКА"
1100 DATA "0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"
1110 FOR IX=0 TO 15 \ READ H$(IX) \ NEXT IX
1200 DATA "NOP", "LXI B", "STAX B", "INX B", "INR B", "DCR B", "MVI B", "RLC"
1210 DATA "0000", "DAD B", "LDAX B", "DCX B", "INR C", "DCR C", "MVI C", "RRC"
1220 DATA "0000", "LXI D", "STAX D", "INX D", "INR D", "DCR D", "MVI D", "RAL"
1230 DATA "0000", "DAD H", "LDAX H", "DCX H", "INR H", "DCR H", "MVI H", "RAR"
1240 DATA "0000", "LXI SP", "SHLD", "INX SP", "INR L", "DCR L", "MVI L", "CHA"
1250 DATA "0000", "DAD H", "LHLD", "INX SP", "INR M", "DCR M", "MVI M", "STC"
1260 DATA "0000", "LXI SP", "STA", "INX SP", "INR A", "DCR A", "MVI A", "CMC"
1270 DATA "0000", "DAD SP", "LDA", "LDC SP", "INX SP", "INR A", "DCR A", "MVI A", "STC"
1280 DATA "MOV B,B", "MOV B,C", "MOV B,D", "MOV B,E", "MOV B,H", "MOV B,L"
1290 DATA "MOV B,M", "MOV B,A", "MOV C,B", "MOV C,D", "MOV C,E", "MOV C,H", "MOV C,L"
1310 DATA "MOV C,M", "MOV C,A", "MOV D,B", "MOV D,C", "MOV D,E", "MOV D,H", "MOV D,L"
1320 DATA "MOV D,M", "MOV D,A", "MOV E,B", "MOV E,C", "MOV E,D", "MOV E,H", "MOV E,L"
1330 DATA "MOV E,M", "MOV E,A", "MOV H,B", "MOV H,C", "MOV H,D", "MOV H,E"
1340 DATA "MOV H,M", "MOV H,A", "MOV L,B", "MOV L,C", "MOV L,D", "MOV L,E", "MOV L,H", "MOV L,M", "MOV L,A"
1350 DATA "MOV L,M", "MOV L,A", "MOV M,B", "MOV M,C", "MOV M,D", "MOV M,E", "MOV M,H", "MOV M,L"
1360 DATA "MOV M,M", "MOV M,A", "MOV A,B", "MOV A,C", "MOV A,D", "MOV A,E"
1370 DATA "MOV A,H", "MOV A,L", "MOV A,M", "MOV A,A"
1380 DATA "ADD B", "ADD C", "ADD D", "ADD E", "ADD H", "ADD L", "ADD M", "ADD A"
1400 DATA "ADC B", "ADC C", "ADC D", "ADC E", "ADC H", "ADC L", "ADC M", "ADC A"
1410 DATA "SUB B", "SUB C", "SUB D", "SUB E", "SUB H", "SUB L", "SUB M", "SUB A"
1420 DATA "SBB B", "SBB C", "SBB D", "SBB E", "SBB H", "SBB L", "SBB M", "SBB A"
1430 DATA "ANA B", "ANA C", "ANA D", "ANA E", "ANA H", "ANA L", "ANA M", "ANA A"
1440 DATA "XRA B", "XRA C", "XRA D", "XRA E", "XRA H", "XRA L", "XRA M", "XRA A"
1450 DATA "ORA B", "ORA C", "ORA D", "ORA E", "ORA H", "ORA L", "ORA M", "ORA A"
1460 DATA "CMA B", "CMA C", "CMA D", "CMA E", "CMA H", "CMA L", "CMA M", "CMA A"
1470 DATA "CMP B", "CMP C", "CMP D", "CMP E", "CMP H", "CMP L", "CMP M", "CMP A"
1480 DATA "RNC", "JNC", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP"
1490 DATA "RZ", "RET", "JZ", "JB", "JNB", "JNC", "CALL", "ACI", "RST 0"
1500 DATA "RNC", "JNC", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP"
1510 DATA "RZ", "RET", "JZ", "JB", "JNB", "JNC", "CALL", "ACI", "RST 1"
1520 DATA "RNC", "JNC", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP"
1530 DATA "RZ", "RET", "JZ", "JB", "JNB", "JNC", "CALL", "ACI", "RST 2"
1540 DATA "RNC", "JNC", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP", "JMP"
1550 DATA "RZ", "RET", "JZ", "JB", "JNB", "JNC", "CALL", "ACI", "RST 3"
1560 FOR IX=0 TO 255 \ READ C$(IX) \ NEXT IX
1570 DATA 1,3,1,1,1,1,2,1,0,1,1,1,1,1,1,2,1,0,3,1,1,1,1,2,1,0,1,1,1,1,1,2,1
1580 DATA 0,3,3,1,1,1,2,1,0,1,3,1,1,1,2,1,0,3,3,1,1,1,1,2,1,0,1,3,1,1,1,1,2,1
1590 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1600 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1610 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1620 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1630 DATA 1,1,3,3,3,1,3,1,1,3,0,3,3,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1640 DATA 1,1,3,1,3,1,2,1,1,3,1,3,0,2,1,1,1,3,1,3,1,2,1,1,1,3,1,3,0,2,1
1650 FOR IX=0 TO 255 \ READ C$(IX) \ NEXT IX
2000 REM
2005 REM |----- ПЕРВЫЙ ПРОХОД -----|
2010 GZ=1 \ MZ=0 \ NZ=0 \ P=0 \ MZ=0
2120 IF END $1 THEN 3000 \ LINPUT $1:$S$ \ NZ=N+1
2025 PRINT $1;"ПЕРВЫЙ ПРОХОД" \ NZ=N \ GOSUB 10000
2030 IF A1X=0 THEN 2020 \ GOSUB 9000 \ IF KX=1 THEN 3000 \ IF A1X=0 THEN 2020
2040 YZ=POS(S$,".",1) \ IF YZ=0 THEN 2060 \ M$=SEG$(S$+1,YZ-1) \ MZ=MZ+1
2050 GOSUB 0000 \ IF AYZ=1 THEN M$(MZ)=H$(A(MZ)-P)
2055 IF BYZ=1 THEN A(MZ)=A(MZ)-1 \ GO TO 2020
2060 IF BYZ=1 THEN 2020 \ IF AYZ=1 THEN 2070 \ GOSUB 7000 \ IF AYZ=0 THEN 2020
2070 P=P+CX(IX)+T \ IF P=65535 THEN 2020 \ P=ABS(65535-P)-1 \ GO TO 2020
3000 REM
3005 REM |----- ВТОРОЙ ПРОХОД -----|
3010 A0Z=0 \ GZ=2 \ NZ=0 \ P=0 \ RESTORE $1 \ PRINT "КОНЕЦ";G$ \ PRINT
3020 IF END $1 THEN 3000 \ LINPUT $1:$S$ \ NZ=N+1 \ S$=S$
3040 GOSUB 10000 \ PRINT $2:0$ \ PRINT $2:USING "###";N;
3050 IF A1X=0 THEN 3700 \ GOSUB 9000 \ IF KX=1 THEN 3000 \ IF BYZ=1 THEN 3030
3060 IF A1X=0 THEN 3700 \ IF AYZ=1 THEN 3070 \ GOSUB 7000 \ IF AYZ=0 THEN 3030
3070 IF FZ=1 THEN 3075 \ D=P \ GOSUB 15000 \ PRINT $2:USING "RRRR",X$;
3075 PRINT $2:TAB(12); \ D=10+Q \ GOSUB 15000
3080 PRINT $2:USING "RR",SEG$(X$,3,4); \ P=P+CX(IX)+T
3090 P=P+CX(IX)+T \ IF P=65535 THEN 3095 \ P=ABS(65535-P)-1
3095 IF AYZ=1 THEN 3000 \ IF CX(IX)=-1 THEN 3700 \ YZ=POS(S$,".",1)
3100 IF YZ=0 THEN 3500 \ M$=SEG$(S$,YZ+1,LEN(S$)) \ GO TO 3505
3520 M$=SEG$(S$,POS(S$,C$(IX),1),LEN(C$(IX)),LEN(S$))
3530 GOSUB 8000 \ IF AYZ=0 THEN D=A(IBZ) \ GO TO 3520
3510 V$=M$ \ A0Z=1 \ GOSUB 12000 \ IF A3Z=1 THEN 3630
3520 GOSUB 15000 \ PRINT $2:USING "RR",SEG$(X$,3,4);
3530 IF CX(IX) < 3 THEN 3700 \ PRINT $2:USING "RR",SEG$(X$,1,2);
3700 PRINT $2:TAB(24);F0$ \ YZ=POS(S$,".",1) \ IF YZ=0 THEN 3030
3710 M$=SEG$(S$,1,YZ-1) \ BYZ=1 \ GOSUB 8000 \ GO TO 3030
3900 IF CX(IX)=2 THEN \ PRINT $2:USING "RR",SEG$(X$,1,2);
3910 GOSUB 13000 \ IF AYZ=0 THEN 3700 \ GO TO 3030
3920 GOSUB 6000 \ PRINT $2 \ PRINT $2; -- ОШИБОК ОБНАРУЖЕНО :";M1Z;

```



УДК 681.325

В. А. Воложанин, В. А. Скворцов,  
Н. Е. Слипень

## ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ ДЛЯ ИНФОРМАЦИОННО- ИЗМЕРИТЕЛЬНЫХ СИСТЕМ СВЯЗИ

В технике связи работа автоматизированных информационно-измерительных систем (АИИС) связана с необходимостью обмена информацией с большим числом внешних устройств, решением многих задач в реальном масштабе времени, большой продолжительностью непрерывной работы при сохранении устойчивости к отказам.

Для использования в АИИС разработан комплекс микропроцессорных средств (КМС), обеспечивающий потребителю возможность создания систем различной конфигурации для АИИС.

Один из вариантов КМС — система автоматизации измерений параметров каналов и трактов связи (САИПКТ), позволяет автоматизировать при создании узлов аппаратуры связи разнообразные измерения (коэффициента передачи, нелинейных искажений, шумов и т. п.), повысить их точность, снизить требования к квалификации оператора. При серийном выпуске аппаратуры связи количество контролируемых параметров значительно меньше, чем при лабораторных исследованиях, поэтому аппаратура контроля может иметь более простую структуру.

Установка для измерения усиления (УИУ-100) автоматически прецизионно измеряет коэффициент усиления систем связи по шлейфовым соединениям и их узлов в диапазоне частот до 100 МГц.

Задачи контроля, управления и сбора информации от большого количества датчиков, освобождая от них центральную ЭВМ, решает автоматизированный комплекс обработки технологической информации (АКОТИ), включенный в локальную вычислительную сеть, состоящая из трех АКОТИ, объединенных в единую систему посредством волоконно-оптических линий связи.

Волоконно-оптические компоненты позволяют развести АКОТИ до расстояния в несколько километров и передавать цифровые сигналы без регенерации с заданным быстродействием.

Адрес для справок: 197022, Ленинград, ул. 3-я Портновой, д. 48, кв. 22.  
Тел. 254-52-72

```

0000 PRINT #2: " ВСЕГО НАЙДЕНА"
0001 PRINT #2: " СТРОК В ПРОГРАММЕ"
0002 PRINT #2: " ТАБ(19)"; " ***** КОНЕЦ АССЕМБЛИРОВАНИЯ *****" \ CLOSE \ STOP
0003 REM
0004 REM | ВВОД ТАБЛИЦЫ СИМВОЛОВ |
0005 PRINT #2 \ PRINT #2 \ PRINT #2
0006 PRINT #2: " ТАБ(30)"; " ТАБЛИЦА СИМВОЛ" \ PRINT #2
0007 PRINT #2: " ТАБ(10)"; " СИМВОЛ"; " ТАБ(20)"; " ЗНАЧЕНИЕ";
0008 PRINT #2: " ТАБ(50)"; " СИМВОЛ"; " ТАБ(60)"; " ЗНАЧЕНИЕ" \ PRINT #2
0009 FOR IX=1 TO MZ STEP 2 \ D=A(IX) \ GOSUB 15000
0010 PRINT #2: " ТАБ(10)"; " M$(IX)"; " ТАБ(21)"; " X$"; \ D=A(IX+1) \ GOSUB 15000
0011 IF (IX+1)=(MZ THEN PRINT #2: " ТАБ(50)"; " M$(IX+1)"; " ТАБ(61)"; " X$
0012 NEXT IX \ PRINT #2 \ RETURN
0013 REM
0014 REM | ПОИСК МНОМОЧКИ В ТАБЛИЦЕ КОМАНД |
0015 A7Z=1 \ FOR IX=0 TO 255
0016 YZ=POS(S$,C$(IX),1) \ IF YZ=0 THEN 7030 \ IOX=IX \ RETURN
0017 NEXT IX \ EX(5)=1 \ GOSUB 13000 \ P=P+1 \ P0=P0+1 \ A7Z=0 \ RETURN
0018 REM
0019 REM | ПОИСК МЕТОК И ИКЕН В ТАБЛИЦЕ СИМВОЛОВ |
0020 AB=1 \ FOR IBX=1 TO MZ \ IF M$(IBX) THEN 8070
0021 IF GZ=1 THEN MZ=MX-1 \ UX=UX+1 \ E0(UX)=P \ GO TO 8060
0022 IF BZ=0 THEN 8060 \ FOR JX=1 TO UX \ IF P=E0(JX)+C$(IX) THEN 8060
0023 NEXT JX \ GO TO 8060
0024 PRINT #2: " E$(3) \ WJZ=MJZ+1
0025 MZ=0 \ BZ=0 \ RETURN
0026 NEXT IBX \ BZ=0 \ RETURN
0027 REM
0028 REM | СЕПАРАТОР ПСЕВДОКОМАНД |
0029 YZ=LEN(S$) \ A9Z=1 \ B9Z=0 \ KX=0 \ Q=0 \ T=0 \ F1Z=0 \ A1Z=1 \ F2Z=0
0030 IF POS(S$, "END", 1)=0 THEN 9020 \ IF YZ=3 THEN IF GZ=1 THEN KX=1 \ RETURN
0031 PRINT #2: " TAB(24)"; " S0$"; \ KZ=1 \ RETURN
0032 OZ=POS(S$, "DRB", 1) \ IF OZ=0 THEN 9040 \ A1Z=0
0033 V$=SEG$(S$, OZ+4, YZ) \ GOSUB 12000 \ GOSUB 9130 \ P=D \ RETURN
0034 OZ=POS(S$, "DS", 1) \ IF OZ=0 THEN 9060 \ V$=SEG$(S$, OZ+3, YZ) \ IX=0
0035 A4Z=1 \ A0Z=1 \ GOSUB 12000 \ GOSUB 9130 \ IX=8 \ IOZ=0 \ T=D \ RETURN
0036 OZ=POS(S$, "DB", 1) \ IF OZ=0 THEN 9080 \ V$=SEG$(S$, OZ+3, YZ) \ IX=0
0037 A0Z=1 \ A4Z=1 \ GOSUB 12000 \ GOSUB 9130 \ Q=D \ IX=0 \ IOZ=0 \ RETURN
0038 OZ=POS(S$, "DW", 1) \ IF OZ=0 THEN 9100 \ V$=SEG$(S$, OZ+3, YZ) \ IX=1
0039 A0Z=1 \ A4Z=1 \ GOSUB 12000 \ GOSUB 9130 \ Q=D \ IX=6 \ IOZ=0 \ RETURN
0040 OZ=POS(S$, "EQU", 1) \ IF OZ=0 THEN A9Z=0 \ RETURN
0041 IX=8 \ V$=SEG$(S$, OZ+4, YZ) \ F1Z=1 \ A4Z=1 \ F2Z=1 \ A0Z=1
0042 GOSUB 12000 \ GOSUB 9130 \ IF DX=255 THEN 9127 \ IX=6
0043 Q=D \ IOZ=0 \ GO TO 9140
0044 IF A3Z=0 THEN 9200 \ B9Z=1 \ RETURN
0045 M$=SEG$(S$, 1, OZ-1) \ M$=TRM$(M$) \ IF M$(M$) THEN 9170
0046 IF GZ=1 THEN 9200 \ EX(6)=1 \ GOSUB 13000 \ GO TO 9130
0047 MZ=MX+1 \ GOSUB 8000 \ IF A8Z=1 THEN 9190 \ MZ=MX-1 \ RETURN
0048 M$(MZ)=M$ \ A(MZ)=D
0049 RETURN
0050 REM
0051 REM | ОТСЕЧЕНИЕ КОММЕНТАРИЕВ, УПЛОТНЕНИЕ СТРОКИ |
0052 A1Z=1 \ YZ=POS(S$, ";", 1) \ IF YZ=0 THEN 10010
0053 IF YZ=1 THEN A1Z=0 \ RETURN
0054 S$=SEG$(S$, 1, YZ-1)
0055 S2$="" \ I1Z=1 \ S$=TRM$(S$) \ IF S$="" THEN A1Z=0 \ RETURN
0056 FOR IX=0 TO LEN(S$) \ S1$=SEG$(S$, IX, IX)
0057 IF I1Z=0 THEN 10060
0058 IF S1$="" THEN 10090 \ IF S1$=T$ THEN 10090 \ I1Z=0
0059 S2$=S2$+S1$ \ GO TO 10090
0060 IF S1$="" THEN I1Z=1 \ GO TO 10050
0061 IF S1$=T$ THEN S1$="" \ I1Z=1
0062 GO TO 10050
0063 NEXT IX \ S$=S2$ \ S$=TRM$(S$) \ IF S$="" THEN A1Z=0
0064 RETURN
0065 REM
0066 REM | ПРЕОБРАЗОВАНИЕ АРГУМЕНТА V$ В ДЕСЯТИЧНОЕ ЧИСЛО D |
0067 REM | ПО ИДЕНТИФИКАТОРУ КОНСТАНТЫ CB, D, O, H, J |
0068 IF V$="" THEN EX(4)=1 \ GO TO 13000
0069 IF SEG$(V$, 1, 1)="" THEN 12015 \ V$=SEG$(V$, 2, LEN(V$)) \ GO TO 12012
0070 N$=SEG$(V$, 1, 1) \ IF N$="" THEN IF N$="" THEN 12025
0071 EX(6)=1 \ GO TO 13000
0072 YZ=LEN(V$) \ N$=SEG$(V$, YZ, YZ) \ X$=SEG$(V$, 1, YZ-1)
0073 IF N$="H" THEN GOSUB 14000 \ GO TO 13000
0074 IF N$="D" THEN GOSUB 16000 \ GO TO 13000
0075 IF N$="Q" THEN GOSUB 18000 \ GO TO 13000
0076 IF N$="O" THEN GOSUB 18000 \ GO TO 13000
0077 IF N$="B" THEN GOSUB 17000 \ GO TO 13000
0078 IF N$="" THEN EX(2)=1 \ GO TO 13000
0079 IF N$="" THEN EX(2)=1 \ GO TO 13000
0080 X$=V$ \ GOSUB 16000
0081 REM
0082 REM | ОБРАБОТКА ОШИБОК |
0083 A3Z=0 \ FOR IX=0 TO 6 \ IF EX(IX)=1 THEN 13030 \ NEXT IX \ RETURN
0084 EX(IX)=0 \ D=0 \ A3Z=1 \ IF GZ=2 THEN 13040
0085 IF CX(IX) THEN 13060 \ P=P+1 \ P0=P0+1 \ GO TO 13080
0086 IF A4Z=1 THEN 13050 \ A4Z=0 \ D=P \ GOSUB 15000
0087 IF F2Z=1 THEN 13050 \ PRINT #2: " USING "RRRR "X$;
0088 IF A0Z=1 THEN PRINT #2: " TAB(12)"; \ PRINT #2: " 00"; \ A0Z=0
0089 IF CX(IX) THEN 13070 \ P=P+1 \ P0=P0+1 \ PRINT #2: " 00";
0090 PRINT #2: " TAB(24)"; " S0$"; \ PRINT #2: " E$(13)"; \ WJZ=MJZ+1
0091 IF F1Z=0 THEN IF A9Z=1 THEN P=P+1 \ P0=P0+1
0092 F1Z=0 \ RETURN
0093 REM
0094 REM | ШЕСТНАДЦАТИРИЧНОЕ СИМВОЛЬНОЕ В ЧИСЛО CX$ -> DJ |
0095 B=0 \ GOSUB 19000 \ I4Z=LEN(CX$) \ IF I4Z=4 THEN EX(1)=1 \ RETURN
0096 FOR JX=0 TO I4Z-1 \ B9Z=SEG$(CX$, I4Z-JX, I4Z-JX)
0097 IF JX=0 TO 15 \ IF N$(JX)=S9Z THEN 14070
0098 NEXT JX \ EX(2)=1 \ RETURN
0099 D=D+J1X*(16^JX) \ NEXT JX \ RETURN
0100 REM
0101 REM | ЧИСЛО В ШЕСТНАДЦАТИРИЧНОЕ СИМВОЛЬНОЕ CD -> X$ |
0102 IF D=65535 THEN EX(1)=1 \ RETURN
0103 X$="" \ I=D \ IF D=0 THEN I=65535+I+1
0104 FOR JX=3 TO 0 STEP -1
0105 I1=16^JX \ I2=INT(I/I1) \ I=I-12*I1 \ X$=X$+I2$

```

## ОБЪЯВЛЕНИЕ — РЕКЛАМА

Всесоюзный научно-исследовательский институт электромеханики (ВНИИЭМ) предлагает в рамках договора комплект технической документации на:

### УСТРОЙСТВО ВВОДА ПРОГРАММ В ЦВМ С ВНЕШНЕЙ ПОСТОЯННОЙ ПАМЯТЬЮ,

включая программное обеспечение (монитор), обеспечивающее через последовательный токовый интерфейс быстрый (9600 бод) ввод в байтовом формате информации из перепрограммируемых микросхем, постоянной памяти серии К573РФ2, К573РФ4, устанавливаемых на сменных кассетах. Общий объем информации составляет 128 Кбайт на кассету, количество которых может быть любым;

### ВХОДНОЙ КОНТРОЛЬ МИКРОЭЛЕКТРОННЫХ ЭЛЕМЕНТОВ,

включая программноносители расчета сигнатур и комплект карт контроля, для системы функционального сигнатурного контроля: цифровых микросхем малой и средней степени интеграции всех ТТЛ и КМОП-серий; аналоговых микросхем операционных усилителей, преобразователей, интегральных стабилизаторов; оптронов, сборок транзисторных (транзисторов) и диодных (диодов), тиристорных; микросхем оперативной и постоянной памяти произвольной емкости и организации;

### УСТРОЙСТВО СОПРЯЖЕНИЯ С МАГНИТОФОНОМ (УСМ) —

— скорость обмена: с телетайпом 300 бод, с дисплеем и магнитофоном 1200 бод; объем информации на одной кассете 150 Кбайт.

Проводится анализ оборудования потребителя и выдаются рекомендации по адаптации УВП, УСМ к нуждам потребителя, проводятся консультации по их внедрению и перспективному развитию применения.

Запросы направлять по адресу: 107817, Москва, ГСП-6, Хоромный тупик, 4. Телефон: 924-70-04, Кащавцеву Ю. А., Новичку Г. Х.

```

15060 NEXT JX \ RETURN
16000 REM
16005 REM ----- ДЕСЯТИЧНОЕ СИМВОЛЬНОЕ В ЧИСЛО (X% -> D) -----
16020 YX=LEN(X%) \ FOR JX=1 TO YX \ X0%=SEG$(X%,JX)
16030 IF X0%("0") THEN EX(2)=1 \ RETURN
16035 IF X0%("9") THEN EX(2)=1 \ RETURN
16040 NEXT JX \ D=VAL(X%) \ IF D>5535 THEN EX(1)=1
16060 RETURN
17000 REM
17005 REM ----- ДВОЙЧНОЕ СИМВОЛЬНОЕ В ЧИСЛО (X% -> D) -----
17010 YX=LEN(X%) \ FOR JX=1 TO YX \ S9%=SEG$(X%,JX)
17040 IF S9%("0") THEN IF S9%("1") THEN EX(2)=1 \ RETURN
17050 NEXT JX \ IF YX>16 THEN EX(1)=1 \ RETURN
17055 D=BIN(X%) \ RETURN
18000 REM
18005 REM ----- ВОСЬМИРИЧНОЕ СИМВОЛЬНОЕ В ЧИСЛО (X% -> D) -----
18020 GOSUB 19020 \ YX=LEN(X%) \ FOR JX=1 TO YX \ X0%=SEG$(X%,JX)
18030 IF X0%("0") THEN EX(2)=1 \ RETURN
18035 IF X0%("7") THEN EX(2)=1 \ RETURN
18040 NEXT JX \ IF YX>6 THEN EX(1)=1 \ RETURN
18050 IF YX=6 THEN IF SEG$(X%,1,1)>"1" THEN EX(1)=1 \ RETURN
18060 D=OCT(X%) \ RETURN
19000 REM
19005 REM ----- ПОДАВЛЕНИЕ НЕЗНАЧАЩИХ НУЛЕЙ -----
19010 IY=LEN(X%)
19020 IF SEG$(X%,1,1)="0" THEN X%=SEG$(X%,2,IY) \ GO TO 19020
19030 RETURN
19999 REM *****XXXXXXXXXXXXX ***** ДК:18000.BAS *****

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

\*\*\* ПРОГРАММА ПЕРЕФОРМИРОВАНИЯ МАССИВА

Символ	Значение	Символ	Значение
E0	2E	I0	EQU 12000
N0		EQU	1200
HF		EQU	150
N0M1		EQU	1010B

\*\*\* РЕЗУЛЬТАТЫ ИЗМЕРЕНИЙ \*\*\*

Символ	Значение	Символ	Значение
AIN	DS	1	АДРЕС МЕТКИ НАЧАЛА
AIB	DS	170	АДРЕС МЕТКИ ВМТ
AIC	DN	50FFH	АДРЕС ID
NATA	DB	65535	
AADN	DS	2	ТЕКУЩИЙ АДРЕС DNI

\*\*\* НАЧАЛО ПРОГРАММЫ \*\*\*

Символ	Значение	Символ	Значение
FORM1	LXI	H,AIB	ОПРЕДЕЛЕНИЕ НОМЕРА
MAY	A,N		
SUI	HF		
LHLD	AIN		ЧИСЛЕНИЕ И
SUB	L		ЗАПОМИНАНИЕ АДРЕСА
CALL	A1		
MVI	H,		
DAD	H		
EQU	Y		ЧИСЛО УРОВНЕЙ
LXI	D,A10		
DW	B1		
SHLD	AADN		ИНИЦИАЛИЗАЦИЯ СТЕКА
LXI	SP,NATA		
SPHL			
CMR	C		
JZ	SA1		
END			КОНЕЦ

СИМВОЛ	ЗНАЧЕНИЕ	СИМВОЛ	ЗНАЧЕНИЕ
I0	2500	N0	0000
HF	000F	N0M1	000A
AIN	4100	AIB	4101
AIC	4110	NATA	4112
AADN	4113	FORM1	5009
SA1	5009	IRQ	0009
A1	501B		

-- ОШИБОК ОБНАРУЖЕНО : 5  
-- ВСЕГО БАЙТОВ : 57  
-- СТРОК В ПРОГРАММЕ : 34

\*\*\*\*\* КОНЕЦ АССЕМБЛИРОВАНИЯ \*\*\*\*\*  
Листинг ассемблера на языке БЕЙСИК

Предлагаемый транслятор служит для учебных целей, поэтому в исходном тексте программы отсутствует модуль, позволяющий образовать и вести загрузочный модуль.

Телефон: 273-89-26, Москва

(Окончание. Начало см. на с. 57)

6. Чижев А. А. Операционная система ЯНУС// Программное обеспечение персональных ЭВМ.— М.: ВЦ АН СССР, 1985.
7. Brudley D. Assembly Language Programming for the IBM Personal Computer. Prentice-Hall Publishing Company, 1984.

8. Брябрин В. М. Системы интерактивного взаимодействия с персональными ЭВМ// Разработка ЭВМ нового поколения: архитектура, программирование, интеллектуализация.— Новосибирск, 1986.
9. Гнездилова Г. Г., Чижев А. А. Управление окнами на персональной ЭВМ.— М.: ВЦ АН СССР, 1986.

Статья поступила 16 сентября 1987

УДК 681.3.06

А. В. Кизуб

### РИСК — РЕАЛЬНО ИСПОЛЬЗУЕМАЯ СЕТКА КОНСТРУКТОРА

#### Цели разработки

Для интенсификации и автоматизации конструкторских работ необходимо широкое использование интерактивных графических систем, в которых общение конструктора и ЭВМ происходит на языке графики, а не команд или подпрограмм.

В статье кратко описан сам метод РИСК и разработанная на его основе программа, предназначенная для создания двумерных чертежей. В РИСКЕ предлагается следующий подход. Вначале в ЭВМ вводится эскиз — неточное и непропорциональное представление чертежа, с возможными ошибками ввода, но отражающее его комбинаторную структуру, наличие и взаимосвязи элементов будущего чертежа. При этом допустимые ошибки ввода удаляются автоматически. Точный чертеж может быть получен, если параметрам эскиза придать определенные, фиксированные значения. Из одного эскиза проставлением различных параметров можно получить множество чертежей. Кроме того, возможно и обычное редактирование эскиза с соответствующими изменениями в выходных чертежах. Таким образом, РИСК дает два средства для решения конструкторских задач при создании чертежа: корректировку (облагораживание эскиза) и параметризацию при получении чертежа.

**Облагораживание эскиза.** Ввод непосредственно графики в ЭВМ (если это не языковое описание) сопряжен с рядом трудностей, порождаемых неточностью и дискретностью вводных устройств (будь то дигитайзер, графический экран, фотосчитыватель или др.). Наиболее распространенный способ преодоления этих трудностей — кодирование на сетке, т. е. стягивание всех вводных точек в ближайшие узлы регулярной сетки, округление абсцисс и ординат к значению, кратному выбранному шагу сетки. Подобные методы используются практически в каждой системе. В некоторых случаях полезной оказывается нерегулярная сетка, отражающая пропорции чертежа. Такая реальная сетка представляет собой систему вертикальных и горизонтальных прямых, настраиваемую и зависящую от каждого конкретного эскиза. Пересечения линий (узлы сетки) расположены в определенных местах, требуемых для кодирования эскиза.

Применение сетки с нерегулярным шагом значительно облегчает кодирование эскиза, устраняет ошибки кодирования, позволяет многократно указывать одну и ту же точку при замыкании контуров и т. д.

Аналоги нерегулярной сетки довольно широко используются при создании чертежных систем [1]. На экране строятся графические элементы (прямые, окружности), которые путем усечений, добавлений, модификации привносятся к множеству элементов, представляющих чертеж. Но в приведенных системах графические элементы множеств не несут никакой функциональной нагрузки. Нерегулярная же сетка играет активную роль, стягивая в свои узлы привязочные точки других графических элементов, т. е. корректируя чертеж. Однако общим недостатком подобных подходов является необходимость соз-

дания основного множества или нерегулярной сетки путем дополнительных построений. Чертеж создается, таким образом, как бы дважды и вручную, ведь при автоматическом создании сетки конструктор не может на нее повлиять, и, как следствие, отсутствует гибкость в сложных ситуациях. Все взаимодействие при таком способе создания графики происходит через чертеж на экране, но не через эскиз. Да и само понятие эскиза исчезает, так как конструктор вынужден вводить уже непосредственно чертеж, не подлежащий редактированию. При изменениях необходимо просто стереть старые элементы и построить новые. Единственно возможной модификацией всего чертежа может быть преобразование (например, аффинное) всей плоскости чертежа, системы его координат, а с ней и самого чертежа. Общение с графикой с помощью ЭВМ, а не с ЭВМ с помощью графики.

В работах [2, 3] рассмотрены некоторые варианты решения задач ввода и параметризации эскиза и чертежа. Однако эти системы используют графические понятия для общения с ЭВМ, что является особенностью реализации интерфейса. Система РИСК, основываясь только на графике, позволяет ввести следующую параметризацию:

1) каждая точка, стянутая к линии сетки, прикрепляется к ней и при изменении положения этой линии (модификации чертежа) перемещается вместе с ней;

2) линии сетки могут быть связаны между собой.

Если линия В связана с линией А, то это значит, что расстояние от А до В должно быть всегда одинаковым и равным Р. Линию А назовем опорной для этой пары, В — зависимой, Р — параметром взаимосвязи. Положение линий на чертеже можно менять, присвоив конкретной опорной линии новое значение (место расположения), изменив параметр или взаимосвязь. Во всех случаях изменится положение не только этой линии, но и линий, зависящих от нее, а также линий, зависящих, в свою очередь, от них и так далее таким образом, чтобы все взаимосвязи и их параметры сохранились. Сетка и координаты всех точек, увязанных на сетке, изменяются — чертеж параметризуется.

В общем виде сама сетка строится следующим образом. Есть привязочные точки эскиза, введенные ранее или предполагаемые к вводу, необходимые для создания комбинаторной структуры, единой для множества чертежей. На эскизе указываются две точки (порождающие две линии сетки) и требуемое расстояние между ними. Таким образом, показывается взаимосвязь и параметр. Если одна из указываемых точек лежит на уже существующей линии сетки, то она становится опорной для данной пары. Если ни одна из точек не принадлежит другим линиям сетки, то первая из них становится опорной для данной пары и базовой для порождаемой ею группы линий. Если обе точки принадлежат существующим линиям сетки, то необходимо сравнить их базы (самые первые опорные линии в цепочках взаимосвязей): общая база — данное указание излишне; разные базы — вторая линия становится зависимой от первой, а цепочка ее взаимосвязей переориентируется в обратную сторону, т. е. база второй линии становится последней зависимой линией от первой базы с сохранением всех взаимосвязей и их параметров.

Конечное множество точек эскиза преобразуется в множества прямых линий сетки, а затем в множество точек чертежа путем подставления параметров — требу-

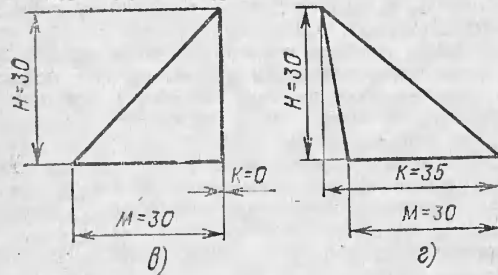
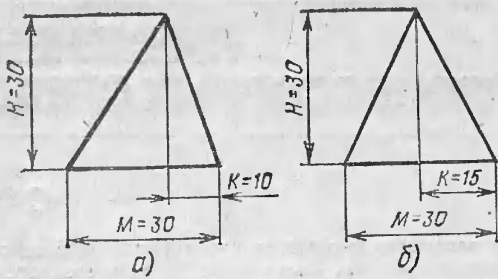


Рис. 1.

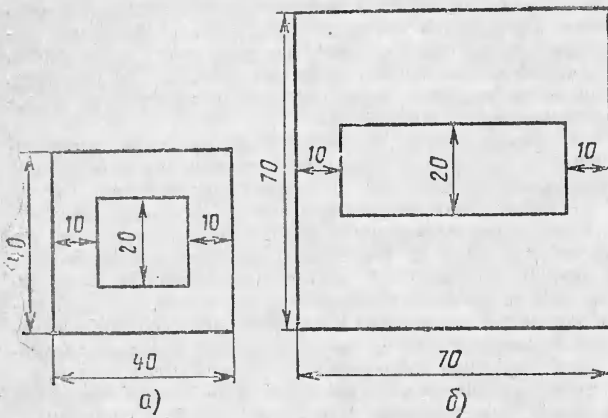


Рис. 2.

мых расстояний между двумя линиями сетки. Все взаимосвязи и их параметры образуют структуру параметров чертежа, в общем случае представляющую собой ориентированное дерево, узлами которого являются линии сетки, а ребрами — взаимосвязи между ними.

**Примеры использования системы**

На рис. 1, а приведен откорректированный эскиз треугольника. Для указания взаимосвязей линий сетки используются размерные стрелки, параметрами являются надписи над ними, т. е. размерные стрелки — семантические указания взаимосвязи, понятные как конструктору, так и машине. На рис. 1, б—г показано, как изменится чертеж треугольника при изменении только одного параметра реальной сетки.

Из приведенного примера, видно, что основа реальной сетки — смысл чертежа; чертеж изменяется путем изменения параметров или их структуры, затрагивая лишь необходимые части рисунка. Обычно в машиностроительном чертеже уже существует или подразумевается реальная сетка с размерами и связями, обозначенными размерными стрелками. Кодирование размерной стрелки и размера над ней указывает наличие двух линий сетки, их взаимосвязь и требуемое расстояние между ними. Подобный подход, дополненный рядом синтаксических правил формирования реальной сетки, прост и естествен.

Размерные стрелки на чертеже указывают не только

реальную сетку, но и семантические взаимосвязи, например, на рис. 2, а горизонтальные стрелки указывают ширину рамки, а вертикальная — высоту проема. С точки зрения синтаксиса задания реальной сетки они тождественны, семантическая же их нагрузка явно видна при простом увеличении размера рамки методом РИСК. Рамка увеличивается, но ее ширина и высота проема остаются постоянными (рис. 2, б).

На рис. 3, а изображен первичный эскиз, а на рис. 3, б—г три чертежа совершенно различных фигур, полученных из этого эскиза: лезвие «жиллет», прокладки или даже пуговицы. Все эти чертежи получены лишь подстановкой различных параметров, без изменения эскиза. Единственным дополнением к описываемому методу на данном чертеже является специальное кодирование сопряжений и окружностей, предназначенное специально для РИСКА. Радиусы дуг также задаются соответствующими параметрами.

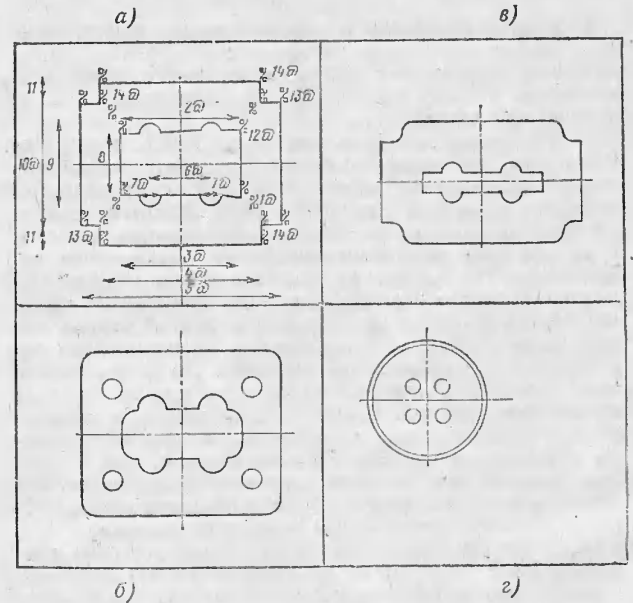


Рис. 3

Метод РИСК — новый подход к получению компьютерной модели среди отечественных систем. Если обычно в ЭВМ можно вводить только точный чертеж, описываемый координатами точек или заранее рассчитанными числами, каким-либо образом попавшими в ЭВМ (в результате вычислений программ или действий оператора), то с помощью РИСКА из приблизительного эскиза можно получить точные результаты, на основании которых будет выводиться окончательный полноценный чертеж или работать программа.

**Программная реализация**

Разработана программа, обрабатывающая графическую информацию в виде основного списочного массива автоматизированного рабочего места АРМ2-01. Программа представляет собой простой двухпроходный транслятор, работающий с массивами, содержащими списки опорных координат. На вход подается массив, описывающий эскиз, на выходе — массив, описывающий чертеж. Скорость обработки информации позволяет использовать программу в системе интерактивного графического диалога.

При первом проходе в массиве идентифицируются размерные линии и надписи над ними, строится модель реальной сетки. Определяющая для линий сетки (ЛС) — размерная линия, соединяющая две ЛС. Наклон самой размерной линии указывает тип ЛС: вертикальные или горизонтальные. Самая первая из указы-

ваемых ЛС является базовой. Ее положение на чертеже не изменится ни при каких значениях параметров до тех пор, пока она остается базовой и не переопределена как зависимая. Зависимые ЛС — все остальные линии, которые ссылаются на базовые прямо или косвенно, через другие зависимые ЛС. При построении сетки она представляется как дерево взаимосвязей.

Для второго прохода сетка преобразуется в два массива пар чисел, представляющих собой значения абсцисс и ординат на эскизе и чертеже. Тем самым сетка строится исключительно из ортогональных линий. Использование только подобной сетки не всегда удобно. В настоящее время ведется работа по использованию в качестве линий реальной сетки наклонных линий и окружностей, что позволит работать в полярной системе координат.

При втором проходе изменяются значения каждой координаты отдельно по абсциссе и ординате. Старое значение по эскизу меняется на новое, соответствующее ему на чертеже. Кроме того, производится облагораживание размерных указаний. Размерные линии становятся параллельными осям координат, выносные линии подтягиваются к размерным и вводятся перпендикулярно им, надписи ставятся над центром размерной линии либо в дополнительно указанном месте. Программа имеет дополнительные возможности обработки размерных указаний, связанных с общей идеей метода. Ведь информацию о параметрах можно не только вкладывать, но и получать из сетки. Если значение параметра не указано, оно берется из длины самой размерной стрелки на эскизе. Лишние указания (общая база у двух повторно указываемых линий сетки) игнорируются, но над размерной линией в этом случае ставится фактически существующий размер, помечаемый звездочкой как справочный.

Возможны и еще интересные применения РИСКА, например кодирование эскиза в любом месте листа, в любых пропорциях, масштабах, сетках и единицах измерения, так как выходной чертеж зависит только от значений параметров и их структуры.

Основой конструктору для общения, редактирования, доработки постоянно служит его собственный эскиз, куда он и вносит изменения. Чертеж, воспроизводимый на экране или бумаге, — это лишь отображение существующей структуры параметров эскиза, никоим образом не влияющее на последний. Следует подчеркнуть, что как размерные стрелки, так и параметры являются обычными графическими элементами и могут быть подвергнуты редактированию. Этим достигается простота и единооб-

разие коррекций, параметризации и прочих действий по редактированию чертежа.

К недостаткам метода следует отнести: необходимость задания размерными стрелками всех параметризуемых линий, что обычно не делается на машинностроительных чертежах; линия сетки действует на весь чертеж на всем своем протяжении, мешая кодированию в других частях чертежа. Наиболее эффективна реальная сетка при кодировании одиночных, семантически увязанных частей чертежа, например библиотечных элементов.

Для иллюстрации принципов РИСКА, а также их возможностей в статье приведены условные примеры. Однако разработанная автором программа используется для проектирования чертежей реальных изделий машиностроения, причем первое же ее применение показало и возможности реального контроля чертежей, правильности и полноты использования параметров и непротиворечивости их структуры. Конструктор, создавая чертеж, может ошибаться: выдавать желаемое за действительное, представлять на эскизе нерелевантные размеры и т. д. При использовании метода РИСК эти несоответствия выявляются уже на чертеже, тем самым повышается качество разработок, экономится труд проектировщиков и рабочих.

В целом реальная параметризуемая сетка позволяет значительно облегчить ввод графической информации в ЭВМ при сочетании наглядности и условности эскиза с любой точностью задаваемых параметров. РИСК, отличаясь простотой реализации и понимания, в большей степени соответствует образу мысли конструктора, когда он выражает образы графически, а не прибегает к специальным языкам. Метод РИСК позволяет естественными и привычными для конструктора приемами графики создавать и редактировать чертежи.

Телефон: 435-19-02, Киев

#### ЛИТЕРАТУРА

1. Jansen H., Meyer B. Reconstruction von volumenorientierten 3D-Modellen aus handskizzierten 2D-Ansichten // In: Geometrisches Modellieren: Fachtagung der GI. ed. Nowacki H., Gnatz R.—Berlin: Informatik-Fachberichte—1983.—Vol. 65.
2. Porter S. New tools for concept engineering // Computer Graphics World.—1986.—Vol. 9, N 11.—pp. 40—42.
3. Grieb Ph. Benutzerorientierte, rechnerunterstützte variantenkonstruktion mit der MEDUSA-parametricbaustein // VDI-Z.—1983.—N 13.—Ss. 546—550.

Статья поступила 16 января 1987

УДК 621.372.061.2

Б. В. Баталов, С. Г. Русаков, В. В. Савин

## ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ АВТОМАТИЗАЦИИ СХЕМОТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

Внедрение ПК в САПР и системы автоматизации инженерных исследований стало практической задачей сегодняшнего дня. Увеличение быстродействия и памяти современных ПК позволило использовать проблемно-ориентированные ППП САПР, которые ранее применялись только на высокопроизводительных ЭВМ. К таким пакетам можно отнести программное обеспечение систем автоматизации схемотехнического проектирования (САСП), которые требуют значительных вычислительных ресурсов для реализации численных методов моделирования. Разработка ПО САСП для ПК и организация на его базе инженерных станций обеспечивает ши-

рокие возможности разностороннего исследования проектируемых схем. Такая станция при наличии должных надежностных характеристик приближает схемотехника к привычному для него процессу разработки новых схем. Первые примеры разработки ППП схемотехнического моделирования для ПК [1] подтвердили эффективность их практического применения. Большинство зарубежных ППП для ПК базируются на математическом обеспечении распространенной программы анализа схем SPICE-2. Примером является программа SPICE-PLUS, входящая в состав пакетов Analog Workbench и PC Workbench для проектирования аналоговых схем на ПК

[1] и использующая версии SPICE-2G.6 и SPICE-3. Наибольшее распространение получил пакет P-SPICE для IBM PC, обеспечивающий выполнение основных видов электрического анализа и графическое представление результатов моделирования схем сложностью до 120 транзисторов.

Разработанный ППП автоматизированного расчета интегральных схем (АРИС) обеспечивает интерактивный режим взаимодействия с пользователем (имеются две версии для ПК: микроАРИС, ориентированный на отечественные микроЭВМ класса ДВК [2], и АРИС-РС — для персональных компьютеров PC IBM). В качестве базового ПО и МО используются находящиеся в практической эксплуатации версии ПО АРИС: ППП микроАРИС выполнен на базе диалоговой системы мини-АРИС для машин класса SM [3], а прототипом АРИС-РС является интерактивная система АРИС-ЭЛ для высокопроизводитель-

ных мини-ЭВМ класса «Электроника 52» и систем проектирования «Кулон 4»\*.

**Функциональные возможности.** Работа ППП АРИС-РС осуществляется в диалоговом режиме. Пользователь может выбрать любой из следующих видов моделирования биполярных или МДП ИС: статический, частотный, динамический, гармонический, а также анализ устойчивости и расчет статической передаточной характеристики. Настройка алгоритма моделирования на конкретную задачу определяется параметрами, указанными в директиве (их значения в ППП АРИС-РС могут выбираться по умолчанию).

Особенность ППП АРИС-РС — наличие справочной информации о директивах системы и параметрах.

В пакет АРИС-РС включены следующие модели компонентов: резистор, конденсатор, диод, постоянный источник напряжений, управляемый источник напряжений, биполярный транзистор, МДП-транзистор. Допускается разбиение схемы, возможны неограниченная вложенность фрагментов и использование личной библиотеки фрагментов.

Для версии микроАРИС максимальная размерность анализируемой схемы — 30 узлов, 100 компонентов; АРИС-РС — 80 узлов при количестве компонентов не более 300.

Пакет P-SPICE ориентирован на использование сопроцессора (аппаратных средств для операций с плавающей запятой); применение ППП АРИС-РС возможно и при его отсутствии в конфигурации технических средств РС ИВМ. Отличается также характер обработки задания на расчет (директивы в программе P-SPICE при трансляции не отделяются от описания анализируемой цепи).

\* В разработке базового ПО АРИС принимали участие В. В. Азаров, В. П. Ватагин, М. М. Жаров, А. А. Лялинский, О. В. Петрова, Н. И. Соколова, С. Л. Ульянов, А. В. Левашов.

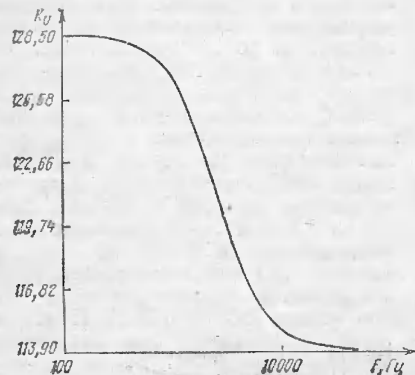


Рис. 1. Рассчитанная амплитудно-частотная характеристика

C) ARIS

SYSTEM ARIS / IBM V 1.0

\* TAKE K140

& K140UD1

T\*1=B(C=1,B=22,E=14); T\*2=B(C=13,B=21,E=14)

T\*3=B(C=14,E=18,B=10)

T\*4=B(C=3,B=13,E=12); T\*5=B(4,12,1)

T\*6=B(10,17,10)

T\*7=B(3,5,4); T\*8=B(6,19,10)

T\*9=B(3,7,6)

MT\*B=EM1(ALFN=0.99)

E\*1=0(21,0); E\*2=INP(22,0)

F\*INP=SIN(A=0.001,F=1E-6)

E\*3=3(3,0); E\*4=-3(9,0)

R\*1=10(2,1); R\*2=9.7(18,9); R\*3=10(2,13)

R\*4=9.7(2,3)

R\*5=3.6(12,0); R\*6=10(10,0); R\*7=2.75(17,9)

R\*8=5(3,4)

R\*9=10(5,6); R\*10=2.1(19,8); R\*11=2.73(7,8)

R\*12=0.59(8,9); R\*13=39(16,9)

C\*1=1500(16,4)

D\*1=A(6,4)

MD\*A=MD1(10-1E-10)

\* STAT

```

$LU=DECOMP. $MDL=CALL $MDL=RET VOLT_ERR CUR_ERR STEP_DIV
% 12 18 0 9.57194E-04 5.52217E-04 -1.00000
  
```

\*\*\* VOLTAGE ON NODES (STATICS) \*\*\*  
 VALUE NODE FRAGMENTS

```

-2.12 1 !
-0.544 14 !
-2.35 8 !
-2.12 13 !
-3.00 16 !
-2.45 18 !
-1.99 10 !
2.42 2 !
1.53 12 !
1.95 4 !
-2.48 17 !
1.43 5 !
1.31 6 !
-2.35 19 !
0.672 7 !
-3.00 9 !
3.00 3 !
0.000E+00 22 !
0.000E+00 21 !
  
```

Программа АРИС-РС позволяет независимо обрабатывать каждую директиву (увеличивается гибкость программы). Возможность вмешательства разработчика в процесс моделирования позволяет менять последовательность расчета в зависимости от визуализации промежуточных результатов моделирования.

Этапы процесса моделирования с помощью ППП АРИС серийно изготавливаемого биполярного операционного усилителя К140УД1: кодировка схемы; подготовка описания (подготовленные заранее файлы используются в сеансе моделирования);

\* SHOW/OP

THE NODES POTENTIALS

NODE	POTENTIAL	NODE	POTENTIAL	NODE	POTENTIAL	NODE	POTENTIAL
1	2.1231	14	-0.54381	22	0.00000E+00	13	2.1231
21	0.00000E+00	18	-2.4517	10	-1.8903	3	3.0000
12	1.5286	4	1.9485	17	-2.4818	5	1.4261
6	1.3075	19	-2.3454	7	0.67182	9	-3.0000
2	2.4214	8	-2.3471	16	-3.0000		

Рис. 2 (Начало)

## RESISTORS

NUMBER	VOLTAGE (V)	CURRENT (MA)	NUMBER	VOLTAGE (V)	CURRENT (MA)
1	0.2982600	2.9825998E-02	2	0.5483479	5.6530714E-02
3	0.2982597	2.9825974E-02	4	-0.5786245	-5.9652008E-02
5	1.528647	0.4246242	6	-1.890265	-0.1890265
7	0.5182462	0.1884532	8	1.051536	0.2103072
9	0.1185632	1.1856318E-02	10	1.6927719E-03	8.0608192E-04
11	3.018904	1.105826	12	0.6529129	1.106632
13	0.0000000E+00	0.0000000E+00			

## D I O D S

NUMBER	VOLTAGE (V)	CURRENT (MA)	NUMBER	VOLTAGE (V)	CURRENT (MA)
1	-0.6409211	-1.0000000E-10			

## B I P O L A R T R A N S I S T O R S

NUMBER	TYPE	VOLTAGE		CURRENT		POWER
		BASE-EMITTER (V)	COL.-EMITTER (V)	COLLECTOR (MA)	BASE (MA)	COL.-EMITTER (MVT)
1	NPN	0.543807	2.66692	2.770294E-02	2.798270E-04	7.38816E-02
2	NPN	0.543807	2.66692	2.770294E-02	2.798270E-04	7.38816E-02
3	NPN	0.561387	1.90785	5.596533E-02	5.653054E-04	0.10677
4	NPN	0.594469	1.47135	0.210190	2.123132E-03	0.30926
5	NPN	0.594468	0.419817	0.210188	2.123132E-03	8.82405E-02
6	NPN	0.591488	0.591488	0.186569	1.884535E-03	0.11035
7	NPN	0.522358	1.57389	1.174687E-02	1.186552E-04	1.84883E-02
8	NPN	0.455129	3.65294	7.980574E-04	8.061179E-06	2.91525E-03
9	NPN	0.635726	2.32818	1.09477	1.105821E-02	2.5488

## THE VOLTAGE SOURCES

NUMBER	CURRENT (MA)	POWER (MVT)	NUMBER	CURRENT (MA)	POWER (MVT)
1	2.7982704E-04	0.0000000E+00	2	-2.7982704E-04	0.0000000E+00
3	1.587142	4.761426	4	-1.351565	4.054694

Рис. 2 (Продолжение)

трансляция (формирование математической модели, определение ошибок описания схемы с точки зрения синтаксиса);

моделирование (выполняется по директивам, результаты представляются в виде таблиц и графиков (рис. 1)).

Для моделирования каждой новой схемы необходима трансляция.

Основные этапы работы ППП АРИС-РС (рис. 2):

трансляция описания схемы, хранящегося в файле с именем К-140;

моделирование статического режима;

просмотр статических характеристик по всем элементам схемы;

анализ амплитудно- и фазо-частотной характеристик.

Длительность этапов моделирования схемы (22 узла 24 компонента):

23 с — трансляция; 15 с — моделирование статического режима; 42 с —

расчет амплитудно- и фазо-частотной характеристик (для РС/АТ без сопроцессора).

Организация ППП АРИС. ППП АРИС-РС — набор программных модулей на языке Фортран-77\*.

Работа ППП АРИС-РС осуществляется под управлением диспетчера, написанного на языке управления операционной системы.

Пакет АРИС-РС (включая руководство по эксплуатации) размещается на шести дискетах (360 Кбайт) и используется в двух вариантах:

на винчестерском диске располагаются все программные модули, которые занимают 1,5 Мбайта, или только служебные файлы, 300 Кбайт (программы видов моделирования на дискетах).

Структура микроАРИС аналогична АРИС-РС: набор программных модулей занимает три дискеты (220 Кбайт).

Пакеты прикладных программ микроАРИС и АРИС-РС можно использовать в качестве базового ПО инженерных станций схемотехнического проектирования на ПК. Перспективы его развития в увеличении сервисных возможностей и расширении функций для обеспечения полного цикла схемотехнических разработок (включение блоков идентификации параметров моделей компонентов и параметрическая оптимизация).

Адрес: 117454, Москва, пр. Вернадского, 78, МИРЭА, кафедра САПР; тел. 434-26-70

## ЛИТЕРАТУРА

1. Беннетт Ив. Пакет для проектирования аналоговых схем на РС

\* Программа графического представления результатов написана на языке Си сотрудником ВЦ АН СССР Тейликиным В. М.

\* FREQ/FBEG=100/FEND=1E6/INP=22/OUT=7/NPOI=4

POINT	FREQUENCY (HZ)	AMPLIFYING	AMPLIFYING(DB)	PHASE	INPUT IMPED. (КОМ)
1	100.000	128.505	42.1784	-0.269424	178.578
2	177.828	128.451	42.1748	-0.477521	178.578
3	316.228	128.275	42.1628	-0.840299	178.578
4	562.341	127.806	42.1310	-1.44726	178.578
5	1000.00	126.489	42.0411	-2.34040	178.578
6	1778.28	123.570	41.8382	-3.23498	178.585
7	3162.28	119.502	41.5475	-3.37834	178.589
8	5623.41	116.310	41.3124	-2.60734	178.593
9	10000.0	114.761	41.1959	-1.66212	178.597
10	17782.8	114.194	41.1529	-0.976008	178.597
11	31622.8	114.009	41.1387	-0.556655	178.597
12	56234.1	113.953	41.1345	-0.314461	178.597
13	100000.	113.924	41.1323	-0.177057	178.597
14	177828.	113.926	41.1324	-9.963425E-02	178.597
15	316228.	113.904	41.1308	-5.597683E-02	178.597
16	562341.	113.914	41.1315	-3.147369E-02	178.597
17	1.000000E+06	113.914	41.1315	-1.771472E-02	178.597

\* EXIT

с)

УДК 681.3.06

О. И. Семенов, В. В. Божуть, Л. А. Гриншпан,  
Я. Т. Малюш, Е. Е. Ткачев, Д. Р. Шерлин

## СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ МУЛЬТИМИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ НА СЕКЦИОНИРОВАННЫХ МИКРОПРОЦЕССОРАХ

Система автоматизированного проектирования (САПР) мультимикропроцессорных устройств (ММПУ) на секционированных микропроцессорах МЕТАМИКРО 3.0, разработанная в Институте технической кибернетики АН БССР, обеспечивает сквозной цикл проектирования ММПУ, включающий разработку и отладку микропрограмм и аппаратуры, подготовку данных для программаторов, выпуск программной и схемной конструкторской документации, формирование машинных моделей электрических схем и блок-схем программ, пригодных для последующей обработки специальными программами. В системе предусмотрена возможность организации индивидуального рабочего места на базе ПЭВМ с подключением до 15 универсальных внутрисхемных эмуляторов (ВСЭ) для отладки микропрограмм и аппаратуры в реальном масштабе времени. Эмуляторы настраиваются на характеристики отлаживаемой части ММПУ программно.

Генерация перемещающих и строчных кроссасемблеров и дизассемблеров для символических языков микропрограммирования и компоновка (сборка) микропрограмм происходят автоматически. Программная и схемная конструкторская документация выпускается в соответствии со стандартами. В систему включен символический отладчик.

Система МЕТАМИКРО 3.0 базируется на ряде концепций, апробированных в предыдущих версиях системы МЕТАМИКРО [1—3]. Разрабатываемое с ее помощью устройство включает несколько блоков управления, каждый из которых содержит кроме прочих узлов память микромайд и, возможно, память мэппинга (карту памяти). Один блок может управлять как одним, так и несколькими микропроцессорами. Все микропроцессоры взаимодействуют друг с другом по алгоритмам, определяемым функциональным назначением устройства,

АТ фирмы IBM // Электроника.— 1986.— № 3.— С. 81—83.

2. Кокорин В. С., Кридинер Л. С., Попов А. А., Хохлов М. М. Тенденция развития диалоговых вычислительных комплексов // Микропроцессорные средства и системы.— 1986.— № 4.— С. 11—15.

3. Баталов Б. В., Русаков С. Г., Фролов В. В. и др. Комплекс программ автоматизированного расчета электрических характеристик интегральных схем на мини-ЭВМ (мини-АРИС) // Микроэлектроника и полупроводниковые приборы.— М.: Радио и связь.— 1984. Вып. 9.— С. 157—174.

Статья поступила 4 июня 1987

Технические средства системы. Ядро САПР МПУ МЕТАМИКРО 3.0 составляет ПЭВМ ДВК-2, ДВК-2М со штатным набором внешних устройств и операционной системой ОС ДВК. Может использоваться также комплекс 15ВУМС 28-025 с операционной системой ФОДОС. К ПЭВМ подключаются до 15 ВСЭ, имеющих интерфейс ИРПР или ИРПС. Эмуляторы соединяются последовательно, каждый из них имеет транслятор интерфейса и уникальный адрес в диапазоне 1...15. Отдельный эмулятор включает эмулирующее оперативное запоминающее устройство (ЭОЗУ), служебную память (СП), трассировочную память (ТП), эмулятор мэппинга (ЭМ), программно управляемый генератор синхронимпульсов, счетчик числа событий и таймер (рис. 1). Максимальная длина слова ЭОЗУ составляет 256 бит, максимальный объем — 64К слов. ЭОЗУ доступно по чтению и записи из ПЭВМ и подключенного к нему ММПУ. Служебная память используется для организации останова, заикливания микропрограмм и управления записью информации в ТП. Трассировочная память запоминает потенциалы сигналов в выбранных пользователями точках отлаживаемого ММПУ.

Эмулятор мэппинга служит для хранения начальных адресов ветвей микропрограмм и используется при работе с микропроцессорными комплектами, имеющими механизм перехода по мэппингу. Четырехбитовый зонд предназначен для останова по событиям. Таймер учитывает время исполнения ветвей микропрограмм. Счетчик необходим для организации останова цикла по заданному числу исполнений любой входящей в него микрокоманды, а также для подсчета числа исполнений отдельной микрокоманды. Программно управляемый генератор синхронимпульсов позволяет формировать на четырехтактных выходах восемь совокупностей сигналов с частотой до 7,5 МГц.

Каждый ВСЭ подключается к одному блоку управления и может использоваться для отладки устройств на секционированных микропроцессорах различных типов. При этом ЭОЗУ подключается вместо памяти микрокоманд, а ЭМ вместо памяти мэппинга. Если блок управления имеет собственный генератор, то его пуски и останова осуществляются ВСЭ, в противном случае блок синхронизируется генератором ВСЭ.

Программное обеспечение системы реализовано в виде пяти водсистем, поддерживающих сквозной цикл проектирования ММПУ. Подсистема автоматической генерации кроссасемблеров и дизассемблеров (ПАГАД) предназначена для



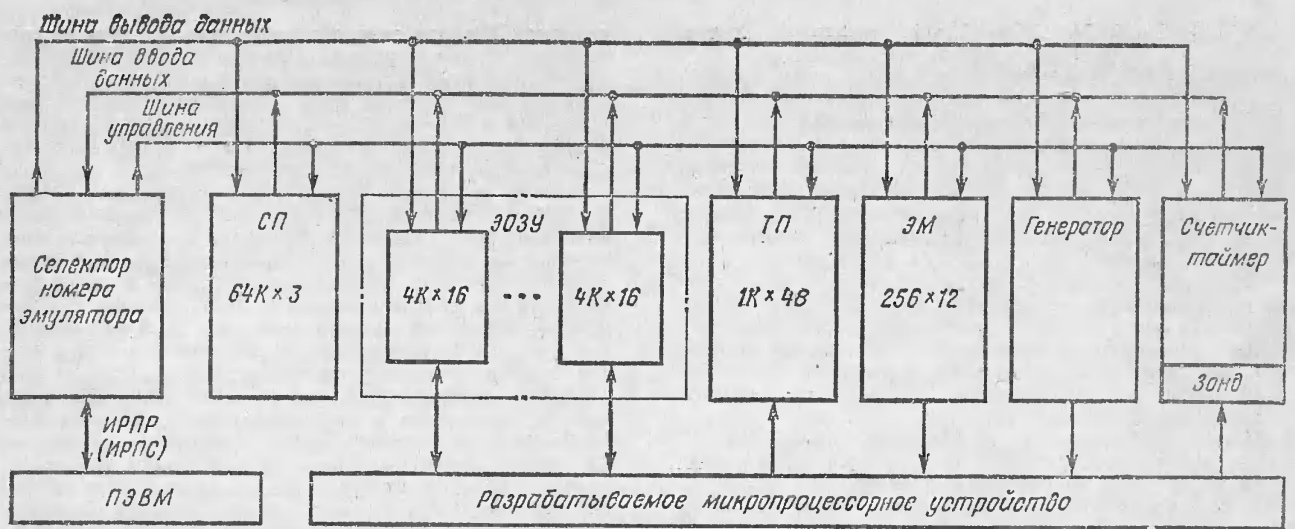


Рис. 1. Структурная схема внутрисхемного эмулятора

автоматического создания перемещающих (обрабатывающих символические метки) и строчных (не обрабатывающих символические метки) кроссассемблеров, а также дизассемблеров для разработанных пользователем символических языков микропрограммирования. Особенности допустимых в САПР ММПУ МЕТАМИКРО 3.0 символических языков описаны ниже. Исходными данными для ПАГАД является описание языка, вводимое в диалоговом режиме. В результате работы ПАГАД порождаются кроссассемблеры и дизассемблер, а также файл листинга характеристик символического языка, оформленный в соответствии с требованиями ЕСПД.

Подсистема трансляции и компоновки микропрограмм (ПТКМ) использует в своей работе перемещающий кроссассемблер, сгенерированный ПАГАД. Подсистема дает возможность вести разработку больших микропрограммных комплексов по частям с последующей сборкой. На основании исходного текста микропрограммы ПТКМ формирует каждую машинную микрокоманду, включая контрольные разряды. Результатом работы подсистемы является загрузочный файл скомпонованной микропрограммы и листинги с исходными и машинными текстами, оформленные в соответствии с требованиями ЕСПД.

Подсистема отладки аппаратуры и микропрограмм (ПОАМП) предоставляет пользователю более 70 директив по настройке эмуляторов, оперативному формированию и редактированию микропрограмм, отладке микропрограмм в аппаратуры, служебным действиям. Загрузочные модули микропрограмм, сформированные ПТКМ, а также строчный ассемблер и дизассемблер, сгенерированные ПАГАД, используются на этапе отладки.

Каждый ВСЭ настраивается на характеристики подключенного к нему блока управления вводом в систему следующей информации: длины микрокоманд, содержимого ЭМ, микропрограмм чтения внутренних регистров ММПУ, формата микрокоманды безусловного перехода, указаний о способе формирования пускового адреса микропрограммы, значений частоты и полярности синхроимпульсов генератора.

Оперативное формирование и редактирование микропрограммы проводятся посредством строчного кроссассемблера и дизассемблера, поддерживающих процедуры символической отладки. Имеется возможность устанавливать необходимые системы счисления адресов и микрокоманд, табулировать символические и машинные микрокоманды, эффективно работать с запоминающими устройствами отдельного ВСЭ,

В процессе отладки допускается назначать столько же точек останова, сколько ячеек (слов) в ЭОЗУ. Останов возможен при достижении точки останова по заданному числу исполнений определенной микрокоманды, а также по событию в аппаратуре, фиксируемому 4-битовым зондом. При исполнении микропрограммы работает трассировочная память, условия занесения данных в которую устанавливаются специальной директивой. ПОАМП позволяет исполнять микропрограмму в пяти автоматических режимах: 1) пуск с заданного адреса, 2) пуск и останов по заданным адресам, 3) пуск с заданного адреса и останов по событию, 4) пуск по заданному адресу и останов по заданному числу исполнений определенной микрокоманды, 5) пошаговый режим, пачинная с заданного адреса.

ПОАМП дает возможность читать содержимое четырех комплектов внутренних регистров ММПУ, не выведенных на контакты микросхем. Каждый комплект содержит 16 регистров, длина которых по желанию пользователя может равняться 8, 16, 24 или 32 битам.

Директивы отладки ПОАМП позволяют подсчитывать число исполнений любой микрокоманды от пуска микропрограммы до останова, измерять времена исполнений различных ветвей, многократно исполнять любую микрокоманду.

Служебные директивы ПОАМП организуют формирование подсказки, информацию о состоянии системы, осуществляют переводы числа в различные системы счисления. Пользователь имеет возможность составить из директив ПОАМП командный файл и тем самым организовать автоматическую процедуру отладки.

Подсистема подготовки данных для программаторов (ППДП) формирует файлы в формате программатора «Электроника МС 9401» из загрузочных модулей микропрограмм для занесения в микросхемы со следующей организацией:  $32 \times 8$ ,  $256 \times 4$ ,  $512 \times 8$ ,  $1024 \times 8$ ,  $2048 \times 8$ . Выводит файл на дисплей в различных системах счисления, распечатывает его, выдает информацию о состоянии ППДП, формирует подсказки.

Подсистема документирования схем (ПДС) обеспечивает построение в диалоговом режиме изображений на экране алфавитно-цифрового дисплея и создание их машинных моделей для последующей обработки программами специального назначения. Позволяет выпускать на штатном печатающем устройстве ПЭВМ конструкторскую документацию (схемы электрические структурные, функциональные, принципиальные) в соответствии с ГОСТ 2.004—79, 2.708—81, 2.743—72,

2.730—73, 2.728—74, 2.751—73, а также блок-схемы программ в соответствии с требованиями ЕСПД.

Директивы ПДС обеспечивают, прежде всего, построение простейших элементов изображения — графических примитивов типа полилиний, символов-маркеров, текстов. Графические примитивы объединяются в составные элементы, рассматриваемые ПДС как именованные фрагменты рисунка. Примитивы и сегменты подвергаются редактированию — их можно передвинуть, размножить, стереть. Сегменты можно дополнить новыми примитивами, удалить примитивы, расформировать.

ПДС дает возможность формировать библиотеки частей схем, чертежей, эскизов, рисунков. Создание библиотек, их корректировка и пополнение осуществляются пользователем. В библиотеки могут быть включены формы конструкторских, технологических или программных документов, условные графические изображения различных электроэлементов и т. д.

Максимальный размер изображения составляет 140 строк по 160 символов каждая. На дисплее пользователь видит часть изображения размером 23 строки по 80 символов, называемую окном. Окно перемещается по изображению, выделяя различные его части. ПДС позволяет выводить изображение на печать с длиной строки 80 или 154 символа в бесформатном виде, а также в форматах А4, А3, А2.

**Символический язык микропрограммирования.** В системе МЕТАМИКРО под символическим языком микропрограммирования понимается язык ассемблерного типа, в котором одной символической микрокоманде соответствует одна физическая (машинная) микрокоманда. Символическая микрокоманда имеет вид [метка : L] (поле 1) [ (поле 2) [ ... ]; [комментарий]

В квадратных скобках указаны элементы, задание которых не обязательно. Метка является символическим именем, получающим при трансляции текущее значение адреса. Микрокоманда состоит из полей, число которых неограничено. Одному полю символической микрокоманды может соответствовать несколько полей физической микрокоманды. Это позволяет упрощать символический язык, объединяя физические поля по функциональному признаку.

Поле задается длиной и типом. Суммарная длина физических полей, соответствующих одному символическому, не должна превышать 32 бит. Длина микрокоманды не более 256 бит.

Для эффективной организации процессов ассемблирования и дисассемблирования символические поля классифицируются по типам следующим образом:

VAR — поле переменных, значение которого определяется подставленным в него символическим именем;

DIG(0) — числовое поле;

DIG(1) — числовое поле с обратным порядком следования битов;

CONST — поле с постоянным значением;

CONTR(0) — поле контрольного бита, дополняющего контролируемые биты до четного числа;

CONTR(1) — поле контрольного бита, дополняющего контролируемые биты до нечетного числа;

SYM(0) — поле адреса перехода. В символической микрокоманде содержит метку, цифровое значение которой записывается в поле перемещающим кроссассемблером;

SYM(1) — поле адреса перехода. Всегда имеет длину 16 бит, цифровое значение определяется перестановкой старшего и младшего байтов, являющихся цифровым эквивалентом метки;

IMG — поле, значение которого формируется переносом (копированием) данных из других битов микрокоманды.

Допускаются любые комбинации полей VAR, DIG, SYM.

Совокупность отличительных особенностей, определяющих правила записи группы микрокоманд, отражается форматом микрокоманды, который задается пользо-

вателем. Для однозначного описания формата микрокоманды требуется указать длину и число полей микрокоманды, порядок следования полей и их длину, символическое имя каждого поля и его тип, содержание полей типов VAR и CONST, а также информацию для формирования полей остальных типов. Допускается неограниченное число форматов микрокоманд.

**Технология работы с системой.** Укрупненная схема функционирования САПР МПУ МЕТАМИКРО 3.0 приведена на рис. 2. Приступая к работе, пользователь имеет возможность создать символические языки микропрограммирования для отдаживаемого ММПУ и сгенерировать для них кроссассемблеры и дисассемблеры. На следующем этапе выполняется настройка ВСЭ на архитектуру управления, к которому он подключен, путем выдачи соответствующих директив. Результатом настройки является файл на диске. Ввод исходных текстов, трансляция и сборка микропрограмм составляют содержание очередного шага технологической цепочки. Затем следует отладка аппаратуры и микропрограмм. Эти этапы повторяются до тех пор, пока не будет завершена отладка. Следует отметить, что в процессе отладки все микропроцессоры ММПУ взаимодействуют по алгоритмам, заложенным разработчиком исходя из назначения ММПУ. Отладка ММПУ с помощью САПР не влияет на алгоритмы взаимодействия микропроцессоров ММПУ и не требует их изменения.

На заключительных стадиях проектирования ведутся подготовка данных для программатора и выпуска схемной документации.

САПР МЕТАМИКРО 3.0 опробована при разработке изделий на базе МПК БИС серий К589, К588, КР1802, КР1804. Система может работать как с одним ВСЭ, так и с несколькими. В последнем случае ВСЭ могут использоваться в качестве имитаторов внешней среды, ге-

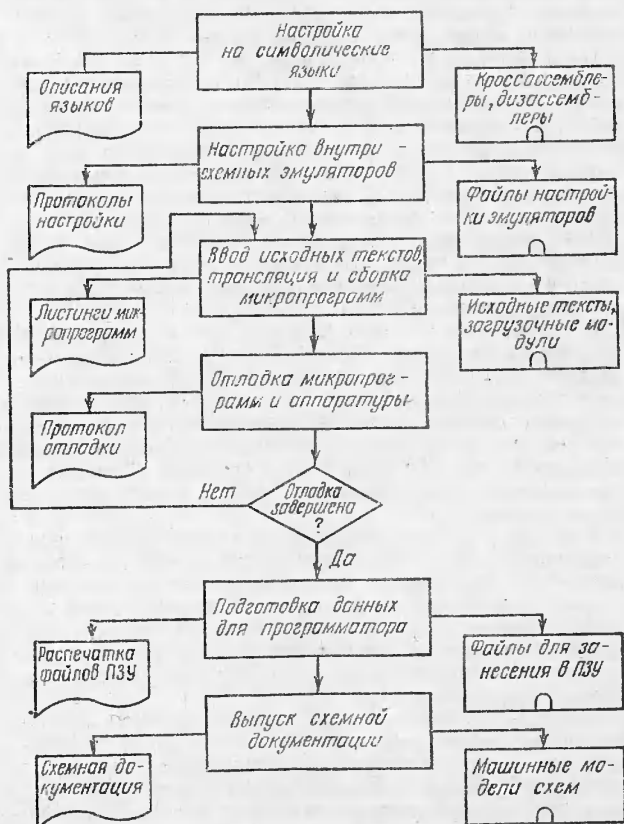


Рис. 2. Алгоритм работы САПР МЕТАМИКРО 3.0

нераторов тестовых последовательностей или трассировочной памяти большого объема. При этом, используя ПАГАД и ПТКМ, можно разрабатывать специальные символические языки и кроссассемблеры для задания внешних воздействий и тестов, а также осуществлять символическое представление данных трассировки с помощью дисассемблеров.

Возможность создания в памяти ПЭВМ машинных моделей электрических схем позволяет развивать САПР ММПУ МЕТАМИКРО 3.0 в направлении создания программ машинного анализа схем, подготовки данных для пакетов проектирования печатных плат, генерации тестов и др. Машинные модели блок-схем микропрограмм могут служить исходными данными для пакетов анализа и верификации микропрограмм.

220605, Минск, ул. Сурганова, 6, СКТЬ с ОП ИТК АН БССР; тел. 39-50-94, 39-50-95

УДК 681.3.06

С. А. Михайлов

### ПАКЕТ САПР КОНСТРУКТОРСКИХ ДОКУМЕНТОВ

В проектировании традиционно различают два крупных этапа — системный и конструкторский. Использование средств автоматизированного проектирования не изменило этой ситуации: на первом этапе определяют все основные характеристики разрабатываемого объекта, второй этап служит для детализации принятых проектных решений, оформления всей необходимой конструкторской документации. На последнем этапе 40...80% от общих затрат времени занимает проектирование, и здесь заключен значительный резерв повышения эффективности конструкторского труда, сокращения длительности цикла разработки конкретной системы. Автоматизация подготовки и выполнения конструкторской документации — изготовление чертежей с помощью средств машинной графики, заготовка и заполнение на ЭВМ спецификаций, таблиц, ведомостей, штампов — позволяет существенно улучшить качество результатов и скорость выполнения проектных решений.

Предлагаемый пакет системы автоматизированного проектирования конструкторской документации (САПР КД) ориентирован на вычислительный комплекс «Искра 1256», исполнение 6, оснащенный символьным дисплеем, матричным печатающим устройством типа «Искра» или «Роботрон 1154» и планшетным графостроителем типа Н-306, Н-307 или ПДП4-002. Программы написаны на символьном языке вычислительного комплекса «Искра 1256». Пакет имеет следующие возможности: выполнение рамок листов стандартных форматов А3, А4, А5; нанесение всех типов основных надписей чертежей; расчерчивание первого и последующих листов бланков спецификаций; стандартное оформление титульного, первого и последующих листов текстовых документов, предварительное редактирование текста. Блок-схема представлена на рис. 1. Подпрограм-

мы, входящие в пакет, описаны в таблице (рис. 2). Полный текст программы содержит свыше тысячи операторов. Необходимый объем оперативной памяти составляет 8 Кбайт.

Функции, выполняемые ЭВМ автоматизировано в процессе диалога с разработчиком: выбор формата чертежа, компоновка элементов чертежа, выбор типа документа — текстовый, чертеж, спецификация, ведомость, формуляр и т. д., определение по-

1. Микропроцессоры: системы программирования и отладки / В. А. Мясников, М. Б. Игнатьев. — М.: Энергоатомиздат, 1985.
2. Семенов О. И., Гриншпан Л. А., Злотник Е. М., Малюш Я. Т., Шерлинг Д. Р. Настраиваемый инструментальный комплекс для разработки систем на специализированных микропроцессорах / Управляющие системы и машины. — 1984. — № 2. — С. 36—39.
3. Гриншпан Л. А., Бокуть В. В., Малюш Я. Т., Шерлинг Д. Р., Гончарик Е. В. Система автоматизированного проектирования микропроцессорных устройств МЕТАМИКРО. — Минск: ИТК АН БССР, 1985.

Статья поступила 3 февраля 1987

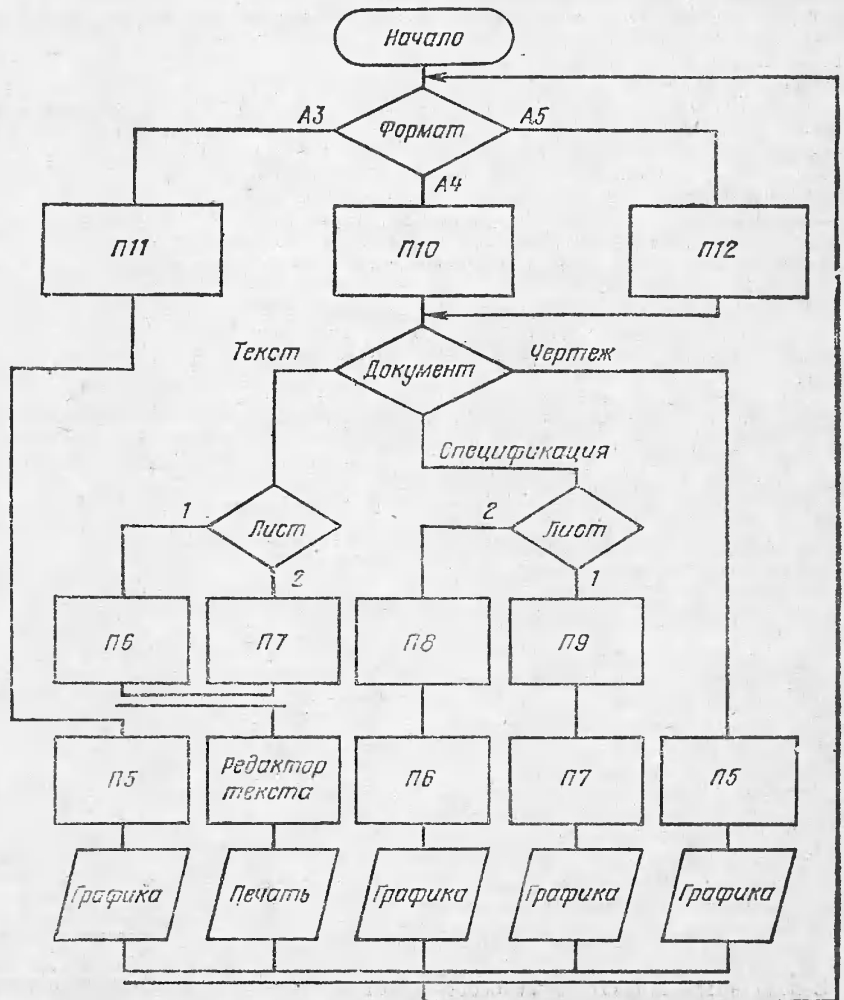


Рис. 1.

СПИСОК ПОДПРОГРАММ ПАКЕТА САПР КД

НОМЕР ПОДПРОГРАММЫ	ИМЯ	ФУНКЦИЯ
П 1	LINE	ЛИНЕЙНОЕ ПЕРЕМЕЩЕНИЕ ПЕРА
П 2	WATE	ОЖИДАНИЕ ВЫПОЛНЕНИЯ КОМАНДЫ
П 3	ON	ОПУСТИТЬ ПЕРО
П 4	OFF	ПОДНЯТЬ ПЕРО
П 5	STAMP1	ОСНОВНАЯ НАДПИСЬ 55 X 185 ММ
П 6	STAMP2	ОСНОВНАЯ НАДПИСЬ 40 X 185 ММ
П 7	STAMP3	ОСНОВНАЯ НАДПИСЬ 15 X 185 ММ
П 8	SPEC1	ПЕРВЫЙ ЛИСТ СПЕЦИФИКАЦИИ
П 9	SPEC2	ПОСЛЕДУЮЩИЕ ЛИСТЫ СПЕЦИФИКАЦИИ
П 10	A4	РАМКА ФОРМАТА А4
П 11	A3	РАМКА ФОРМАТА А3
П 12	A5	РАМКА ФОРМАТА А5
П 13	NOB	ОТСУТСТВИЕ ТРЕБУЕМОГО ДОКУМЕНТА
П 14	EDITOR	ТЕКСТОВЫЙ РЕДАКТОР
П 15	CYCLE	ЦИКЛИЧНОЕ ПРОДОЛЖЕНИЕ РАБОТЫ
П 16	ZERO	НАЧАЛЬНАЯ УСТАНОВКА

Рис. 2.

рядкового номера листа документа, редактирование и расположение на листе текста. Время изготовления конструкторской документации напрямую связано с качеством его выполнения и определяется, в основном, работой периферийных устройств — печатающего и графопроектировщика. Так, наиболее сложная основная надпись чертежа производится за 1,5...2 мин, а рамка листа формата А5 — за 5...10 с.

Опыт использования пакета САПР КД показывает, что любое число конструкторских документов ЭВМ выполняется с одинаково высоким качеством, точно и аккуратно, а разработчик освобождается от работы рутинного характера. Пакет может найти применение как подсистема в САПР печатных плат, кристаллов микропроцессоров, компоновки, размещения и трассировки полузаказных БИС и в других подобных задачах, когда требуется получение результатов автоматизированного проектирования в виде конструкторских документов.

Адрес: 270029, Одесса,

ул. Дидрихсона, 8,

Одесское высшее инженерное морское училище им. Ленинского комсомола, тел. 32-47-14

Сообщение поступило 12 ноября 1987

УДК 681.327

Б. Л. Горшков

## ОСОБЕННОСТИ ПОСТРОЕНИЯ ИЗМЕРИТЕЛЬНЫХ ТРАКТОВ ДЛЯ СИСТЕМ АВТОМАТИЗАЦИИ НА ОСНОВЕ ЭВМ СЕМЕЙСТВА «ЭЛЕКТРОНИКА 60»

Создание прецизионных измерительных трактов, предназначенных для автоматизации процессов, использующих силовое оборудование, генерирующее мощные помехи, и выполненных в стандарте микроЭВМ семейства «Электроника 60», потребовало решения проблем, связанных с характеристиками измеряемых сигналов и конструктивными особенностями ЭВМ.

Пути решения проблем:

прецизионное выделение и измерение малых сигналов датчиков на фоне мощных помех промышленной частоты 50 Гц;

эффективная защита входов устройств от случайно возникающих повышенных напряжений (например, в случае аварий на оборудовании); от помех, генерируемых системными устройствами ЭВМ, и пульсаций в цепи питания, порождаемых источником БПС-6 (до 200 мВ при напряжении 12 В);

обеспечение комплекса высоких метрологических характеристик (точность, линейность, стабильность) при минимальных затратах;

создание прецизионных аналоговых устройств, работающих при пониженном номинале питающего напряже-

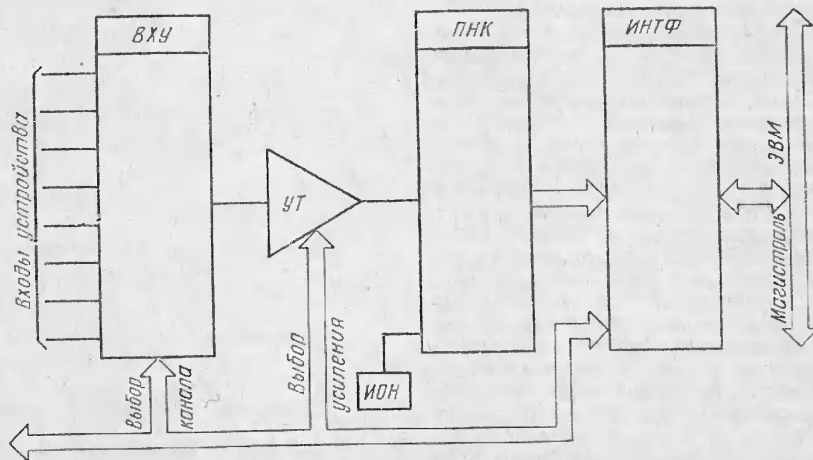


Рис. 1. Структура универсального измерителя:

ВХУ — входное устройство; УТ — усилительный тракт; ПНК — преобразователь напряжения — код; ИОН — источник опорного напряжения; ИНТФ — интерфейс системной магистрали

## ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ

ния (12 В и ниже) (ограничение на выбор элементной базы);

высокая технологичность устройств при массовом производстве (в первую очередь, за счет минимизации процесса настройки).

Анализ традиционной структурной схемы устройства аналогового ввода [1, 2] позволяет указать узлы (рис. 1), которые в наибольшей мере оказывают влияние на качество и прецизионность всего устройства в целом.

Усилительный тракт определяет ка-

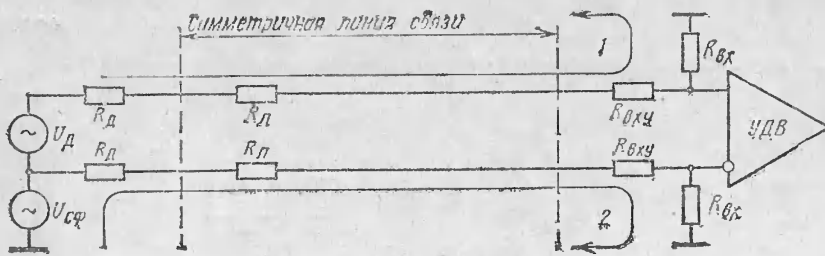


Рис. 2. Эквивалентная схема входной цепи усилителя с дифференциальным входом:

УДВ — усилитель с дифференциальным входом;  $U_d$  — эквивалентный генератор сигнала датчика;  $U_{сф}$  — эквивалентный генератор сигнала синфазной помехи;  $R_d$  — выходное сопротивление датчика;  $R_л$  — эквивалентное сопротивление линии связи;  $R_{вх}$  — последовательное сопротивление во входной цепи усилителя;  $R_{вх}$  — входное сопротивление усилителя

чество выделения сигналов датчиков на фоне помех промышленной частоты, а также погрешность, связанную с шумами, нелинейностью, температурной и долговременной нестабильностью отдельных элементов тракта. Погрешность измерения, вызываемая ИОН, обусловлена температурной и долговременной нестабильностью опорного напряжения и проявляется в виде изменения масштаба преобразования аналогового сигнала.

Высокие метрологические характеристики (разрядность, линейность, стабильность) обеспечиваются применением современных больших интегральных схем БИС АЦП.

Симметрирование контура датчика линии связи-входное устройство позволяет преобразовать сигнал помехи в синфазный и подавить усилителем с дифференциальным входом (рис. 2).

Дифференциальный измерительный усилитель [3, 4], в дальнейшем — измерительный усилитель (рис. 3), состоит из парафазного каскада (A1, A2), усиливающего дифференциальный сигнал датчика ( $U_d$ ) в  $K_n$  раз ( $K_n = (R1 + R1' + R2)/R2$ ), передающего синфазный сигнал помехи без усиления (повышая отношение сигнал-помеха на своем выходе); и вычитающего (A3), преобразующего усиленный дифференциальный сигнал датчика в однофазный и подавляющего синфазный сигнал помехи. Эффективность подавления помехи при  $K_n = 1$  определяется точностью подбора резисторов  $R3 = R3'$  и  $R4 = R4'$ . При увеличении  $K_n$  повышается вклад парафазного усилителя в подавление синфазной составляющей сигнала. Парафазный усилитель определяет качество усилительного тракта в целом, и для его реализации необходимо выбирать прецизионные операционные усилители (ОУ), удовлетворяющие следующим требованиям:

$R_{вх} = \max$  (исключается необходимость симметрирования внешней це-

ни со стороны генератора синфазной помехи ( $U_{сф}$ );

значение коэффициента ослабления синфазного сигнала максимально;

значения долговременного и температурного дрейфов напряжения и тока смещения минимальные.

Использование современных ОУ (K140УД17) позволяет создавать прецизионные измерительные усилители с потенциальной схмотехникой, которые по дрейфам (около 1 мкВ/°С) не уступают, а по скорости превосходят (0,1 В/мкс) традиционные в этом применении сложные и громоздкие МДМ каскады.

Для ограничения уровня шумов в структуре измерительного тракта предусмотрен фильтр низкой частоты (ФНЧ). Настройка частоты среза фильтра зависит от конкретных условий. Например, если шумы на выходе измерительного усилителя определяются только шумами входных элементов (ОУ, входные цепи), то (ФНЧ) настраивают так, чтобы полный размах шумовой дорожки не превышал значения единицы младшего разряда устройства ввода (1МЗР).

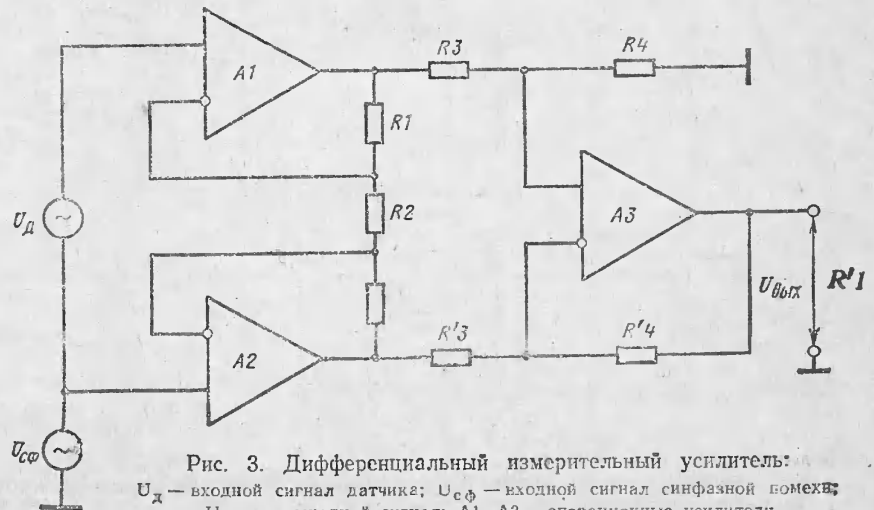


Рис. 3. Дифференциальный измерительный усилитель:  $U_d$  — входной сигнал датчика;  $U_{сф}$  — входной сигнал синфазной помехи;  $U_{вх}$  — выходной сигнал; A1...A3 — операционные усилители

Если на входе усилительного тракта присутствует помеха, не подавленная в достаточной мере измерительным усилителем (высокочастотная помеха), частотный спектр помехи должен попасть в полосу ослабления. Во многих случаях достаточно использования простого однозвенного R-C фильтра.

Одним из основных источников шумов для усилительного тракта является напряжение ( $\pm 12$  и  $\pm 5$  В), вырабатываемое импульсным блоком ЭВМ. Полный размах высокочастотной помехи составляет 200 мВ при 12 В и 400 мВ при 5 В. Помехи этого вида подавляются включением в структуру устройств фильтров питания.

Наиболее удобны активные транзисторные фильтры (рис. 4). Для уменьшения габаритов необходимо использовать супер-бета транзисторы, позволяющие увеличить значение фильтрующих базовых резисторов

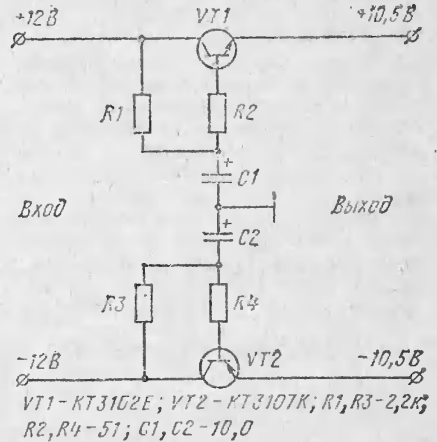


Рис. 4. Активные транзисторные фильтры для питания аналоговых узлов измерителя

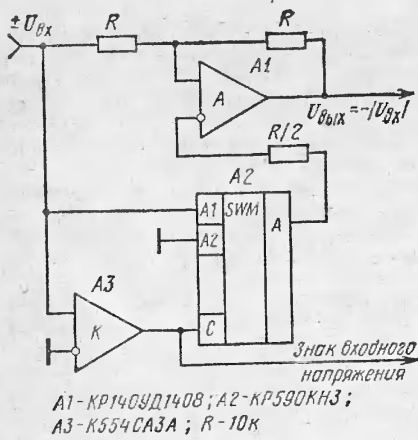


Рис. 5. Устройство аналоговой свертки:

SWM — аналоговый КМОП коммутатор; К — компаратор

(R1, R3) с соответствующим уменьшением номинала, а следовательно, и габаритов конденсаторов (C1, C2). За устойчивую работу фильтров на емкостную нагрузку отвечают резисторы R2 и R4. Построенный таким образом фильтр обеспечивает малоснижение питающего напряжения (с 12 до 10,5 В на выходе) и подавление высокочастотной помехи до 20 мВ.

Экспериментально установлено, что один из основных источников шумов усилительного тракта — напряжение системного блока питания (+5 В),

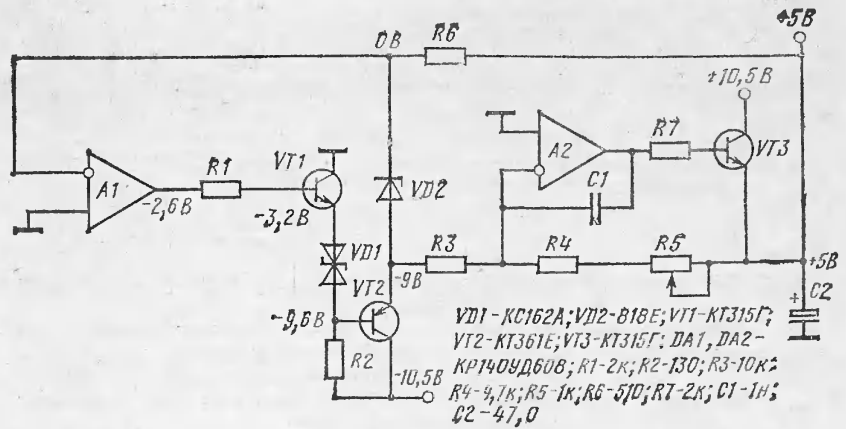


Рис. 6. Источник опорного напряжения

используемого в цепях управления коммутирующими элементами. Для устранения этой составляющей шумов необходимо питать логические цепи коммутаторов от отдельного встроенного аналогового источника (+5 В).

При использовании для питания пониженного напряжения (10,5 В) динамический диапазон входных-выходных аналоговых сигналов следует ограничить величиной ±5 В; применить БИС АЦП последовательного приближения К572ПВ1А в 10-разрядном включении, соответствующем паспортному значению по уровню интегральной нелинейности, равной 0,1% [5]; создать биполярный преобразователь напряжения кода и вы-

сокостабильный источник опорного напряжения.

Традиционным приемом построения биполярного преобразователя напряжение-код на базе униполярного АЦП является смещение биполярных сигналов в униполярную область [2], например, сигналы динамического диапазона ±5 В смещаются в область динамического диапазона 0... +10 В, что в условиях пониженного уровня питания (10,5 В) неприемлемо, так как линейный динамический диапазон операционных усилителей составляет 8 В. Эффективным решением проблемы является включение в состав преобразователя напряжение-код устройства аналоговой свертки, преобразующего би-

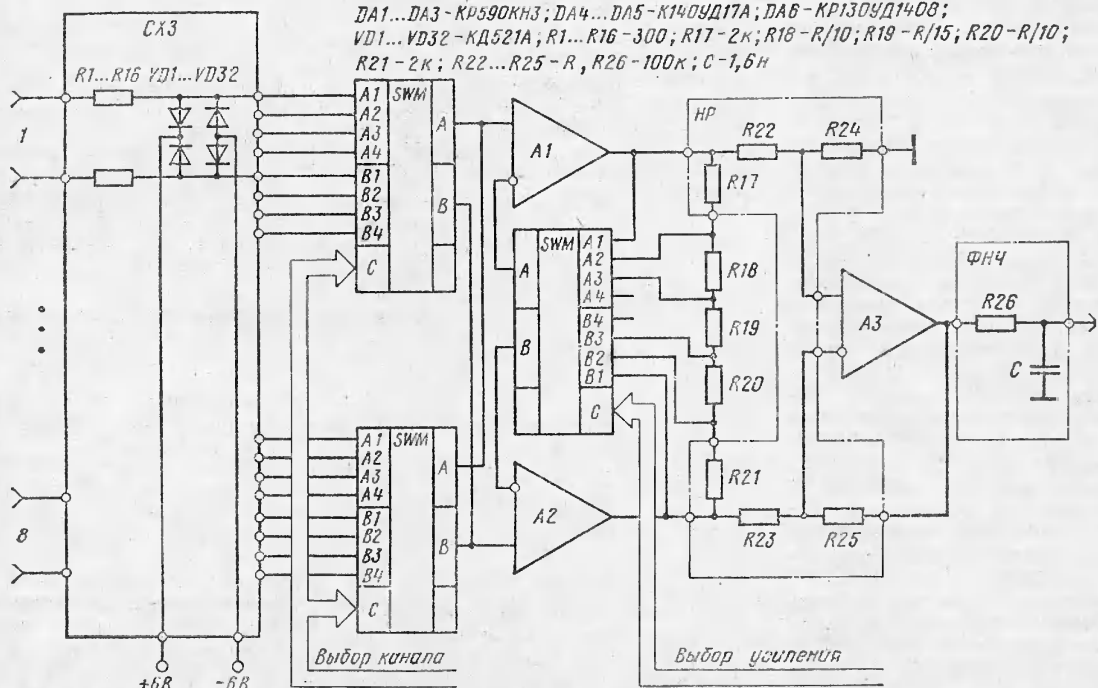


Рис. 7. Схема входной части устройства аналогового ввода

лярный входной сигнал в отрицательный униполярный модуль сигнала с указанием знака входного сигнала по логическому уровню (рис. 5). Автоматическое устройство аналоговой свертки составлено из компаратора, определяющего знак входного сигнала, аналогового ключа, управляемого компаратором, и усилителя с коэффициентом передачи  $\pm 1$ , в зависимости от состояния ключа.

Проблема создания прецизионного источника опорного напряжения в условиях пониженного напряжения питания (10,5 В) возникает из-за того, что годные к применению в источнике прецизионные стабилизаторы (Д818Е, КС162Ф) имеют высокое (9 В) напряжение стабилизации, близкое к питающему напряжению. Выход из создавшейся ситуации — построение 2-каскадного источника с положительной обратной связью по выходному напряжению (рис. 6).

После первого каскада (А1, VT1, VT2, VD2) получается стабильное опорное напряжение, близкое к напряжению источника питания (ток стабилизации подается на прецизионный стабилизатор VD2 со стороны инвертирующего входа А1).

Второй каскад (А1, VT3) осуществляет масштабирование выходного напряжения источника до требуемой величины и подачу (через R6) стабильного тока на прецизионный стабилизатор VD2. Резистор R2 обеспечивает «холодный запуск» источника при включении питания, а в статическом режиме задает ток опорного диода VD1, смещающего рабочую точку выходного каскада А1 в линейную область (−2,6 В). Стабильность такого источника опорного напряжения определяется стабильностью резисторов обратной связи R3, R4, R6. При использовании резисторов типа С2-29В и выходном опорном напряжении, равном 5 В, температурный дрейф не превышает значения 500 мкВ/град. С (0,01 %/°С).

Пример построения входной части устройства аналогового ввода, включающей в себя входное устройство, программируемый измерительный усилитель и ФНЧ, показан на рис. 7.

Параллельные диодные ограничители схемы защиты обеспечивают защиту входов дифференциальных КМОП коммутаторов от случайных выбросов напряжения. Основной критерий выбора ограничивающих диодов VD1...VD32 — небольшой ток утечки (не более тока смещения операционного усилителя). Хорошие результаты дает применение диодов КД521А и КД102А.

Набор коэффициентов передачи измерительного усилителя определяется резисторами R17...R21 и составляет  $K=1,16,64$ . Программный выбор необходимого коэффициента передачи осуществляется дифференциальным коммутатором. Использование потенциальный (бестоковый) способ переключения коэффициента передачи исключает влияние конечного сопротивления открытых каналов КМОП коммутаторов на точность установки.

В качестве резисторов обратной связи каскада (для исключения на-

стройки на механизм измерительного усилителя по подавлению синфазной составляющей входного сигнала) используется прецизионная интегральная сетка резисторов R=2R класса точности 0,01 % в нестандартном включении (на рис. 7 показано пунктиром).

Проблема минимизации влияния наводок от системных модулей и собственной виброу части устройства решается на стадии конструкторской проработки топологии печатной платы устройства благодаря сокращению длины соединительных линий в аналоговой части устройства, разделению контура аналоговой и цифровой общей шины («земли») и пространственному разделению цифровой и аналоговой частей устройства.

Изложенные решения были использованы при реализации набора измерительных трактов для систем автоматизации на базе микроЭВМ семейства «Электроника 60» [6].

#### Характеристики измерительного усилителя

Коэффициент ослабления синфазной составляющей входного сигнала на частоте 50 Гц, дБ	не менее 80
Входное сопротивление, МОм	не менее 3
Температурный дрейф нуля, приведенный ко входу, мкВ/°С	не более 2
Среднеквадратичное шумовое напряжение, приведенное ко входу в полосе 1 кГц, мкВ	не более 4
Интегральная нелинейность, %	не более 0,1

#### Характеристики устройства аналогового ввода

Диапазон входных сигналов на максимальном пределе, В	$\pm 5,115$
Разрядность, двоичных разрядов	1С+13наковый
Интегральная нелинейность, %	не более 0,1
Число шкал измерения	10
Шар квантования, мВ/квант	5К, где К=1,2, ..., 512
Время преобразования, мкс	100

#### ЛИТЕРАТУРА

1. Бернштайн Н. // Электроника. — 1978. — Т. 51. — № 2. — С. 38—48.
2. Александров А. Л. и др. // ПТЭ. — 1985. — № 1. — С. 62—66.
3. Clayton G. B. Operational amplifiers. Sec. Ed. — 1979. — Sidney, Toronto, Butterworth and Co Ltd.

Телефон: 2-22-87, Гатчина Ленинградской обл.

4. Фолкенберри Л. Применение операционных усилителей. — М.: Мир, 1985.
5. Арсениев В. М. и др. // Электронная промышленность. — 1983. — Вып. 4 (121). — С. 48—49.
6. Горников Б. Л. и др. // Микропроцессорные средства и системы. — 1987. — № 5. — С. 47.

Статья поступила 19 марта 1987

УДК 681.327

К. А. Глушенко, В. Б. Дудыкевич, О. Б. Котыло

## РЕАЛИЗАЦИЯ ФУНКЦИИ ИЗМЕРЕНИЯ ЧАСТОТЫ В МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ

При построении микропроцессорных систем и приборов часто возникает необходимость в реализации функции измерения частотно-временных параметров сигналов (период, частота).

Сущность метода измерения частоты заключается в квантовании интервала времени  $T_x$  дискретным интервалом воспроизводящим единицу времени  $T_{опр}$  ( $T_{опр} \ll T_x$ ). Число импульсов опроса подсчитывается и

результатирующее значение частоты измеряемого сигнала определяется следующей зависимостью:

$$f_x = \frac{K_{опр}}{K_n T_{опр}}$$

где  $K_{опр}$  — число импульсов опроса за целое значение периодов входной частоты;  $K_n$  — число периодов (целое) измеряемой частоты;  $T_{опр}$  — период частоты опроса.

Предлагаемая реализация метода измерения частоты требует минимума

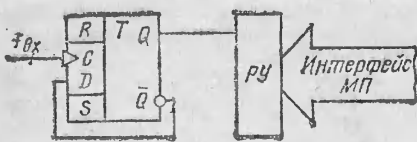


Рис. 1. Схемная реализация функции измерения частоты

аппаратурных затрат и обладает достаточно высокой точностью (рис. 1). Входной сигнал подается на С-вход счетного триггера, формирующего на выходе сигнал типа «меандр». Этот сигнал по определенному алгоритму опрашивается процессором через регистр управления, в качестве которого можно использовать регистр внешних устройств, имеющий один свободный разряд, предназначенный для считывания. При этом подсчитываются значения  $K_{опр}$  и  $K_n$  и вычисляется результирующее значение частоты. Образцовая мера времени — программно-формируемый период, с которым процессор опрашивает состояние триггера, дополнительно ведет подсчет величин  $K_{опр}$  и  $K_n$  за целое число периодов, определяет окончание цикла измерения частоты. Если МП тактируется импульсами от кварцевого генератора, то стабильность этого периода определяется стабильностью генератора. Введение счетного триггера позволяет расширить частотный диапазон в два раза, уменьшить погрешность от неравенства длительности полупериодов и исключить погрешность, возникающую при большой скважности входного сигнала. Верхний предел измеряемой частоты (рис. 2) ограничивается частотой опроса  $f_{опр}$ , нижний — зависит от количества импульсов и может определяться программно.

Программа преобразования входной частоты в цифровое значение реализована для МП типа К1801 (К1806). Считывание входной частоты происходит через регистр управления (адрес=4000<sub>8</sub>), который является внешним устройством для МП. В начале программы осуществляется обнуление текущего значения числа полупериодов в регистре R3 и загрузка счетчика количества опросов ( $СЧ_{опр}$ ) (рис. 3). Для сокращения времени опроса входной частоты содержимое  $СЧ_{опр}$  находится в двух регистрах: прямом (R5) и инверсном (R0) кодах. Изменение содержимого  $СЧ_{опр}$  (для R5 — уменьшение до нуля, для R0 — увеличение до нуля) происходит в последовательности, необходимой для управления логикой работы программы.

Процесс измерения частоты начинается с поиска среза при максимально высокой частоте опроса. При этой же частоте осуществляется поиск фронта, после нахождения которого начинается процесс подсчета числа полупериодов. Время опроса измери-

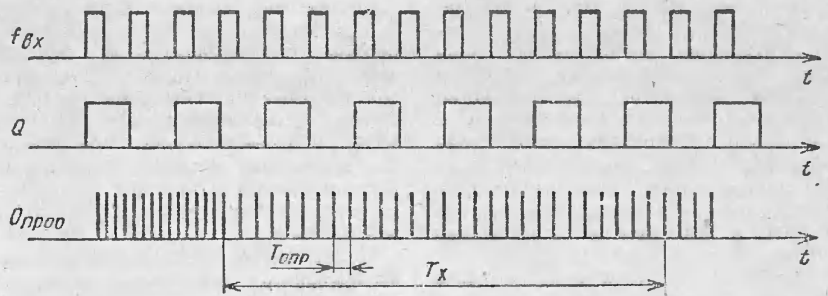


Рис. 2. Временные диаграммы работы МП при измерении частоты

смой частоты увеличивается на время выполнения вспомогательных операций (запоминание числа импульсов опроса при фиксированном значении полупериодов, увеличение на единицу импульсов  $СЧ_{опр}$  и его про-

верка на окончание счета). Времена выполнения команд при считывании нулевого и единичного потенциалов должны быть равны для обеспечения равномерности заполнения импульсами опроса. Подсчет числа полу-

```

; *****
; АСЕМБЛЕР MACRO-11/РАФОС
; ПОДПРОГРАММА ВЫЧИСЛЕНИЯ ЧАСТОТЫ
; *****
SACP1: CLR R1 ;ЗАГРУЗКА НАЧАЛЬНЫХ УСЛОВИЙ
CLR R3
MOV #D2000,R5
MOV R5,R0
S0H: NEG R0
S0H: TST @#40000 СРЕЗ ?--НА S1H
BPL S1H
BR S0H
S1H: TST @#40000 / ФРОНТ?--НА S11H
BPL S1H
S11H: TST @#40000
BPL S3H ; СРЕЗ ?--НА S3H
MOV R0,R0 ;ВСПОМОГАТЕЛЬНЫЕ ОПЕРАЦИИ
INC R1 ;К.ОПР.=К.ОПР+1
DEC R5 ;СЧ.ОПР=СЧ.ОПР-1
INC R0
BMI S11H ;СЧ.ОПР. < Т.ИЗМ --НА S11H
BR S6H
S3H: MOV R1,R4 ;ЗАПОМИНАНИЕ К.ОПР.
INC R3 ;К.ПП=К.ПП+1
INC R0
DEC R5 ;СЧ.ОПР=СЧ.ОПР-1
BMI S6H ;СЧ.ОПР. >= Т.ИЗМ. --НА S6H
S21H: TST @#40000
BMI S4H ; ФРОНТ? --НА S4H
MOV R0,R0 ;ВСПОМОГАТЕЛЬНЫЕ ОПЕРАЦИИ
INC R1 ;К.ОПР.=К.ОПР+1
INC R0 ;СЧ.ОПР=СЧ.ОПР-1
DEC R5
BPL S21H ;СЧ.ОПР. < Т.ИЗМ. -- НА S21H
BR S6H
S4H: MOV R1,R4 ;ЗАПОМИНАНИЕ К.ОПР
INC R3 ;К.ПП=К.ПП+1
DEC R5 ;СЧ.ОПР.=СЧ.ОПР.-1
INC R0
BPL S6H ;СЧ.ОПР.>= Т.ИЗМ. --НА S6H
SA1H: TST @#40000
BPL S3H ; СРЕЗ ? --НА S3H
SA13H: MOV R0,R0 ;ВСПОМОГАТЕЛЬНЫЕ ОПЕРАЦИИ
INC R1 ;К.ОПР.=К.ОПР+1
DEC R5 ;СЧ.ОПР.=СЧ.ОПР-1
INC R0
BMI S11H ;СЧ.ОПР.< Т.ИЗМ. --НА S11H
BR S6H
S6H: ADD R3,R4 ;КОРРЕКЦИЯ КОЛИЧЕСТВА ИМПУЛЬСОВ
; ОПРОСА
; ВЫЧИСЛЕНИЕ ЧАСТОТЫ Fx
; ВЫХОД ИЗ ПОДПРОГРАММЫ ВЫЧИСЛЕНИЯ ЧАСТОТЫ

```

Рис. 3. Подпрограмма вычисления частоты



периодов необходимо выполнять при каждом переходе через фронт входного сигнала (импульсов опроса — непрерывно). Для выравнивания времени в один из циклов измерения вводится холостая команда пересылки, эквивалентная по времени выполнения команде запоминания числа импульсов опроса в R4. Если при изменении на единицу величина  $СЧ_{опр}$  равна заданному значению (в данной программе 2000 выборок), осуществляется выход из блоков счета и переход к вычислению результата. В противном случае продолжается опрос входной частоты и, при изменении фронта, заново начинается число импульсов опроса, увеличивается на единицу число полупериодов. Затем происходит изменение содержимого  $СЧ_{опр}$  и его проверка на окончание счета.

Вычисление результата начинается коррекцией числа импульсов опроса. В цикле, где велся подсчет полупериодов, отсутствовала команда подсчета импульсов опроса, поэтому искомое значение:  $К_{опр} = K_n + K_{опр}$ .

В разработанной программе число команд перехода сведено к минимуму, время выполнения условных команд перехода сбалансировано. Аппаратная реализация предложенного метода позволила измерять частоты до 29 кГц при тактовой частоте МП К1806 2 МГц (время опроса составило 34,97 мкс). Погрешность измерения определяется числом импульсов опроса и для  $К_{опр} = 2000$  относительная погрешность составляет  $\delta = 1/2000 \times 100\% = 0,05\%$ .

Адрес для справок: 290646, Львов, ул. Мира, д. 12, ЛПИ, кафедра «Автоматика и телемеханика»; тел. 39-81-97.

Статья поступила 14 мая 1987

\* Орнатский П. П. Автоматические измерения и приборы. — К: Вища школа, 1986.

УДК 681.772.7.049.771.14

А. В. Хромов, В. В. Покровский

## ВОСЬМИКАНАЛЬНЫЙ МИКРОПРОЦЕССОРНЫЙ УЛЬТРАЗВУКОВОЙ ДАЛЬНОМЕР

Рассматривается система, содержащая ультразвуковые дальномерные модули (УДМ) и блок управления и обработки информации на основе БИС КР580ИК80А. Связь блока управления с УДМ осуществляется через программируемый параллельный интерфейс (ППИ) КР580ВВ55.

В УДМ в качестве источника и приемника используется серийный пьезорезонансный микрофон МУП-1, работающий на частоте 40 кГц. Передающая часть УДМ содержит управляемый генератор и согласующий каскад на транзисторе. Включение генератора осуществляется подачей сигнала низкого уровня в порт С ППИ.

Приемная часть УДМ объединяет три каскада операционных усилите-

лей, детектор и компаратор. Выход компаратора соединен с первым входом логического элемента 2ИЛИ—НЕ, второй вход и выход которого связаны соответственно с выводами портов В и А ППИ (рис. 1). Таким образом, программное управление УДМ и прием информации осуществляется по трем линиям. К одному ППИ может быть одновременно подсоединено восемь УДМ.

Система совместного управления восьмиканальным ультразвуковым дальномером работает следующим образом. По программе выполняется запись кода 00 в порт С ППИ (рис. 2). Через время  $t$  выполняется запись кода FF в тот же порт. При этом происходит подача и снятие сигналов «запуск» низкого уровня,

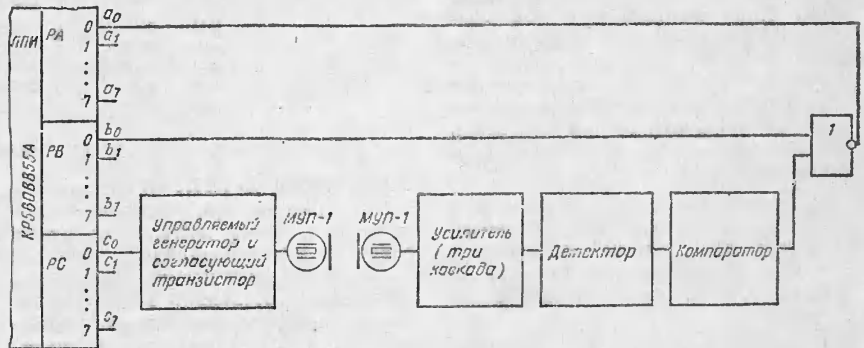
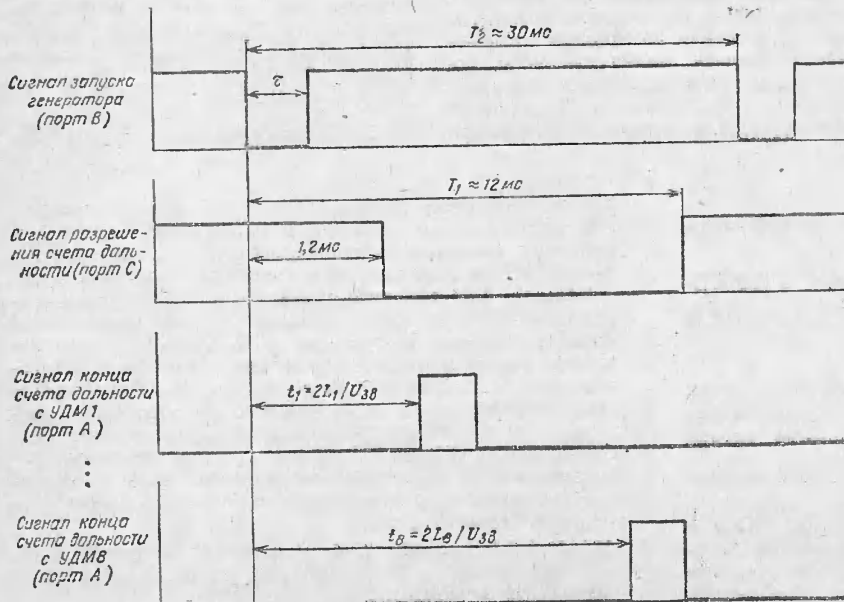


Рис. 1. Структурная схема подключения ультразвукового дальномера к микропроцессорной системе



поступающих на входы управляемых генераторов, что обеспечивает формирование пачек импульсов резонансной частоты передатчика длительностью  $t$ , излучаемых в нужных направлениях. В зависимости от диапазона измеряемых расстояний  $t$  принимает значения от 0,3 до 1,0 мс.

Через 1,2 мс (время затухания помех) в порт В ППИ записывается код 00, что обеспечивает подачу сигналов «разрешение», поступающих в виде сигналов низкого уровня на вторые входы элементов 2ИЛИ—НЕ. Таким образом открывается канал приема отраженных сигналов с УДМ. С этого момента программа осуществляет циклический опрос состояния

Рис. 2. Временные диаграммы работы микропроцессорного блока управления ультразвуковыми дальномерными модулями:  $L_1$ ,  $L_8$  — расстояние от УДМ1, УДМ8 до отражающей поверхности;  $U_{сз}$  — скорость звука,

порта А ППИ и счет числа выполненных обращений. В случае прихода отраженного сигнала от одного или нескольких УДМ накопившееся число опросов, пропорциональное расстоянию до отражающей поверхности, вместе с адресами соответствующих УДМ заносится в ОЗУ. В дальнейшем информация, поступающая от УДМ, данные с которых были записаны в ОЗУ, игнорируется.

Программа измерения расстояний оформлена в виде подпрограммы (рис. 3) и может входить в состав программного обеспечения системы управления. По истечении времени  $T_1=12$  мс, соответствующего удвоенному максимально возможному расстоянию до отражающих объектов, происходит выход из подпрограммы измерения дальности. В результате работы программы в ОЗУ формируется массив из адресов УДМ и соответствующих дальностей, расположенных в порядке поступления отраженных сигналов. Для получения результата к информации о дальности в ОЗУ надо прибавить константу, соответствующую времени затухания помех. Максимальное число ячеек ОЗУ, необходимых для хранения измерительной информации, равно 16 байт (адреса в ОЗУ с 21 00 по 21 0). Эта информация обрабатывается, и через время  $T_2=30$  мс, достаточное для затухания кратных отражений, весь цикл измерения может быть повторен.

Благодаря введению согласующих временных задержек обеспечивается равенство длительностей выполнения команд при записи информации в ОЗУ и при проверке конца счета циклов.

Максимальная дальность действия системы измерения расстояний определяется типом применяемых ультразвуковых приемопередатчиков и в данном случае составляет 2 м. Минимальная дальность действия соответствует времени затухания помех и равно 0,2 м. Минимальная дискретность измерения расстояния определяется временем выполнения цикла

ADRES	EQU	210FH	; КОНЕЧНЫЙ АДРЕС ОЗУ.
DAI-N:	MVI	A, 82H	; ЗАПИСЬ В РУС.
	OUT	RUS	; СЛОВА-СОСТОЯНИЯ.
	XRA	A	;
	LXI	H, ADRES	;
A1:	MOV	M, A	;
	DCR	L	; НАЧАЛЬНЫЕ
	JP	A1	; УСТАНОВКИ.
	MOV	C, L	;
	MOV	B, A	;
	MOV	D, A	;
	OUT	PORT C	; ВКЛЮЧИТЬ ГЕНЕРАТОРЫ УДМ.
	MVI	A, 28H	; ЗАДЕРЖКА НА ВРЕМЯ
A2:	DCR	A	; ФОРМИРОВАНИЯ ПАЧКИ
	JP	A2	; ИМПУЛЬСОВ.
	OUT	PORT C	; ВЫКЛЮЧИТЬ ГЕНЕРАТОРЫ УДМ.
	MVI	A, 80H	; ЗАДЕРЖКА НА ВРЕМЯ
A3:	DCR	A	; ЗАТУХАНИЯ
	JNZ	A3	; ПОМЕХ.
	OUT	PORT B	; ПОСЛАТЬ СИГНАЛ РАЗРЕШЕНИЯ.
A4:	MVI	A, C5	; ПРОВЕРКА КОНЦА
	CMF	D	; ВЫПОЛНЕНИЯ
	RZ		; ПОДПРОГРАММЫ.
	MVI	A, 07H	; СОГЛАСУЮЩАЯ
A5:	DCR	A	; ВРЕМЕННАЯ
	JNZ	A5	; ЗАДЕРЖКА.
	IN	PORT A	; ПРОВЕРИТЬ НАЛИЧИЕ ОТРАЖЕНИЙ.
	INR	D	; УВЕЛИЧИТЬ КОЛИЧЕСТВО ЦИКЛОВ ОПРОСА.
	ANA	C	; ИСКЛЮЧИТЬ ПОВТОРНУЮ ИНФОРМАЦИЮ.
	CMF	B	; ПРОВЕРИТЬ НАЛИЧИЕ
	JZ	A4	; ИНФОРМАЦИИ.
	INX	H	; ЗАПИСАТЬ АДРЕСА
	MOV	M, A	; УДМ В ОЗУ.
	INX	H	; ЗАПИСАТЬ ТЕКУЩЕЕ КОЛИЧЕСТВО
	MOV	M, D	; ЦИКЛОВ ОПРОСА В ОЗУ.
	CMA		; УЧЕСТЬ НОМЕРА УДМ,
	ANA	C	; ИНФОРМАЦИЯ ОТ КОТОРЫХ
	MOV	C, A	; ЗАПИСАНА В ОЗУ.
	MVI	A, 04H	; СОГЛАСУЮЩАЯ
	JMP	A5	; ВРЕМЕННАЯ ЗАДЕРЖКА.
	END		

Рис. 3. Текст программы измерения дальности

опроса порта А ППИ. В рассматриваемой системе при тактовой частоте 2 МГц время выполнения цикла равно 60 мкс и подобрано таким образом, что дискретность измерения расстояния составляет 1 см. Уменьшение дискретности нецелесообразно, так как она становится значительно меньше ошибки датчиковой части.

Систему предполагается использовать в робототехническом комплексе для исключения возможных столкновений звеньев манипулятора с подвижным технологическим оборудованием, элементами строительных конструкций и человеком. Благодаря

гибкости система обеспечивает адаптивное управление и реализацию режимов работы, зависящих от конфигурации манипулятора и скорости движения его звеньев, что способствует увеличению рабочей зоны. Кроме того, имеется возможность введения статистической обработки и фильтрации, в частности вобуляции частоты импульсов запуска.

Адрес для справок: 194064, Ленинград, пр. Тихорецкий, 21, ЦНИИ робототехники и технической кибернетики ЛПИ им. М. И. Калинина; тел. 552-41-81

Статья поступила 12 августа 1987

УДК 658.5.011.53:331.015.11

А. Н. Пипченко, С. П. Гончарук, В. В. Пономаренко, А. Е. Савельев

## РАСПРЕДЕЛЕННАЯ СИСТЕМА УПРАВЛЕНИЯ СУДОВОЙ ЭЛЕКТРОСТАНЦИЕЙ НА БАЗЕ КТС ЛИУС-2

На базе микропроцессорного комплекса технических средств (КТС) ЛИУС-2 разработана двухуровневая распределенная система управления режимами работы многоагрегатной электростанции.

Задачи управления распределены «по вертикали» между локальным координатором (ЛК) на верхнем уровне и локальными подсистемами (ЛП1... ЛПп) — на нижнем. Для повышения функциональной надежности каждый генераторный агрегат (ГА) имеет свою локальную подсистему контроля, защиты, диагностики

ГА и управления пуском, остановкой, синхронизацией. ЛК решает более сложные и глобальные задачи управления, связанные с адаптацией судовой электростанции (СЭС) в нормальных и аварийных режимах и др.

Каждая ЛП объектно-ориентированной системы управления СЭС содержит в своем составе, кроме шестнадцати типовых модулей из КТС ЛИУС-2, дополнительно разработанные модули преобразования синусоидальных сигналов в цифровые. ЛК содержит десять типовых модулей и дополнительно разработанный модуль задания очередности включения-отключения ГА. Связь между ЛК и ЛП, ЛК и другими судовыми локальными МП системами реализуется через межблочную интерфейсную магистраль с помощью контроллера связи линейного типа.

270101, Одесса, ул. 25-й Чапаевской дивизии, д. 1, кв. 172. Телефон: 44-38-25 (после 19.00), Пипченко Александр Николаевич,

УДК 681.322 — 681.3.06

Р. А. Бронштейн

## ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА РАСШИРЕНИЯ ВОЗМОЖНОСТЕЙ МИКРОЭВМ «ЭЛЕКТРОНИКА 60» и ДВК

Сервисные программы, заложенные в ПЗУ и работающие в адресах страницы ввода-вывода (160000 и выше), улучшают функциональные и эксплуатационные характеристики вычислительных комплексов на базе микроЭВМ. Возможности, предоставляемые такими программами, воспринимаются пользователями как аппаратные свойства машины, так как доступны сразу после включения ЭВМ и не зависят от содержимого оперативной памяти. Характерным примером могут служить мониторы пультажного режима, выполняющие минимальное подмножество команд программ отладки (открыть ячейку, открыть регистр, продолжить выполнение программы и т. д.), а в ЭВМ ряда ДВК — небольшой набор тестов.

В статье описано специализированное устройство системного ПЗУ (СПЗУ), содержащее набор сервисных программ. Применение его в стационарных вычислительных комплексах повышает производительность труда; особенно эффективно использование в микроЭВМ, встраиваемых в системы управления.

### Состав и назначение программ СПЗУ

Программы СПЗУ позволяют пользователям загружать операционную систему с различных носителей, тестировать вычислительный комплекс, отлаживать прикладные программы.

Выбор конкретной функции и запуск ее осуществляются интерактивно, с клавиатуры консоли, в формате: SS[N][AAAAA]<BK>, где: SS — двухсимвольный код директивы (все символы могут быть введены на любом регистре); N, AAAAA — факультативный аргумент, специфицирующий номер устройства в группе однотипных, и стандартный адрес регистра состояния устройства соответственно.

Примеры синтаксически правильных директив: MX<BK>; MY2<BK>; DX,177174<BK>; QD6,176500<BK>.

Ввод неопознанной комбинации символов вызывает печать краткой инструкции.

Директивы загрузки операционной

системы обеспечивают обращение к следующим носителям:

стандартному пакетному диску RK емкостью 2,5 Мбайт (ИЗОТ 1370, СМ 5400);

стандартному 8-дюймовому (203 мм) флоппи-диску DX;

133-мм мини-флоппи-диску с «дорожечной» структурой записи MX, входящему в комплектацию ДВК-2, ДВК-3;

133-мм мини-флоппи-диску с «секторной» структурой записи MY, входящему в комплектацию ДВК-4;

электронному квазидиску QD [1, 2].

В процессе загрузки контролируется правильность работы внешнего устройства; обнаружение ошибок вызывает повторение операции. Загрузку, затянувшуюся из-за устойчивых ошибок устройства, можно интерактивно прервать с клавиатуры консоли.

Попытка обращения к несуществующему устройству вызывает печать диагностики TRAP 4 и перезапуск программы СПЗУ.

Тестирование вычислительного комплекса, выполняемое при запуске программ СПЗУ и после завершения директив, не вызывающих передачи управления за пределы СПЗУ, охватывает проверку команд процессора, способов адресации и процессорных прерываний. По окончании теста определяется объем наличной оперативной памяти и производится контрольное суммирование СПЗУ; результаты выдаются на консоль.

Помимо контрольных функций, выполняемых автоматически, программы СПЗУ содержат процедуры тестирования, запускаемые по директиве пользователя. К ним относятся тест оперативной памяти и тесты внешних запоминающих устройств.

Тесты всех поддерживаемых магнитных запоминающих устройств осуществляются проверкой полного тракта «носитель - дисковод-контроллер-процессор»; при тестировании содержимое носителя не разрушается (производится только чтение); по ходу теста на консоли индицируется успешность обращения к каждому сектору в указанном пользователем диапазоне дорожек носителя.

Тесты оперативной памяти и электронного диска разрушают содержимое запоминающих устройств, поэтому запуск их осуществляется после дополнительного подтверждения пользователем. Длительность одного прохода теста оперативной памяти на микроЭВМ «Электроника 60» — около 45 с (при объеме ОЗУ 28К слов), текста электронного диска — около 170 с (при объеме диска 256 Кбайт).

Длительность тестов на ДВК-3 примерно в 4 раза меньше.

Отладка программ пользователей — самая важная функция СПЗУ, занимающая примерно половину его объема. Наличие в ЭВМ мощной программы, постоянно готовой к применению и не использующей оперативного адресного пространства, обеспечивает пользователям эффективный рабочий инструмент. Достоинства заложенного в ПЗУ отладчика особенно отчетливо проявляются при разработке автономных прикладных программ для неполных конфигураций вычислительного комплекса (т. е. для встроенных ЭВМ) и при разработке комплексов операционной системы (драйверов и т. п.).

Командный язык СПЗУ-отладчика близок к языкам отладочных программ перфокарной операционной системы и ОС РАФОС; список команд включен в текст краткой инструкции, выводимой на консоль.

Отладчик может запускать программы пользователя на исполнение в пошаговом и непрерывном режимах (в последнем случае можно задать до восьми контрольных точек, в которых отладчик перехватывает управление и предоставляет пользователю возможность анализировать и модифицировать состояние программы); имеет встроенный дизассемблер, резко повышающий производительность труда даже квалифицированных программистов.

Он позволяет исследовать и задавать содержимое слов и байтов в различных форматах: символьном, RADIX50, десятичном и восьмеричном (в прямом коде со знаком или дополнительно).

Вспомогательные команды отладчика:

групповая печать регистров процессора и содержимого контрольных точек;

вычисление относительного смещения между двумя адресами;

представление и модификация указанной области памяти как текста в коде КОИ-7;

поиск адресных ссылок на заданный диапазон адресов (с выводом на консоль признака способа адресации);

вычисление суммы слов или байтов;

поиск слов или байтов, совпадающих с образцом по заданной маске (или отличающихся от него);

сравнение содержимого двух областей памяти равной длины;

заполнение памяти в указанных границах заданным 8- или 16-разрядным кодами;

перенос содержимого указанной области памяти в заданное место;

разрешение обработки аварийных прерываний либо отладчику, либо программе пользователя.

### Структура и электрическая схема устройства

В адресном пространстве машины СПЗУ занимает 4400 байт (1 1/8 К слов). Этот отрезок содержит три зо-

ны адресов: ПЗУ объемом 4000 байт (здесь и далее все числа, относящиеся к адресам СПЗУ, выражены в восьмеричной системе), регистр коммутации (два байта) и системное ОЗУ (376 байт).

Системное ОЗУ небольшого размера служит для размещения стека и служебной информации программ СПЗУ. Доступ к этой зоне адресов устройства открыт только при работе программ СПЗУ. Начальный адрес системного ОЗУ смещен на 4002 относительно базового адреса ПЗУ.

Зона ПЗУ — адресное пространство, в котором работают программы описываемого устройства. Пусковой адрес программы смещен на 3000 от начала области. При работе на микроЭВМ «Электроника 60» и ДВК-3 естественно назначить для СПЗУ область адресов 170000...173777 с пусковым адресом 173000. При работе на ДВК-2 базовым адресом СПЗУ приходится назначать 160000, так как область с адреса

173000 в этой машине занята.

Интервал адресов ПЗУ разбит на две неравные части с относительными адресами 0000...2777 и 3000...3777. Вторая из них всегда присутствует на общей шине ЭВМ (всегда доступна для обращения). Первая часть становится доступной для обращения только после начала работы программ СПЗУ. Доступ прекращается при передаче управления за пределы СПЗУ. Операционная система РАФОС в процессе своего запуска адаптируется к конкретной конфигурации ЭВМ, в частности проверяет доступность по чтению абсолютного адреса 172000 (регистр графического дисплея), и для предотвращения неправильной самонастройки стандартной ОС РАФОС программы СПЗУ отключают доступ к младшей части своих адресов.

Полный объем программ, размещенных в системном ПЗУ, равен 4К слов (20000 байт). Для того чтобы сохранить в разумных пределах дли-

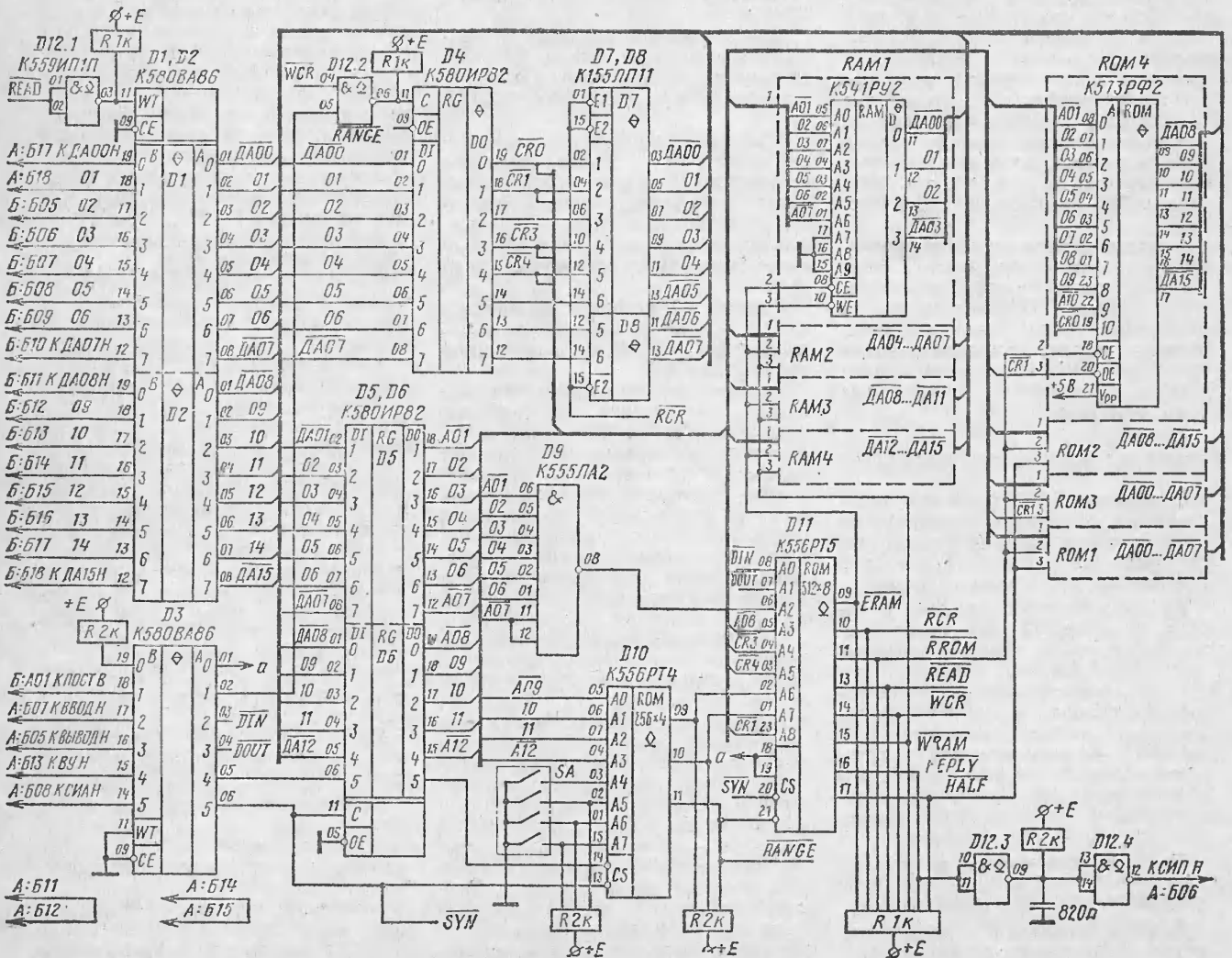


Рис. 1. Электрическая схема устройства

ну области, занятой устройством в адресном пространстве ЭВМ, программа разбита на четыре секции, поочередно отображаемые в зону адресов (в окно) ПЗУ. Номер секции, «видимой» в данный момент на общей шине в окне ПЗУ, определяется состоянием регистра коммутации.

Регистр коммутации, имеющий адрес 4000 и постоянно доступный на общей шине ЭВМ, позволяет программно управлять работой устройства. Наиболее важная функция регистра — переключение секций ПЗУ.

Назначение используемых битов регистра:

00,01 — номер отображаемой секции ПЗУ;

03 — разрешение доступа к системному ОЗУ;

04 — разрешение доступа к младшей части (0000...2777) области адресов ПЗУ;

05 — управление обработкой прерываний по вектору 10, возникающих в программе пользователя: («0» — обработка СПЗУ-отладчиком, «1» — программой пользователя);

06 — управление обработкой прерываний по вектору 04 (аналогично биту 05);

07 — флаг выполненного сохранения динамического состояния отлаживаемой программы.

Пользователям разрешено изменять состояние битов 5, 6.

Электрическая схема системного ПЗУ приведена на рис. 1. Шинные формирователи, выполненные на микросхемах D1, D2, рассчитаны на ток канала до 32 мА, то есть на работу в машинах ряда ДВК; для применения устройства в микроЭВМ «Электроника 60» следует заменить каждую из микросхем D1, D2 на два корпуса К589АП16 и исключить инвертор D12.1.

Регистр коммутации реализован элементом D4. Начальный адрес области ПЗУ задается в диапазоне 160000...173000 с точностью до 1000 переключателями SA; их со-

A7-A4	МЛАДШИЕ БИТЫ АДРЕСА A3-A0															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	7	7	7	7	7	7	7	7	7	2	0	1	1	1	1	1
10	7	7	7	7	7	7	7	7	2	0	1	1	1	1	7	7
20	7	7	7	7	7	7	2	0	1	1	1	7	7	7	7	7
30	7	7	7	7	7	2	0	1	1	7	7	7	7	7	7	7
40	7	7	7	7	7	2	0	1	1	7	7	7	7	7	7	7
50	7	7	7	7	2	0	1	1	1	7	7	7	7	7	7	7
60	7	7	7	2	0	1	1	7	7	7	7	7	7	7	7	7
70	7	7	7	2	0	1	1	7	7	7	7	7	7	7	7	7
80	7	7	2	0	1	1	7	7	7	7	7	7	7	7	7	7
90	7	2	0	1	1	7	7	7	7	7	7	7	7	7	7	7
A0	7	2	0	1	1	7	7	7	7	7	7	7	7	7	7	7
В0	2	0	1	1	7	7	7	7	7	7	7	7	7	7	7	7
С0	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
D0	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
E0	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
F0	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

Рис. 2. Таблица прошивки дешифратора базового адреса D10

A8-A4	МЛАДШИЕ БИТЫ АДРЕСА A3-A0															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF
010	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF
020	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF
030	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF
040	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF
050	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF	FF	BF	B3	FF
060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	9E	B5	FF
090	FF	FF	FF	FF	FF	FF	FF	FF	FF	AF	B5	FF	FF	9E	B6	FF
0A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	AF	B5	FF	FF	9E	B6	FF
0B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	AF	B5	FF	FF	9E	B6	FF
0C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
100	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F
110	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F
120	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F
130	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F
140	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F
150	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F	7F	3F	33	7F
160	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F
170	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F
180	7F	7F	7F	7F	7F	7F	7F	7F	7F	2F	35	7F	7F	1E	36	7F
190	7F	7F	7F	7F	7F	7F	7F	7F	7F	2F	35	7F	7F	1E	36	7F
1A0	7F	7F	7F	7F	7F	7F	7F	7F	7F	2F	35	7F	7F	1E	36	7F
1B0	7F	7F	7F	7F	7F	7F	7F	7F	7F	2F	35	7F	7F	1E	36	7F
1C0	7F	7F	7F	7F	7F	7F	7F	7F	7F	2F	35	7F	7F	1E	36	7F
1D0	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F
1E0	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F
1F0	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F

Рис. 3. Таблица прошивки дешифратора адресов и операций D11

стояния анализируются предварительным дешифратором D10, выполненным на микросхеме К556РТ4 (рис. 2).

Окончательная дешифрация адресов осуществляется элементом D11 (К556РТ5), который управляет доступом к структурам устройства с учетом текущего состояния регистра коммутации (рис. 3).

Собственно ПЗУ представлено четырьмя микросхемами К573РФ2 (или К573РФ5). Поскольку внутренние магистрали устройства инверсны, коды программ перед записью в ПЗУ должны быть переформатированы: расположены в обратном порядке адресов и инвертированы.

Системное ОЗУ выполнено на микросхемах К541РУ2 с организацией 1К×4 бит. Для применения устройства на микроЭВМ «Электроника 60» серьезными преимуществами обладает другой вариант — выполнение системного ОЗУ на микросхемах динамической памяти; емкость их достаточно велика для того, чтобы на этих же микросхемах реализовать и основную память машины. При использовании БИС К565РУ5Д1 (К565РУ5Д2) на полуплате размерами 135×252 мм легко размещаются еще контроллер регенерации на основе БИС К1801ВП1-30 и селектор банков (общий объем устройства 38 микросхем). Преимущество такого технического решения:

две стандартные платы памяти ПЗ заменены одной платой гораздо более широкого назначения;

возникает возможность отключения регенерации, управляемой процессором (на 10...12% возрастает скорость исполнения программ, а предельная частота опроса процессоров, доступная системам сбора и обработки данных, повышается с 3...5 до 20...50 кГц);

незначительное усложнение селектора банков позволяет регулировать размер ОЗУ не только квантами по 4К слов, но и дополнительно квантами по 1К слов в седьмом банке (в странице ввода-вывода); в предельном случае удается расширить ОЗУ до 30К слов, сохраняя работоспособность стандартной операционной системы.

#### Особенности программирования СПЗУ

Функции, выполняемые программой СПЗУ, не являются неизменными. В зависимости от конфигурации и назначения конкретной вычислительной установки может оказаться необходимым дополнить программы загрузчиком с электронного диска другой архитектуры, тестом специфического внешнего устройства, функцией телезагрузки и т. п.

Требуемые изменения могут быть выполнены в любой операционной системе на языке Макроассемблер.

При разработке или модификации программ СПЗУ необходимо учитывать специфику устройства: поочередное отображение четырех секций программы в одно адресное окно. В условиях столь необычной аппарату-

ры необходимы специальные меры для выполнения самых распространенных действий. обращения к переменным в области адресов системного ОЗУ и вызова подпрограмм из области адресов ПЗУ.

Рассмотрим приемы, разработанные для программирования на этой аппаратуре.

Программный комплекс СПЗУ состоит из шести файлов исходного текста на языке Макроассемблер: четырех файлов отдельных секций (SECT0, ... SECT3), префиксного файла PREFIX и постфиксного файла PSTFIX. Результирующий двоничный текст программы СПЗУ образуется при компоновке четырех объектных файлов секций, каждый из которых создается при трансляции секций по схеме:

```
MACRO PREFIX+SECT*/OBJ+
+PSTFIX
```

Обращение к переменным в области адресов системного ОЗУ должно выполняться с относительным способом адресации, так как аппаратное исполнение устройства позволяет изменять его базовый адрес. Коды перекрывающихся секций расположены в поле адресов микросхем ПЗУ последовательно; последовательно они размещены и в результирующем файле, содержащем эти коды для записи в ПЗУ; транслятор МАКРО также интерпретирует коды секций как расположенные в адресном пространстве ЭВМ последовательно. Следовательно, имеются различия между статическими (времени трансляции) и динамическими (времени исполнения) смещениями от команд, расположенных в ПЗУ, до операндов, размещенных в системном ОЗУ.

Для преодоления этого расхождения необходимо искусственным приемом так сформировать имена операндов, чтобы компенсировать указанные различия.

С этой целью в префиксный файл введены определения:

```
BODYLEN = 3000 ; Длина запираемой части области адресов ПЗУ
COMMLEN = 1000 ; Длина перманентной .. ..
FULLLEN = BODYLEN + COMMLEN ; 1К слов
RAMLEN = 400 ; Длина системной области ("MAGIC" + RAM)
;
.MACRO GATHER FIRST SECOND THIRD
FIRST'SECOND'THIRD
.ENDM GATHER
;
; Порядок следования программных секций
Q.NUMB = 0
.REPT 4
GATHER < .PSECT BODY \Q.NUMB <, OVR >
" = + BODYLEN
GATHER < .PSECT COMM \Q.NUMB <, OVR >
GATHER < START \Q.NUMB <: >
" = + COMMLEN
Q.NUMB = Q.NUMB + 1
.ENDR
.PSECT RAM, OVR ; Регистр коммутации, ОЗУ
" = + RAMLEN
```

Каждый из файлов секций построен по определенной схеме. Пример нулевой секции:

```
.TITLE 173SC0
Q.NUMB = 0
.PSECT BODY0
" = 0
.....
.PSECT COMM0
" = 0
.....
.....
```

Префиксный файл завершается операторами:

```
AGV = (3-Q.NUMB)*FULLLEN ; Коррекция смещения адресов до "RAM"
AQ0 = (Q.NUMB-0)*FULLLEN ; \
AQ1 = (Q.NUMB-1)*FULLLEN ; \ Коррекция адресов
AQ2 = (Q.NUMB-2)*FULLLEN ; / "других" секций
AQ3 = (Q.NUMB-3)*FULLLEN ; /
.PSECT RAM
" = 0
MAGIC = " - AGV ; Регистр коммутации
.BLKW 1
STACK = MAGIC + RAMLEN
POWER = " - AGV ; Система счисления для ввода аргументов
.BLKW 1
INPSTRN = " - AGV ; Буфер ввода директивы
.BLKW 7
.....
.....
```

При таком определении имен объектов в области адресов системного ОЗУ смещения к ним формируются правильно — соответственно ситуации времени исполнения, а не трансляции.

Обращения к подпрограммам не вызывают никаких проблем, если вызываемый адрес и точка вызова расположены в одной секции ПЗУ. В случае произвольного размещения вызываемой подпрограммы прямое обращение к ней из других секций

невозможно — необходимо обратиться к промежуточной процедуре, передав ей в качестве параметров адрес подпрограммы и номер содержащей ее секции. Промежуточная процедура должна запомнить адрес и номер секции точки вызова, осуществить требуемое переключение секций, затем обратиться к вызываемой подпрограмме, а после завершения ее работы снова переключить секции и вернуть управление в точку вызова. Промежуточная процедура должна правильно работать в момент смены номеров секций, когда в адресном окне ПЗУ «скачком» меняется содержимое читаемых слов.

Подпрограммы, запускаемые через

седьмой регистр (счетчик команд), могут быть вызваны из любой секции обращением к промежуточной процедуре OVERLAY с передачей ей в качестве параметров имени подпрограммы и номера секции, в которой она расположена.

Технически макротексты программного комплекса СПЗУ не содержат операторов прямого вызова подпрограмм CALL и явных обращений к процедуре OVERLAY. Во всех точках, где по смыслу требуется обращение к подпрограмме с именем NAME, стоят макровыводы ROUTIN с именем NAME в качестве одного из параметров. Макровыводы ROUTIN всегда расширяется в два машинных слова: оператор CALL (если обращение к подпрограмме происходит внутри секции) или экстракод прерывания в слово параметра.

В программном комплексе СПЗУ широко используется вызов функций через программное прерывание по резервной инструкции, возникающее при интерпретации машинных кодов 7000...7777. Диапазон этих кодов разбит на группы, определяющие конкретную функцию. Одной из таких функций является обращение к промежуточной процедуре OVERLAY, служащей для вызова подпрограмм из «других» секций.

Макроопределение ROUTIN находится в префиксном файле:

```

EXTRA = 7000 ; Базовое значение экстракода
SWITCH = 400 ; Группа функций: вызов подпрограмм
  .MACRO ROUTIN NAME SCTNUM
    ; NAME - имя вызываемой подпрограммы
    ; SCTNUM - номер содержащей ее секции
  .IF EQ SCTNUM-Q.NUMB
    CALL NAME
  .IFF
  .WORD EXTRA + SWITCH + SCTNUM
  GATHER < .WORD NAME + AQ>, \SCTNUM, < - .>
  .ENDC
  .ENDM ROUTIN

```

Процедура переключения секций OVERLAY является частью программы обработки прерываний по резервной команде, размещенной в постфиксном файле:

```

EXTCOD: MOV R0, - (SP) ; Входная точка обработки прерывания по резервной команде
MOV R1, - (SP) ;
MOV R2, - (SP) ;
MOV 6(SP), R1 ; Адрес возврата из прерывания
MOV -2(R1), - (SP) ; Экстракод
BIT #SWITCH, (SP)
BNE OVERLAY
; Обработка прочих функций,
; вызываемых через экстракоды
OVERLAY: MOV SP, R2
ADD #12, R2 ; Адрес хранения PSW в стеке
MOV R1, - (SP)
ADD (R1)+, (SP) ; Адрес вызываемой подпрограммы
MOV R1, (R2) ; Адрес возврата в точку вызова
MOV MAGIC, - (R2) ; Регистр коммутации вызова
BIC #C(3), (R2) ; Номер вызываемой секции
MOV - (R2), R0 ; Восстановлен R0
MOV PC, (R2) ; Адрес возврата в "OVERLAY"
4x: ADD # (8x-4x), (R2) ; из вызываемой подпрограммы
MOV - (R2), R1 ; Восстановлен R1
MOV (SP)+, (R2) ; Адрес вызываемой подпрограммы
MOV - (R2), R2 ; Восстановлен R2
MOV (SP)+, (SP) ; Номер вызываемой секции
8x: BIC #3, MAGIC ; Сформирован регистр коммутации
BIS (SP)+, MAGIC
RTS PC

```

OVERLAY изменяет значения кодов условий (при передаче управления в вызываемую подпрограмму и при возврате из нее) и добавляет в стек два дополнительных слова по сравнению с прямым вызовом через оператор CALL; в остальном она «прозрачна».

Правильное функционирование процедуры OVERLAY в момент переключения секций обеспечивается простым способом: имеются четыре идентичных экземпляра этой процедуры, размещенных в каждой из секций по одинаковому относительному адресу.

К сожалению, не удалось придумать способа автоматически обеспечить совпадение начальных адресов различных экземпляров процедуры; имеется лишь прием, позволяющий выполнять эту задачу за два прохода трансляции.

Каждый из файлов секции завершается операторами, описывающими поле выравнивания длины секции:

PLACE=\*\*\* — некоторое абсолютное значение, например 0;  
REPT PLACE

.BYTE 0 — поле выравнивания длины секции.

ENDR

Постфиксный файл содержит вышеописанную программу обработки

После любого изменения программного текста какой-либо секции необходимо выполнить его пробную трансляцию по формуле

MACRO/NOOBJ PREFIX+  
+SECT\*+PSTFIX,

затем повторно отредактировать файл, изменив размер поля выравнивания длины PLACE в соответствии с напечатанными при трансляции сообщениями, и только после этого выполнить рабочую трансляцию и сборку.

Заключение

Описанное несложное устройство предоставляет пользователям эффективный набор услуг. Дополнительный сервис, обеспечиваемый устройством, воспринимается как аппаратное свойство машины.

Важнейшие сервисные функции: загрузчик электронного диска позволяет полностью реализовать преимущества энергонезависимого электронного носителя операционной системы [2] (готовность к работе через 3...6 с после включения питания ЭВМ и независимость от магнитных накопителей);

СПЗУ-отладчик резко сокращает затраты времени на доработку и отладку автономных и системных программ.

Адрес для справок: 103062, Москва, ул. Обуха, д. 10, НИФХИ им. Л. Я. Карпова, тел.: 227-00-14, доб. 23-89

#### ЛИТЕРАТУРА

1. Бронштейн Р. А., Кашкадзев В. А., Клименко С. В. и др. Одноплатный электронный диск для микроЭВМ «Электроника 60» и ДВК // Микропроцессорные средства и системы.— 1988.— № 2.— С. 84.
2. Бронштейн Р. А., Евтехов А. С. Электронный диск для микроЭВМ «Электроника 60», ДВК с энергонезависимым хранением информации // Микропроцессорные средства и системы.— 1988.— № 2.— С. 82.  
Статья поступила 16 декабря 1987

```

GATHER < FREECOMM = . - START> \Q.NUMB < - COMLEN>
.NLIST LOC, SEQ
.LIST TTM
GATHER <.PRINT ;> \Q.NUMB <-я секция СПЗУ>
WPLACE = PLACE - FREECOMM
.IF LT FREECOMM
.PRINT -FREECOMM ; Свободное пространство в конце секции
.PRINT WPLACE ; Требуемое значение "PLACE"
.ENDC
.IF GT FREECOMM
.ERROR FREECOMM ; Превышение длины секции
.IF GE WPLACE
.PRINT WPLACE ; Требуемое значение "PLACE"
.IFF
.ERROR -WPLACE ; ! ! Абсолютный избыток !
.ENDC
.ENDC

```

С. К. Гершкович, В. В. Крынкин, А. Ю. Востропят

# АВТОМАТИЗИРОВАННЫЙ УЧЕБНО-ЛАБОРАТОРНЫЙ КОМПЛЕКС

Создание автоматизированных учебно-лабораторных комплексов (АУЛК), способствующих качественному преобразованию процесса обучения, тормозится из-за отсутствия доступных средств связи ЭВМ с оборудованием лабораторных стендов.

Наиболее подходящим для учебных целей является ИВК на базе микроЭВМ «Ириша», но и он ориентирован на конкретную ЭВМ, что сильно затрудняет его применение.

Авторы предлагают «облегченный» стандарт на устройства связи с реальными объектами (УСО) для применения в учебных измерительно-вычислительных комплексах, ориентированный на связь с ЭВМ по стандартному ИРПР. Применение разработанного УСО позволяет строить АУЛК на базе практически любых имеющихся ЭВМ без существенных изменений в устройстве.

Электронное свободнопрограммируемое устройство (ЭСПУ) — комплекс технических и программных средств, предназначенный для выполнения лабораторных работ (рис. 1).

Функции ЭСПУ:  
управление оборудованием учебных стендов;  
имитационное моделирование работы типовых узлов и систем;  
информационное обеспечение;  
автоматизация элементов процесса обучения и контроля знаний.

Принцип работы устройства связи (рис. 2) основан на централизованном управлении всеми функциональными блоками от внешней ЭВМ или от специализированного блока процессора (СБП), входящего в его состав (СБП инициирует сигнал «ЗАХВ» и управляет обменом по внутренней магистрали устройства). Структура УС обладает достаточной гибкостью; формат служебного слова позволяет адресовать до 64 конечных функциональных блоков (т. е. возможно расширение устройства в пределах шестизрядного адреса). Функ-

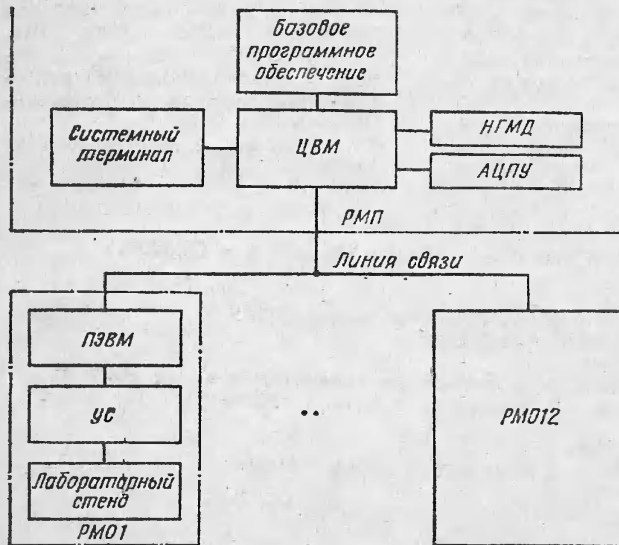


Рис. 1. Электронное свободнопрограммируемое устройство

циональные блоки подразделяются на активные (передатчики) и пассивные (приемники), предназначенные для передачи данных в ЭВМ и приема данных из ЭВМ. В каждом блоке может находиться до восьми субблоков (активных и пассивных). Алгоритмы обмена между блоками УС и ЭВМ стандартизованы (существует единый алгоритм обмена со всеми активными устройствами и единый алгоритм обмена с пассивными блоками).

Унифицированная структура магистрали позволяет строить функциональные блоки различного типа — от простейших релейных коммутаторов до самоуправляемых микропроцессорных модулей (способы подключения могут быть отличными от ИРПР).

Активные блоки могут работать как в режиме принудительного опроса, так и в режиме прерывания, пассивные — только в режиме принудительного опроса. Работа блоков в режиме принудительного опроса:

в начале каждого обмена по сигналу ЭВМ «Сброс ВУ» (начало нового цикла обмена) во внутреннюю магистраль передается сигнал КО (конец обмена);

все дешифраторы адреса в функциональном блоке устанавливаются в исходное состояние;

на внутренней магистрали появляется сигнал АДР, характеризующий поступление на шины данных служебного слова;

ЭВМ передает служебное слово (байт), сопровождаемое стробом;

в функциональных блоках дешифрируется адресная и операционная части служебного слова;

выбранный функциональный блок передает в магистраль сигнал готовности (ГПР);

ЭВМ передает служебное слово — байт информации, в состав которого входят три основных поля: трехрядный адрес функционального блока (А), трехрядный адрес субблока (Б) и код операции (В).

Возможны четыре операции: общий сброс (00) — начальная установка и обнуление регистров, чтение регистров выбранного блока (01), запись (10), обнуление регистра данных (11).

В различных вариантах УС возможно неоднозначное использование адресных полей А и Б. В основном варианте эти поля трехрядные — адресуются восемь блоков и восемь субблоков. Возможны варианты 16,4 и 32,2, что позволяет изменить структуру УС.

Цикл обмена считается законченным после передачи сигнала КО (начало нового цикла).

Работа блоков в режиме прерывания:

по сигналу блока «Запрос прерывания» (ЗПР) ЭВМ разрешает прерывание (РПР);

блок передает на линии данных байт состояния, имеющий структуру, аналогичную служебному слову (байт состояния содержит полный адрес блока, вызвавшего прерывание, и код требуемой операции), и формирует (по фронту РПР) строб СТРА, по которому байт состояния принимается ЭВМ.

Активный цикл работы закончен. Запрос блока ставится в очередь на обслуживание ЭВМ. Опрос (или запись) блока происходит в режиме принудительного обмена.



Рис. 2. Устройство связи со стендом



Особенность внутренней магистрали — структура шин дополнительной логики и шин питающих напряжений.

Состав магистрали:  
 шины данных (8 шип данных-адреса ДА0...ДА7 для передачи адреса блоков и информации для записи в регистры пассивных блоков, а также 8 информационных шин ДИ0...ДИ7 для данных активных блоков);  
 шины управления: конец обмена (КО), сброс регистра (СБРРГ), общий сброс (СБР), выходной сигнал разрешения прерывания (РПРвых), готовность приемника (ГПР), ввод-вывод (ВВ-ВЫВ), захват магистрали (ЗАХВ), строб адреса (АДР), строб приемника (СТРП), входной сигнал разрешения разрешения прерывания (РПРвх), запрос прерывания (ЗПР);

шины дополнительной логики (шесть линий тактовой частоты Ч00...Ч12), по которым во все блоки УС передаются частотные сигналы, вырабатываемые таймером;  
 шины питающих напряжений (+5В, +15В, 20В) и сигналы аналоговой и логической общих шин.

Блок связи УС с ИРПР (контроллер крейта) (рис. 3) предназначен для взаимного преобразования сигналов стандартного ИРПР и внутренней магистрали, дешифрации служебного слова и формирования управляющих сигналов.

По сигналу ИРПР «Сброс ВУ» дешифратор (ДС) устанавливается в состояние «ожидание», на линию ЗП-П подается сигнал Лог.0 через ЛК: блок связи готов к приему служебного слова.

На выходную шину данных ИРПР ЭВМ передает служебное слово, сопровождаемое стробом СТР-П. Дешифратор устанавливается в состояние «Работа», команда анализируется на ДФКС, при необходимости формируются импульсные сигналы «СБР» и «СБРРГ». «Команда действия» (ввод-вывод) запоминается и передается во внутреннюю магистраль. К шине ЗП-П ИРПР через логический коммутатор ЛК подключается ГПР внутренней магистрали и передается код адреса. Блок связи остается в режиме «Работа» до прихода сигнала «Сброс ВУ».

Приемники и передатчики (МПЛ, МПУС, МПрУС, МПрД) управляют сигналом ЗАХВ внутренней магистрали (если обмен осуществляется под управлением внутреннего специпроцессора, то сигнал Лог.0 на линии ЗАХВ отключает от магистрали блок связи).

Через входной регистр (рис. 4) в ЭВМ вводятся данные с активных элементов стенда (концевых выключателей, логических датчиков и т. д.). Каждый входной регистр может работать в двух режимах:

с внешним стробированием информации — запись в БР

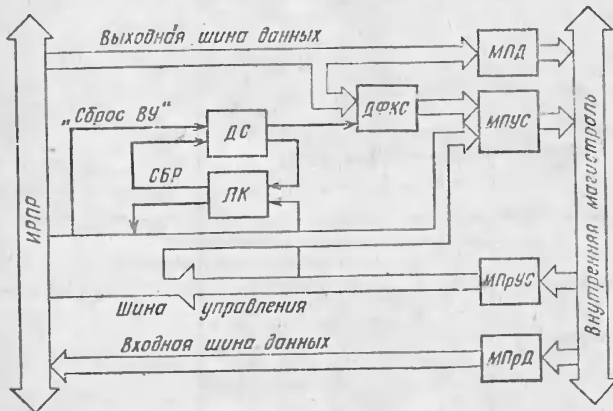


Рис. 3 Блок связи (контроллер крейта):

ДС — дешифратор состояний; ЛК — логический коммутатор; ДФКС — дешифратор-формирователь команд; МПД — магистральные передатчики данных; МПУС — магистральные передатчики управляющих сигналов; МПрУС — магистральные приемники управляющих сигналов; МПрД — магистральные приемники данных

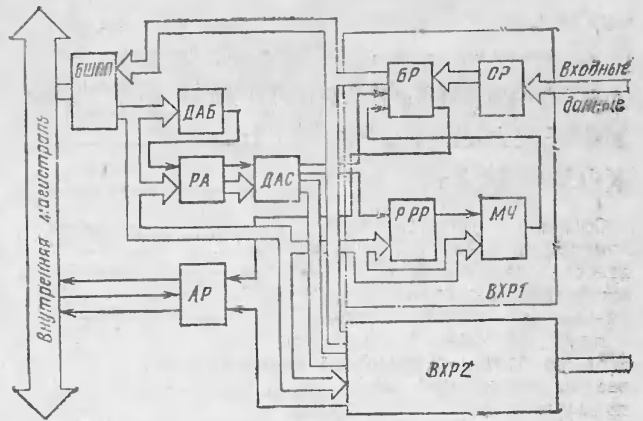


Рис. 4. Входной регистр:

БШПП — блок шинных приемопередатчиков; ДАБ — дешифратор адреса блока; РА — адресный регистр; ДАС — дешифратор адреса субблока; АР — регистр прерываний; ВХР1, ВХР2 — блоки входного регистра; БР — буферный регистр данных; ОР — блок оптронов; РРР — регистр регистрации режима; МЧ — мультиплексор частот

через блок ОР происходит по внешнему стробу, АР выставляет требование на прерывание и передает адрес конкретного регистра (1 или 2); ЭВМ обрабатывает прерывание и передает адрес блока регистра; ДАБ дешифрирует адрес, в АР записывается адрес субблока; адрес субблока дешифрируется в ДАС и информация из БР через БШПП передается во внутреннюю магистраль и, далее, в ЭВМ;

со стробированием от линии фиксированных частот таймера — запись внешней информации происходит по стробу, формируемому по одной из шести фиксированных частот таймера. Выбор частоты происходит на мультиплексоре (МЧ). Каждый блок имеет два типа регистров: пассивный регистр РРР (записывается перед началом работы блока код режима работы (0 или 1) и номер одной из линий частот, по которой будет происходить стробирование во втором режиме) и буферный регистр данных (БР).

Выходной регистр (рис. 5) управляет внешними устройствами дискретного действия, имеет релейно-контактные выходы, позволяющие осуществлять коммутацию достаточно мощных устройств и цепей, обеспечивает гальваническую развязку схем регистра от коммутируемых цепей.

Блок выходного регистра (по характеру алгоритмов обмена) — пассивное устройство. Запись байта данных в ВР1 и ВР2 происходит по адресу выбранного субблока (0 или 1).

В соответствии с единичными разрядами записанного байта замыкаются контакты реле, на внешние цепи подается напряжение, подключенное к соответствующему контакту разъема.

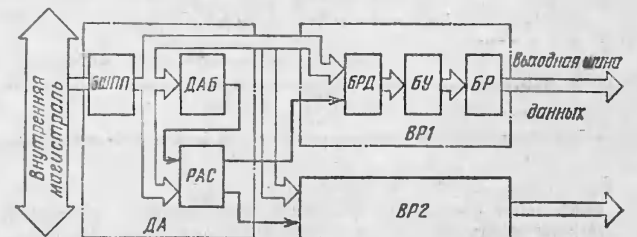


Рис. 5. Выходной регистр:

БШПП — блок шинных приемопередатчиков; ДАБ — дешифратор адреса блока; РАС — регистр адреса субблока; ВР1, ВР2 — блоки выходного регистра; БРД — буферный регистр данных; БУ — блок управления; БР — блок реле

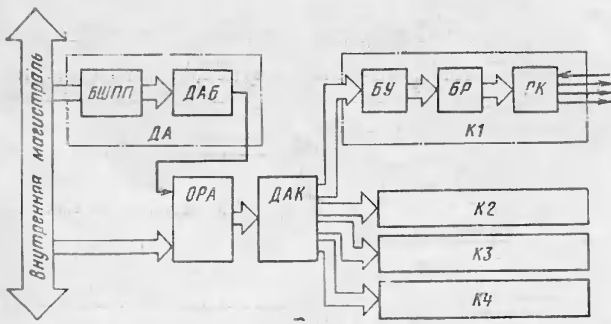


Рис. 6. Выходной мультиплексор:

ОРА — объединенный регистр адреса, ДАК — дешифратор адреса коммутатора; К1...К4 — коммутаторы; БУ — блок усилителя; БР — блок реле; РК — контактный коммутатор

Выходной мультиплексор (рис. 6) коммутирует внешние логические и аналоговые цепи через контакты реле.

Тип мультиплексора 4×4 (четыре блока работают по принципу 4×1 — один из входных контактов может быть соединен с любым из четырех выходных).

Работа схемы согласования с магистралью (рис. 7).

Адресный байт (первый байт обмена на шине ДА) через приемники ПР подается на дешифратор адреса блока ДАБ и вход данных регистра РД. На один из входов данных регистра поступает бит совпадения адреса (РАБ) с дешифратора ДАБ. По сигналу АДР в регистр РД записывается адрес субустройства, поступающий с приемника ПР, и бит РАБ с дешифратора ДАБ. При условии выбора данного блока значение

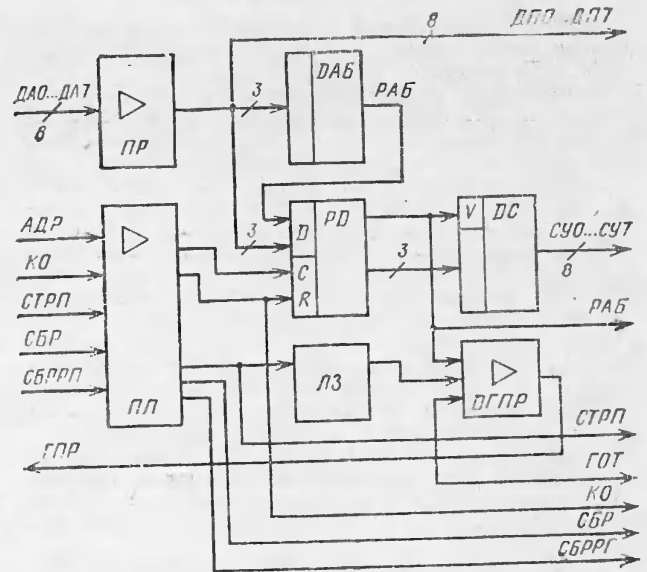
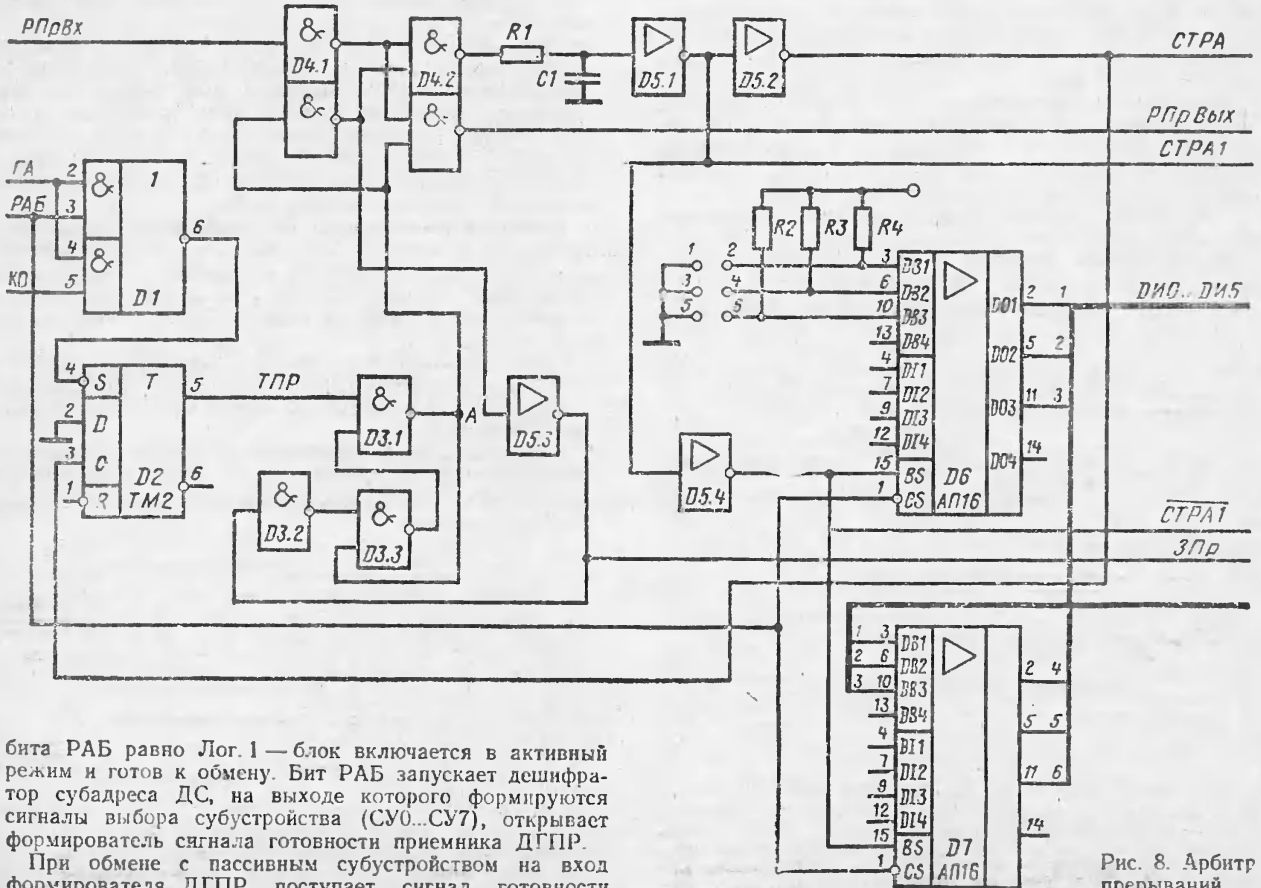


Рис. 7. Схема согласования с магистралью:

ПР — магистральные приемники; ПЛ — приемники логических сигналов; ДАБ — дешифратор адреса блока; РД — регистр данных; ДС — дешифратор субадреса; ДГПР — дешифратор-формирователь сигнала готовности; ЛЗ — линия задержки

приемника (ГОТ), передаваемый на линию ГПР, на шинах ДА0...ДА7 появляется информация, сопровождаемая сигналом СТРА1.



бита РАБ равно Лог.1 — блок включается в активный режим и готов к обмену. Бит РАБ запускает дешифратор субадреса ДС, на выходе которого формируются сигналы выбора субустройства (СУ0...СУ7), открывает формирователь сигнала готовности приемника ДГПР.

При обмене с пассивным субустройством на вход формирователя ДГПР поступает сигнал готовности

Рис. 8. Арбитр прерываний

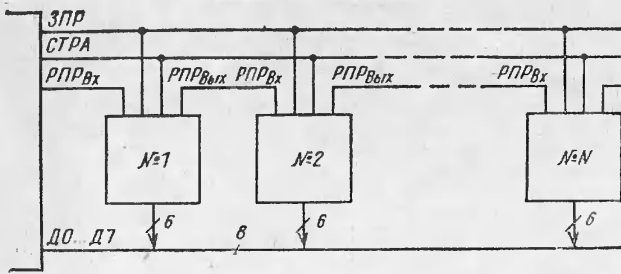


Рис. 9. Схема относительных приоритетов

Запись информации в субустройство с шин ДП0...ДП7 происходит по сигналам СУ0...СУ7.

Сигнал СТРП после задержки в блоке ЛЗ сбрасывает готовность приемника ГПР — информация принята.

По окончании обмена на шине управления появляется сигнал КО, субустройство отключается от магистрали и функциональный блок готов к обращению по следующему адресу.

Сигналы СБР и СБРРГ, поступающие с приемника ПЛ, могут использоваться по необходимости.

Арбитр прерываний (рис. 8) предотвращает конфликты на шине запроса прерываний (ЗПР). Наибольшим приоритетом обладает функциональное устройство, ближе всех расположенное к контроллеру крейта (рис. 9).

В исходном состоянии сигнал РПР ВХ = Лог. 1, ГА = Лог. 0 (требование прерывания отсутствует), ТПР = Лог. 0 (триггер прерываний не взведен); ЗПР = Лог. 0 (магистраль не занята), РПР ВЫХ = Лог. 1. Активное устройство по готовности выдает сигнал ГА; КО взводит триггер прерываний, ТПР = Лог. 1, в точке А Лог. 0, формируется сигнал запроса ЗПР = Лог. 0.

Из сигнала РПР ВХ формируется строб активного устройства СТРА. Сигнал А = 0 запрещает прохождение РПР ВХ (сигнал РПР ВЫХ отсутствует). По сигналу СТРА открываются шинные формирователи Д6, Д7 и происходит передача вектора прерывания функционального устройства. После считывания сигнал РПР ВХ сбрасывается, исчезает СТРА, сбрасывается

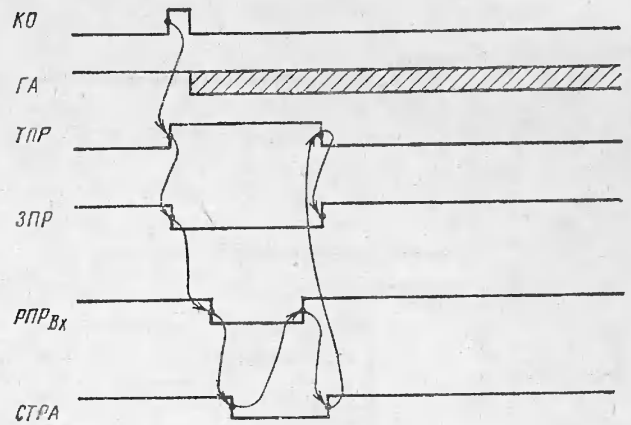


Рис. 10. Диаграмма цикла обслуживания прерываний

триггер прерываний — обработка прерывания от данного устройства закончена (рис. 10).

ЦАП содержит четыре независимых канала: два из них могут автоматически линейно изменять выходной аналоговый сигнал (рис. 11).

Входная часть ЦАП представляет схему согласования с магистралью (ССМ) (рис. 7).

Устройство имеет в своем составе шесть субрегистров, запись в нужный осуществляется блоком ЛК с помощью конъюнкции одного из сигналов СУ0...СУ5 и сигнала СТРП.

В автоматическом режиме в четырехразрядный регистр тактовой частоты РТЧ по субадресу 000 записывается номер выбранной опорной частоты 400...412 и Лог. 1 в разряд запуска этого регистра. Мультиплексор МХ подключает ко входу счетчика СЧ одну из опорных частот. Таким образом на цифровых входах ЦАП код изменяется на единицу каждый период тактовой

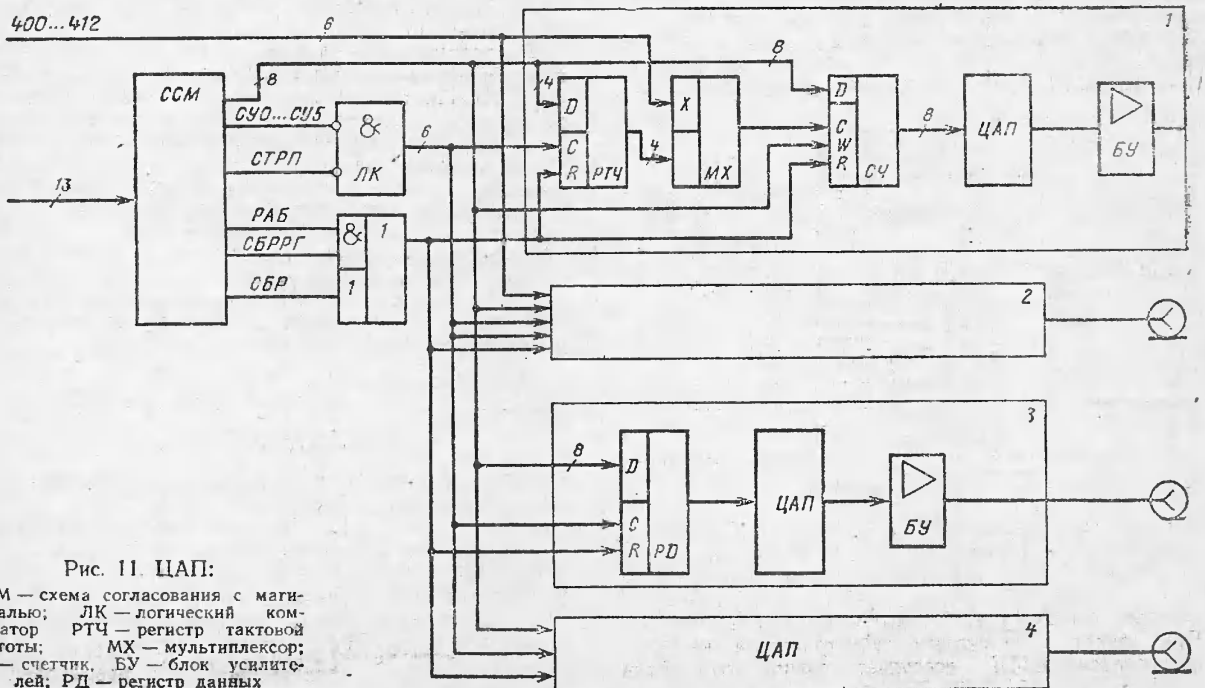


Рис. 11. ЦАП:

ССМ — схема согласования с магистралью; ЛК — логический коммутатор; РТЧ — регистр тактовой частоты; МХ — мультиплексор; СЧ — счетчик; БУ — блок усилителей; РД — регистр данных

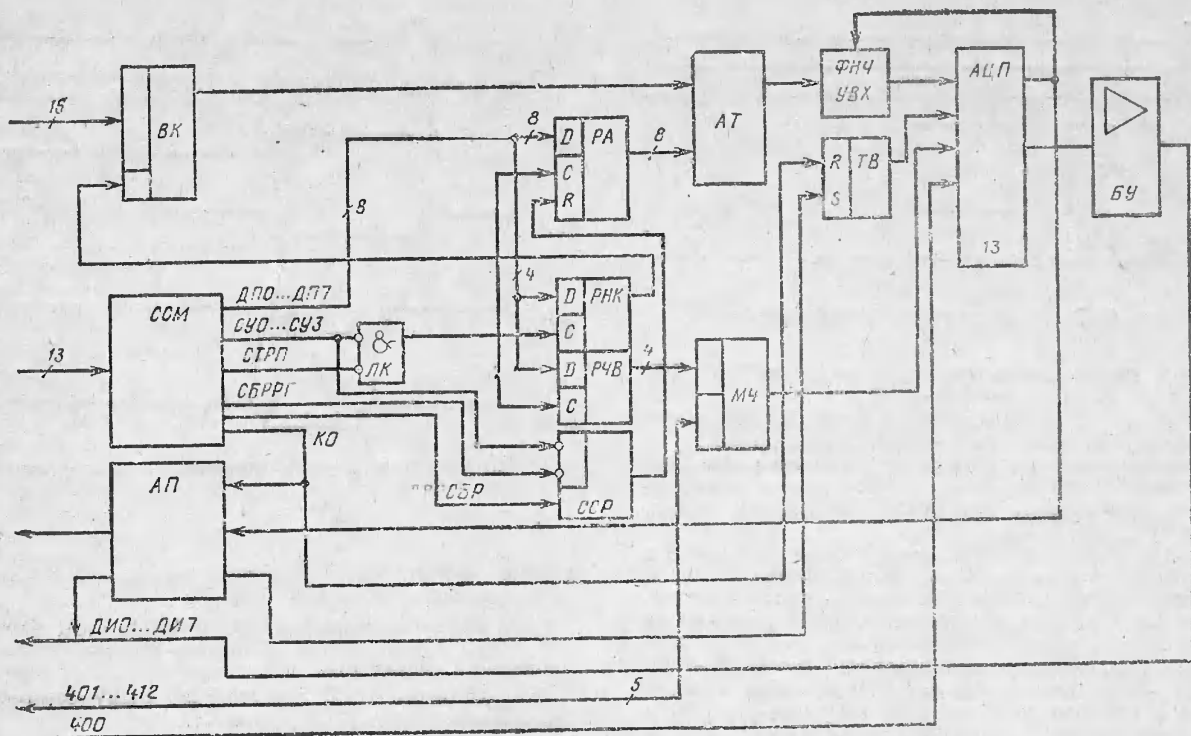


Рис. 12. АЦП:

ВК — входной коммутатор; ЛК — логический коммутатор; РА — регистр коэффициента ослабления аттенюатора; РНК — регистр номера канала; РЧВ — регистр частоты выборки; ССР — схема сброса регистра; АТ — аттенюатор; МЧ — мультиплексор частот; ФНЧ УВХ — устройство выборки и хранения с фильтром низкой частоты; ТВ — триггер номера байта.

частоты (на выходе ЦАП появляется ступенчатое напряжение).

Счетчик СЧ при записи Лог.0 в разряд запуска регистра РТЧ работает как обыкновенный регистр.

ЦАП 3 и 4 имеют входной регистр РД кода напряжения, ЦАП 3.2 и буферный усилитель БУ. Субадрес входного регистра ЦАП 3 равен 100, ЦАП 4 — 101.

АЦП имеет шестнадцать аналоговых входов униполярного напряжения (рис. 12). Со стороны внутренней магистрали АЦП представлен тремя субадресами 0, 2, 3.

Перед началом преобразований функциональный блок АЦП необходимо запрограммировать. Для этого в регистр РА по адресу 000 записывается код коэффициента ослабления аттенюатора АТ, в регистр РЧВ по адресу 011 — номер одной из линий опорных частот, в регистр РНК по адресу 2 — номер измеряемого канала. Программирование регистров АЦП может осуществляться в любом порядке.

Программируемый таймер (рис. 13) вырабатывает шесть попарно-зависимых частот в диапазоне от 10 Гц до  $5 \cdot 10^5$  Гц. Эти частоты формируются на двух интегральных таймерах КР580ВИ53 в типовом включении. Таймеры имеют дешифратор внутри микросхемы, поэтому два разряда регистра субадреса в схеме согласования с магистралью ССМ подключены непосредственно к ним. Третий разряд позволяет выбрать один из таймеров для обращения.

Регистр РВ блокирует все выходы программируемого таймера при появлении магистрального сигнала СБР или магистрального СБРРГ и внутреннего сигнала СТРП РАБ. Таймер запускается по первому сигналу СТРП в магистрале.

Адрес для справок: 111024, Москва, 3-я Кабельная ул., д. 1, ВСНПО «Союзвузприбор»; тел. 273-16-96

#### ЛИТЕРАТУРА

1. Стандартные интерфейсы для измерительной техники // Г. Науман и др. — М.: Мир, 1982.
2. Страшун Ю. П. Средства связи микроЭВМ семейства СМ 1800 с объектом. // Микропроцессорные средства и системы. — 1987. — № 2. — С. 70.
3. Гарасов Н. А. Устройство обработки информации с малым энергопотреблением // Микропроцессорные средства и системы. — 1985. — № 3. — С. 46.

Статья поступила 24 августа 1987

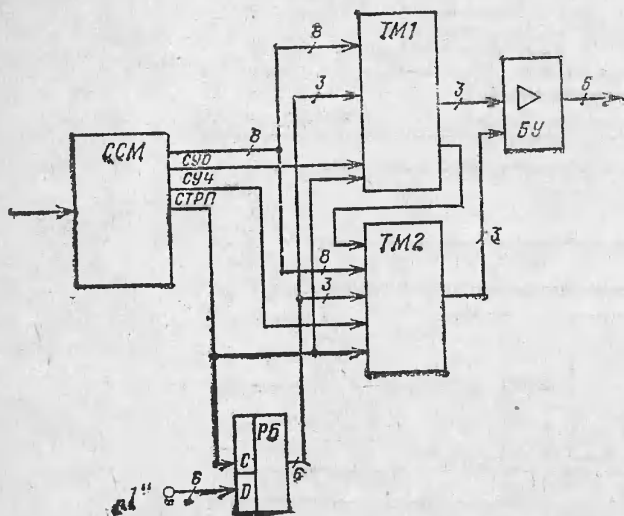


Рис. 13. Таймер

## ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ РАСШИРЕННОЙ ПАМЯТЮ ДЛЯ ПРОЦЕССОРА КР580ИК80А

Предлагаемый способ организации управления расширенной памятью позволяет использовать модули памяти, разрядность адресной шины которых больше, чем непосредственно адресует процессор, сохраняя непрерывность адресного пространства модулей. При этом модули системной памяти и «электронного диска» [1—3] могут быть идентичны или даже совмещены на одной плате.

Для организации обмена между банками используется прямой доступ к памяти (ПДП). При этом не нужно заботиться о том, чтобы активная программа оставалась постоянно подключенной к процессору, так как регистр расширенного адреса будет менять свое значение только при операциях с ПДП, а при процессор-

ных обращениях на старшие (расширенные) разряды адреса включен регистр системного банка. Его значение устанавливается программно процессором при включении питания или рестарте ЭВМ, когда определяются номера подключаемых банков ОЗУ и их количество.

Приведенная на рис. 1 схема позволяет обслужить до 1 Мбайта ОЗУ. Можно дополнительно расширить адресное пространство процессора до 16 Мбайт путем замены 4-разрядных регистров (D5, D6) на 8-разрядные или удвоением их количества. Схема состоит из контроллера ПДП (D1), регистра стробирования старших разрядов адреса АВ8... АВ15 (D2), буфера управляющих сигналов системной магистрали (D3), бу-

ферного регистра данных (D4), через который происходит обмен данными при ПДП в циклах чтения и записи в память. Микросхемы К555ИР22 можно заменять на КР580ИР82. Регистры D5 и D6 с тремя состояниями на выходе предназначены для хранения и выдачи старших (расширенных) разрядов адреса АВ16... АВ19 на системную магистраль. Микросхема D7 — ПЗУ — используется для дешифрации адреса и может быть запрограммирована на любые свободные в ЭВМ адреса внешних устройств.

Кроме того, в ЭВМ должен быть системный регистр, выставляющий на старшие разряды адреса АВ16... АВ19 номер банка системного ОЗУ, которое выбирается при процессорных обращениях и переводится в третье состояние при работе в режиме ПДП, т. е. каждый канал контроллера ПДП дополняется регистрами расширенного адреса и, таким образом, контроллер выставляет на шину сразу 20-разрядный адрес памяти, старшие биты которо-

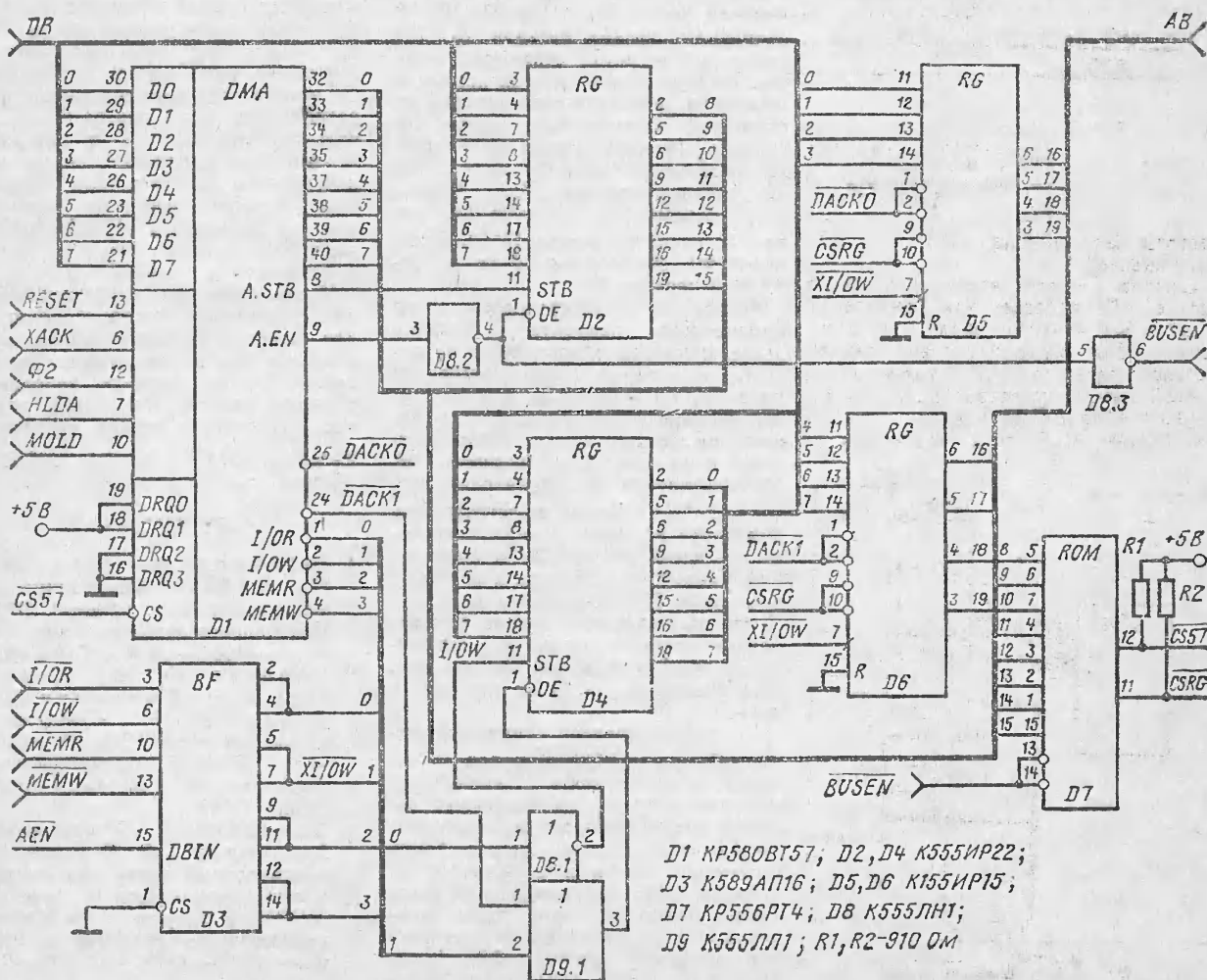


Рис. 1. Принципиальная схема устройства управления «электронным диском»

го содержат номер выбранного банка.

Канал 0 контроллера программируется на чтение из памяти во внешнее устройство, которым является буферный регистр D4, а канал 1 — на запись из внешнего устройства (D4) в память. На рис. 2 приведен фрагмент программы-драйвера «электронного диска» такой системы, включенной в секцию ввода-вывода (BIOS) ОС (стандарт для CP/M), которая выполняет чтение и запись сектора длиной 128 байт. Входными данными являются: в регистре В — номер дорожки, в регистре С — номер сектора на дорожке, в регистровой паре HL — адрес начала буферной области объемом 128 байт в системном банке памяти, куда считывается сектор «электронного диска» или откуда записывается сектор на «электронный диск».

```

;M80 ASSEMBLER CP/M
CDMA0 EQU 40H ;CHANNEL 0
CDMA1 EQU 41H
CDMA2 EQU 42H ;CHANNEL 1
CDMA3 EQU 43H
CDMAS EQU 44H ;CONTROL REGISTER
XDMA EQU 49H ;EXTERN REGISTER

; B- TRACK, C- SECTOR, HL- ADDRESS
WRITE: CALL TRSEC

XCHG
RRC
RRC
RRC
RRC
RRC
JMP STDMA

READ: CALL TRSEC

STDMA: OUT XDMA ;DE=INP,R=OUT MEM
MOV A,E
OUT CDMA0
MOV A,D
OUT CDMA0
MOV A,L
OUT CDMA2
MOV A,H
OUT CDMA2
MVI A,7FH
OUT CDMA3
MOV A,40H
OUT CDMA3
MVI A,7FH
OUT CDMA1
MVI A,80H
OUT CDMA1
MVI A,73H ;REGIM
OUT CDMAS
XRA A
RET

TRSEC: PUSH H
MOV A,C ;SECT
RRC
MOV D,A
ANI 0FH
MOV E,A
MOV A,D
MVI C,0FH
MOV D,A
MOV A,B ;TRSEC
RRC
RRC
RRC
RRC
MOV B,A
LDA C
MOV B,A
XRA B
ORA D
MOV D,A ;DE-ADR SECT
LDA B,H
MOV B,BANK
SUB B
POP B ;A-NUM. OF BANK
RET ;DMA

BANK: DB 0
END

```

Рис. 2. Драйвер «электронного диска»

Как видно из рисунков, в регистр адреса канала 0 контроллера ПДП выводятся младшие 16 бит адреса, с которого будет читаться информация, а в регистр адреса канала 1 контроллера — 16 бит адреса, на который будут записываться прочитанные данные, а старшие 4 бита адреса (номера банков) читаемой и записываемой областей компонуются в один байт и выводятся в регистры расширения адреса (D5 и D6). Длина передаваемой области задается одинаковой для обоих каналов и может быть произвольной вплоть до 16 Кбайт. Эти числа записываются в регистры управления каналов контроллера ПДП. Режим контроллера (байт 73H) задает циклический приоритет с расширенной записью и автоматическим окончанием работы для двух каналов (нулевого и первого).

После вывода начальных адресов и размеров передаваемых массивов, а также управляющего слова, задающего режим и разрешающего работу каналов, первым начинает работать канал 0, так как он имеет более высокий приоритет. При появлении единичного уровня сигнала  $BUSEN$  системный регистр отключает старшие биты адреса системной памяти и процессор переводит свои сигналы на системную магистраль в третьем состоянии. Нулевой канал контроллера выставляет на шину адреса все 20 разрядов адреса (на  $AB16...AB19$  открывается регистр D5, так как установлен сигнал  $DACK0$ ), по которому считывается байт в буферный регистр D4.

Контроллер запрограммирован на циклический приоритет. Поэтому после прочтения одного байта по адресу, на который запрограммирован канал 0, он переходит к обслуживанию запроса от канала 1, т. е. записи по адресу памяти, установленному в канале 1 и регистре D6, открывающемся при появлении сигнала  $DACK1$ . Затем по сигналу чтения порта на шину данных открывается буферный регистр D4 и появится информация, прочитанная в предыдущем цикле ПДП по адресу канала 0. После обслуживания канала 1 контроллер опять переходит к обслуживанию канала 0 и так до тех пор, пока заданный массив не передается до конца.

По концу передачи снимается единичный сигнал  $BUSEN$  и для считывания процессором следующей команды открывается системный регистр, который выдает на разряды расширенного адреса номер банка системной памяти и подключает ее.

Таким образом, время пересылки одного байта без учета тактов ожидания памяти равняется восьми тактам опорной частоты Ф2 (один цикл контроллера ПДП равен четырем тактам), т. е. при максимальной

частоте процессора (2,5 МГц) — 3,2 мкс, а скорость обмена — 300 Кбайт/с.

При разбиении на секторы и дорожки под управлением ОС CP/M учитывались некоторые особенности дисковых устройств, работающих под этой ОС, и минимизация обслуживающей программы-драйвера. Было выбрано такое разбиение, при котором на одной дорожке находится 32 сектора, т. е. одна «дорожка» включает в себя 4 Кбайта ОЗУ, а один банк памяти состоит из 16 «дорожек».

При включении питания программы проверяются на наличие все банки ОЗУ, номер системного банка, номера банков и размер «электронного диска». Полученная информация помещается в ячейку BANK (см. рис. 2) и номер банка системной памяти выводится в системный порт.

Как показывает практика работы с таким «электронным диском» под управлением ОС CP/M, достаточным для работы практически со всеми программами является общий объем ОЗУ в 256 Кбайт, из которых 64 отводится под системное ОЗУ, а 192 — под «электронный диск». Время работы системы и программ существенно уменьшилось и ГМД стали необходимы только для хранения разрабатываемых программ и начальной загрузки ОС и программ-утилит. Для микроЭВМ, снабженной накопителем на кассетном магните, вообще отпадает необходимость в применении ГМД, так как указанные функции ГМД можно производить и с НКМЛ.

Кроме описанного применения данного контроллера можно использовать для регенерации динамического ОЗУ, а также при использовании только канала 1 для быстрого заполнения областей памяти константой, например для очистки экрана дисплея.

Телефон для справок: 466-17-09, Москва

#### ЛИТЕРАТУРА

1. Зеленко Г. В., Панов В. В., Попов С. Н. Электронный квадрик для персональной ЭВМ // Микропроцессорные средства и системы. — 1984. — № 4. — С. 79—82.
2. Сорокин Ю. Ю., Лаврентьев В. В., Максимак С. П., Субач В. В. «Электронный диск» для микроЭВМ «Электроника 60M» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 92.
3. Злотник Е. М., Стежко И. К., Анищенко В. В., Киркоров С. И. «Электронный диск» для вычислительных комплексов на базе микроЭВМ «Электроника 60» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 90—92.

Статья поступила 28 марта 1987

УДК 681.3.01:51

## ИНТЕРФЕЙСНЫЕ БИС МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА K1801\*

Микропроцессорный комплект БИС серии K1801 наряду с однокристальными микропроцессорами [1, 2] содержит ряд универсальных интерфейсных БИС для управления блоками оперативной памяти (K1801ВП1-030) и внешними устройствами (K1801ВП1-035), организации интерфейсных устройств параллельного ввода-вывода (K1801ВП1-033, K1801ВП1-034). Все интерфейсные БИС разработаны на основе базового матричного кристалла (БМК) K1801ВП1 с применением библиотечных элементов: инверторов, многоходовых элементов И-НЕ, ИЛИ-НЕ, триггеров и т. д. Собственно БМК состоит из ячеек трех типов.

Изготовление БИС на основе БМК заключается в формировании с помощью двух верхних слоев (слой металлизации и слой контактов) разветвленных связей необходимой конфигурации между ячейками БМК. Процесс проектирования и изготовления БИС осуществ-

\* Цикл статей, рассчитанный на публикацию с продолжением в последующих номерах журнала в разделе «Справочная информация».

ляется в несколько этапов (техническое задание, ввод исходной информации, логическое моделирование, синтез тестов, трассировка связей, расчет паразитных элементов связей, расчет электрической схемы) с помощью автоматизированной системы проектирования. На основе БМК K1801ВП1 спроектировано несколько сотен различных заказных схем в основном частного применения, т. е. предназначенных для конкретной аппаратуры.

В разделе «Справочная информация» будут рассмотрены универсальные интерфейсные БИС, получившие наиболее широкое распространение: K1801ВП1-030 — устройство управления блоками ОЗУ динамического типа;

K1801ВП1-033 — многофункциональное устройство, позволяющее реализовать интерфейс накопителя на гибких магнитных дисках, контроллер интерфейса параллельного ввода-вывода, контроллер байтового параллельного интерфейса;

K1801ВП1-034 — многофункциональное устройство, предназначенное для организации устройства передачи данных, устройства выдачи вектора прерывания и компаратора адреса, буферного регистра данных;

K1801ВП1-035 — контроллер внешних устройств, работающих на линию связи с последовательной передачей данных.

Таблица 1

Основные электрические параметры интерфейсных БИС серии K1801ВП1

Параметр, единица измерения	Обозначение	Значение		Температура, °С	Режим измерения
		не менее	не более		
Выходное напряжение низкого уровня, В	$U_{OL}$	—	0,4	$+25 \pm 10$	$U_{CC} = 4,75$ В
		—	0,5	$-10 \pm 3$ $+70 \pm 3$	$I_{OL} = 3,2$ мА
Выходное напряжение высокого уровня, В	$U_{OH}$	2,5	—	$+25 \pm 10$	$U_{CC} = 4,75$ В
		2,4	—	$-10 \pm 3$ $+70 \pm 3$	$I_{OH} = 1,0$ мА
Ток утечки «Лог. 0» и «Лог. 1», мкА	$I_{LIL}$	—	1	$+25 \pm 10$	$U_{CC} = 5,25$ В
	$I_{LIH}$	—	10	$+70 \pm 3$	$U_{IL} = 0$ В $U_{IH} = U_{CC}$
Ток потребления, мА	$I_{CC}$	—	180	$-10 \pm 3$	$U_{CC} = 5,25$ В
		—	200	$+70 \pm 3$	$U_{IL} = 0$ В $U_{IH} = U_{CC}$
Напряжение питания, В	$U_{CC}$	4,75	5,25	—	—
Емкость нагрузки, пФ	$C_L$	—	100	—	—
Длительность фронта выходного сигнала, нс	$t_{\phi}$	—	50	—	$C_L = 100$ пФ

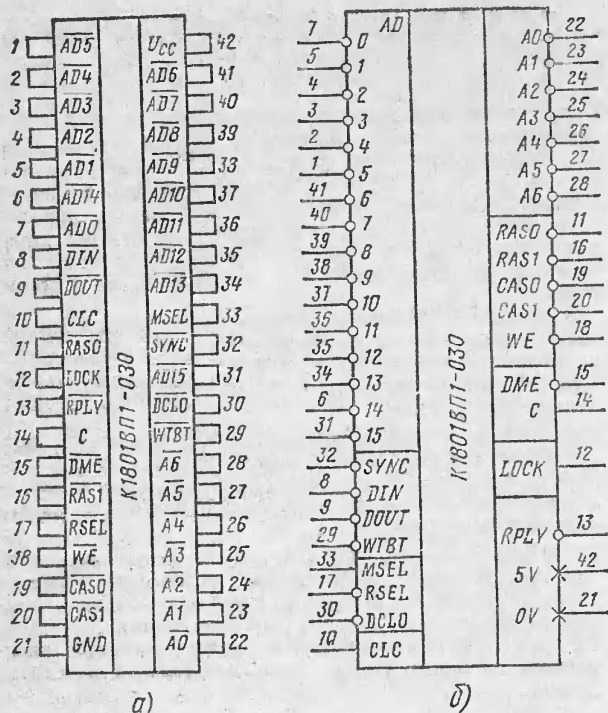


Рис. 1. Условное графическое обозначение микросхемы К1801ВР1-30 по порядку расположения (а) и функциональному назначению (б) выводов

Все микросхемы изготавливаются по n-канальной МОП-технологии в керамических 42-выводных корпусах типа 429.42-5. Основные параметры микросхем приведены в табл. 1.

В этом номере журнала представлена микросхема К1801ВР1-030. Остальные интерфейсные БИС комплекта К1801 и рекомендации по их применению будут рассмотрены в последующих номерах.

### Микросхема К1801ВР1-030

Предназначена для использования в блоках ОЗУ, выполненных на основе микросхем К565РУ3 с динамическим хранением данных, в качестве устройства управления (контроллера) обменом данных. Условное графическое обозначение микросхемы дано на рис. 1, назначение выводов приведено в табл. 2. Номер ТУ: ОК.348.740-0301ТУ. Контроллер рассчитан на управление блоком ОЗУ емкостью 16К 16-разрядных слов, выполненном на 16 микросхемах серии К565РУ3, или двумя полублоками. Микросхема осуществляет следующие функции:

- прием, хранение и преобразование адреса для накопителя ОЗУ;
- регенерацию памяти;
- обмен информацией по системной магистрали микроЭВМ «Электроника 60»;
- выработку вспомогательного сигнала блокирования LOCK при обращении в область внешних устройств (старшие 4К слов). Сигнал LOCK предназначен для использования БИС К1801ВР1-030 совместно с микропроцессором К1801ВМ1.

Структурная электрическая схема контроллера ОЗУ (рис. 2) включает счетчик адресов регенерации (СЧАР),

Таблица 2

Назначение выводов микросхемы К1801ВР1-30

Вывод	Обозначение	Тип вывода	Назначение
1...5	$\overline{AD5} \dots \overline{AD1}$		Адресные входы
6	$\overline{AD14}$	Вход	Четырнадцатый разряд адреса-данных
7	$\overline{AD0}$	Вход	Нулевой разряд адреса-данных
8	DIN	Вход	Чтение данных
9	DOUT	Вход	Запись данных
10	CLC	Вход	Синхронизация
11	$\overline{RAS0}$	Выход	Сигнал сопровождения адреса строки накопителя (1-ый полублок)
12	LOCK	Выход	Блокировка
13	RPLY	Выход	Ответ
14	C	Выход	Сигнал стробирования записи в буферный регистр данных (БРД)
15	$\overline{DME}$	Выход	Выборка данных памяти
16	$\overline{RAS1}$	Выход	Сигнал сопровождения адреса строки накопителя
17	$\overline{RSEL}$	Вход	Выборка регистра режима
18	$\overline{WE}$	Выход	Сигнал сопровождения записи в накопитель ОЗУ
19	$\overline{CAS0}$	Выход	Сигнал сопровождения адреса колонки накопителя (1-й полублок)
20	$\overline{CAS1}$	Выход	Сигнал сопровождения адреса колонки накопителя (2-й полублок)
21	GND	—	Общий
22	$\overline{A0}$	Вход	Нулевой разряд адреса накопителя ОЗУ
23...28	$\overline{A1} \dots \overline{A6}$	Выходы	Разряды адреса накопителя ОЗУ
29	WTBT	Вход	Запись-байт
30	DCLO	Вход	Авария источника питания
31	$\overline{AD15}$	Вход	Пятнадцатый разряд адреса-данных магистрали
32	SYNC	Вход	Обмен
33	MSEL	Вход	Выборка памяти
34...41	$\overline{AD13} \dots \overline{AD6}$	Входы	Разряды адреса-данных
42	UCC	—	Напряжение источника питания



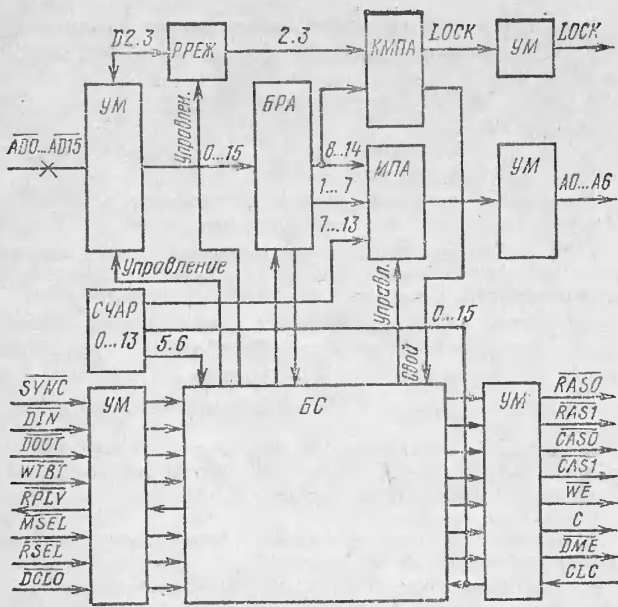


Рис. 2. Электрическая структурная схема микросхемы К1801ВР1-030

буферный регистр адреса (БРА), мультиплексор адресов (МПА), регистр режима (РРЕЖ), компаратор адресов (КМПА), блок синхронизации (БС), буферные усилители мощности (УМ).

Счетчик адресов СЧАР объединяет 7-разрядный делитель тактовой частоты (разряды 0...6) и собственно 7-разрядный счетчик адресов регенерации (разряды 7...13). В режиме регенерации содержимое разрядов 7...13 через мультиплексор поступает на выходы А0...А6 и является адресом регенерации.

Буферный регистр БРА (разряды 0...15) предназначен для хранения адреса, поступающего по системной магистрали на выходы AD0...AD15. Содержимое разрядов 1...7 в фазе выдачи адреса строки поступает через мультиплексор на выходы А0...А6 в режиме обмена.

Содержимое разрядов 8...14 поступает на выходы А0...А6 в фазе выдачи адреса столбца и на компаратор для выработки сигнала LOCK. Содержимое разряда 0 используется в блоке синхронизации для определения номера байта при выполнении процедуры записи байта. Запись данных в БРА происходит высоким уровнем сигнала SYNC. Фиксируется запись низким уровнем этого сигнала.

Мультиплексор МПА служит для раздельной во времени выдачи адреса ОЗУ в виде 7-разрядных адресов строки и столбца в циклах обмена с накопителем ОЗУ, а также 7-разрядного адреса регенерации.

Вспомогательный регистр режима РРЕЖ предназначен для дублированного хранения служебных признаков режимов работы «останов» и «расширенная арифметика» микропроцессора К1801ВМ1. Содержит два разряда РРЕЖ2 и РРЕЖ3, доступных по чтению и записи из канала (разряды AD2 и AD3 соответственно). Режиму «останов» соответствует значение единицы в разряде РРЕЖ2, режиму «расширенная арифметика» — в разряде РРЕЖ3. Информация, хранящаяся в РРЕЖ, влияет на установку сигнала LOCK.

Компаратор адресов КМПА вырабатывает сигнал блокировки LOCK, который служит для выборки областей адресов из системного ПЗУ и блокировки этих областей в адресном пространстве внешних устройств.

Сигнал LOCK устанавливается в следующих случаях:  
 адрес обращения находится в диапазоне 160000<sub>8</sub>...163777<sub>8</sub> (режим «останов»);  
 адрес обращения — в диапазоне 160000<sub>8</sub>...173777<sub>8</sub> (режим «расширенная арифметика»);  
 адрес обращения — в диапазоне 173000<sub>8</sub>...173777<sub>8</sub> (независимо от состояния режимов РРЕЖ2 и РРЕЖ3).  
 Компаратор вырабатывает признак «свой» для блока

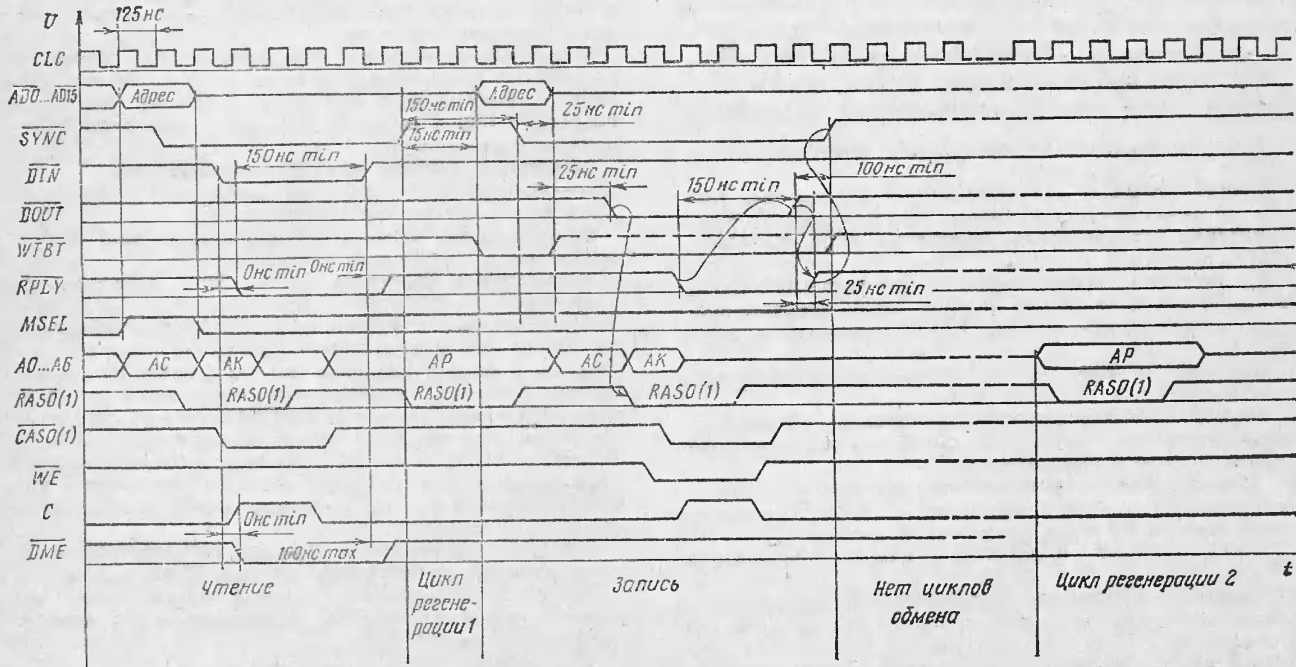


Рис. 3. Временная диаграмма работы микросхемы управления ОЗУ К1801ВР1-030

синхронизации при обращении по адресу в диапазоне 177600<sub>8</sub>...177677<sub>8</sub>, который позволяет осуществлять запуск БС в отсутствие сигнала MSEL.

Блок синхронизации БС вырабатывает сигналы управления внутренними узлами микросхемы, накопителем ОЗУ, внешним буферным регистром данных, а также обрабатывает сигнал управления по системной магистрали ЭВМ.

#### Режим работы контроллера

Регенерация (восстановление) информации в памяти динамического типа происходит по принципу: один цикл регенерации — по одному адресу строки в течение 15,6 мкс. Полная регенерация по всем адресам наступает через 2 мс. В БС введена схема подсинхронизации запуска текущего цикла регенерации к окончанию цикла обмена информацией (цикл регенерации 1), что делает память динамического типа максимально прозрачной для процессора (рис. 3). При отсутствии циклов обмена с памятью в течение 8 мкс после получения БС запроса на регенерацию очередной цикл регенерации наступает принудительно, без подсинхронизации (цикл регенерации 2).

Сигналы RAS0 и RAS1 вырабатываются одновременно.

Чтение из памяти возможно при воздействии сигналов SYNC, задающих адрес обращения AD0...AD15 в буферном регистре и MSEL, которые фиксируются в элементах хранения БС. Кроме того, цикл чтения наступает при отсутствии сигнала WTBT в адресной части и отсутствии очередного цикла регенерации памяти.

Временная диаграмма работы БИС К1801ВП1-030 в режиме чтения показана на рис. 3. Выходные сигналы DME и RPLY микросхема вырабатывает только при наличии сигнала DIN. Сигнал RAS0 устанавливается в фазе выдачи адреса, если AD15=1. При AD15=0 формируется сигнал RAS1. Сигналы CAS0 и CAS1 вырабатываются одновременно.

Запись в память вызывают следующие сигналы: SYNC, фиксирующий адрес обращения AD0...AD15 в БРА;

MSEL и WTBT, которые фиксируются в элементах хранения БС;

сигнал DOUT, а также отсутствие очередного цикла регенерации памяти.

Временная диаграмма работы контроллера в режиме записи так же приведена на рис. 3.

Наличие сигнала WTBT в фазе выдачи адреса является признаком записи слова, а в фазе выдачи данных — признаком записи байта. При модификации «запись байта» БС вырабатывает сигнал CAS0, если AD0=1 в фазе выдачи адреса. Если AD0=0 устанавливается сигнал CAS1. При модификации «запись слова» сигналы CAS0 и CAS1 вырабатываются одновременно.

Блокировка. Сигнал блокировки устанавливается в фазе выдачи адреса. Задержка сигнала относительно момента выдачи адреса на выходы AD0...AD15 — не более 100 нс.

Начальная установка. Микросхема приходит в состояние готовности за время, соответствующее длительности прохождения семи синхронизирующих импульсов CLC. Сигнал DCLO устанавливает СЧАР в нулевое состояние и PPEЖ в режим «станов».

#### ЛИТЕРАТУРА

1. Однокристалльные микропроцессоры комплекта БИС серии К1801/В. Л. Дшхунян, Ю. И. Борщенко, В. Р. Науменков и др. // Микропроцессорные средства и системы.— 1984.— № 4.— С. 12.
2. Однокристалльный микропроцессор КМ1801ВМ3 /Р. И. Волков, В. П. Горский, В. Л. Дшхунян и др. // Микропроцессорные средства и системы.— 1986.— № 4.— С. 37.

Тел. 208-73-23, Москва, Г. Г. Глушкова

#### «РАТЕКС»

Московское объединенное консультативно-исследовательское кооперативное предприятие. Тематика — ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ: проектирование, разработка и внедрение «под ключ», консультации, маркетинг

Выбор и техническое обоснование проектных решений;

Разработка аппаратно-программных комплексов, блоков и узлов информационных систем, средств автоматизации (контроллеров, подсистем АСУ ТП, АСНИ, ЛВС и т. д.);

Тестирование, приобретение и продажа программ, в том числе пакетов машинной графики, игровых программ, экспертных систем, САПР, речевого синтеза и т. д.);

Контроль и обеспечение совместимости ЭВМ разных классов;

Разработка и консультации по системам, обеспечивающим создание, накопление, обработку и распространение патентной информации;

Анализ рыночной конъюнктуры (внутренней и внешней): оценка альтернативных путей коммерческой реализации вновь создаваемых систем;

Консультативная поддержка экспортно-импортных операций;

Правовое обеспечение проектирования, разработки и коммерческой реализации средств информационной технологии.

«РАТЕКС» заключает договоры на абонементное обслуживание, в том числе:

Ежегодные прогностические обзоры основных секторов отечественной и зарубежной информационной технологии: вычислительная техника и информатика, связь, видеотехника и др.;

Консультации ведущих специалистов по вопросам разработки и внедрения средств вычислительной техники и программного обеспечения (по телефону, телеграфу, почтой, а также с выездом эксперта на предприятие к заказчику).

Качество продукции «РАТЕКС» гарантирует: конкурсный отбор ведущих специалистов по каждому заказу, принятому к исполнению, экономические санкции к исполнителю при любых обоснованных претензиях потребителя.

К участию в постоянно действующем конкурсе исполнителей «РАТЕКС» допускаются высококвалифицированные специалисты в области разработки и внедрения средств информационной технологии из Москвы и других регионов страны (необходимым условием является надежная телефонная связь с Москвой). Преимущества при конкурсном отборе специалистов для участия в заказных исследованиях, разработках и экспертизе «РАТЕКС» имеют авторы статей, опубликованных в журнале «МП», и конкретных работ, обсуждавшихся на семинаре «МИКРОПРОЦЕССОРЫ В ПОЛИТЕХНИЧЕСКОМ».

«РАТЕКС» принимает заказы от государственных и кооперативных организаций, центров НТМ и т. д.

Заказы на договорные услуги и предложения по участию в работе «РАТЕКС» принимаются по адресу: 109052, Москва «РАТЕКС»

Справки по телефону: 208-73-23 по вторникам и средам с 18<sup>00</sup> до 21<sup>00</sup>. Матвеев Сергей Семенович,

## РЕФЕРАТЫ СТАТЕЙ

УДК 681.322.1+681.325.5

Кушнир В. Е., Панфилов Д. И., Шаронин С. Г. Многофункциональный комплекс программно-аппаратных средств для семейства однокристальных ЭВМ серии K1816 // Микропроцессорные средства и системы.—1988.—№ 4.—С. 3.

Рассмотрены программные средства многофункционального комплекса для разработки и исследования микроконтроллеров на основе ОЭВМ серии K1816. Описаны директивы отладочного монитора микроЭВМ УМПК-48/ВМ, входящей в его состав, и приведен его исходный текст.

УДК 681.326—181.4

Нунупаров Г. М. Сопряжение микроЭВМ K1827BE1 с внешним ППЗУ // Микропроцессорные средства и системы.—1988.—№ 4.—С. 6.

Рассмотрены особенности работы микроЭВМ K1827BE1 и принципиальная схема ее сопряжения с ППЗУ K573PФ2.

УДК 681.322.1

Иванов Е. А. Совместимость ПЭВМ с помощью аппаратно-программных сопроцессоров // Микропроцессорные средства и системы.—1988.—№ 4.—С. 17.

Дается определение используемой терминологии, рассматривается взаимодействие прикладной программы, загруженной в память сопроцессора, с устройствами ввода-вывода базовой ПЭВМ, приводится анализ известных архитектур сопроцессоров (с общей памятью, «компьютер в компьютере», «компьютер в компьютере» с контроллерами УВВ) с рекомендациями по применению, приводятся характеристики сопроцессора БА 86М.

УДК 681.03

Гехман Б. И. Диалоговая система на микроЭВМ «Электроника ДЗ-28» для подготовки графических иллюстраций // Микропроцессорные средства и системы.—1988.—№ 4.—С. 23.

Рассматривается комплекс аппаратных и программных средств для микроЭВМ «Электроника ДЗ-28» с объемом ОЗУ 128 Кбайт, обеспечивающий подготовку на фотоленке цветных и черно-белых иллюстраций в виде графиков и гистограмм для докладов, статей и отчетов.

УДК 681.32

Злотник Е. М., Киркоров С. И., Стежко И. К. Графический адаптер для ППЭВМ ЕС 1840 // Микропроцессорные средства и системы.—1988.—№ 4.—С. 25.

Описан графический адаптер, предназначенный для организации на базе профессиональной персональной ЭВМ ЕС 1840 рабочей станции широкого назначения. Приведены основные технические характеристики и структурная схема графического адаптера, позволяющего отображать графическую информацию на экране растрового цветного монитора.

УДК 681.142.2

Глухов В. Н., Долбилов Л. А., Дудников Е. Е., Лоозе П. МикроDOS — адаптивная система программного обеспечения для 8-разрядных микроЭВМ // Микропроцессорные средства и системы.—1988.—№ 4.—С. 33.

Рассматривается созданное на базе ОС МикроDOS программное обеспечение для 8-разрядных микроЭВМ. Дается характеристика ОС МикроDOS и различных пакетов (прикладных программ обработки и передачи данных, обслуживающих программ, программного обеспечения интерфейса оператора с ОС, языков программирования высокого уровня и др.).

Отмечены особенности генерации ОС МикроDOS на конкретных моделях микроЭВМ.

UDC 681.322.1+681.325.5

Kushnir V. E., Panfilov D. I., Sharonin S. G. Multifunction software-hardware toolkit for K1816 single-chip microcomputer family // Microprocessor devices and systems.—1988.—N. 4.—P. 3.

The software support of multifunction toolkit designed for education purposes and for development of microcontrollers using K1816 single-chip computer family is described. The source of the Monitor/Debugger program for УМПК-48/ВМ is listed and its directives are explained.

UDC 681.326—181.4

Nunuparov G. M. External EPROM interfacing to K1827BE1 single-chip computer // Microprocessor devices and systems.—1988.—N. 4.—P. 6.

The details of operation of K1827BE1 LSI and its interfacing to external ROM are explained. The circuit diagram of K573PФ5 connection to single chip microcomputer is shown.

UDC 681.322.1

Ivanov E. A. The software compatibility obtained with hardware/software coprocessors // Microprocessor devices and systems.—1988.—N. 4.—P. 17.

The main glossary of terms related to the problem is given. The interaction of the program loaded into coprocessor RAM with host computer I/O devices is examined. The analysis of well-known coprocessor architecture implementations (common memory field approach, "computer in the computer", "computer in the computer" with I/O controllers) is made. Some practical recommendations on the application and main technical specifications of the BA 86M coprocessor are given.

UDC 681.03

Gehman B. I. Dialog graphic editor system on microcomputer // Microprocessor devices and systems.—1988.—N. 4.—P. 23.

The hardware and software system for 128K "Elektronika D3-28" microcomputer enables preparation of colour and black-and-white slides to be shown during conference sessions, as well as for articles and reports. Data are presented in the form of plots and histograms.

UDC 621.32

Zlotnik E. M., Kirkorov S. I., Stezhko I. K. Graphic adapter for EC-1840 personal computer // Microprocessor devices and systems.—1988.—N. 4.—P. 25.

The graphic adapter unit intended for use in the multipurpose workstation based on EC-1840 personal computer is described. The main technical data are listed and block diagram of the adapter displaying graphics on a colour TV monitor is shown.

UDC 681.142.2

Glukhov V. N., Dolbilov L. A., Dudnikov E. E., Looze J. MicroDOS — the adaptive software system for 8-bit microcomputers // Microprocessor devices and systems.—1988.—N. 4.—P. 33.

The description of software based on MicroDOS operating system is given. Main features of MicroDOS and problem-oriented utilities (data communication and handling, support utilities, user interface to the OS, high-level language systems etc.) are discussed. The system alternation procedure for some selected computers is explained.

**УДК 681.3.06**

Анисимов А. А., Анисимов Г. А. Режим разделения времени в системах с однозадачным монитором // Микропроцессорные средства и системы.—1988.—№ 4.—С. 37.

Рассматривается задача организации централизованного управления встроенной микропроцессорной системой группой автономных объектов в режиме разделения времени.

**УДК 681.3.06**

Брылев С. П. Программирование встроенных микропроцессоров с применением компилятора Паскаль/MT+ // Микропроцессорные средства и системы.—1988.—№ 4.—С. 50.

Рассмотрены особенности применения пакета программ Паскаль/MT+ для программирования встроенных модулей на основе микропроцессора КР580ВМ80. Кратко описана технология разработки программ.

**УДК 681.3.06**

Филин А. Л. Организация программ динамической структуры при разработке интегрированных систем с использованием языка высокого уровня // Микропроцессорные средства и системы.—1988.—№ 4.—С. 55.

Рассматривается построение интегрированной системы из программ на языке высокого уровня с помощью модулей для динамической загрузки программ ядром и функций для обращения к его компонентам из загруженной программы. Описывается программная реализация для операционной системы MS DOS и языка Си (компилятор MSC=4.0) на ЭВМ типа EC-1840 (IBM PC).

**УДК 681.3.06**

Кизуб А. В. РИСК — реально используемая сетка конструктора // Микропроцессорные средства и системы.—1988.—№ 4.—С. 61.

Дано краткое описание метода реально используемой сетки конструктора (РИСК) и программы обработки графической информации, созданной на его основе. РИСК позволяет из неточных эскизов получать правильные чертежи с возможностью их параметризации.

**УДК 621.372.061.2**

Баталов Б. В., Русаков С. Г., Савин В. В. Пакет прикладных программ автоматизации схемотехнического проектирования для персональных компьютеров // Микропроцессорные средства и системы.—1988.—№ 4.—С. 63.

Пакет ориентирован на интерактивный режим расчета электрических характеристик интегральных схем. Рассмотрены особенности версий ППП АРИС для персональных компьютеров IBM PC и микроЭВМ класса ДВК.

**УДК 681.327**

Горшков Б. Л. Особенности построения измерительных трактов для систем автоматизации на основе ЭВМ семейства «Электроника 60» // Микропроцессорные средства и системы.—1988.—№ 4.—С. 70.

Анализируются проблемы создания прецизионных аналоговых трактов для микроЭВМ семейства «Электроника 60», предназначенных для автоматизации управления промышленным оборудованием в условиях мощных промышленных помех.

**UDC 681.3.06**

Anisimov A. A., Anisimov G. A. Time-sharing mode in the single-job monitor systems. // Microprocessor devices and systems.—1988.—N. 4.—P. 37.

The implementation of centralized control of multiple autonomous objects by the single built-in microprocessor system in a time-sharing mode is described.

**UDC 681.3.06**

Brylev S. P. Built-in microprocessor programming using PASCAL MT+ compiler. // Microprocessor devices and systems.—1988.—N. 4. P. 50.

The details of PASCAL MT+ application for ROM-resident software development dedicated for the built-in microprocessor systems with КР580ВМ80 CPU are explained. The software development process is described in brief.

**UDC 681.3.06**

Filin A. L. Dynamically-structured programs creation in high-level language for integrated software packages. // Microprocessor devices and systems.—1988.—N. 4.—P. 55.

The procedure of an integrated software system building using dynamically-swapped high-level language "shell" programs loaded by the kernel module, as well as program requests to its principal components are explained. The inside look at the implementation of MS DOS and MSC-4.0 compiler on EC-1840 an IBM PC is made.

**UDC 681.3.06**

Kizub A. V. RISK — really-used designer's scale grid. // Microprocessor devices and systems.—1988.—N. 4.—P. 61.

A brief description of the method of "really-used scale grid" and its application in some programs is given. The method enables conversion of draft drawings into correct high-quality artwork and the possibility of its parametrization.

**NDC 621.372.061.2**

Batalov B. V., Rusanov S. G., Savin V. V. Software package for automated integrated circuit design on personal computer. // Microprocessor devices and systems.—1988.—N. 4.—P. 63.

An interactive software package supporting integrated circuit design and computing electrical characteristics is proposed. Two versions of ARIS package for IBM PC and DBK personal computers are discussed in detail.

**UDC 681.327**

Gorshkov B. L. Specific circuit design of analog I/O units for automation systems with LSI-11 compatible computers. // Microprocessor devices and systems.—1988.—N. 4.—P. 70.

The detailed analysis of circuit design methods applicable for analog I/O units working in industrial environment under high level of electromagnetic interference. The units have O-bus system interfaces and are designed for industrial control systems.

От редакции. После того как мы опубликовали в «Пятиугольнике МП», № 1, 1986 г. письмо одного из наших подписчиков, который хотел бы стать нашим автором и поэтому не согласен с отказом редакции в публикации его статьи «Алгоритмы и программы встроенного микропроцессора для очистки настольных поверхностей», резко увеличился поток писем от авторов неопубликованных по разным причинам статей.

В то же время часть читателей с недоверием восприняла этот материал и требует от редакции прекратить «эти розыгрыши или оставить их только для первоапрельских номеров». Поэтому ниже мы публикуем подлинник очередного сердитого письма и саму статью «Комбинированное устройство: письменный стол — ПЭВМ» с возражением на наш отказ принять ее к публикации и просим читателей ответить авторам непосредственно или через редакцию.

**Письмо авторов в редакцию.** Направляя для опубликования в Вашем журнале статью «Комбинированное устройство: письменный стол — ПЭВМ» (письмо НИИУ автопрома от 22.10.86 № 13/2345), мы исходили из того, что данная статья относится к следующим разделам Вашего журнала:

применение микропроцессорных средств; персональные компьютеры, — и, следовательно, вполне соответствует направлению журнала, указанному в № 1, 1984 г. Поэтому нам непонятен ответ журнала с утверждением о том, что статья не отвечает профилю и направленности журнала. Кроме того, на наш взгляд удачно, статья служит развитию темы, поднятой в статье Гиглавога А. В., Котлярова В. П. «Возможен ли идеальный персональный компьютер?» (о которой мы при подготовке своей статьи не знали).

В нашей статье изложено описание принципиально нового изделия микропроцессорной техники — комбинированного устройства письменный стол —

ПЭВМ (столем), которое, по нашему мнению, может сравниться к концу века по распространенности с холодильниками или магнитофонами. Как указано в статье, применение столов имеет значительное преимущество (в ряде случаев) перед ПЭВМ уже сложившихся форм. По нашим оценкам, экономический эффект от применения столов при их массовом выпуске может достигать сотен миллионов рублей в год. И чем скорее найдутся люди (после прочтения статьи в журнале), которые смогут подключиться к организации выпуска столов, тем быстрее страна сможет получить этот экономический эффект.

Если, по Вашему мнению, положения статьи спорны, она может быть опубликована в порядке обсуждения. Если все же мы не правы, хотелось бы получить более развернутое и обоснованное изложение позиции журнала.

603600, г. Горький, ГСП-852, телефон: 33-79-99, к. т. н. Бронфельд Г. Б., Соболев А. И.

УДК 681.322.1

Г. Б. Бронфельд, А. И. Соболев

## КОМБИНИРОВАННОЕ УСТРОЙСТВО: ПИСЬМЕННЫЙ СТОЛ — ПЭВМ

В статье изложен критический анализ возможностей использования существующих персональных ЭВМ (ПЭВМ) пользователем, имеющим в качестве рабочего места письменный стол. Предложен новый подход к организации рабочего места с применением ПЭВМ и описано комбинированное устройство письменный стол — ПЭВМ, служащее для этих целей. Приведен эскиз изделия и указаны источники экономической эффективности от его применения.

В последние годы за рубежом, а теперь и в СССР начался массовый выпуск ПЭВМ. С точки зрения организации рабочего места для пользователя, сидящего за письменным столом (далее будем называть просто столом), использование существующих ПЭВМ [1—3] имеет следующие недостатки:

стационарные настольного исполнения — занимают много места на столе или же требуют дополнительного стола; не могут быть быстро убраны при ненадобности; имеют существенно разные конструктивы (даже подчас ПЭВМ одного типа), а поэтому блоки не взаимозаменяемы;

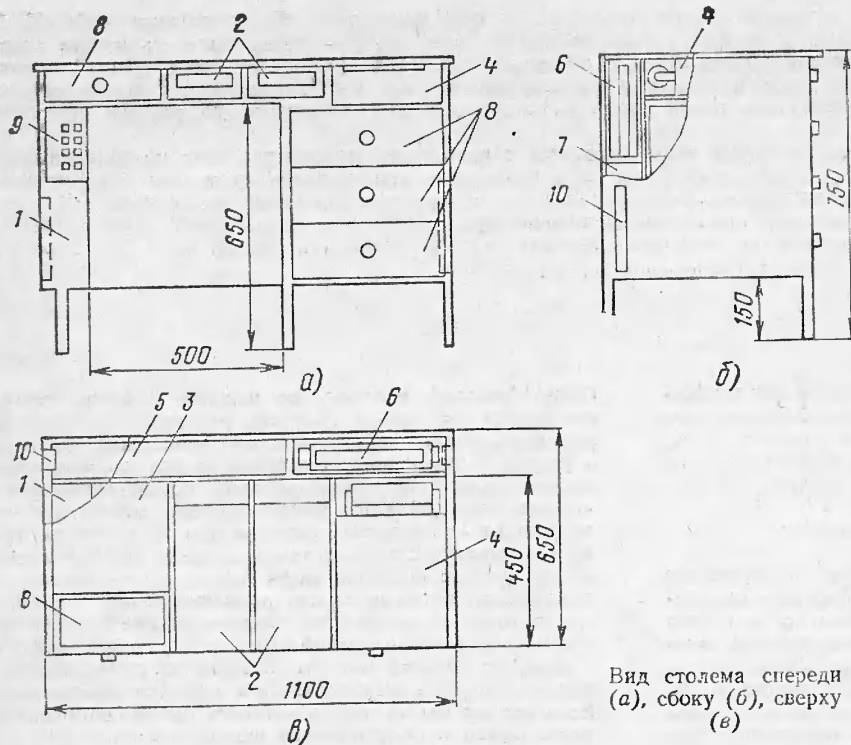
стационарные встроенного исполнения — занимают весь объем столешницы [2, 3], не оставляя существенного места для ящиков стола — традиционных мест хранения; поверхность столешницы обычно плохо приспособлена для работы с рукописными документами и книгами (имеет выступы, различные наклон поверхности и т. п.); часть места постоянно занимает дисплей; бумага, выступающая из печатающего устройства с результатами расчетов, вы-

ходит обычно за пределы стола, что требует больших площадей для размещения ПЭВМ, а пользователю необходимо вставать, чтобы собрать распечатку;

портативные — имеют малые возможности и их тоже требуется куда-то убирать после использования, а это действие обычно не увязано с традиционными формами и размерами стола.

Для удовлетворения пользователя необходимо встроить ПЭВМ в стол так, чтобы он имел традиционные ящики, свободную поверхность стола, привычные эргономические характеристики [3, 4]. Это можно осуществить, используя, в отличие от традиционной компоновки [4, 5], встраивание отдельных блоков в объемные конструктивы стола (в левую и правую боковины стола, и в заднюю и верхнюю части стола) и применяя некоторые современные технические достижения в области технических средств ПЭВМ.

Для образуемого таким образом рабочего места, сочетающего возможности письменного стола и ПЭВМ,



Вид стола спереди (а), сбоку (б), сверху (в)

предлагается название «столем» (совуочно сочетанию «стол—электронная машина»).

Рассмотрим один из вариантов возможной компоновки стола (см. рисунок). Стол состоит из следующих составных частей: процессора и ОЗУ (1), встраиваемых в левую боковину стола; ВЗУ на гибких МД (2) и МД типа «винчестер» (3); печатающего устройства (4), выдвигаемого при печати или заправке бумаги; устройства преобразования речевого ввода информации и блока связи с локальной вычислительной сетью или модема для связи по телефонным каналам, встроенных в заднюю часть стола (5), оставляющих свободное место в виде закрывающегося откидной крышкой кармана для плоской дисплейной панели (6) с накладным сенсорным экраном; общего блока питания (7), встроенного под карманом для плоской дисплейной панели. Плоская клавиатура с беспроводной инфракрасной линией связи при необходимости может складываться вдвое в виде книги и убираться вместе с микрофоном для речевого ввода информации в один из ящиков (8) стола, которые имеются как в пра-

вой тумбе (три ящика), так и в верхней левой части стола (один ящик). Дисплейная панель имеет самфиксирующуюся планку, что позволяет устанавливать ее в любом удобном месте стола. Впереди на левой боковине стола имеется пульт (9), оснащенный элементами индикации состояния устройств стола и кнопками включения питания, сброса ошибок, управления подачей бумаги в печатающем устройстве, управления накопителями на МД и т. д.

С левой и правой сторон стола в углублениях (10) имеются два одинаковых блока разъемов, чтобы увеличить число вариантов установки стола в помещениях. При работе используется лишь один из блоков разъемов. Через него подается питание, осуществляется подключение к локальной или региональной вычислительной сети, могут подсоединяться микрофон или дополнительные устройства.

Боковые стенки стола и крышка могут сниматься, что обеспечивает удобный подход ко всем блокам. При необходимости отвод тепла обеспечивается подачей воздуха вентиляторами через прорези в стенках столе-

ма и фильтры в нижней части. Подключение всех блоков посредством кабелей с разъемами дает возможность периодически заменять устройства на более современные и мощные, используя старые конструктивы стола. Поскольку стол, как принципиально новый вид изделия, может просуществовать несколько десятилетий и выпускаться в количествах, измеряемых миллионами единиц в год, то для реализации указанной выше возможности имеет смысл стандартизировать конструктивы и присоединительные размеры стола.

Конструктивные размеры рассмотренной компоновки стола, существенные с эргономической точки зрения [4, 5], приведены на рисунке. Большинство из них считаются оптимальными.

Столы позволяют:

оснастить специалистов психологически привычным и удобным элементом рабочего места;

экономить площади для размещения специалистов, так как обычная ПЭВМ или занимает большую часть письменного стола, или не приспособлена для использования в качестве письменного стола;

экономить на конструктивах и накладных расходах за счет выпуска совмещенного устройства;

обеспечить преемственность блоков ПЭВМ в течение длительного времени постепенной заменой лишь отдельных элементов стола.

## ЛИТЕРАТУРА

1. Яковлев Ю. С., Новиков Б. В., Нестеренко Н. В. Особенности применения архитектурно-структурной организации персональных ЭВМ (обзор) // Управляющие системы и машины.— 1984.— № 5.— С. 7—12.
2. Романов Ф. И., Шахнов В. А. Конструктивные системы микроЭВМ.— М.: Радио и связь, 1983.
3. Романов Г. М. Человек и дисплей.— Л.: Машиностроение, 1986.
4. Смирнов Е. Л. Справочное пособие по НОТ.— М.: Экономика, 1981.
5. Дитрих Я. Проектирование и конструирование: системный подход.— М.: Мир, 1981.

# 50-я книга в серии «Библиотечка программиста» издательства «Наука»

Серия «Библиотечка программиста», начавшаяся в 1968 г. книгой А. Л. Брудно «Алгол», сразу взяла курс на теоретические разработки в области математического обеспечения ЭВМ и их применение в конкретных вычислительных системах. Круг читателей определился сразу — научные сотрудники в области разработки вычислительных систем, программисты, студенты вузов, инженеры — электронщики.

Интерес к этой серии не ослабевает вот уже 20 лет. К настоящему времени — времени всеобщей компьютеризации, особое значение приобретает новое звучание серии в качестве вспомогательного средства для образования специалистов, приобщающихся к использованию вычислительной техники.

1. А. Л. Брудно. Алгол.—1968.
2. Ж. Бертэн, М. Риту, Ж. Ружиё. Работа ЭВМ с разделением времени / Пер. с франц. под ред. С. С. Лаврова.—1970.
3. Лавров С. С., Гончарова Л. И. Автоматическая обработка данных (хранение информации в памяти ЭВМ).—1971.
4. А. Л. Брудно. Алгол.—Изд. 2-е.—1971.
5. В. А. Васильев. Язык алгол-68. Основные понятия / Под ред. С. С. Лаврова.—1972.
6. Ю. А. Первин. Основы фортрана.—1972.
7. Ж. Бертэн, М. Риту, Ж. Ружиё. Работа ЭВМ с разделением времени.—Изд. 2-е.—1972.
8. И. Л. Братчиков. Синтаксис языков программирования / Под ред. С. С. Лаврова.—1975.
9. Л. Дж. Коэн. Анализ и разработка операционных систем / Пер. с англ. под ред. В. Ф. Тюрина.—1975.
10. Д. Уолш. Руководство по созданию документации для математического обеспечения / Пер. с англ. под ред. Н. И. Козлова.—1975.
11. Б. М. Павлов, И. Н. Посохов. Математическое обеспечение ЭВМ типа М-20.—1975.
12. В. Я. Карпов. Алгоритмический язык фортран. Фортран-Дубна / Под ред. Н. Н. Говоруна.—1976.
13. А. И. Салтыков, Г. И. Макаренко. Программирование на языке фортран / Под ред. Н. Н. Говоруна.—1976.
14. В. Н. Бусленко. Автоматизация имитационного моделирования сложных систем / Под ред. и с послесловием Н. П. Бусленко.—1977.
15. В. Ф. Демин, Л. В. Добролюбов, В. А. Степанов. Системы программирования на алголе / Под ред. Е. А. Гребеникова.—1977.
16. Г. М. Адельсон-Вельский, В. Л. Арлазаров, М. В. Донской. Программирование игр.—1978.
17. В. В. Кобелев. Машинная графика для системы БЭСМ-алгол / Под ред. Г. Г. Рябсва.—1978.
18. С. С. Лавров, Г. С. Силагадзе. Автоматическая обработка данных. Язык лисп и его реализация.—1978.
19. Г. Л. Мазный. Программирование на БЭСМ-6 в системе Дубна / Под ред. Н. Н. Говоруна.—1978.
20. Г. Г. Белоногов, А. П. Новоселов. Автоматизация процессов накопления, поиска и обобщения информации.—1979.
21. Ф. П. Брукс мл. Как проектируются и создаются программные комплексы. Мифический человек-месяц. Очерк по системному программированию / Пер. с англ. под ред. А. П. Ершова.—1979.
22. Н. П. Брусенцов. Миникомпьютеры / Под ред. Л. Н. Королева.—1979.
23. В. Ф. Жиров. Математическое обеспечение и проектирование структур ЭВМ / Под ред. Л. Н. Королева.—1979.
24. Транслятор альфа-6 в системе «Дубна» / Под ред. А. П. Ершова.—1979.
25. С. А. Усов. Диалоговый монитор Димон / Под ред. В. М. Брябина.—1980.
26. Ю. М. Безбородов. Сравнительный курс языка PL-1 (на основе алгола-60).—1980.
27. В. Р. Хисамудинов, В. С. Авраменко, В. И. Легоньков. Автоматизированная система информационного обеспечения разработок.—1980.
28. Н. И. Козлов. Организация вычислительных работ.—1981.
29. М. Крейн, О. Лемуан. Введение в регенеративный метод анализа моделей / Пер. с англ. под ред. В. В. Калашникова.—1982.
30. Ю. М. Безбородов. Индивидуальная отладка программ.—1982.
31. Л. А. Осипов. Язык аналитик и его сравнение с языками алгол и фортран.—1982.
32. В. М. Пентковский. Автокод Эльбрус / Под ред. А. П. Ершова.—1982.
33. С. Н. Бушев, М. С. Бесфамильный. Программно-аппаратные методы управления данными / Под ред. С. В. Емельянова.—1982.
34. К. Жаблон, Ж. К. Симон. Применение ЭВМ для численного моделирования в физике / Пер. с франц. под ред. В. В. Александрова и Ю. С. Вишнякова.—1983.
35. З. Д. Усманов, Т. И. Хаитов. Программирование состояний коллекций.—1983.
36. Г. Г. Белоногов, Б. А. Кузнецов. Языковые средства автоматизированных информационных систем.—1983.
37. В. Н. Пильщиков. Язык плэнер.—1983.
38. Ю. М. Безбородов. От фортрана к PL-1.—1984.
39. А. И. Салтыков, Г. И. Макаренко. Программирование на языке фортран.—Изд. 2-е.—1984.
40. А. В. Гуляев, Н. В. Макаров-Землянский, И. В. Машечкин. Диалоговый комплекс программ Краб / Под ред. Л. Н. Королева.—1985.
41. В. А. Евстигнеев. Применение теории графов в программировании / Под ред. А. П. Ершова.—1985.
42. В. Ф. Тюрин. Операционная система Дислак.—1985.
43. А. Н. Андрианов, С. П. Бычков, А. И. Хорошилов. Программирование на языке симула-67.—1985.
44. Ю. И. Баяковский, В. А. Галактионов, Т. Н. Михайлова. Графор: Графическое расширение фортрана.—1985.
45. С. А. Абрамов. Элементы анализа программ.—1986.
46. В. И. Зуев, В. М. Крюков, В. И. Легоньков. Управление данными в вычислительном эксперименте.—1986.
47. Е. Г. Ойхман. Графические системы для СМ ЭВМ.—1986.
48. В. И. Зверев, Ю. Л. Кетков, В. С. Максимов. Алфавитно-цифровые дисплеи ЕС-7920 в диалоговых системах.—1986.
49. В. А. Успенский, А. Л. Семенов. Теория алгоритмов: основные открытия и приложения.—1987.
50. Г. К. Боровин, М. М. Комаров, В. С. Ярошевский. Ошибки-ловушки при программировании на фортране.—1987.

В ближайшее время в серии «Библиотечка программиста» выйдут из печати книги, отражающие новые перспективные направления развития языков и методов программирования (программирование мультипроцессорных супер-ЭВМ, использование абстрактных типов данных, распределенное программирование) и таких компонент программного обеспечения, как средства построения сетей ЭВМ, СУБД и САПР. Все это, а также выпуск книг по основам информатики, применению языков программирования позволит расширить читательскую аудиторию.

## ВНИМАНИЕ ИДЕТ ПОДПИСКА НА 1989 ГОД

Подписка на журнал «Микропроцессорные средства и системы» принимается отделениями «Сюзпечать» без каких-либо ограничений на всей территории Советского Союза, а за рубежом — в отделениях «Межкнига».

Журнал позволит Вам первыми узнавать о новинках в области микропроцессорной техники, а также регулярно знакомиться со схемами и примерами программ для самостоятельного создания конкретных устройств автоматизации на базе микроЭВМ и микропроцессоров.

За первые пять лет издания журнал стал настольным, рабочим пособием для профессионалов в области ЭВМ и программирования и вызвал активный практический интерес у самого широкого круга специалистов отраслей народного хозяйства, делающих первые шаги в новый для них мир микропроцессорных средств автоматизации.

Циклы статей из раздела «Учебный центр» помогут Вам на простых примерах практически освоить поистине неисчерпаемые возможности микропроцессорной техники.

На страницах журнала регулярно обсуждаются актуальные проблемы программирования, новейшие тенденции развития информационной технологии. Дискуссии ученых и специалистов по наиболее острым проблемам развития вычислительной техники помогают читателям контролировать «горячие точки» науки и технологии в этой бурно развивающейся области.

Напомним, что в розничную продажу журнал не поступает, а достать его в библиотеке, как утверждают читатели, оказывается весьма непросто.

Журнал «Микропроцессорные средства и системы» — Ваш первый путеводитель в мир микропроцессорной техники — мир техники будущего. Не забудьте своевременно оформить подписку на 1989 год!

Наш индекс по каталогу «Союзпечать» — 70588. Цена одного комплекта журналов (шесть номеров) — 6 руб. 60 коп.