МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

5 | 1988

ISSN 0233-4844

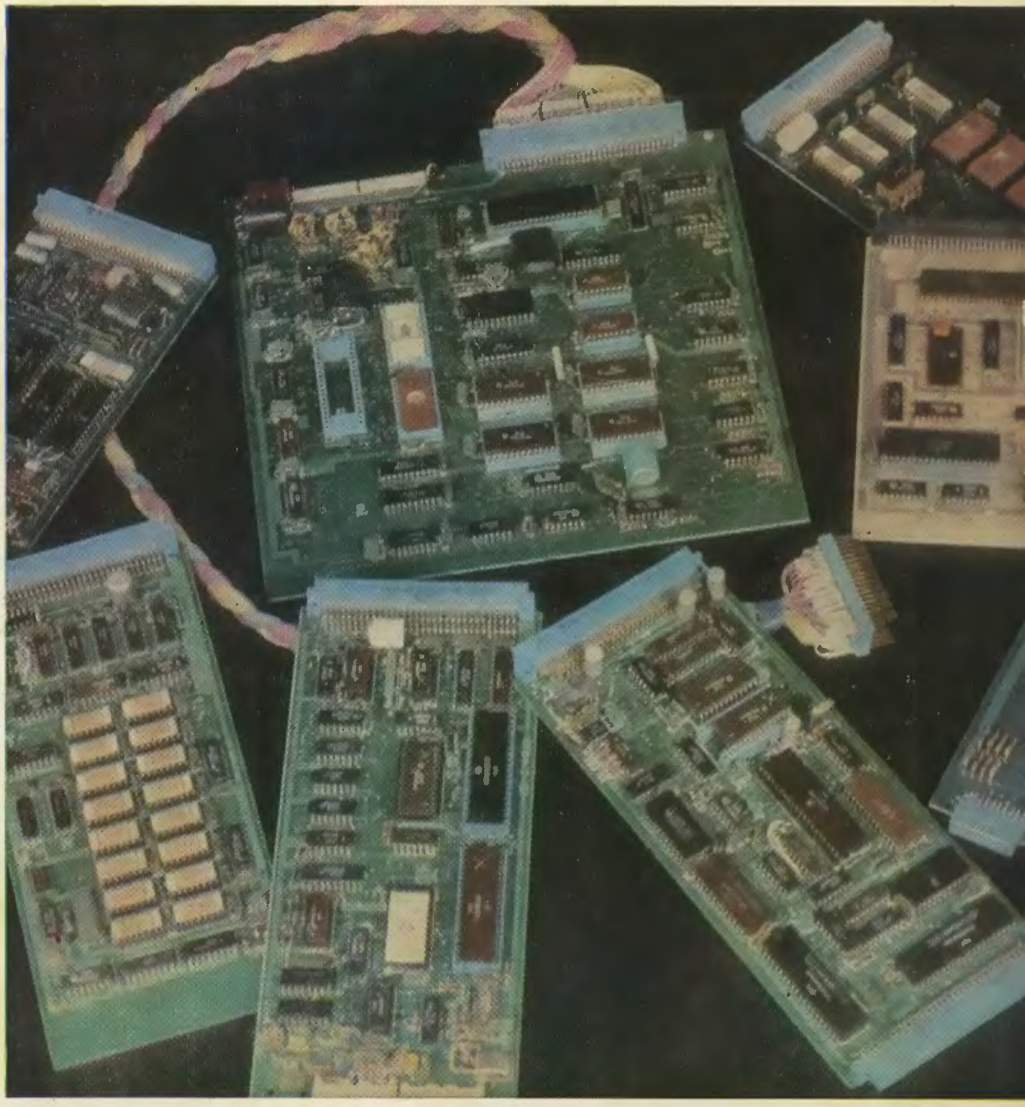
Однокристальная ЭВМ КМ1813ВЕ1 содержит встроенные аналоговые устройства ввода-вывода и репрограммируемую память с УФ-стиранием для построения систем цифровой обработки сигналов и цифрового динамического управления

Волоконно-оптические системы сбора и передачи аналоговой и цифровой информации «Электроника МС8201» и «Электроника МС4101» позволяют опрашивать до 1600 контрольных датчиков на электроустановках

Многопользовательский отладочный комплекс для инструментальной поддержки всех этапов проектирования микропроцессорных устройств

ЭВМ и информатизация общества: старые мифы и новые проблемы

Набор подпрограмм управления экраном монитора ПЭВМ «Искра 1030» открывает широкие возможности по организации диалогового интерфейса, приспособленного к «человеческим» методам решения задач силами пользователей — прикладников



СИСТЕМА ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА БАЗЕ МИКРОЭВМ, СОВМЕСТИМОЙ С IBM PC

(К ст. Сасова А. Ю.)

На Геологическом факультете МГУ создана система цифровой обработки изображений. Краткая информация о ней и примеры ее применения приводились в «МП» № 4, 1988 г. на 1-й и 2-й стр. обложки.

Аппаратная часть системы — одноплатный интерфейс для ввода изображений с черно-белых или цветных ТВ-камер и с электронного микроскопа. Программная часть обеспечивает коррекцию изображения, преобразование, обработку методами математической морфологии, а также анализ черно-белых цветных изображений, стереопар, и реализует реконструкцию трехмерных объектов по проекциям.

На этой странице показан один из примеров обработки изображений с растрового электронного микроскопа на данной системе: анализ зоны разрушения образца из алюминиевого сплава.

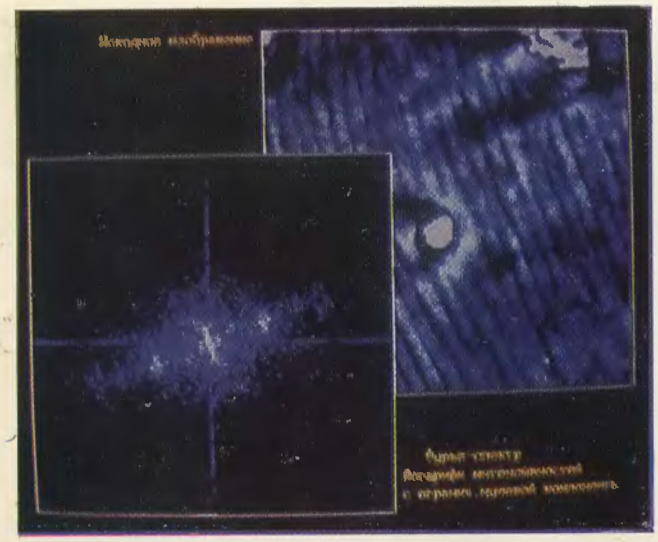
а — исходное полутоновое изображение и спектральный анализ сигнала вдоль одной строки;

б — двумерное преобразование Фурье о исходного полутонового изображения;

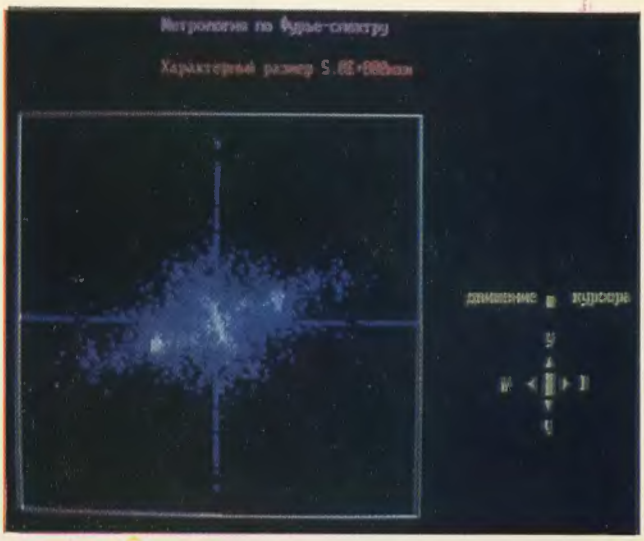
в — метрологический анализ периодичности объекта по двумерному Фурье-спектру;

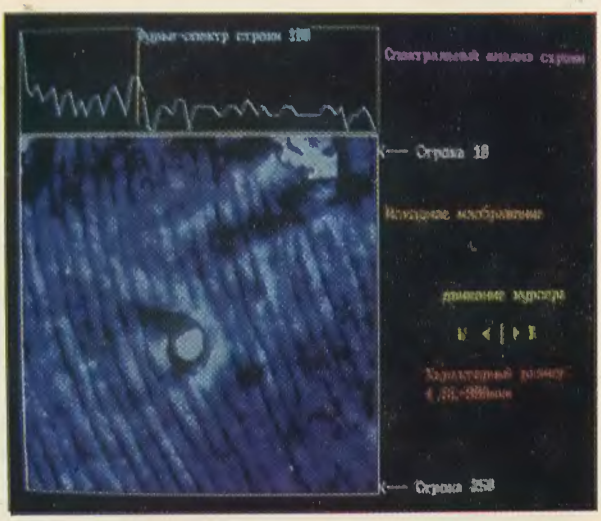
г — анализ ориентационных и гранулом рических характеристик образца по Фурье-спектру

а

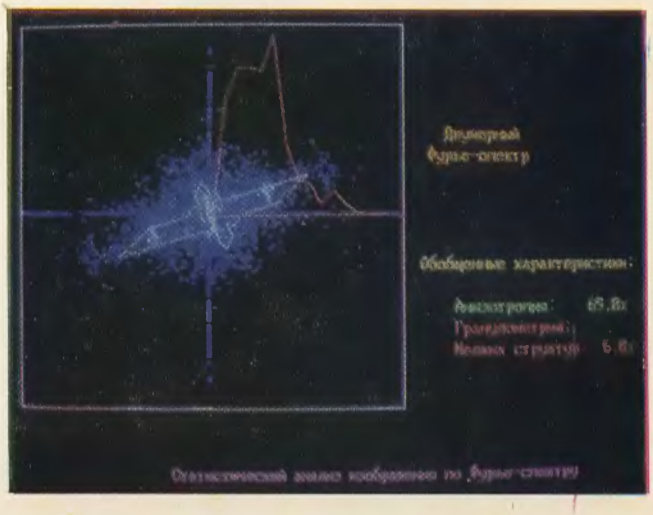


в





б



г

Пролетарии всех стран, соединяйтесь!

ОРГАН ГОСУДАРСТВЕННОГО КОМИТЕТА СССР ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ И ИНФОРМАТИКЕ

Издается с 1984 года

МП МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД

НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 5 / 1988 МОСКВА

СОДЕРЖАНИЕ

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Средства отладки МП систем

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Многомашинные комплексы

В блокнот разработчика

Форум «МП»

Справочная информация

Громов Г. Р. — Колонка редактора 2
Донев В. С. — Микропроцессорный комплект БИС серии СМ600 3
Гуревич М. Х., Кузнецов А. А., Луцкий И. Ю., Страутманис Г. Ф. — Однокристальная ЭВМ КМ1813ВЕ1 с аналоговыми устройствами ввода-вывода. . 5
Таланов В. А. — МикроЭВМ с маркой ВЭФ 11
Домарацкий С. Н., Шраго И. Л. — Организация экрана монитора ПЭВМ «Искра 1030» в программах автоматизации научных исследований . . . 13
Попов А. Л. — Комплекс технологических дискет для ПЭВМ 20
Гвездилова Г. Г., Гончаров О. А., Сенин Г. В. — К типологии компьютерных игр 22
Погода А. В. — Пакет подпрограмм на ассемблере для арифметических вычислений 27
Журавский Д. А., Барков С. Н. — Операционная система CP/M в ПЗУ . . 28
Александров А. Б. — Автоматизированная система по выбору элементной базы для технологических контроллеров 30
Пройдаков Э. М. — Дисассемблер, управляемый данными 33
Берков А. В., Поротов В. Н. — Аппаратно-программный комплекс программирования и отладки контроллеров на основе микропроцессоров серии К1801 35
Паньгин П. И., Рождественский С. М., Шагурин И. И. — Микропрограммируемый комплекс для тестирования и отладки микропроцессорных систем 39
Бухвостов Д. Ю., Жиляев Ю. П., Иванов А. И., Казанцев В. Ю., Печенкин А. Н. — ДВК-2 — «Щель» — простой программно-аппаратный комплекс для отладки встраиваемых управляющих микроЭВМ на базе МПК БИС серии К588 41
Конько В. В., Штурман А. Н., Недорубан В. А. — Аппаратно-программный отладочный комплекс 44
Кривцов А. Н. — Сигнатурный анализатор 46
Бродин В. Б., Шагурин И. И. — Микропрограммируемый схемный эмулятор для отладки микропроцессорных систем 49
Муренко Л. Л. — Отладочный комплекс на базе ПЭВМ типа ДВК для однокристальных микроЭВМ серий КМ1814, КМ1820, КМ1816 54
Тарасов В. В. — Организация передачи управления монитору и прикладным программам в отладочном устройстве 56
Сасов А. Ю. — Система цифровой обработки изображений на базе микроЭВМ, совместимых с IBM PC 58
Жуков А. А., Кузин В. Н., Панов М. Ю., Сомов А. В., Хмелевский А. М. — Микропроцессорная автоматизированная система учета рабочего времени 62
Богатырев В. А., Данилова Л. Д., Иванов Л. С., Каменев А. В., Куконин А. Ю., Охапкин А. В. — Устройство для построения многомашинных вычислительных систем на базе микроЭВМ «Электроника 60» 64
Богатырев В. А., Куконин А. Ю., Иванов Л. С. — Контроллер многомашинной вычислительной системы с общей магистралью 66
Бух-Винер П. В. — Сеть микроЭВМ в режимах реального времени . . . 68
Жданов В. И., Бобылев В. Н., Гринь Н. Ф., Уткина Т. Г. — Автономные обучающие устройства для ОЭВМ серии К1816 72
Тененбаум Я. П., Туркот А. И. — Организация вычислительного процесса в управляющем вычислительном комплексе на базе микроЭВМ «Электроника 60» 76
Вдовин С. Е., Волынчук В. Н., Зибров И. Н., Ковальчук В. Т. — Применение БИС К1802ВР2 для построения многоразрядных устройств деления . . . 79
Бокарев А. В., Кабанов А. И. — Адаптер-преобразователь адреса . . . 80
Громов Г. Р. — ЭВМ и информатизация общества: старые мифы и новые проблемы 85
Интерфейсные БИС микропроцессорного комплекта К1801: микросхема К1801ВП1-033 87

© Московский экспериментальный вычислительный центр «Элекс»

Главный редактор
А. П. ЕРШОВ
Редакционная коллегия:
А. Г. Алексенко
В. М. Брябрин
А. А. Васенков
(зам. главного редактора)
И. В. Вельбицкий
А. Б. Венгеров
Г. Р. Громов
(ответственный секретарь)
В. П. Иванников
М. Б. Игнатьев, 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. Шаруненко
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. Прангишвили
Л. Н. Преснухин
В. В. Пржиялковский
Н. Л. Прохоров, Г. Г. Рябов
В. И. Хохлов
Н. Н. Шереметьевский
В. В. Шильдин, А. В. Яковлев
Э. А. Якубайтис
Номер подготовили:
Е. И. Бабич, Г. Г. Глушкова,
В. М. Ларионова,
С. С. Матвеев
Кор. Е. М. Кучерявенко
Технический редактор
Г. И. Колосова
Адрес редакции журнала:
103051, Москва, Малый Сухаревский пер., д. 9А
Телефон: 208-73-23

Сдано в набор 30.07.88. Т-18568.
Подписано к печати 31.10.88.
Формат 84×108 1/16.
Высокая печать.
Усл. печ. л. 10,08. Уч.-изд.л. 14,6. Тираж 100.650.
Заказ 185. Цена 1 р. 10 к.
Орган Государственного комитета СССР по вычислительной технике и информатике
Московская типография № 13 ПО «Периодика» ВО «Союзполиграфпром» Госкомиздата СССР
107005, Москва, Денисовский пер., дом 30.

На первой странице обложки: микропроцессорные модули, входящие в набор комплекса МП-средств медицинской техники. (См. ст. Мамджяна Г. Г. и др. в № 6, 88)

ЗАКОН О КООПЕРАЦИИ И МАЛЫЕ ИННОВАЦИОННЫЕ ФИРМЫ

Нередко мы слышим сетования руководителей отраслевой и академической науки о том, как им трудно преодолевать многочисленные ведомственные барьеры на пути к внедрению результатов ПЛАНОВЫХ научных исследований в практику. Что же тогда говорить о проблемах отдельного инженера — электронщика или инициативной группы программистов в том «неплановом» случае, когда их техническая идея не только не завоевала еще сторонников в академических кругах, но и более того, рассматривается крупнейшими учеными и руководящими специалистами отрасли как очевидная ересь? Причем, это отнюдь не обязательно отрицательно характеризует какую-либо из драматически резко несогласных между собой сторон, научного конфликта. К примеру, один из наиболее авторитетных в США экспертов в области вычислительной техники Ф. Брукс (автор широко известного, в том числе и в нашей стране, программистского бестселлера «Мифический человеко-месяц») в начале 80-х годов энергично доказывал, что «влияние персональных компьютеров на развитие вычислительной техники будет нулевым или отрицательным»...

Понятно, что ученые и руководители промышленности, как и люди иных профессий, могут заблуждаться, причем, увы, в области своей основной профессиональной компетенции тоже. Это очевидное обстоятельство сегодня становится одним из критически важных для понимания сути общей задачи создания модели инновационного механизма, обостренно чуткого к нововведениям, полностью свободного от пут проржавевшей командно-бюрократической системы.

Есть основания предполагать, что принципиально новые возможности для независимого от тирании авторитетов и чинов творческого самовыражения открывает для тысяч независимо действующих инициативных бригад инженеров и программистов Закон о кооперации. В инновационном механизме страны в этих условиях может сложиться организационно гибкая структура «малых форм», которая позволит вскрыть для практического использования в народном хозяйстве мощный пласт индивидуально генерируемых «неплановых» идей и технических решений в области информационной технологии.

При этом, однако, возникает широкий спектр сложных организационных вопросов, требующих неотложного решения, среди которых безусловно первый — как обеспечить эффективное сопряжение государственного и кооперативного секторов инновационного механизма отрасли? Может ли отрасль эффективно и взаимовыгодно сотрудничать с большим числом независимо от нее существующих малых консультативно-исследовательских и производственных организаций? Причем сотрудничать так, чтобы не задушить их в дружеских объятиях, например, при попытке силовым образом «помочь» им различными инструкциями и нормативной документацией на виды и формы дозволенных действий или какими-то другими директивно навязываемыми «услугами».

Чтобы более полно понять и практически в реальной работе испытать «изнутри» социально-экономические проблемы и творческие возможности кооперативной «фирмы» организаторы и авторский актив постоянно действующего семинара журнала МП создали осенью 1988 года консультативно-исследовательский кооператив «РАТЕКС» (см. МП, 1988, № 4, с. 92). В одном из ближайших номеров журнала мы надеемся рассказать читателям в чем заключаются первые социально-экономические проблемы роста таких организаций, какие исследовательские, производственные и коммерческие проекты могут быть, с нашей точки зрения, наиболее эффективно реализованы уже на первом этапе становления такого типа организации. Приглашаем всех участников самых разнообразных компьютерных кооперативов, созданных в Москве и других городах страны, а также их ассоциаций (там, где они уже созданы) принять активное участие в этом обсуждении.

Выбор пути ускоренного развития инновационного механизма отраслей информационной технологии — многосложная разноплановая задача, центр тяжести которой находится не в научно-технической, а в социально-экономической области. Создать эффективно действующий хозяйственный механизм, в котором заблуждения (не важно, добросовестные или нет) любого уровня руководителей промышленности и самых крупных ученых отражаются лишь на их личной карьере, но не могут затормозить процесс внедрения в народное хозяйство ценных нововведений — это и значит принципиально, на уровне социально-экономической системы, а не личных качеств отдельных функционеров, решить задачу ускорения научно-технического прогресса в стране.

Г. Громов

УДК 621.328.3.049.776

В. С. Донев

МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКТ БИС СЕРИИ СМ600

Нарастающая изо дня в день интеграция в области приборостроения между социалистическими странами приводит к тому, что все чаще советским специалистам приходится сталкиваться с использованием зарубежных электронных компонентов и изделий на их основе. Недостающая во многих случаях информация о возможностях и способах применения БИС является препятствием для полноценного усвоения приобретенной аппаратуры. В этом плане настоящий материал — попытка ознакомить советских разработчиков микропроцессорных устройств с составом и возможностями выпускаемого в НРБ микропроцессорного комплекта БИС серии СМ600.

Л. Л. Утяков
доктор техн. наук, профессор

В настоящее время выпускается много комплектов микропроцессорных БИС. Каждый из них характеризуется определенными достоинствами и недостатками, и эффективность использования в основном зависит от их полного познания.

Рассматриваемый здесь набор больших интегральных схем — это развитие семейства 8-разрядных микропроцессоров и по своим функциональным возможностям очень похож на выпускаемый в СССР комплект К580. Отличия имеют второстепенное значение и относятся прежде всего к сигналам управления. В СМ600 учтены недостатки К580. Более упрощенная внутренняя структура микросхем, однако, оплачивается некоторой потерей в возможностях при программировании. В целом, системы, построенные на основе обоих комплектов, эквивалентны по производительности и внешней организации и, как показывает опыт разработчиков в НРБ, перейти с одной на другую не трудно.

Комплект СМ600 — универсальный 8-разрядный набор БИС, отличающийся гибкой организацией обмена данными с внешней средой. Комплект выполнен на базе N-канальной MOS-технологии с питанием от одного источника напряжением +5 В. Часть микросхем функционально обособлены, относительно самостоятельны и могут использоваться в составе систем, построенных на основе других микропроцессорных комплектов, например К580. Все входные и выходные сигналы комплекта БИС имеют стандартные ТТЛ-уровни.

В системе (рис. 1), построенной на основе БИС комплекта СМ600, весь обмен данными осуществляется с помощью трех магистралей:

адресной — 16 однонаправленных шин, исходящих из управляющего микропроцессора (МП). Все устройства в системе подключены равноправно к этой магистрали;

данных — 8 двунаправленных шин, по которым собственно осуществляется обмен данными. Управление магистрали предоставлено МП или (при определенных условиях) внешнему устройству;

управления — включает линии системных тактов Φ_1 и Φ_2, начальной установки $\overline{RES}$, доступного адреса VMA, управления чтением/записью R/$\overline{W}$, заявки на прерывание $\overline{IRQ}$, заявки на немаскируемое прерывание $\overline{NMI}$, остановки $\overline{HALT}$, управления третьего состояния шин TSC, разрешения шин данных DBE, доступности шин данных BA. Одна часть этих сигналов генерируется МП (VMA, BA, R/$\overline{W}$), а другая — служит для управления его работой (Φ_1, Φ_2, $\overline{RES}$, $\overline{IRQ}$, $\overline{NMI}$, $\overline{HALT}$, TSC, $\overline{DBE}$).

В состав комплекта входят: МП СМ601; интерфейсные схемы СМ602, СМ603, СМ604; вспомогательные микросхемы СМ605, СМ606.

СМ601 — монолитный, 8-бит МП универсального назначения. Система команд включает в себя 72 инструкции арифметических, логических и управляющих операций и полностью совместима с системой команд МП фирмы Motorola — MC6800. Максимальная частота системного такта 1 МГц. При этой частоте среднее время выполнения одной инструкции 2...6 мкс. Адресное пространство составляет 65536 ячеек памяти. Внутренняя структура и расположение выводов показаны на рис. 2 и 3. В СМ601 входят три 16-разрядных и три 8-разрядных программно доступных регистров (рис. 4). Требования к системным тактам Φ_1 и Φ_2 примерно

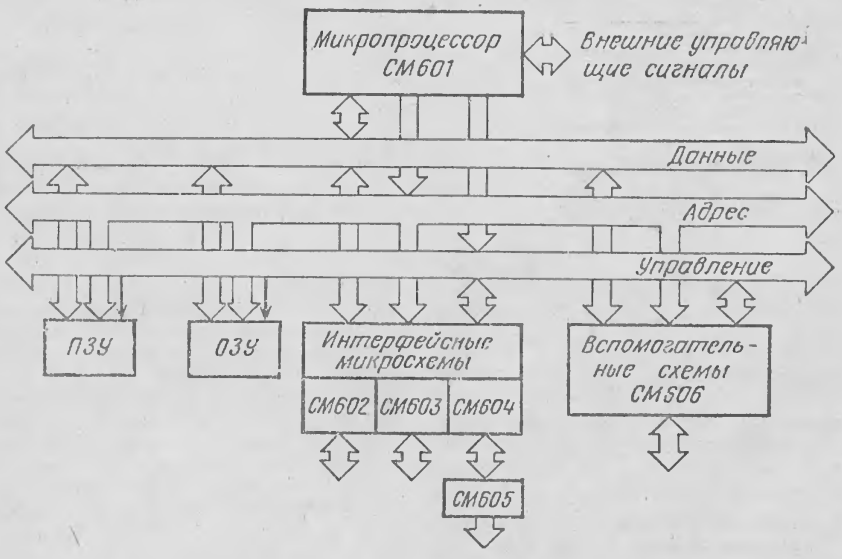


Рис. 1. Структурная схема системы на основе БИС СМ600

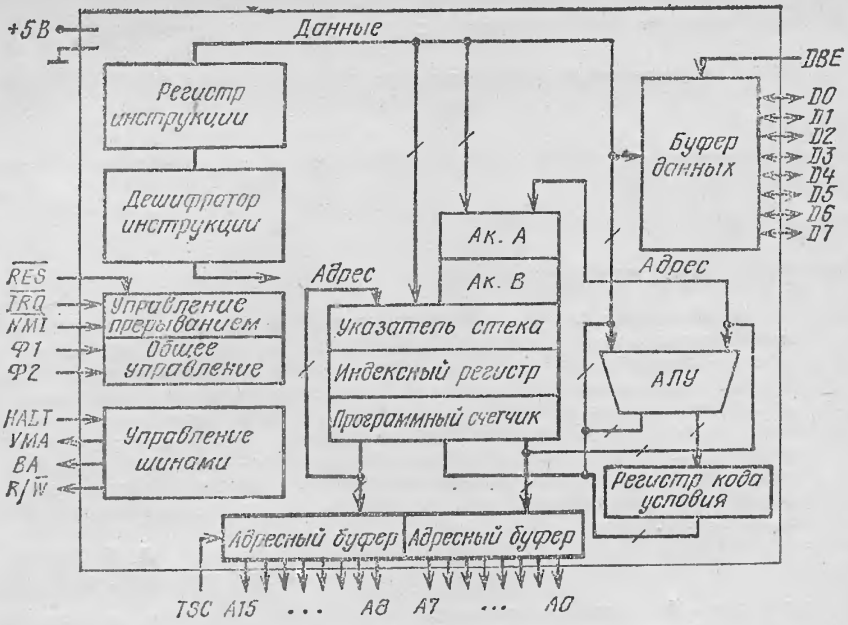


Рис. 2. Внутренняя структура СМ601

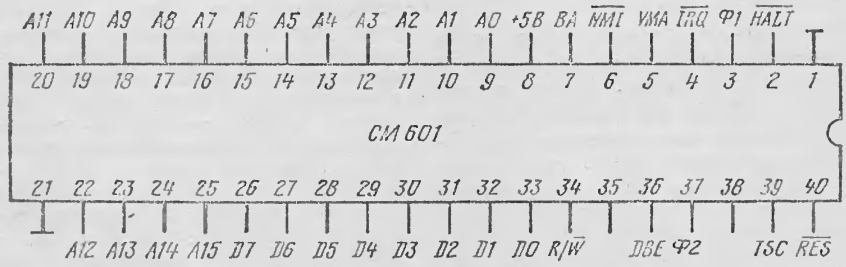


Рис. 3. Расположение выводов СМ601



Рис. 4. Внутренние регистры СМ601

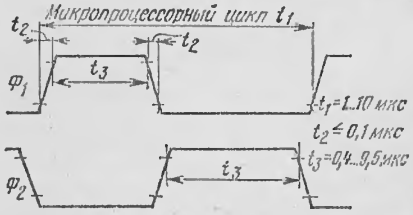


Рис. 5. Системные такты СМ601

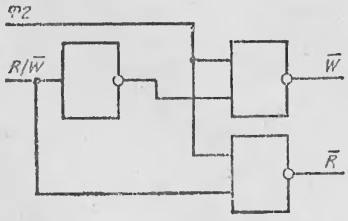


Рис. 6. Получение отдельных сигналов для чтения и записи

такие же, как и в комплекте К580, с той разницей, что пауза между тактами необязательна (рис. 5). Получение отдельных сигналов для чтения и записи от сигнала R/$\overline{W}$ возможно с помощью простой схемы (рис. 6).

Обмен информацией с внешними устройствами производится с помощью включенных в комплект интерфейсных БИС.

СМ602 — периферийный интерфейсный адаптер (PIA) для параллельного обмена данными с другими устройствами и системами (рис. 7). Обмен осуществляется по 16 (2×8) двунаправленным программируемым и четырем управляющим шинам (две с программируемым направлением). В состав микросхемы входят шесть программно доступных регистров. Регистры выбираются с помощью выводов CSO, CS1, $\overline{CS2}$, RS0, RS1, R/$\overline{W}$. Структура микросхемы состоит из двух частей (А и В), функционирование которых относительно самостоятельное. Выходы со стороны В имеют увеличенную нагрузочную способность: до 1 мА при 1,5 В.

СМ603 — асинхронный последовательный интерфейсный адаптер ($\overline{ASIA}$) для преобразования параллельного формата данных в последовательный и наоборот. Адаптер обеспечивает выполнение всех функций, связанных с синхронизацией и управлением последовательным обменом, максимальная скорость которого 500 Кбит/с. При параллельно-последовательном преобразовании автоматически формируется передаваемая комбинация битов, состоящая из старт-бита (0), семи или восьми информационных битов, контрольного бита четности или нечетности, 1; 1,5 или 2 стоп-битов [1]. При приеме последовательной информации автоматически проверяется ее правильность (для наличия бита четности-нечетности). Режим работы и формат обмениваемых последовательных слов находятся полностью под программным контролем. Управление и обмен осуществляются через четыре программно доступных регистра. Адаптер имеет три вывода для обеспечения совместной работы с модемом — $\overline{CTS}$, $\overline{RTS}$, $\overline{DCD}$ (рис. 8).

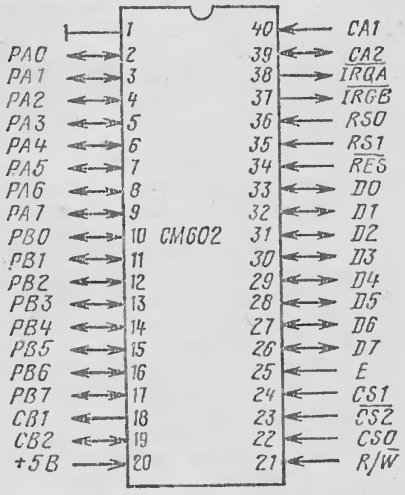


Рис. 7. Расположение выводов СМ602

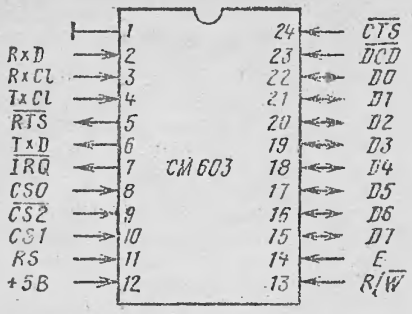


Рис. 8. Расположение выводов СМ603

СМ604 — синхронный последовательный интерфейсный адаптер (SSDA) для обмена данными. Адаптер обеспечивает выполнение всех операций по синхронизации и управлению обменом, максимальная скорость которого 600 Кбит/с. В процессе приема автоматически проверяется правильность данных. Состояние адаптера и процесс обмена контролируются через семь программно доступных регистров (рис. 9).

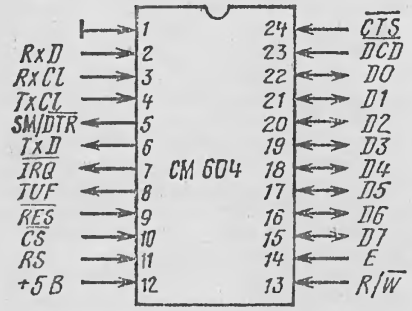


Рис. 9. Расположение выводов СМ604

СМ605 — среднескоростной цифровой модем для осуществления дифференциально-фазовой модуляции передаваемых последовательных данных (рис. 10). Имеет встроенный генератор 511-бит псевдослучайной контрольной последовательности. Модем работает на двух фиксированных скоростях — 1200 бод/с и 2400 бод/с.

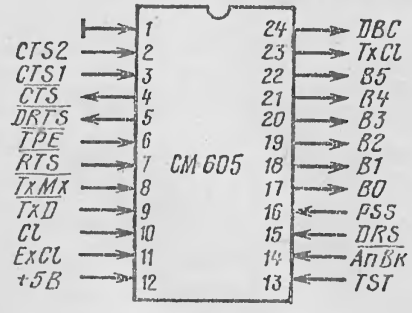


Рис. 10. Расположение выводов СМ605

Возможно самостоятельное использование в системах передачи данных.

СМ606 — программируемый таймерный модуль для генерирования и измерения интервалов (рис. 11). В состав микросхемы включены три независимых 16-разрядных таймера, все режимы которых задаются программно. Управление и контроль состояния каждого таймера производятся индивидуально с помощью отдельных регистров. Основные режимы работы отдельных таймеров: непрерывный, генерирование отдельного импульса, сравнивание частот, сравнивание интервалов.

Рассмотренные микросхемы комплекта СМ600 выпускаются серийно. Все корпусы керамические, типа DIL.

Поскольку настоящий материал имеет ознакомительный характер, автор с удовольствием предоставит интересующимся дополнительную информацию, относящуюся к особенностям отдельных микросхем, способам их программирования и включения в составе различных микропроцессорных систем.

Телефон 77-23-92, Варна-9000

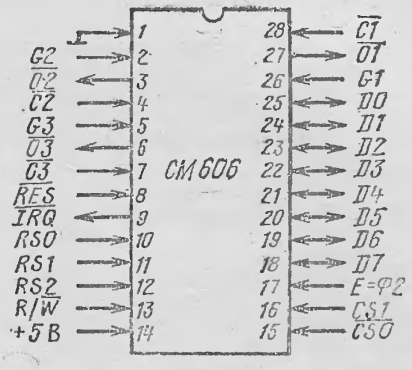


Рис. 11. Расположение выводов СМ606

ЛИТЕРАТУРА

1. Техническая информация/ НПСК по Полупроводниковой технике.— Ботевград.— 1986.
2. Станчев В. М., Бакърджиев Е. Микропроцессорная система СМ600 — описание, программирование, приложение.— София: Техника, 1984.— 264 с.

Статья поступила 11 марта 1988

УДК 681.323

М. Х. Гуревич, А. А. Кузнецов, И. Ю. Луцкий, Г. Ф. Страутманис

ОДНОКРИСТАЛЬНАЯ ЭВМ КМ1813ВЕ1 С АНАЛОГОВЫМИ УСТРОЙСТВАМИ ВВОДА-ВЫВОДА

Микросхема КМ1813ВЕ1 представляет собой однокристальную ЭВМ (ОЭВМ) со встроенными репрограммируемой памятью программ с ультрафиолетовым стиранием и аналоговыми устройствами ввода-вывода [1—3]. Набор команд ОЭВМ в сочетании с высокой точностью работы АЛУ (25 двоичных разрядов) позволяет строить на ее основе сложные системы цифровой обработки сигналов и цифрового динамического управления, промышленная автоматика, гидроакустика, геофизика, биомедицина, анализ и синтез речи.

Микросхема КМ1813ВЕ1 выполнена по n-канальной МДП-технологии с поликремниевыми затворами; выпускается в 28-выводном металлокерамическом корпусе типа 2121.28-6(7) с прозрачной для ультрафиолетового излучения крышкой и имеет следующие основные характеристики:

Способ синхронизации	От внешнего генератора или кварцевого резонатора
Разрядность АЛУ, бит	25
Емкость РПЗУ, бит	192×24
Емкость ОЗУ, бит	40×25
Память констант, бит	16×4
Возможность масштабирования	$2^2 \ldots 2^{-13}$
Устройства ввода-вывода	Аналоговые и цифровые
Число разрядов АЦП и ЦАП	9 (8+ знак)
Число каналов	
аналогового ввода	4
аналогового вывода	8
цифрового ввода	1

Возможность непосредственного цифрового ввода-вывода информации позволяет объединять микросхемы в сложные мультипроцессорные системы. Типовые функции, реализуемые ОЭВМ, связаны с обработкой сигналов (цифровые и адаптивные фильтры, корреляторы, модемы), обработкой и распознаванием речи (вокодеры, синтезаторы), реализацией устройств управления объектами и генерацией сигналов произвольной формы. Основные области применения ОЭВМ: низкочастотная связь,

Электрические параметры ОЭВМ приведены в табл. 1. Микросхема КМ1813ВЕ1 функционирует в двух режимах — «программирование» и «работа» (табл. 2). Условное графическое обозначение микросхемы показано на рис. 1.

Структурная организация. Программа, записанная в РПЗУ, управляет работой всех узлов микросхемы (рис. 2). По заданной программе микросхема отсчитывает значения аналогового или цифрового сигнала по одному из четырех входных каналов и

Таблица 1

Электрические параметры ОЭВМ КМ1813ВЕ1
(Т = —10 ... + 70° С)

Параметр, единица измерения	Норма	
	не менее	не более
Входное сопротивление, кОм, в режиме		
хранения	1000	
выборки		1
Напряжение входного аналогового сигнала, В	—2	+2
Нелинейность аналого-цифрового преобразования, дифференциальная нелинейность аналого-цифрового преобразования, нелинейность цифро-аналогового преобразования, МЗР	—1	+1
Дифференциальная нелинейность цифро-аналогового преобразования, МЗР	—0,5	+0,5
Коэффициент аналого-цифрового преобразования, В/В	0,98	1,02
Коэффициент цифро-аналогового преобразования, В/В	0,8	1,0 6,67
Тактовая частота, МГц	0,5	
Напряжение опорного источника, В	1	2
Напряжение источника питания, В		
U_{CC1}	4,75	5,25
U_{CC2}	—5,25	—4,75
Ток потребления, мА		
I_{CC1}		50
I_{CC2}		170
Число циклов перезаписи РПЗУ	20	

Таблица 2

Назначение выводов ОЭВМ КМ1813ВЕ1

Вывод	Обозначение	Назначение
	Программирование	
1, 26, 27, 28	D3...D0	Ввод-вывод 4-разрядного слова
4, 8, 12, 20	0В	Цифровая земля
17, 18, 25	5В	Напряжение питания
21	$\overline{RST}$	Сброс счетчика РПЗУ
22	$\overline{INCR}$	Наращивание счетчика РПЗУ
23	VSP	Программирующее напряжение
24	PROG	Запись-чтение данных
	Работа	
1, 2, 3, 5, 6	OUT3 ... OUT7	Аналоговые выводы
26, 27, 28	OUT0 ... OUT2	
4	OV⩘	Аналоговая земля
7,9	C1, C2	Подключение внешнего конденсатора
8	U_{REF}	Опорное напряжение
10	IN0/DI	Аналоговый вход — цифровой вход
11	IN3/$\overline{DO}$	Аналоговый вход — цифровой выход
12	U_{CC2}	Напряжение питания минус 5 В
13	IN2/$\overline{RD}$	Аналоговый вход-выход синхроимпульса «чтение»
14	IN1/$\overline{WR}$	Аналоговый вход-выход синхроимпульса «запись»
15, 16	CR1, CR2	Входы тактирования схемы
17, 20, 23	OV⩗	Цифровая земля
18	U_{CC1}	Напряжение питания 5 В
19	$\overline{CCLK}$	Выход синхронизации
21	$\overline{RST}$/$\overline{EOP}$	Сброс счетчика — конец программы
22	$\overline{OF}$	Выход переполнения АЛУ
24, 25	M2, M1	Режим работы входов-выходов

запоминает результат во входной схеме выборки и хранения. После АЦП цифровое значение входного сигнала поступает в процессор для дальнейшей обработки в соответствии с программой. Результат обработки передается на ЦАП. Аналоговое представление результата обработки через выходной демультиплексор и выходную схему выборки и хранения поступает на один из 8 выходных каналов. Специальный режим позволяет организовать обработку цифровых сигналов без использования АЦП и ЦАП.

Частота выборки аналогового сигнала определяется числом команд в программе и временем командного цикла (четыре такта тактового генератора). Конвейерный принцип построения ОЭВМ обеспечивает одновременную (параллельную) работу цифровой и аналоговой частей микросхемы, высокую производительность выполнения операций (около 2 млн. операций/с) и возможность обработки сигналов в реальном масштабе времени.

Цифровая часть микросхемы содержит РПЗУ, ОЗУ, двоичное масштабирующее устройство (МУ) и АЛУ. Аналоговая часть микросхемы включает 4-канальный входной аналоговый мультиплексор, схему выборки и хранения (одну на все четыре канала), компаратор, ЦАП, 8-канальный выходной демультиплексор, восемь выходных схем выборки и хранения. Связь между цифровой и аналоговой частями микросхемы осуществляется через специальный регистр DAR (цифро-аналоговый регистр).

Программная память КМ1813ВЕ1 представляет собой РПЗУ с ультрафиолетовым стиранием. В режиме «работа» РПЗУ организована как матрица 192×24-разрядных слов, в режиме «программирование» — 1152×4-разрядных слов D3, D2, D1, D0. Цикл «чтение — выполнение команд» построен по конвейерному принципу, глубина конвейера — четыре команды. Это означает, что четыре команды считываются из РПЗУ программ за время выполнения предыдущих четырех команд. После выполнения текущей команды содержимое счетчика

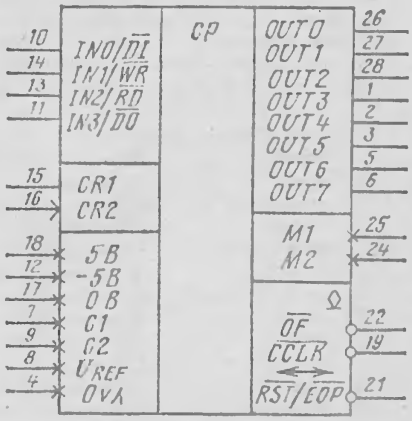


Рис. 1. Условное графическое обозначение ОЭВМ КМ1813ВЕ1

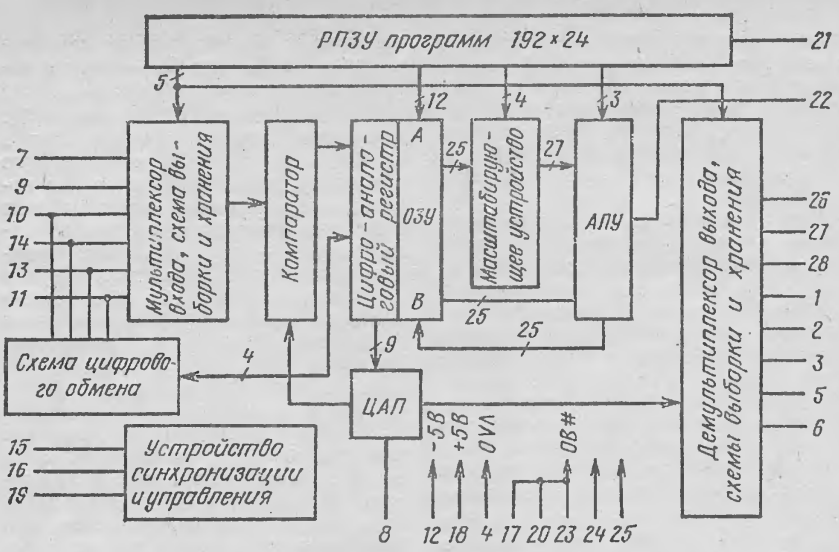


Рис. 2. Структурная схема ОЭВМ КМ1813ВЕ1

команд, предназначенного для формирования адреса, увеличивается на единицу. Оно может также изменяться скачкообразно на число, кратное четырем, при выполнении команд безусловного возврата на начало программы, условного возврата на начало программы и условного перехода на 32 команды вперед.

Командное слово длиной 24 разряда разбито на пять полей (рис. 3), одно из которых управляет работой аналоговой части микросхемы, а остальные четыре — цифровой (ОЗУ, МУ, АЛУ). Арифметическо-логическое устройство представляет собой параллельное 25-разрядное устройство, позволяющее выполнять арифметические и логические операции над данными, представленными в двоичном дополнительном коде с фиксированной запятой.

В состав АЛУ входят собственно АЛУ, схема коррекции результата при переполнении, флаг разрешения коррекции результата при переполнении, триггер хранения предыдущего переноса СУ, дешифратор команд. Вычисления выполняются в дополнительном коде с 25-разрядной точностью над масштабированным операндом А и операндом В, поступающим из ОЗУ. Старший разряд переменной является знаковым, а обычный диапазон переменной X находится в пределах $-1{,}0 \leqslant X < 1{,}0$. Минимальное приращение любой переменной составляет $5{,}96 \cdot 10^{-8}$. Устройство имеет схему коррекции результата для выполнения операций сдвига МУ и при переполнении результата в ходе арифметической операции. Схема коррекции в зависимости от кода команды, знаков операндов А и В, переноса в знаковый разряд и состояния 26-го и 27-го разрядов операнда А (после масштабирующего устройства) выдает сигнал переполнения (вывод $\overline{OF}$) и определяет результат вычисления в АЛУ.

Если флаг разрешения коррекции результата (устанавливается программно) включен, то результату выполнения команды присваивается предельное значение (—1,0, если переполнение вызвала отрицательная переменная; +1,0 — положительная переменная), а на выводе $\overline{OF}$ на время следующей команды устанавливается низкий уровень. Если коррекция результата запрещена, то по адресу В записываются 25 младших разрядов результата, а на выводе $\overline{OF}$ уровень остается высоким до тех пор, пока командами EOP и XOR CND не будет включен флаг разрешения коррекции. Выключается флаг командой ABA CND.

Таблица 3

Мнемокоды и значения констант

Мнемоника константы	Значение константы		Мнемоника константы	Значение константы	
	десятичное	двоичное		десятичное	двоичное
KP0	0	0,000	KM1	—0,125	1,111
KP1	+0,125	0,001	KM2	—0,250	1,110
KP2	+0,250	0,010	KM3	—0,375	1,101
KP3	+0,375	0,011	KM4	—0,500	1,100
KP4	+0,500	0,100	KM5	—0,625	1,011
KP5	+0,625	0,101	KM6	—0,750	1,010
KP6	+0,750	0,110	KM7	—0,875	1,001
KP7	+0,875	0,111	KM8	—1,000	1,000

Таблица 4

Мнемокоды управления масштабирующим устройством

Мнемокод сдвига	Код операции	Эквивалентный множитель	Мнемокод сдвига	Код операции	Эквивалентный множитель
L01	1110	$2^1=2{,}0$	R06	0101	$2^{-6}=0{,}015625$
L02	1101	$2^2=4{,}0$	R07	0110	$2^{-7}=0{,}0073125$
R00	1111	$2^0=1{,}0$	R08	0111	$2^{-8}=0{,}00390625$
R01	0000	$2^{-1}=0{,}5$	R09	1000	$2^{-9}=0{,}001953125$
R02	0001	$2^{-2}=0{,}25$	R10	1001	$2^{-10}=0{,}0009765625$
R03	0010	$2^{-3}=0{,}125$	R11	1010	$2^{-11}=0{,}00048828125$
R04	0011	$2^{-4}=0{,}0625$	R12	1011	$2^{-12}=0{,}000244140625$
R05	0100	$2^{-5}=0{,}03125$	R13	1100	$2^{-13}=0{,}0001220703125$

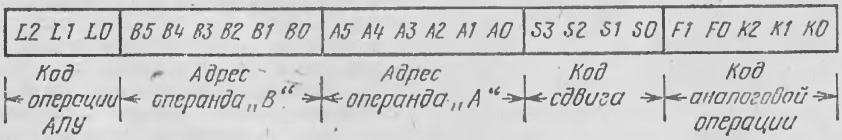


Рис. 3. Кодовое слово ОЭВМ КМ1813ВЕ1

Оперативное запоминающее устройство представляет собой двухадресное ОЗУ статического типа с произвольной выборкой и записью, организован-

ное из 40 слов×25 разрядов каждое, с двумя портами-регистрами для хранения операндов источника A и приемника B. Порт источника (порт операнда A) доступен только для чтения информации, которая через МУ передается на один из входов АЛУ. Через порт приемника (порт операнда B) информация поступает на второй вход АЛУ и принимаются результаты вычислений из АЛУ.

Каждый порт адресуется независимо через отдельные 6-разрядные поля командного слова. Расширенное поле адресов ОЗУ (64 вместо 40) используется для формирования констант и организации адреса регистра.

Память констант содержит 16 4-разрядных чисел, занимающих в поле адреса операнда A адреса 48 ... 63. Признаки обращения к памяти констант указываются в поле адреса IISXXX: II — признак обращения к константам, S — знак константы, XXX — значение константы (табл. 3).

Масштабирующее устройство представляет собой арифметический двоичный сдвигатель, расположенный между выходом порта операнда A ОЗУ и входом АЛУ. Данные в МУ могут сдвигаться влево и вправо, что позволяет масштабировать операнд A на любое значение, лежащее в пределах $2^2 \ldots 2^{-13}$ (сдвиг влево на два разряда, сдвиг вправо на 13 разрядов). При сдвиге влево освободившиеся правые разряды заполняются нулями, а старшие разряды числа A заносятся в два дополнительных разряда (26-й и 27-й). При сдвиге влево на один разряд значение 24-го разряда числа A заносится в 26-й разряд, а 27-й сохраняет значение знакового разряда.

При сдвиге вправо освободившиеся левые разряды заменяются значениями знакового разряда (дополнительный код). Такой сдвиг эквивалентен умножению операнда A на 2^k. Мнемокоды сдвига МУ и соответствующие значения эквивалентного множителя операнда A приведены в табл. 4.

Устройство управления и синхронизации предназначено для выработки сигналов, обеспечивающих выполнение команд. Оно включает генератор и формирователь сигналов управления и сигнала $\overline{CCLK}$. Генератор является резонансной схемой, работающей в диапазоне частот 0,5 ... 6,67 МГц. Выводы CR1 и CR2 используются для подключения внешнего кварцевого резонатора. На вывод CR2 можно подавать сигналы от внешнего генератора импульсов со скважностью Q = =2±5 %.

Формирователь сигналов управления осуществляет деление частоты генератора на четыре, выработку тактовых сигналов и сигнала $\overline{CCLK}$. Период следования сигнала $\overline{CCLK}$ соответствует четырем циклам команды и равен 16 T_c, где T_c — период колебаний сигналов генератора. Длительность сигнала $\overline{CCLK}$ равна 4 T_c.

Система команд, используемая в ОЭВМ, ориентирована на цифровую обработку сигналов. Одна группа команд предназначена для обработки цифровых данных (цифровые команды), другая — для аналоговых сигналов (аналоговые команды). Группа цифровых команд (табл. 5) содержит команды сложения ADD, вычитания SUB, пересылки LDA, логических операций XOR, AND, ограничения LIM, абсолютной величины ABS и сложения с абсолютной величиной ABA. Все перечисленные команды выполняются над операндами A и B. В качестве операндов используются данные из ОЗУ, задаваемые в двух полях адресов операндов, или константы, задаваемые только адресом операнда A.

Условные цифровые команды включают команды условного сложения ADD CND, вычитания SUB CND и пересылки LDA CND, которые используются при выполнении операций умножения и деления.

Команды переходов содержат команду безусловного возврата на начало программы EOP и команды условного возврата на начало программы RNZ и перехода на 32 команды вперед INZ.

Аналоговые команды ввода-вывода (табл. 6) предназначены для ввода-вывода аналоговых и цифровых данных через цифро-аналоговый регистр DAR. В ОЭВМ КМ1813ВЕ1 команды АЛУ и ввода-вывода могут выполняться одновременно.

Устройства ввода-вывода предназначены для обмена данными между ОЭВМ КМ1813ВЕ1 и ВУ. На них реализованы режимы ввода аналоговой информации через 9-разрядный АЦП (знак +8 разрядов), вывода аналоговой информации через 9-разрядный ЦАП (знак + 8 разрядов), вывода ТТЛ-уровней, последовательного ускоренного ввода-вывода цифровых данных.

В состав устройств ввода и вывода входят 4-канальный входной мультиплексор, схема выборки-хранения (СВХ) с внешним конденсатором хранения выборки, компаратор, цифро-аналоговый преобразователь (знак +8 разрядов), цифро-аналоговый регистр DAR и 8-канальный выходной демультиплексор со схемами выборки-хранения.

Аналого-цифровое преобразование аналогового сигнала в схеме выборки и хранения отсчета реализуется программным путем в соответствии с алгоритмом последовательного приближения с точностью, соответствующей девяти двоичным разрядам, старший из которых используется в качестве знакового. Результат АЦП формируется в регистре DAR, через который осуществля-

Таблица 5

Цифровые команды ОЭВМ КМ1813ВЕ1

Цифровые команды	L2, L1, L0	Операция	Примечание
		Условные команды	
ADD	110	$A\times2^k+B\longrightarrow B$	
SUB	101	$B-A\times2^k\longrightarrow B$	
DLA	111	$A\times2^k\longrightarrow B$	
XOR	000	$A\times2^k+B\longrightarrow B$	
ABS	011	$A\times2^k\longrightarrow B$	
ABA	100	$A\times2^k+B\longrightarrow B$	
LIM	010	$\pm1\longrightarrow B$	Если $A\geqslant0$, то «+1» Если $A<0$, то «—1»
AND	001	$A\times2^k\&B\longrightarrow B$	
		Команды перехода	
ADD CND(N)		$A\times2+B\longrightarrow B$ $B\longrightarrow B$	$DAR(N)=0$ $DAR(N)=0$
SUB CND(N)		$B-A\times2^k\longrightarrow B$ $CY_{i-1}\longrightarrow DAR(N)$	$CY_{i-1}=1$
.		$B+A\times2^k\longrightarrow B$ $CY_i\longrightarrow DAR(N)$	$CY_{i-1}=0$
LDA CND(N)		$A\times2^k\longrightarrow B$ $B\longrightarrow B$	$DAR(N)=1$ $DAR(N)=0$
RNZ CND(N)		$PC\leftarrow0$	$DAR(N)=1$
		$PC=PC+1$	$DAR(N)=0$
JNZ CND(N)		$PC=PC+32$ $PC=PC+1$	$DAR(N)=1$ $DAR(N)=0$

Примечания: 1. Операция $A\times2^k$ (k=2, 1, 0...—12, —13) реализуется с помощью масштабирующего устройства. 2. DAR(N) — анализируемый N-й разряд цифро-аналогового регистра, где N=5, 7, 6 ... 1, 0. 3. PC — счетчик команд. 4. CY_i — значение переноса из знакового разряда АЛУ.

Таблица 6

Аналоговые команды ОЭВМ КМ1813ВЕ1

Аналоговые команды	F0, F1	K2, K1, K0	Примечание
IN(Q) Q=0...3	00	0...3	Выборка сигнала со входа Q-го канала
OUT(P) P=0...7	10	0...7	Выход ЦАП на P-й канал
CVTS	00	6	Определение бита знакового разряда
CVT(K) K=0...7	01	0...7	АЦ-преобразователь K-го разряда
EOP	00	5	Конец программы
NOP	00	4	Нет операции
CND(K) K=0...7	11	0...7	Условная операция по биту K-го разряда
CNDS	00	7	Условная операция по биту знакового разряда

ется обмен с АЛУ и ОЗУ. Процесс АЦП выполняется по программе по командам CVT (N), где N=S, 7, 6 ... 1,0 — номера разрядов регистра DAR, S — знаковый разряд DAR (табл. 7).

Для установления необходимых временных соотношений в процессе АЦП используются команды NOP (табл. 8). Подключение входного канала производится по команде IN(Q), где Q=0,3 — номер входного канала. Сопротивление аналогового входа в открытом состоянии соответствует 1 кОм. Время установления сигнала на выходе мультиплексора входа должно быть равно примерно 6-кратному значению постоянной времени RC и составлять 4,5 ... 6 мкс. Это время определяет число команд IN(Q) для получения выборки сигнала с 9-разрядной точностью. Время установления ЦАП составляет 1,5 ... 1,8 мкс и определяет число команд NOP (не менее одной) перед командами CVT (N).

Цифро-аналоговое преобразование выполняется по методу поразрядного уравновешивания

Таблица 8

Команды устройства ввода-вывода

№ п/п	Команда АЛУ	B	A	Сдвиг	Аналоговая команда
1	LDA	DAR	Y	R00	NOP
2	.	.	.	.	NOP
3	.	.	.	.	NOP
4	.	.	.	.	NOP
5	.	.	.	.	OUT (P)
6	.	.	.	.	OUT (P)
7	.	.	.	.	OUT (P)
8	.	.	.	.	OUT (P)
9	.	.	.	.	OUT (P)
10	.	.	.	.	OUT (P)
11	.	.	.	.	OUT (P)

Таблица 7

Организация ввода-вывода аналоговой информации

№ п/п	Команда АЛУ	B	A	Аналоговая команда	№ п/п	Команда АЛУ	B	A	Аналоговая команда
1	LDA	DAR	KP0	IN(Q)	19	.	.	.	CVT5
2	.	.	.	IN(Q)	20	.	.	.	NOP
3	.	.	.	IN(Q)	21	.	.	.	NOP
4	.	.	.	IN(Q)	22	.	.	.	CVT4
5	.	.	.	IN(Q)	23	.	.	.	NOP
6	.	.	.	IN(Q)	24	.	.	.	NOP
7	.	.	.	IN(Q)	25	.	.	.	CVT3
8	.	.	.	IN(Q)	26	.	.	.	NOP
9	.	.	.	NOP	27	.	.	.	NOP
10	.	.	.	CVTS	28	.	.	.	CVT2
11	.	.	.	NOP	29	.	.	.	NOP
12	.	.	.	NOP	30	.	.	.	NOP
13	.	.	.	CVT7	31	.	.	.	CVT1
14	.	.	.	NOP	32	.	.	.	NOP
15	.	.	.	NOP	33	.	.	.	NOP
16	.	.	.	CVT6	34	.	.	.	CVT0
17	.	.	.	NOP					
18	.	.	.	NOP					

двоичного кода числа, находящегося в регистре DAR после обработки в процессоре. Результат ЦАП передается на один из восьми выходных каналов. Выбор выходного канала осуществляется программно по командам OUT (P), где P=76 ... 1,0 — номер выходного канала. Пределы выходного напряжения определяются значением опорного напряжения, умноженного на коэффициент ЦАП G_i (см. табл. 1). Типовое значение времени установления выходного сигнала составляет 4,2 ... 4,8 мкс и определяет число команд OUT (P), необходимых для получения выходного напряжения с 9-разрядной точностью.

Для организации вывода ТТЛ-уровней на любой из восьми выходных каналов опорное напряжение должно быть не менее 1,7 В. Режим вывода задается управляющими напряжениями на выводах M1 и M2 (табл. 9).

Режим последовательного ускоренного ввода-вывода цифровой информации осуществляется подачей напряжения 5В на вывод M1, а на вывод M2 — минус 5 В. В этом случае выводы IN0 ... IN3 переходят в режим последовательного ввода-вывода. К выводам IN2/RD, IN1/WR, IN3/DO необходимо подключить нагрузочные резисторы 10 кОм на 5 В. Ввод аналоговой информации невозможен.

Для организации **последовательного ускоренного ввода** по команде CVT (N), где N=S, 7, 6 ... 1,0 — номер разряда регистра DAR, в N-й разряд DAR инверсно (за исключением знакового разряда, который вводится в прямом коде) со входа IN0/DI вводится один бит информации. При вводе информации в N-й разряд DAR N — 1-й разряд всегда устанавливается в единицу. В течение команды CVT (N) на вывод IN2/RD выводится синхроимпульс низкого уровня.

Последовательный ускоренный вывод осуществляется по команде OUT (P), где P=0,1 ... 7, из P-го разряда DAR на вывод IN3/DO. Одновременно на вывод IN1/WR подается синхроимпульс низкого уровня. Порядок следования команд OUT (P) произвольный. Следует отметить, что во время команды OUT (P) на вывод OUT (P) подается сигнал, соответствующий находящемуся в DAR значению кода.

Рекомендации по применению. Основное назначение КМ1813ВЕ1 — цифровая обработка аналоговых (непрерывных) сигналов, базирующаяся на теореме отсчетов, согласно которой для достоверной обработки сигнала частота отсчетов должна быть в два раза больше максимальной частоты обрабатываемого сигнала. Теорема отсчетов в комплексе с параметрами КМ1813ВЕ1 устанавливает основные ограничения по применению данной микросхемы.

Частота отсчетов определяется длиной программы (числом командных

Таблица 9

Режим ввода-вывода информации

Напряжение на выводе М1, В	Напряжение на выводе М2, В	Режимы входов	Режимы выходов
5	5	Аналоговые	Аналоговые
5	—5	Последовательный ускоренный ввод-вывод	Аналоговые (OUT0 ... OUT3) Цифровые ТТЛ (OUT4 ... OUT7)
—5	5	Аналоговые	Цифровые ТТЛ (OUT0 ... OUT3) Аналоговые (OUT4 ... OUT7)
—5	—5	Аналоговые	Цифровые ТТЛ (OUT0 ... OUT7)

слов), записанной в РПЗУ, и временем цикла выполнения одной команды $f_s = 1/m \cdot t_s$, где f_s — частота взятия отсчетов; m — число командных слов в программе; t_s — время цикла. Максимально допустимая частота обрабатываемого непрерывного сигнала $f_{max} \leqslant f_s/2$.

Важным требованием является сохранение постоянной длины программы при взятии каждого последующего отсчета. Это требование необходимо соблюдать также при использовании команд RNZ и INZ.

Функциональная сложность узлов, выполняемых на КМ1813ВЕ1, определяется емкостью программной и оперативной памяти. Так, например, для реализации одного полюса элементарного цифрового фильтра требуется одно слово оперативной памяти, т. е. емкости ОЗУ хватит на 40 полюсов. Если учесть, что в самом простом случае на один цифровой фильтр 2-го порядка потребуется 10 команд ОЭВМ, то предельная эквивалентная функциональная сложность микросхемы характеризуется 19-ю звеньями цифрового фильтра 2-го порядка. Увеличить этот показатель можно, используя команды RNZ для фильтра с одинаковыми передаточными характеристиками звеньев.

Ниже приводятся примеры некоторых типовых программ КМ1813ВЕ1. **Программа умножения на константу.** Умножение выполняется с использованием операции сдвига. Константа-множитель представлена в канонической знакоразрядной системе счисления, число команд, необходимых для умножения, определяется при этом числом единиц в двоичном представлении константы и зависит от требуемой точности.

Сущность алгоритма проще всего пояснить на примере. Пусть переменная x умножается на константу C = =0,4726562, т. е. выполняется операция $y = C \cdot x$. В двоичном и знакоразрядном представлениях константа C имеет вид C=0,01111001 и 0,10001001. Последний код может быть представлен с помощью степеней двойки как $C = 2^{-1} - 2^{-5} + 2^{-8}$. Тогда операцию умножения запишем следующим образом:

$$y = x \cdot 2^{-1} - x \cdot 2^{-5} + x \cdot 2^{-8}.$$

Соответствующая этой записи программа

LDA	y	x	R01
SUB	y	x	R05
ADD	y	x	R08

Деление на константу. Деление переменной на константу выполняется умножением этой переменной на величину, обратную константе.

Умножение на переменную. Пусть требуется выполнить операцию y=xz, где x, z — переменные ($-1 \leqslant x, z < +1$). Точность представления множителя — восемь двоичных разрядов. Операция выполняется по командам условного сложения с поочередным анализом разрядов множителя. Если разряд имеет значение 1, то выполняется суммирование сдвинутого соответствующим образом множимого с частным произведением; если 0, то суммирование не выполняется. Программа умножения на переменную выглядит следующим образом:

LDA	DAR	x	R00	NOP	загрузка множителя в DAR
LDA	y	KP0	R00	NOP	очистка ячейки результата
ADD	y	z	R01	CND7	перемножение кодов
ADD	y	z	R02	CND6	
⋮	⋮	⋮	⋮	⋮	
ADD	y	z	R08	CND0	
SUB	z	z	L01	NOP	определение знака произведения
ADD	y	z	R00	CNDS	

Программа деления на переменную. Пусть требуется выполнить операцию y=x/z. Если обе переменные x и z могут принимать положительные и отрицательные значения (деление в этом случае называется 4-квадрантным), то, используя команду XOR, можно определить знак частного. Дальнейшие операции выполняются над абсолютными значениями переменных. Операция деления заключается в последовательном условном вычитании делителя из делимого, а частное накапливается в регистре DAR. Выполнение операции начинается с безусловного вычитания. Поскольку результат меньше единицы, то делитель предварительно масштабируется таким образом, чтобы наименьшее его значение всегда превышало наибольшее значение делимого. Тогда результат первого вычитания всегда будет отрицательным. Последовательность команд 4-квадрантного деления приведена ниже:

LDA	T	z	R13	NOP

— формирование знака частного

XOR	T	x	R13	NOP
ABS	x	x	R00	NOP

— выделение абсолютных значений переменных x, z

ABS	z	z	R00	NOP

— обнуление DAR

LDA	DAR	KP0	R00	NOP

— безусловное вычитание x

SAB	x	z	R00	NOP
SUB	x	z	R01	CND7
⋮	⋮	⋮	⋮	⋮
SUB	x	z	R08	CND0

— получение значащих цифр частного в DAR

XOR	DAR	T	R13	NOP

— восстановление знака результата

Результат деления, включающий восемь значащих разрядов и знак, запоминается в DAR. Если требуется большая точность, то содержимое DAR должно быть занесено в рабочую ячейку, затем DAR обнуляется и восстанавливается значение переноса CY_1. Далее продолжаются условные вычитания делителя с накоплением в DAR младших разрядов частного.

Телефон 52-01-64, Рига

ЛИТЕРАТУРА

1. Страутманис Г. Ф., Кузнецов А. А. Однокристальная микроЭВМ с аналоговыми устройствами ввода-вывода — новое средство цифровой обработки сигналов и управления. — Киев: Общество «Знание», 1986.
2. Цифровой процессор обработки сигналов с аналоговыми устройствами ввода-вывода/Под ред. А. А. Ланнэ. — Л.: ВАС, 1985.— С. 1—88.
3. А. С. № 1195364 МКИ³. Микропроцессор. Дычаковский В. Б., Кузнецов А. А., Ланнэ А. А. — Опубл. 1985, Бюл. № 44.

Статья поступила 31 марта 1988

УДК 681.327.8

В. А. Таланов

МИКРОЭВМ С МАРКОЙ ВЭФ

На Рижском производственном объединении (РПО) ВЭФ им. В. И. Ленина для выпуска квазиэлектронных автоматических телефонных станций, новейших телефонных аппаратов, популярных «Спидол», «Сигм» используют микроЭВМ и микропроцессоры на всех стадиях изготовления — от разработки до контроля готовой продукции.

В основании сложной системы взаимодействия компьютеров различных уровней находятся микропроцессоры и микроЭВМ, управляющие непосредственно процессами выпуска изделий, а на верхнем уровне системы — мощная центральная ЭВМ, координирующая производство в целом.

В РПО ВЭФ был разработан и внедрен универсальный микропроцессорный комплекс аппаратных и программных средств для построения технологических, социально-бытовых, медицинских и спортивных информационно-управляющих систем.

Сейчас в РПО ВЭФ создаются и внедряются новые образцы вычислительной техники для дальнейшего повышения эффективности производства и качества продукции.

В распределенных системах управления, автономных контрольно-измерительных устройствах, системах отладки и в качестве персональных ЭВМ широко применяются компактные настольные микроЭВМ семейства «ВЭФ-1024» (см. таблицу).

Семейство микроЭВМ „ВЭФ-1024"

Модель	Устройство ввода-вывода	Число посадочных мест	Габаритные размеры, мм
„ВЭФ-микро-1024"	ЭЛТ с зеленым свечением (31 см по диагонали); два устройства НГМД ЕС5088	14	480×490×265
„ВЭФ-микро-1024А"	ЭЛТ (23 см по диагонали); кассетный накопитель (блок магнитофона от магнитолы „Сигма")	11	490×420×295
„ВЭФ-микро-1024Ц"		6	410×310×140
„ВЭФ-микро-1024М"	Два устройства НГМД ЕС5088	5	350×290×255

МикроЭВМ «ВЭФ-микро-1024» содержит БИС серии КР580ИК80А, ПЗУ — 16 Кбайт, ОЗУ — 64 Кбайт, интерфейс бытового магнитофона, параллельные порты для клавиатуры и печатающего устройства, блок обработки прерываний, порты ИРПС/С2, таймер, генератор звукового сигнала, адаптер видеоконтрольного устройства — дополнительное ОЗУ объемом 4 Кбайта со схемой регенерации растрового монохромного или цветного изображения на специальном мониторе или бытовом телевизоре. Адаптер обеспечивает текстовый и графический режимы работы. Формат изображения 80 символов, 25 строк или две страницы (40 символов, 25 строк). Матрица 8×8 содержит 224 символа: прописные и строчные буквы русского и латинского алфавитов, знаки, цифры и спецсимволы, в том числе для псевдографики. Для каждого знакоместа изображения устанавливается набор атрибутов: цвет символа и цвет фона — всего восемь цветов или градаций яркости, мерцание. В графическом режиме разрешающая способность — 160×100 четырехцветных элементов изображения.

Текстовая клавиатура на 86 клавишах вырабатывает 256 символов в коде КОИ-8 и сигналы управления ЭВМ. Адаптеры обеспечивают использование НГМД «Электроника ГМД 70» (7012) или четырех устройств ЕС5074 и ЕС5088. Для расширения возможностей по сбору и накоплению используется модель передачи данных: восемь портов ИРПС/С2, адаптер накопителя на магнитной ленте (два НМЛ СМ5300.01), дополнительное ОЗУ (до восьми блоков объемом по 64 Кбайт).

Математическое обеспечение микроЭВМ стандартного типа содержит ДОС, ОС реального времени, интерпретаторы языков высокого уровня, резидентный ассемблер и редакторы текста.

Универсальная микроЭВМ «ВЭФ-микро-1025» — управляющая ЭВМ для обработки и передачи информации в системах ГАП и АСУ. Она предназначена для широкого применения в качестве базовой аппаратуры блочной конструкции открытой модульной архитектуры.

Архитектура микроЭВМ использует принцип распределения обработки данных и вычислительной мощности между микропроцессорами (рис. 1).

Модуль центрального интегрированного процессора осуществляет логические и арифметические операции, генерирует команды для управления внешней памятью и портами ввода-вывода внешних устройств, система команд определяется типом микропроцессора (БИС КР580ИК80А). ОЗУ состоит из двух модулей динамической оперативной памяти, построенных на микросхемах КР565 (общей емкостью 64 Кбайт). Генератор формирования сигналов работает с тактовой частотой 18 МГц, что позволяет процессору обращаться к памяти без цикла ожидания.

Модуль контроллера магнитной памяти (МКМП) работает под управлением собственного микропроцессора КР580ИК80А и модуля интерфейса (МИ) диска. Накопитель на гибких магнитных дисках имеет два дисковода и модуль сопряжения, формирующий под управлением контроллера НГМД сигналы выбора соответствующего дисковода и работы головки записи-чтения.

Модуль контроллера видеотерминала (МКВ) управляет интерфейсом монитора, читает коды, выданные клавиатурой оператора, обеспечивает связь с ЦП через порт последовательного ввода-вывода. Программа дисплея хранится в ПЗУ объемом 6 Кбайт. Модуль интерфейса видеомонитора (МИВ) предназначен для вывода алфавитно-цифровой и псевдографической информации на экран. С его помощью информация об отображаемых символах и регенерация изображения записывается и хранится в ОЗУ, вырабатываются кадровые и строчные синхроимпульсы. Для отображения знаков применяется система телевизионной развертки.

Основное поле клавиатуры имеет 112 клавиш, распределенных на функциональные группы: основные алфавитно-цифровые, редактирования, установки режима видеотерминала, шестнадцатеричных цифр, функциональные, и может быть дополнено через разъем входа.

К микроЭВМ можно подключать перфоратор, считыватель с перфоленты, накопитель на магнитной ленте, телетайп, широкоформатное печатающее устройство и пр.

Технические характеристики микроЭВМ „ВЭФ-микро-1025"

Объем памяти, Кбайт	64
Разрядность шин, байт	8
Размер экрана по диагонали, см	31
Формат изображения, символы	80×24
Потребляемая мощность, В·А	500
Габаритные размеры, мм (масса, кг)	
микроЭВМ ВЭФ-микро-1025»	595×320×275 (20)
автономный дисплей «ВЭФ-микро-7025»	505×320×275 (15)
клавиатура «ВЭФ-микро-6025»	255×175×40 (1)
НГМД «ВЭФ-микро-5125»	340×410×360 (25)

На основе микроЭВМ «ВЭФ-микро-1025» возможно создание многомашинных комплексов со сложной вы-

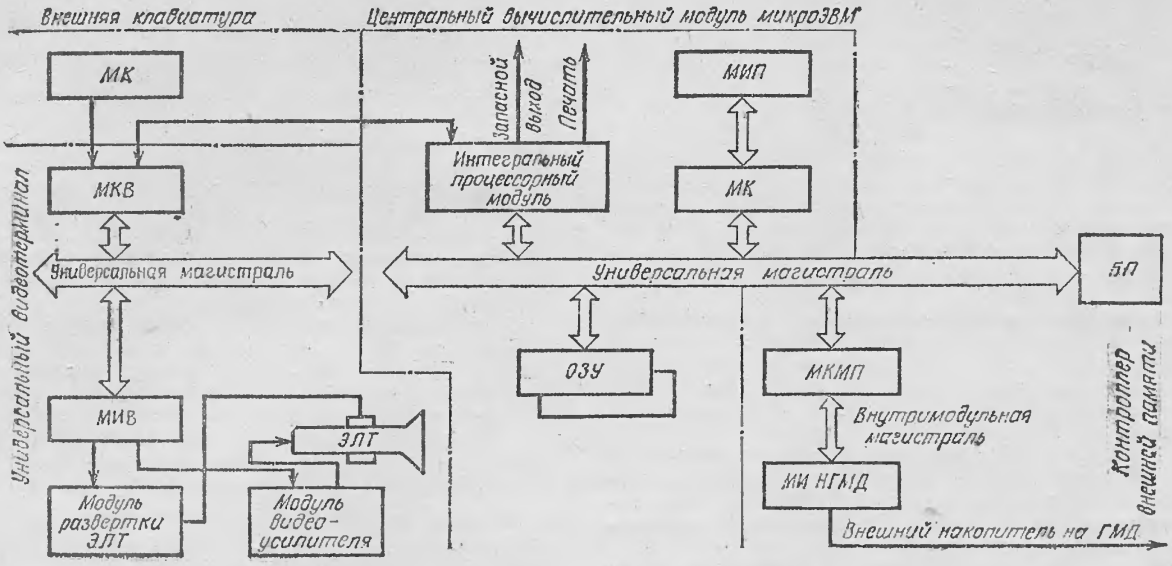


Рис. 1. Структурная схема «ВЭФ-микро-1025»

числительной структурой. Она может подключаться к промышленным локальным сетям и быть сопряжена со множеством периферийных устройств. МикроЭВМ защищена тремя авторскими свидетельствами.

В последних образцах «ВЭФ-микро-1025» удалось объединить видеотерминал и микроЭВМ в единый корпус.

В качестве базовой аппаратуры микроЭВМ «ВЭФ-микро-1025» была использована при создании микропроцессорной системы «ВЭФ-микро-МКС-803/25» для центров управления модулями и комплексами гибких производств, автоматизации конторских работ, организации банков данных. Она с успехом применяется для отладки при разработке программных микропроцессорных средств, редактирования и трансляции программ в процессе проектирования микропроцессорных комплексов, интеллектуальных терминалов, контроллеров, измерительных приборов, конструируемых на базе БИС.

Система имеет автономный или встроенный дисплей, накопитель на гибких магнитных дисках «ВЭФ-микро-5325» и последовательное широкоформатное печатающее устройство. Внешние накопители подключаются с помощью контроллера НГМД с двумя физическими каналами. К каждому из них могут одновременно подключаться два разных дисковода. Например, к первому каналу — два «ВЭФ-микро-5325», ко второму — два «ВЭФ-микро-5125». Максимальный объем памяти при таких накопителях — 320 Кбайт. Расширение памяти возможно с помощью более мощного НГМД «ВЭФ-микро-5074» (до 1 Мбайт).

Большое внимание на РПО ВЭФ уделяется контролю качества выпускаемой продукции. С помощью системы контроля параметров радиоприемников «ВЭФ-ПАРАМЕТР» (см. вкладку) в условиях массового производства осуществляется многосторонняя проверка изготавливаемых изделий (программа оценки записана жестко). Проверяются диапазоны принимаемых частот, чувствительность, ограниченная усилением и шумами, односигнальная избирательность по зеркальному и соседнему каналам.

Производительность «ВЭФ-ПАРАМЕТР» не менее 120 радиоприемников за смену, то есть на проверку каждого затрачивается около 4 мин, результаты контроля фиксируются автоматически и выдаются распечатанными.

Технические характеристики микропроцессорной системы „ВЭФ-микро-МКС-803/25“

Объем памяти, Кбайт	64
Объем ППЗУ, Кбайт	512
Размер экрана по диагонали, см	31
Формат изображения, символы	80×24
Потребляемая мощность, В·А	250
Габаритные размеры, мм	
микроЭВМ со встроенным дисплеем	601×450×322
клавиатура	495×191×322
НГМД «ВЭФ-микро-5325»	461×400×160
Масса общая, кг	60

Технические характеристики системы контроля „ВЭФ-ПАРАМЕТР“

Погрешность измерения, %	
в КВ-диапазоне	20
в ДВ-и СВ-диапазонах	30
Потребляемая мощность, В·А	500
Габаритные размеры, мм	1690×805×910
Масса, кг	300

В объединении действует система СОТ-1М — автоматизированная система оценки качества труда и продукции членов бригады рабочих и контролеров ОТК на производственном участке. Она предназначена для сбора и обработки оперативной информации, характеризующей качественные и количественные показатели бригады, ведет автоматизированный учет продукции, принятой с первого предъявления, а также бракованной, несоответствующей требованиям ТУ, классификацию по видам дефектов продукции.

СОТ-1М обеспечивает выполнение директив, направленных с диалогового устройства, обменивается информацией с пультами контролеров ОТК, обрабатывает и накапливает в памяти результаты, оценивает труд и, фиксируя лучших, выдает данные для составления отчетов.

Системы оценки труда СОТ-1М обеспечивают непрерывный учет и дифференцированную оценку качества выполняемой производственной операции.

Технические средства системы СОТ-1М: управляющая микроЭВМ, канальный процессор, пульт ввода цифровой информации с рабочих мест, система «ВЭФ-ИНФОРМ» с информационным табло и собственной микроЭВМ, внешний накопитель информации, устройство печати, дисплей.

Применение микропроцессорной техники непосредственно на производстве дает значительный экономический эффект — несколько десятков миллионов рублей по объединению.

(Окончание см. на с. 26)

УДК 681.32.06

С. Н. Домарацкий, И. Л. Шраго

ОРГАНИЗАЦИЯ ЭКРАНА МОНИТОРА ПЭВМ «ИСКРА 1030» В ПРОГРАММАХ АВТОМАТИЗАЦИИ НАУЧНЫХ ИССЛЕДОВАНИЙ

При разработке программ автоформализации профессиональных знаний [1] для систем автоматизации научных исследований (АНИ) на базе ПЭВМ, реализующих эвристические алгоритмы, повышенное внимание следует обращать на организацию интерактивного взаимодействия человека с системой [2]. Работающие «один на один» с системой пользователи не являются, как правило, профессионалами в области обработки данных и заинтересованы в возможно большей простоте работы. Они рассматривают экран монитора как основной индикатор, а клавиатуру ПЭВМ — как орган управления прибором. Поэтому способ общения даже «гениальной» программы с пользователем во многом определяет ее распространение в потенциальной предметной области.

Появление отечественных ПЭВМ «Искра 1030.11» [3] и ЕС 1840 (исполнение 5) с графическим дисплеем и передачей цветов полутонами открывает широкие возможности по организации диалогового интерфейса, приспособленного к «человеческим» методам решения задач из конкретной предметной области.

Целью настоящей статьи является описание подпрограмм управления монитором, а также опыта авторов по разработке программ ПЭВМ «Искра 1030.11» и ее аналогов для ряда систем АНИ. Авторы надеются, что это хотя бы отчасти будет способствовать процессу автоформализации [1] или, по крайней мере, поможет заинтересованным читателям сэкономить время на разработку собственных программ.

Обобщенная структура иерархии выбора альтернатив при интерактивном взаимодействии пользователя с интегрированным пакетом АНИ в абстрактной предметной области приведена на рис 1. Число уровней иерархии может быть и бо́льшим, но показанные на рисунке устойчиво присутствуют в пакетах программ ПЭВМ для АНИ.

Авторам известны интерфейсы пользователя — около ста пакетов программ ПЭВМ для различных предметных областей, в которых можно выделить три основных способа управления задачей посредством:

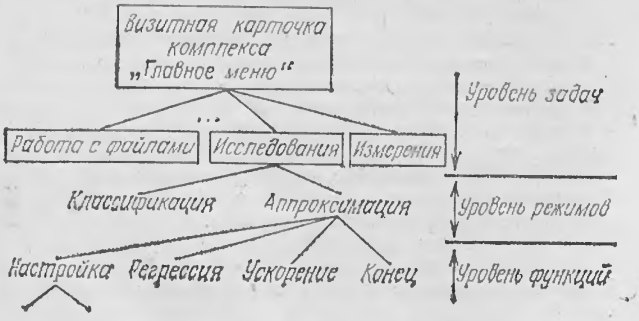


Рис. 1. Типовая структура интегрированных пакетов автоматизации научных исследований

ввода с клавиатуры ПЭВМ команд с определенными мнемониками и синтаксисом. Это характерно для входных языков оболочек операционных систем, например PC DOS, АДОС [3, 4], систем управления базами данных, а также интерпретирующих систем (БЕЙСИК, ДИАСП и т. п.), для режимов управления изображением в пакетах САПР (например, AutoCAD фирмы Autodesk, США);

ввода с клавиатуры ЭВМ кодов символов или функциональных клавиш, соответствующих альтернативным вариантам. Этот способ чаще встречается в режимах конфигурации задач и на самом нижнем уровне иерархии выбора (например, в пакете сервисных программ в [4]);

изменения координат элемента слежения (выделенного цветом, рисунком, яркостью участка экрана), каждое положение которого соответствует одному из альтернативных вариантов. Элемент слежения управляется одним из типов микроманипуляторов или клавишами управления курсором. Сам элемент может быть определен строкой текста (словом), пиктограммой и т. д. Вариации этого способа с горизонтальной, вертикальной или табличной формой задания меню выбора альтернативных вариантов применяются в большинстве распространенных пакетов программ для ПЭВМ. При этом само меню находится на экране постоянно либо появляется по мере надобности и восстанавливает экран после выбора альтернативы. Есть и более удобный для восприятия, но сложный в смысле программирования способ, применяемый в пакетах типа Microsoft Windows. Однако для использования его преимуществ необходим микроманипулятор типа мышь.

Большое значение для восприятия пакета человеком имеет хорошо продуманная текстовая подсказка, вызываемая пользователем на экран.

В разрабатываемых для ПЭВМ интегрированных пакетах программ АНИ с учетом известного психологического ограничителя на одновременное восприятие источников информации 7 ± 2 [5] и собственного опыта авторы пришли к обобщенной модели экрана монитора с независимыми окнами (рис. 2).



Рис. 2. Удобное для восприятия распределение экрана монитора ПЭВМ

В системах АНИ на базе отечественных ПЭВМ управление интегрированным пакетом программ осуществляется изменением координат элемента слежения. В отличие от 2-го способа этот способ эффективно опирается не на запоминание цепочек управляющих последовательностей или функциональных клавиш, а на врожденные ориентационные способности и память человека, что существенно ускоряет его работу в системе АНИ. Сведения о динамике процесса в символьной или графической форме, на наш взгляд, целесообразно выводить в больших окнах в средней части экрана. При этом слева лучше воспринимаются сведения о текущем шаге по программе или бегущие значения параметров, характеризующих ее состояние, а справа — контролируемые значения параметров процесса (регулируемые величины, погрешности).

Однако вывод более трех бегущих значений в одном окне приводит к быстрой утомляемости пользователя. Необходимость демонстрации динамики контролируемого процесса удовлетворяется выводом в большом окне в середине экрана графической интерпретации параметров в виде гистограмм, годографов или мультипликации, отображающей смысл происходящего с информационными структурами задачи в представлении пользователя (например, изображение «живого» поперечного разреза печи, геомагнитного поля или табло приборов). Верхние окна желательно оставить для вывода информации о месте процесса на графе (см. рис. 1), а в нижнем окне удобно организовать меню выбора альтернатив воздействия на текущий процесс. Само меню можно организовать в виде строки слов — логотипов, отражающих смысл предлагаемых альтернатив. Элемент слежения для восприятия удобно отмечать обратной засветкой, а управлять им — клавишами управления курсором. Одновременно с перемещением по строке меню элемента слежения должна меняться строка быстрой подсказки, которая, например, дает расширенное толкование логотипа меню. В ПЭВМ «Искра 1030.11» окна можно выделять как рамками из символов псевдографики, так и тонированием площади окна путем изменения атрибута цвета фона.

На рис. 3 в качестве примера использования описанной модели приведен вид экрана ПЭВМ «Искра 1030.11» для режима аппроксимации функциональных зависимостей (см. рис. 1) разработанного авторами интегрированного пакета программ автоформализации знаний в области рентгеноспектрального анализа [6].

Прикладные пакеты программ АНИ в различных областях знаний чаще разрабатываются на языках высокого уровня (Фортран, Паскаль, Си). Поэтому процедуры организации вывода на экран монитора должны удовлетворять общему для процессоров всех перечисленных языков межмодульному интерфейсу.



Рис. 3. Заполнение экрана монитора ПЭВМ, выполняемое процедурой SAMPLE (листинг 12)

В зарубежных пакетах подпрограмм для компиляторов языков высокого уровня, реализующих различные функции управления монитором и клавиатурой, часто требуется включение в прикладные программы или в оболочки ОС оригинальных мониторов или драйверов виртуальных устройств. Не говоря уже о доступности рядовому пользователю зарубежных пакетов, их совместимость с отечественными ПЭВМ требует в каждом случае проверки, так как, кроме прочего, не совпадают кодовые таблицы (в переменной части) [7] и внутреннее соответствие кодов клавиш и символов.

Во избежание лицензионных трудностей и для упрощения задачи программирования собственных диалоговых интерфейсов авторы предлагают читателю набор своих подпрограмм организации экрана монитора на языке ассемблера ПЭВМ «Искра 1030.11» с обращением к базовой системе ввода-вывода в ПЗУ. Эти подпрограммы не требуют никаких мониторов и связываются редактором связей с объектным модулем вызывающей их прикладной программы. Предлагаемый набор подпрограмм — лишь база, которую можно применять или наращивать по образу и подобию. Большинство подпрограмм можно использовать и на совместимых с «Искра 1030.11» отечественных и зарубежных ПЭВМ. Межмодульный интерфейс ориентирован на передачу адресов параметров. Это единственно доступный интерфейс для пользователей Фортрана, он реализуется и компиляторами других языков. В дальнейшем будем полагать для простоты, что прикладная программа написана на Фортране. Тело процедур применимо и для ассемблер-программ. В этом случае отпадает необходимость в передаче адресов параметров.

Сводка процедур управления экраном монитора ПЭВМ «ИСКРА 1030.11»

1. Установка режима работы монитора (листинг 1)

call SETMOD (nummode),

где nummode — номер режима:
- =0 — текстовый 40×25 — черно-белый
- =1 — текстовый 40×25 — цветной
- =2 — текстовый 80×25 — черно-белый
- =3 — текстовый 80×25 — цветной
- =4 — графический 320×200 — цветной
- =5 — графический 320×200 — черно-белый
- =6 — графический 640×200 — черно-белый
- =7 — монохромный одностраничный.

2. Смена активной страницы (листинг 2)

call NEWPAG (numpage),

где numpage — номер выбираемой страницы:
- =0 ... 3 — при работе в режимах 80×25
- =0 ... 7 — при работе в режимах 40×25;

процедура неприменима при работах с монохромным алфавитно-цифровым монитором ЕС 1840 и в графических режимах.

3. Чтение текущего состояния монитора (листинг 3)

call VIDEOT (numpage, nummode, length).

Процедура возвращает текущие значения параметров:
- numpage — номер активной страницы монитора;
- nummode — номер режима работы монитора;
- length — длина строки монитора в символах.

4. Чтение координат курсора на заданной странице монитора (листинг 4)

call CURPOS (numpage, numrow, numcolumn).

Процедура возвращает значения номера строки и столбца текущего положения курсора:
- numrow — номер строки;
- numcolumn — номер столбца.

5. Позиционирование курсора заданной страницы (листинг 5)

call MOVCUR (numpage, numrow, numcolumn).

6. Процедура создания и сдвига содержимого окна на активной странице монитора (листинг 6)

call SCROLL (direct, windowdecl, numbscroll, attrib), где direct — направление сдвига (0 — вверх, 1 — вниз); windowdecl — описатель окна (4 байта) ⟨столбец, строка левого верхнего угла и столбец, строка правого нижнего угла⟩; numbscroll — число сдвигов (0 = очистке всего окна); attrib — атрибуты цвета заднего и переднего планов, устанавливаемые при сдвиге. Неприменима в графическом режиме.

7. Функция чтения символа и его атрибутов с заданной страницы (листинг 7)

```
INTEGER*2 i
CHARACTER chat(2)
EQUIVALENCE (i,chat(1))
i=ITSTCH (numpage),
```

где с chat (1) — код символа; с chat (2) — атрибуты цвета символа (число 0 ... 255).

8. Вывод строки символов с ограничителем в текущих цветовых атрибутах (листинг 8)

```
call TYPE (fon, string, terminator),
```

где fon—атрибуты цвета фона при работе в графическом режиме; string — строка символов, заканчивающаяся ограничителем; terminator — символ-ограничитель.

9. Вывод строки символов с ограничителем в инверсных цветовых атрибутах (листинг 9)

```
call TYPINV (string, terminator).
```

10. Процедура заполнения экрана заданным символом (листинг 10)

```
call FILLCH (numpage, numbsimbol, simbol),
```

где numpage — номер страницы; numbsimbol—число повторов символа; simbol — символ заполнения. Если курсор установлен на начало страницы (координаты 0,0), в numbsimbol длина строки экрана*25. После выполнения этой процедуры весь экран будет заполнен заданным символом. Курсор положения не изменяет.

11. Процедура сохранения строки экрана (листинг 11)

```
call READPG (numpage, length, string).
```

Процедура возвращает строку string длины length, содержащую последовательность символов на указанной странице numpage, начиная с позиции курсора на заданной странице.

Пояснения к текстам подпрограмм

Базовая система ввода-вывода в ПЗУ «Искра 1030.11» предоставляет возможность вызова конкретных программ управления клавиатурой посредством запросов на внутренние прерывания ПЭВМ с соответствующими кодами. Ввод-вывод на монитор имеет код прерывания 10, ввод с клавиатуры —2, 16 (числа 16-ричные). Код функции устройства передается в базовую систему ввода-вывода через регистр AH; для передачи параметров используются и другие регистры процессора.

Пример применения описанных процедур на Фортране приведен в листинге 12. Текст комментария отделяется от операторов языка в листингах 12—14 символом!. Процедура SAMPLE оформлена в виде подпрограммы, которую можно поместить непосредственно в текст собственной прикладной программы. Она реализует диалоговый интерфейс [6] и вначале организует экран монитора (рис. 3), затем в бесконечном цикле выводит в три окна гистограмму и ее числовые характеристики, а также ожидает прерывание цикла по нажатию клавиши с кодом 27 (здесь и далее ESC), после которой переходит на меню (метка 20), позволяющее выбрать одну из четырех альтернатив или завершить процедуру (метка 25). Среди предлагаемых альтернатив есть переход на следующий уровень иерархии управления (меню метки 24), с которого нажатием клавиши ESC можно вернуться назад (на более высокий уровень иерархии).

Таким образом обеспечивается иерархия управления. Выбор альтернативы из меню при движении по графу (см. рис. 1) может быть организован, например, процедурой листинга 13. Листинг 14 показывает один из вариантов вывода на экран текстов полной подсказки при его многостраничной организации. Подсказка вызывается нажатием клавиши F1 и занимает весь экран. Внешний вид подсказки зависит от способа ее создания (набор текста) и загрузки на страницы монитора (атрибутированная загрузка). Выход из подсказки и восстановление нулевой страницы монитора (см. рис.3) выполняются нажатием ESC; нажатие остальных клавиш клавиатуры вызывает переход на следующую страницу подсказки или возврат на первую.

При исполнении описанных подпрограмм в программах на языках Паскаль и Си необходимо учитывать передачу параметров:

в подпрограммы из Паскаль-программы только по наименованию. Объявление процедуры SETMOD в Паскаль-программе должно выглядеть, например, так:

```
PROCEDURE SETMOD (VARS MODE:ИNTEGER2);
```

в С-программы через их указатели с учетом возможного изменения порядка передачи, принятого конкретным компилятором. Кроме того, в С-описаниях приводимых функций следует дать признак fortran, а также согласовать модуль генерируемой задачи, пользуясь руководством по компиляции. Для большой модели в Microsoft Си 3.0 объявление будет иметь вид

```
fortran setmod (mode)
int * mode;
```

(* — здесь можно дать описатель long mode; *)

При передаче строк с ограничителем следует помнить, что в языке Си существует стандартный ограничитель строк— нулевой байт. Таким образом, различия языков не требуют модификации предлагаемых подпрограмм, а влияют только на спецификации подпрограмм и на вид вызова.

Листинг 1

```
;*****************************************
;
;    Установка режима работы монитора
;
     TITLE     SETMOD
     NAME SETMOD
;
;*****************************************
;    Описание стека
;*****************************************
FRAME STRUC
;
SAVEDS    DW   ?    ;Область хранения DS
SAVEBP    DW   ?    ;-"-          -"-  BP
RETADDR   DD   ?    ;Адрес возврата
MODADDR   DD   ?    ;Адрес параметра режима
;
FRAME     ENDS
;
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC    SETMOD
;
SETMOD    PROC FAR        ;Точка входа
;
;------ Тело процедуры ------------------
     PUSH  BP             ;;Хранение BP
     PUSH  DS             ;;   -"-   DS
     MOV   BP,SP          ;;Параметры через BP
;
     LES   BX,[BP].MODADDR     ;
     MOV   AL,ES:[BX]          ;Код режима в AL
     XOR   AH,AH               ;AH=0
     INT   10H                 ;Переключение режима
;
     POP   DS             ;;;
     POP   BP             ;;;
     RET   4
SETMOD    ENDP
MYSEG     ENDS
          END
```

Листинг 2

```
;*****************************************
;
;     Переключение страниц монитора
;
     TITLE     NEWPAG
     NAME NEWPAG
;
;*****************************************
;     Описание стека
;*****************************************
FRAME STRUC
;
SAVEDS    DW   ?    ;Область хранения DS
SAVEBP    DW   ?    ;-"-        -"-   BP
RETADDR   DD   ?    ;Адрес возврата
PAGADDR   DD   ?    ;Адрес номера страницы
;
FRAME     ENDS
;
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC    NEWPAG
;
NEWPAG    PROC FAR       ;Точка входа
;
;------- Тело процедуры ---------------------
     Вставить строки листинга 1 с пометкой ;;
     LES   BX,[BP].PAGADDR    ;
     MOV   AL,ES:[BX]         ;Номер страницы в AL
     MOV   AH,5               ;Смена страницы
     INT   10H                ;монитора
;
     Вставить строки листинга 1 с пометкой ;;;
     RET   4                  ;Выход из процедуры
NEWPAG    ENDP
MYSEG     ENDS
          END
```

Листинг 3

```
;*****************************************
;
;     Чтение текущего состояния монитора:
;          номер активной страницы;
;          режим работы монитора;
;          длина строки монитора в символах
;
     TITLE     VIDEOT
     NAME      VIDEOT
;*****************************************
;     Описание стека
;*****************************************
;
FRAME STRUC
;
SAVEDS    DW   ?    ;Область хранения DS
SAVEBP    DW   ?    ;-"-        -"-   BP
RETADDR   DD   ?    ;Адрес возврата
COLUMAD   DD   ?    ;Адрес длины строки монитора
MODADDR   DD   ?    ;Адрес режима работы монитора
PAGADDR   DD   ?    ;Адрес номера активной
;                    страницы
;
FRAME     ENDS
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC    VIDEOT
;
VIDEOT    PROC FAR       ;Вход в процедуру
;
;------- Тело процедуры ---------------------

     Вставить строки листинга 1 с пометкой ;;
      MOV AH,15       ;Чтение текущего
;                      состояния монитора
     INT  10H
     LES  SI,[BP].MODADDR ;Адрес режима в ES:SI
     MOV  ES:[SI],AL      ;Передача номера режима
     LES  SI,[BP].COLUMAD ;Адрес длины строки в
;                          ES:SI
     MOV  ES:[SI],AH      ;Передача длины строки
;                          монитора
     LES  SI,[BP].PAGADDR ;Адрес номера страницы в
;                          ES:SI
     MOV  ES:[SI],BH      ;Передача номера
;                          активной страницы
     Вставить строки листинга 1 с пометкой ;;;
     RET  12              ;Выход из процедуры
VIDEOT    ENDP
MYSEG     ENDS
     END
```

Листинг 4

```
;*****************************************
;
;     Чтение текущих координат
;     курсора - строка/столбец
;       на заданной странице монитора
;
     TITLE     CURPOS
     NAME      CURPOS
;
;*****************************************
;     Описание стека
;*****************************************
FRAME STRUC
;
SAVEDS    DW   ?    ;Область хранения DS
SAVEBP    DW   ?    ;-"-        -"-   BP
RETADDR   DD   ?    ;Адрес возврата
COLADDR   DD   ?    ;Адрес номера столбца
ROWADDR   DD   ?    ;Адрес номера строки
PAGADDR   DD   ?    ;Адрес номера страницы
;
FRAME     ENDS
;
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC    CURPOS
;
CURPOS    PROC FAR       ;Точка входа
;
;------- Тело процедуры ---------------------
     Вставить строки листинга 1 с пометкой ;;
;
     LES   BX,[BP].PAGADDR ;
     MOV   BH,ES:[BX]      ;Номер страницы в BH
     MOV   AH,3            ;Чтение координат
     INT   10H             ;курсора
;
     LES   BX,[BP].ROWADDR ;Запись номера
     MOV   ES:[BX],DH      ;строки
     LES   BX,[BP].COLADDR ;Запись номера
     MOV   ES:[BX],DL      ;столбца
;
     Вставить строки листинга 1 с пометкой ;;;
     RET   12         ;Выход из процедуры
CURPOS    ENDP
MYSEG     ENDS
     END
```

Листинг 5

```
;*****************************************
;
;     Позиционирование курсора на заданной
;     странице монитора по заданной строке/столбцу
;
     TITLE   MOVCUR
     NAME    MOVCUR
;
;*****************************************
;     Описание стека
;*****************************************
;
FRAME STRUC
;
SAVEDS    DW   ?     ;Область хранения DS
SAVEBP    DW   ?     ; -"-       -"-    BP
RETADDR   DD   ?     ;Адрес возврата
COLUMAD   DD   ?     ;Адрес номера столбца
ROWADDR   DD   ?     ;Адрес номера строки
PAGADDR   DD   ?     ;Адрес номера страницы
;
FRAME     ENDS
;
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC    MOVCUR
;
MOVCUR    PROC FAR       ;Вход в процедуру
;
```

```
;------- Тело процедуры ----------------------------------
     Вставить строки листинга 1 с пометкой ;;
;
     LES  BX,[BP].ROWADDR ;Адрес номера
                          строки в ES:BX
;
     MOV  DH,ES:[BX]      ;Номер строки в DH
     LES  BX,[BP].COLUMAD ;Адрес номера
                          столбца в ES:BX
;
     MOV  DL,ES:[BX]      ;Номер столбца DL
     LES  BX,[BP].PAGADDR ;Адрес номера
                          ;страницы в ES:BX
;
     MOV  BH,ES:[BX]      ;Номер страницы BH
     MOV  AH,2            ;Позиционирование
     INT  10H             ;курсора
     Вставить строки листинга 1 с пометкой ;;;
     RET  12              ;Выход из процедуры
MOVCUR   ENDP
MYSEG    ENDS
     END
```

Листинг 6

```
;*************************************************
;
;     Сдвиг (scroll) окна экрана монитора
;     вверх или вниз
;
     TITLE     SCROLL
     NAME      SCRGLL
;
;*************************************************
;     Описание стека
;*************************************************
;
FRAME STRUC
;
SAVEDS   DW   ?    ;Область хранения DS
SAVEBP   DW   ?    ;-"-          -"-   BP
RETADDR  DD   ?    ;Адрес возврата
ATRADDR  DD   ?    ;Адрес атрибутов окна
NLNADDR  DD   ?    ;Адрес количества сдвигов
WNDADDR  DD   ?    ;Адрес описателя окна
DIRADDR  DD   ?    ;Адрес направления
;                    (0-вверх,1-вниз)
FRAME    ENDS
;

MYSEG    SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC   SCROLL
;
SCROLL   PROC FAR       ;Вход в процедуру
;
;------- Тело процедуры ----------------------------------
     Вставить строки листинга 1 с пометкой ;;
     LES   BX,[BP].DIRADDR ;
     MOV   AH,ES:[BX]      ;Направление в AH
     ADD   AH,6            ;Код функции сдвига
     LES   BX,[BP].NLNADDR ;
     MOV   AL,ES:[BX]      ;Количество сдвигов
     LES   BX,[BP].WNDADDR ;
     MOV   CX,ES:[BX]      ;Описатель окна в
     MOV   DX,ES:[BX][2]   ; <CX,DX>
     LES   BX,[BP].ATRADDR ;
     MOV   BH,ES:[BX]      ;Атрибуты окна в BH
     INT   10H
     Вставить строки листинга 1 с пометкой ;;;
     RET   16         ;Выход из процедуры
SCROLL   ENDP
MYSEG    ENDS
     END
```

Листинг 7

```
;*************************************************
;
;     Чтение текущего символа и его атрибутов
;     с заданной страницы монитора
;
     TITLE     ITSTCH
     NAME ITSTCH
;
```

```
;*************************************************
;     Описание стека
;*************************************************
;
FRAME STRUC
;
SAVEDS   DW   ?    ;Область хранения DS
SAVEBP   DW   ?    ;-"-          -"-   BP
RETADDR  DD   ?    ;Адрес возврата
PAGADDR  DD   ?    ;Адрес параметра номера
;                    страницы монитора
;
FRAME    ENDS
;
;

MYSEG    SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC   ITSTCH
;
ITSTCH   PROC FAR       ;Вход в функцию
;
;------- Тело функции ------------------------------------
     Вставить строки листинга 1 с пометкой ;;
;
     LES   BX,[BP].PAGADDR ;Адрес параметра
;                            в ES:BX
     MOV   BH,ES:[BX]      ;Номер страницы в BH
;
     MOV   AH,8            ;Символ/атрибуты в
     INT   10H             ;     AL/AH
;
     Вставить строки листинга 1 с пометкой ;;;
     RET   4               ;Выход из функции
ITSTCH   ENDP
MYSEG    ENDS
     END
```

Листинг 8

```
;*************************************************
;
;     Вывод строки ограниченной <$>,в текущую
;     страницу монитора при заданном фоне
;
     TITLE     TYPE
     NAME      TYPE
;
;*************************************************
;     Описание стека
;*************************************************
;
FRAME STRUC
;
SAVEDS   DW   ?    ;Область хранения DS
SAVEBP   DW   ?    ;-"-          -"-   BP
RETADDR  DD   ?    ;Адрес возврата
EOSADDR  DD   ?    ;Адрес ограничителя строки
BUFADDR  DD   ?    ;Адрес строки
FONADDR  DD   ?    ;Адрес фона
;
FRAME    ENDS
;
MYSEG    SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC   TYPE
;
TYPE  PROC    FAR          ;Вход в процедуру
;
;------- Тело процедуры ----------------------------------
     Вставить строки листинга 1 с пометкой ;;
     LES   BX,[BP].EOSADDR ;Признак конца
     MOV   DL,ES:[BX]      ;строки в DL
     LES   BX,[BP].FONADDR ;Цвет фона в
     MOV   BL,ES:[BX]      ;графич. режиме в BL
     LES   SI,[BP],BUFADDR ;Адрес строки- ES:SI
     CLD                   ;Очистка D-флага PSW
     MOV   AH,14           ;Телетайпный вывод
TTYPE:
     LODSB                 ;Загрузка символа
                           строки в AL
     CMP   AL,DL           ;Проверка на признак
     JE EXIT               ;конца строки
     INT   10H             ;Вывод символа
     JMP   SHORT TTYPE     ;Повтор вывода
EXIT:
     Вставить строки листинга 1 с пометкой ;;;
     RET   12              ;Выход из процедуры
TYPE  ENDP
MYSEG    ENDS
     END
```

Листинг 9

```
;***********************************************
;
;     Вывод строки инверсно к установленным
;     атрибутам цветности заполняемых позиций
;     экрана
;
     TITLE     TYPINV
     NAME      TYPINV
;
;***********************************************
;     Описание стека
;***********************************************
;
FRAME STRUC
;
SAVEDS    DW   ?    ;Область хранения DS
SAVEBP    DW   ?    ;-"-       -"-   BP
RETADDR   DD   ?    ;Адрес возврата
EOSADDR   DD   ?    ;Адрес признака конца стр
BUFADDR   DD   ?    ;Адрес строки
;
FRAME     ENDS
;
;
DATA SEGMENT PUBLIC 'DATA'
;
HEX  DB     16      ;Константа=16
SAVE DB      0      ;Хранение атрибутов
COD  DB      0      ;Признак конца стр.
;
DATA ENDS
DGROUP GROUP DATA
;
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG,DS:DGROUP,ES:DGROUP
PUBLIC    TYPINV
;
TYPINV    PROC FAR          ;Вход в процедуру
;
;------- Тело процедуры ----------------------------
     Вставить строки листинга 1 с пометкой ;;
     LES   BX,[BP].EOSADDR  ;Признак конца
     MOV   DL,ES:[BX]       ;строки в DL
     MOV   COD,DL           ;Признак в COD
;
     LES   SI,[BP].BUFADDR  ;Адрес строки- ES:SI
;
     MOV   AH,15            ;Состояние монитора
     INT   10H              ;номер активной стр.
;                           ;в BH
     MOV   AH,3             ;Координаты курсора
     INT   10H              ;DH/DL-строка/столб.
;
     XOR   CX,CX            ;Ноль в CX
     INC   CL               ;Единица в CX
     CLD                    ;Очистка D-флага PSW
TTYPINV:
     CALL INVERSE           ;Инверсные атрибуты
;                           ;цвета в BL
   LODSB                    ;Загрузка символа
;                           ;строки в AL
     CMP   AL,COD           ;Проверка на признак
     JE    EXIT             ;конца строки
;
     MOV   AH,9             ;Вывод символа
     INT   10H              ;с атрибутами цвета
;
     INC   DL               ;Сдвиг курсора
     MOV   AH,2             ;вправо без проверки
     INT   10H              ;конца строки
;
     JMP   SHORT TTYPINV    ;Повтор вывода
;
EXIT:
     Вставить строки листинга 1 с пометкой ;;;
     RET   8                ;Выход из процедуры
TYPINV    ENDP
;
;    Чтение атрибутов цветности текущей позиции
;    курсора на активной странице монитора (BH)
;    и формирование инверсных атрибутов цветности
;
INVERSE PROC   NEAR         ;Вход в процедуру
;
     MOV  AH,8              ;Чтение символа
     INT  10H               ;и атрибутов
;
     XOR  AL,AL             ;Инверсия атрибутов
     XCHG AL,AH             ;цветности
     DIV  HEX               ;
     MOV  SAVE,AL           ;ФОН*16+ЦВЕТ
     XCHG AL,AH             ;переводится
     IMUL HEX               ;в
     MOV  BL,SAVE           ;ЦВЕТ*16+ФОН
     ADD  BL,AL             ;в BL
     RET
INVERSE   ENDP
MYSEG     ENDS
     END
```

Листинг 10

```
;***********************************************
;
;     Заполнение строки символов
;
     TITLE    FILLCH
     NAME FILLCH
;
;***********************************************
;     Описание стека
;***********************************************
;
FRAME STRUC
;
SAVEDS    DW   ?    ;Область хранения DS
SAVEBP    DW   ?    ;-"-       -"-   BP
RETADDR   DD   ?    ;Адрес возврата
CHADDR    DD   ?    ;Адрес ASCII кода символа
NUMADDR   DD   ?    ;Адрес количества повторов
PAGADDR   DD   ?    ;Адрес номера страницы
;
FRAME     ENDS
;
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC    FILLCH
;
FILLCH    PROC FAR        ;Точка входа
;
;------- Тело процедуры ----------------------------
     Вставить строки листинга 1 с пометкой ;;
     LES   BX,[BP].CHADDR   ;Адрес символа
     MOV   AL,ES:[BX]       ;Символ в AL
     LES   BX,[BP].NUMADDR  ;Адрес числа символов
     MOV   CX,ES:[BX]       ;Число символов в CX
     LES   BX,[BP].PAGADDR  ;Адрес номера
;                           ;страницы
     MOV   BH,ES:[BX]       ;Номер страницы BH
;
     MOV   AH,10            ;Писать символ
     INT   10H              ;CX раз
;
     Вставить строки листинга 1 с пометкой ;;;
     RET   12               ;Выход из процедуры
FILLCH    ENDP
MYSEG     ENDS
     END
```

Листинг 11

```
;***********************************************
;
;     Чтение строки символов, записанной
;     в заданную страницу монитора ПК от текущей
;     позиции курсора
;
     TITLE     READPG
     NAME      READPG
;
;***********************************************
;     Описание стека
;***********************************************
;
FRAME STRUC
;
SAVEDS    DW   ?    ;Область хранения DS
SAVEBP    DW   ?    ;-"-       -"-   BP
RETADDR   DD   ?    ;Адрес возврата
BUFADDR   DD   ?    ;Адрес массива хранения
LENADDR   DD   ?    ;Адрес длины строки
PAGADDR   DD   ?    ;Адрес номера страницы
;
```

```
FRAME     ENDS
;
;
MYSEG     SEGMENT 'CODE'
 ASSUME CS:MYSEG
PUBLIC    READPG
;
READPG    PROC FAR          ;Вход в процедуру
;
;------ Тело процедуры ----------------------------------
     Вставить строки листинга 1 с пометкой ;;
     LES   BX,[BP].PAGADDR
     MOV   BH,ES:[BX]        ;Номер страницы в BH
;
     MOV   AH,3              ;Чтение позиции
     INT   10H               ;курсора в (DH,DL)
;
     PUSH  DX                ;Хранение DX
     LES   SI,[BP].LENADDR   ;
     MOV   CX,ES:[SI]        ;Длина строки в CX
;
     LES   DI,[BP].BUFADDR   ;Адрес массива
                             ; в ES:DI
;
     CLD                     ;Очистка D-флага PSW
READ:
     MOV   AH,8              ;Чтение текущего
     INT   10H               ;символа
;
     STOSB                   ;Хранение символа
     INC   DL                ;Новая координата
                              курсора
;
     MOV   AH,2              ;Сдвиг курсора
     INT   10H
;
     LOOP  READ              ;Цикл по длине
                              сохраняемой строки
;
     POP   DX                ;Восстановление
     MOV   AH,2              ;начальной позиции
     INT   10H               ;курсора
;
     Вставить строки листинга 1 с пометкой ;;;
     RET   12                ;Выход из процедуры
READPG    ENDP
MYSEG     ENDS
     END
```

Листинг 12

```
     subroutine SAMPLE(...)
c
c    пример описания и создания окон рис.3
c
     character*1 wnd1(4),wnd2(4),wnd3(4),wnd4(4)
     character*1 wnd5(4),wnd6(4),wnd7(4)
     character*1 terminator,simbol
     character*80 string
     integer*2 fon1,color1,...,fon7,color7
c    wnd          -описатель окна (координаты углов)
c    fon          -цвет фона в окне (0..7)
c                  (8..15 -мигание переднего плана)
c    color        -цвет переднего плана (0..15)
c    интерпретация атрибута цветности зависит от
c    типа монитора. В "Искре 1030.11" цвета
c    передаются полутонами
c    terminator-ограничитель строки (в C =0)
c    simbol       -псевдографический символ
c
     data wnd1/.../,...,wnd7/.../  ! описание
     data fon1,...,fon7/.../       !координат и
                                    атрибутов
c
     data color1,...,color7/.../   !цвета окон
     data terminator/'$'/,simbol/'Z'/
c
c    создание  семи окон
c
     call SCROLL(0,wnd1,0,fon1*16+color1)!чистка и
                                          установка
c
     ...
     callSCROLL(0,wnd7,0,fon7*16+color7)!атрибутов
                                          цветности
c
c    вывод сообщений в окна
     call MOVCUR(0,wnd1(2),wnd1(1)+2)   !установка
                                          курсора
c
     call TYPE(0,'ИССЛЕДОВАНИЯ$',terminator)!вывод
c                                          строки
     ...
     call MOVCUR(0,wnd3(2),wnd3(1))
     call TYPE(0,'АППРОКСИМАТОР$',terminator)
     ...
c    вывод данных в окно-столбик  (левый столбец
c     рис.3) и гистограмм
c
     i=0
1    i=i+1
     write(string,10)i,terminator   !вывод   в
c                                   строку памяти
10   format(I5,A1)
     call   MOVCUR(0,wnd5(4),wnd5(1))   !курсор  в
c                          левый нижний угол окна 5
     call TYPE(0,string,terminator)
     call    SCROLL(0,wnd5,1,fon5*16+color5)!сдвиг
c                          содержимого окна 5 вверх
     ...
     write(string,20)sigma,terminator    !вывод в
c                                   строку памяти
20   format(E10.4,A1)
     call   MOVCUR(0,wnd7(4),wnd7(1))   !курсор  в
c                          левый нижний угол окна 7
     call TYPE(0,string,terminator)
     call    SCROLL(0,wnd7,1,fon7*16+color7)!сдвиг
c                          содержимого окна 7 вверх
     ...
c    вывод гистограмм
c
     call MOVCUR(0,wnd6(4),wnd6(1))   !курсор  в
c                          левый нижний угол окна 6
     length=...                      !длина строки
c                                    гистограммы
     call    FILLCH(0,length,simbol)!вывод  копии
c                          символа без сдвига курсора
     call SCROLL(0,wnd6,1,fon6*16+color6)   !сдвиг
c                          содержимого окна 6 вверх
     if(NWKEY.NE.27)goto 1          !проверка буфера
c    клавиатуры без ожидания ввода.  Если не была
c    нажата клавиша ESC, то продолжаем процесс
c    вывода в окна иначе:
c
c    меню уровня с набором команд  и  поясняющих
c    текстов
c
20   i=MENU('МЕНЮ$',4,
     + 'НАСТРОЙКА$РЕГРЕССИЯ$УСКОРЕНИЕ$КОНЕЦ$',
     + 'Установка...$Аппроксимация уравнения
     + 'полиномом$...$Оформление результата$')
     goto (21,22,23,24,25),i!переход на
c    альтернативу; по клавише ESC на метку 25
c    НАСТРОЙКА
21   ...
     goto 20 ! возврат на меню
c    РЕГРЕССИЯ
22   ...
     goto 20 ! возврат на меню
     ...
c    КОНЕЦ - меню подуровня
c
24   i=MENU('МЕНЮ$',4,'ХРАНИТЬ$АРХИВ$ВЫХОД$',
     + 'Запись в файл$Запись в архив$Выход$')
     goto (31,32,25,20),i !переход на альтернативу
c    по клавише ESC возврат на меню уровня...
31   ...
     goto 24
32   ...
     goto 24
25   return
     end
```

Листинг 13

```
     function MENU(mname,numbcom,comlist,expl)
c
c    Процедура строчного меню
c    ========================
c    mname      имя меню  формат:='xxxxxx$'
c    numbcom    число команд меню
c    comlist    список команд
c    формат:='kom1$kom2$...kom(numbcom)$'
c    expl текст быстрой подсказки
c    формат:='expl1$expl2$...expl(numbcom)$'
c    $          ограничитель строки для Фортрона
c    ========================
c
     ...          Текст настройки процедуры
     ...          и вычисления позиций текста

     do 7 i=1,numbcom              ! Вывод команд
     l=tabcom(1,i)                 ! меню на 22-й
     call MOVCUR(0,22,l)           ! строке экрана
     call TYPE(0,comlist(1),'$')   ! в позициях
                                   ! из tabcom
     ...
```

```
      ibascom=tabcom(1,imenu)       ! Позиция kom1,
      lencom=tabcom(2,imenu)        ! ее длина
      ibasexpl=tabexpl(1,imenu)     ! Позиция expl1,
      lenexpl=tabexpl(2,imenu)      ! ее длина
      ...
c     Выделение готовой к активизации
c     команды меню негативным цветом
c
1     call MOVCUR(0,22,ibascom)  !Засветка альтер-
      call TYPINV(comlist(ibascom),'$')    !нативы
c
c     Вывод текста быстрой подсказки для
c     альтернативы imenu
      call MOVCUR(0,23,0) !Курсор ниже строки меню
      call TYPE(0,expl(ibasexpl)
      + ,'$')                !Вывод текста подсказки
c
c     Ожидание кода от оператора (<--/-->)
c
      icod=INKEY() ! Ввод кода с клавиатуры (АДОС)
      if(icod.EQ.27)then  !Проверка кода ESC
      MENU=numbcom+1      !Выход из меню по
      return              !нажатой клавише ESC
      endif
c
      if(icod.EQ.315)call HELP !Функциональная
c     клавиша F1 вызывает режим полной подсказки
      ...  Обработка введенного кода,
      ...  вычисление сдвига или
      ...  выход с номером активной команды
c
c     Возвращение нормальных цветовых атрибутов
c     ранее выделенной команде меню
c
      call MOVCUR(0,22,ibascom)
      call TYPINV(comlist(ibascom),'$')
      ...       Вычисление нового
      ...       значения ibascom
      goto 1
      end
```

Листинг 14

```
      subroutine HELP
c
c        АЛГОРИТМ:
c     1. Ищем слово HELP в левом верхнем углу
c        первой страницы,
c        если нет, то сообщаем об отсутствии и на шаг 5.
c     2. Переключаемся на следующую страницу.
c     3. Ожидание ввода с клавиатуры,
c        если введен код ESC, отмена режима и на шаг 6.
c     4. Выполняем шаг 1 для страницы i+1,
c        если нет слова HELP, готовим
c        переключение на 1-ю страницу.
c     5. Идем на шаг 2.
c     6. Переключаемся на      0-ю страницу,
c        выходим из процедуры.
c
      character*4 HELP,buffer
      integer PGNUMBER,ESC,NUL
      data HELP/'HELP'/,ESC/27/,NUL/0/
c
      call MOVCUR(1,NUL,NUL)
c
c     Проверка наличия HELP
c
      call READPG(1,4,buffer)   !Чтение страницы 1
      if(HELP.NE.buffer)return !Нет текста пояснения
      i=1
21    call NEWPAGE(i)             !Смена страницы
      j=INKEY()                   !Ожидание ввода
      if(j.EQ.ESC)goto 3
      i=i+1
      call MOVCUR(1,NUL,NUL)    !Курсор в начало
      if(i.EQ.4)goto 2
c
c     Проверка наличия HELP
c
      call READPG(1,4,buffer)    !Чтение страницы i
      if(HELP.EQ.buffer)goto 21
      goto 2
3     call NEWPAG(NUL)            !На страницу 0
      return
      end
```

Телефон 252-69-89, Ленинград

ЛИТЕРАТУРА

1. Громов Г. Р. Автоформализация профессиональных знаний // Микропроцессорные средства и системы.— 1986.— № 3.— С. 80—91.
2. Громов Г. Р. и др. ДИАНЭД-4 — инструмент для создания диалоговых систем — Пущино: НЦБИ АН СССР, 1981.— С. 40.
3. Ярошевская М. Б.— Персональная ЭВМ «Искра 1030.11» // Микропроцессорные средства и системы.— 1986.— № 4.— С. 23—24.
4. Norton P. Programmers Guide to IBM PC.— Maryland: Prentice-Hall, Bowie. 1986.
5. Миллер Дж. А. Магическое число семь плюс или минус два. О некоторых пределах нашей способности перерабатывать информацию // Сб. статей «Инженерная психология».— М.: Прогресс, 1984.— С. 192—225.
6. Домарацкий С. Н., Лозовой Л. Н. Аппаратные и программные средства новой серии многоканальных портативных рентгеновских спектрометров РОСА // ПСУ.— 1987.— № 3.— С. 21—23.
7. Брябрин В. М., Ландау И. Я., Неменман М. Е. Система кодирования для персональных ЭВМ // Микропроцессорные средства и системы.— 1986.— № 4.— С. 61—63.

Статья поступила 24 апреля 1987

УДК 681.06

А. Л. Попов

КОМПЛЕКС ТЕХНОЛОГИЧЕСКИХ ДИСКЕТ ДЛЯ ПЭВМ

У пользователей современных отечественных ПЭВМ две основные проблемы, существенно снижающие производительность труда:

низкая оперативность поиска и загрузки программных средств (ПС), необходимых для решения поставленной задачи. После удачного поиска ПС приходится компоновать их на одной или нескольких дискетах, а это требует многократного выполнения однообразных операций копирования и смены дискет в накопителях, что вызывает дополнительные затраты времени;

необходимость обязательного знания основных команд взаимодействия с системными и прикладными ПС. Эти команды определяют методы системного обслуживания дискет и файлов, устройств печати, ОС, а также различных систем программирования, обработки текстов, управления базами данных, информационного поиска. Все это связано с затратами времени на поиски необходимой справочной информации об используемых для данной задачи системных и прикладных ПС и требует многократного ввода одних и тех же команд и их сочетаний, что само по себе не рационально.

В составе комплекса технологических дискет (КТД) комплект дискет трех классов: «целевая», «прикладная» и «рабочая».

Дискеты класса «целевая» и «прикладная» содержат ПС, предназначенные для решения одной или нескольких задач, определенных для конкретной дискеты. Каждой дискете КТД соответствует идентификатор, определяющий функциональное назначение ПС, включенных в ее состав. Любая дискета этих классов должна иметь средства автоматизации многократного выполнения команд, сохраненных в ее системе. В то же время такая система должна обладать наглядными возможностями визуального выбора из функций, назначенных для исполнения.

Дискета класса «целевая» — автономная системная дискета, содержащая обязательный минимальный набор системных ПС, а также выбранные прикладные ПС, необходимые для решения одной или нескольких целевых задач,

Дискета класса «прикладная», дополняющая дискету класса «целевая», не содержит системных ПС и предназначена для решения прикладных задач (демонстрационных, информационных, игровых, вывода текстовых документов на экран или печать), не требующих внешней памяти для организации файлов.

Дискета класса «рабочая» выполняет функции внешней рабочей области, предназначенной для организации файлов под управлением ПС дискеты класса «целевая».

Для разработки КТД можно использовать ПЭВМ класса ДВК (ДВК-2 и ДВК-3) в среде ОС RT-11 или совместимой (РАФОС, ФОДОС, ОС ДВК).

Пользователь КТД взаимодействует с ПС, включенными в состав дискет класса «целевая» и (или) «прикладная», через интерфейс типа «Система меню». Этот интерфейс разработан с помощью ПС, включенных в состав «Системы меню» программного комплекса «РТК-МИКРО».

Состав ПС «Системы меню»:

«Интерпретатор меню» организует взаимодействие пользователя с ПС дискет класса «целевая» и(или) «прикладная».

«Редактор меню» создает и редактирует структуру отдельного меню. Меню — структурный файл из обязательных («Имя файла меню», «Заголовок меню», «Список работ», «Отдельная работа») и вспомогательных («Пароль», «Стартовая работа») компонентов.

«Имя файла меню» определяет вход в отдельное меню для программы «Интерпретатор меню», «Заголовок меню» указывает основное функциональное назначение, «Список работ» включает наименования отдельных работ данного меню. «Отдельная работа» определяется одним из трех типов: «Справка», «Команды», «Меню», «Справка» и «Меню» используют имя файла соответствующего типа (QUE и MEN) и определяют направление дальнейшей интерпретации меню. «Меню» — базовая работа для построения системы меню иерархического типа, «Команды» — конечное исполнительное звено любого меню, содержащего работы этого типа. В работе используется группа команд монитора, которая выполняет соответствующие функции. Команды монитора могут содержать формальные параметры, заменяемые пользователем на действительные в процессе интерпретации меню. «Справка» имеет информационное значение, так как файл типа QUE — обычный текстовый файл ограниченного объема (до 51 блоков). Он создается и редактируется средствами стандартного редактора текста.

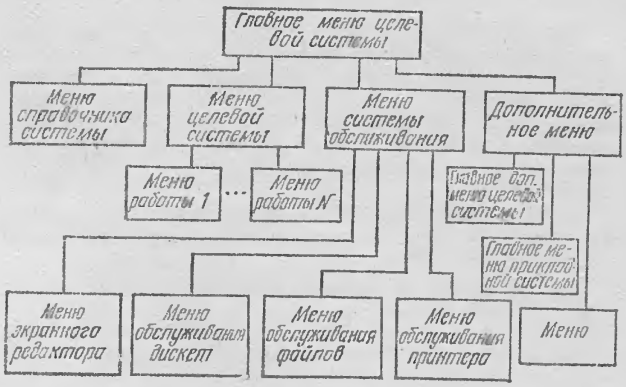
Вспомогательные компоненты «Пароль» и «Стартовая работа» могут отсутствовать в конкретном меню. «Пароль» меню — совокупность заданных и не отображаемых символов, которая определяет санкционированный доступ в данное меню. «Стартовая работа» — отдельная работа одного из двух типов: «Справка» или «Команды»; она автоматически выполняется при входе в данное меню.

«Печать меню» выводит структуру определенного меню на указанное внешнее устройство (принтер, дискету или терминал). Эта информация необходима для отладки всей иерархии меню.

Структура системы меню дискеты класса «целевая»

Дискета класса «целевая» — базовая компонента КТД (см. рисунок). Главное меню — вершина структуры, включающая четыре вида работ: справочник системы; целевую работу, определяющую основное назначение данной дискеты; систему обслуживания и дополнительное меню.

«Система обслуживания» — базовая работа для любой дискеты класса «целевая», включает многочисленные и наиболее необходимые функции обслуживания дискет, файлов, принтера, программы системного и прикладного назначения.



Фрагмент структуры системы меню

«Целевая работа» вызывает систему меню, специфичную для дискеты класса «целевая». Явно заданное наименование этой работы и состав включенных в нее функций определяют основное целевое назначение и возможности данной дискеты.

«Дополнительное меню» обеспечивает выбор одного из трех режимов: вызов «Меню новой целевой системы», обращение к «Меню прикладной системы» и вход в «Определенное меню». Первый режим предусматривает смену дискеты класса «целевая», второй требует установки дискеты класса «прикладная» в свободный дисковод, третий режим позволяет оперативно войти в любое отдельное меню.

Сервисные особенности дискеты класса «целевая»

Стартовый командный файл базовой ОС дискеты класса «целевая» обеспечивает запрос и запоминание в системе даты ее запуска, сообщение дня недели, автоматический запуск программы «Интерпретатор меню» и обращение к главному меню дискеты.

Работа в системе «целевой» дискеты может быть приостановлена для выхода в режим монитора, где доступны любые команды монитора для непосредственного и оперативного использования системных и прикладных ПС, входящих в данную дискету. Для ОС RT-11 версии V5.0 используется редактор командной строки, обеспечивающий режимы: запоминание в кольцевом буфере введенных командных строк, быстрый выбор необходимой командной строки из кольцевого буфера, повтор предыдущей командной строки, редактирование текущей командной строки, построение строки из нескольких команд для последовательного исполнения. После окончания работы в режиме монитора возможен быстрый возврат в точку выхода из меню или в главное меню системы.

При использовании ОС RT-11 версии V5.0 в качестве базовой ОС целевой дискеты две специальные команды (на базе редактора командной строки) включают и выключают печать текста с экрана дисплея.

«Интерпретатор меню» выбирает необходимую работу из списка и обеспечивает возврат в высший уровень меню. Для оперативного возврата в главное меню используется специальная работа «Возврат в главное меню» — последняя работа любого меню, кроме главного.

Использование двух сторон 8″ дискеты (с дополнительным, симметричным стандартному, маркерным отверстием в конверте диска) может существенно расширить ее целевые функции.

Технические возможности КТД практически не ограничены и определяются составом ПС, входящих в отдельные дискеты комплекса.

Примерный состав КТД: дискеты класса «целевая» («Администратор» — универсальная система обслуживания, «Построитель меню», «Паскаль», «Макроассемблер», «Фортран», «Р-Паскаль», «Р-Макроассемблер», «Р-Фортран», «Реляционная база данных», «Текстовый процессор», «Информационно-поисковая система»); дискеты класса «прикладная» («Система обучения», «Игротека»).

Процесс построения КТД — трудоемкая работа только на начальном этапе построения первой дискеты класса «целевая», поэтому рекомендуется выбрать дискету «Построитель меню», в которой автоматизированы функции программ «Редактор меню» и «Печать меню», а также реализована «Система обслуживания». Последующие дискеты этого класса разрабатываются с помощью начального переноса (копирования) файлов базовой ОС и «Системы обслуживания» дискеты «Построитель меню», а также новых файлов прикладных ПС целевого назначения, выбранных для данной дискеты. После этих операций с помощью функций дискеты «Построитель меню» производится разработка соответствующих структур меню и отладка их взаимодействия. КТД может разрабатываться и использоваться на совместимых микроЭВМ: СМ-3, СМ-4, СМ-1420, ВУМС, «Электроника 60» и др.

Базовой ОС построения дискет класса «целевая» является ОС RT-11 или совместимая с ней: РАФОС, ФОДОС, ОС ДВК.

Телефон 162-91-11, Москва (с 22 до 23 ч), Попов Александр Леонидович

Статья поступила 8 октября 1987

УДК 681.3.06

Г. Г. Гнездилова, О. А. Гончаров, Г. В. Сенин

К ТИПОЛОГИИ КОМПЬЮТЕРНЫХ ИГР

Согласно «Советскому энциклопедическому словарю» [1] игра — это «вид непродуктивной деятельности, мотив которой заключается не в ее результатах, а в самом процессе. ... тесно связана со спортом, военными и другими тренировками, искусством (особенно его исполнительскими формами). ... Имеет важное значение в воспитании, обучении и развитии детей как средство психологической подготовки к будущим жизненным ситуациям». Это определение, в основном, приложимо и к компьютерным играм.

Сейчас довольно большую часть времени компьютеры затрачивают на игры. По зарубежным источникам свыше 60 % компьютеров приобретается в развлекательных целях. В настоящее время для ЭВМ разработан целый спектр игр, вызывающих интерес у самого широкого круга лиц, от детей двух- или трехлетнего возраста до серьезных ученых или администраторов.

Основы классификации игр

Игровые программы, написанные для компьютеров, можно разделить на три класса: собственно игры, игрушки и живые картинки.

В процессе **игры** человек должен достичь некоторой заранее определенной цели. Достижение этой цели обычно называют выигрышем. Традиционный пример игры — шахматы. Цель игры — поставить мат королю соперника.

В отличие от игр **игрушки** не ставят перед играющим какой-либо определенной цели, но лишь предоставляют ему возможности для манипуляций с ними. В качестве примера традиционных игрушек можно привести обычные детские автомобили. Примером этого класса компьютерных игр могут служить игры «Жизнь» и «Паучья графика».

Игра «Жизнь» моделирует жизнь поколений гипотетической колонии живых клеток, которые выживают, размножаются или погибают в зависимости от того, сколько соседей из восьми возможных оказывается у клетки. Так, например, если у клетки только один сосед или вовсе ни одного, она погибает «от одиночества», если клетка имеет четырех или более соседей, она погибает «от перенаселения», и т. д. Разместив в начале игры клетки исходной популяции, игрок созерцает эволюцию колонии. Существуют конфигурации статические, повторяющиеся, перемещающиеся в определенном направлении. Поражает то, что простейшие правила «жизни» в зависимости от начальных условий порождают чрезвычайно разнообразные комбинации, которые совершенно невозможно предсказать. Иногда это выглядит как довольно сложное «поведение» колонии и вполне заслуживает названия «Жизнь». Играющий в «Паучью графику» порождает узор на экране, управляя перемещением «паука». Построенная картинка отчасти напоминает паутину.

К рассматриваемому классу игр, по-видимому, относятся и недавно появившиеся диалоговые фильмы. Эти фильмы демонстрируются на видеомагнитофоне, подключенном к персональным ЭВМ (ПЭВМ). В ходе демонстрации диалогового фильма зритель имеет возможность вмешиваться в ход развития сюжета с помощью соответствующих команд, подаваемых с ПЭВМ. Чтобы обеспечить эту возможность, сюжет фильма разветвляется на отдельные подсценарии, которые переключаются командами непосредственно в ходе сеанса.

Живые картинки — вид развлечения, бурно расцветший после появления у компьютеров графических терминалов. В живых картинках от человека не требуется почти никакой деятельности. Нужно только смотреть. Прототипами живых картинок можно считать различные виды калейдоскопов. Орнаменты, паркеты, закономерности, действующие в живой природе, математические зависимости, простейшие геометрические фигуры или сложные пространственные тела необычной формы, самые разнообразные объекты могут стать основой картинки, формируемой на экране дисплея. Картинка становится действительно «живой», если в нее встраиваются движущиеся элементы.

Перечислим несколько программ [3].

Программа «Пять кривых» строит на экране кривые, хорошо известные любителям математики: спираль Архимеда, улитку Паскаля, кардиоиду, трилистник и четырехлистник. Рисунок, вычерчиваемый на экране программой «Убегающий квадрат», образован вложенными квадратами. Вершины каждого следующего квадрата делят стороны предыдущего в заданном отношении, тем самым квадраты не только становятся все меньше и меньше, но и поворачиваются на некоторый угол. Узор, который рисует программа «Кружева», образован отрезками, соединяющими каждую вершину правильного восемнадцатиугольника со всеми другими его вершинами. НЛО, созданный программой «Прыгающий НЛО», появляется то в одной позиции экрана, то в другой, изредка издавая звуковой сигнал и меняя цвет при каждом «прыжке».

Наиболее широкий класс игровых программ — собственно игры — можно подвергнуть дальнейшей классификации и выделить: логические игры, игры на «везение», игры на «ловкость», обучающие и тренирующие игры.

Логические игры близки к традиционным математическим загадкам, головоломкам, упражнениям, но облеченным в игровую форму. Отличительная особенность этих игр заключается в том, что вам и вашему сопернику (соперникам) предоставляется возможность выбора из нескольких вариантов: нужно принимать решение. Примерами таких игр могут быть «Ним» и «Коровы и быки». Как и все логические игры, эти игры требуют не столько скорости мышления, сколько его глубины и точности, умения анализировать варианты. Быть может, наиболее приемлемы логические игры для людей, тяготеющих к сосредоточенной мыс-

лительной работе, с небыстрым, но глубоким складом ума.

Играть в «Ним» начинают с трех кучек каких-либо предметов. Ход игрока состоит в удалении произвольного числа предметов из любой одной кучки. Игроки ходят по очереди до тех пор, пока не уберут все предметы. Игрок, сделавший последний ход, выигрывает.

В игре «Коровы и быки» один из играющих задумывает четырехзначное число, другой должен его угадать. На каждом шаге тот, кто отгадывает, называет какое-либо число, загадавший сообщает, сколько цифр числа угадано («коровы») и сколько из них стоит на нужном месте («быки»).

Компьютерные логические игры могут быть разделены на две группы в зависимости от того, какие функции возлагаются в игре на компьютер. В играх первой группы основное назначение компьютера — поддержка необходимой игровой обстановки. Компьютер порождает исходную ситуацию и организует диалог с игроком, обеспечивая ему возможность сделать очередной ход. Обязанностью компьютера являются контроль правильности сделанного хода и соответствующее изменение игровой ситуации. От него требуется также слежение за тем, достигнута ли целевая ситуация. В играх второй группы компьютер помимо всех действий, выполняемых им в играх первой группы, должен еще делать игровые ходы. В этих случаях перед программистом встает проблема, как запрограммировать игру компьютера, как выбрать стратегию, которой тот должен придерживаться. Назовем четыре наиболее употребительных способа программирования компьютера.

1. Оптимальная стратегия, т. е. стратегия, ведущая к победе или, если таковая невозможна, к минимальному проигрышу. Естественно, что такую стратегию можно реализовать только для тех игр, для которых она известна и вычислима за приемлемое время. Примером игры с хорошо изученной выигрышной стратегией является уже упоминавшаяся игра «Ним».

2. Перебор вариантов. Компьютер перебирает возможные варианты продолжения игры и выбирает лучший. Сила переборных программ определяется тем, на сколько ходов вперед может «заглянуть» программа.

3. Перебор с применением эвристик. При переборе вариантов ходов используются некоторые соображения (эвристики), специфические для данной игры, позволяющие отбросить заведомо «плохие» ходы и выбрать «хорошие». Проблема в том, чтобы придумать эвристики. Иногда это удается.

4. Вероятностная стратегия. Компьютер делает очередной ход наугад. Эта стратегия наиболее проста в реализации и часто оправдывает себя в играх, где число допустимых ходов невелико и вероятность случайного выбора оптимального хода высока.

Только первая стратегия употребляется в чистом виде. Остальные, как правило, комбинируются друг с другом. Например, вероятностную стратегию можно сочетать с использованием эвристик: если эвристика применима в данной ситуации, то компьютер делает ход в соответствии с ней, иначе ход выбирается наугад.

Игры «на везение» — это азартные игры: рулетка, очко и т. п. Для подобных игр большого мастерства не требуется, достаточно благосклонности фортуны. Примером компьютерной игры «на везение» может служить игра «Фруктовая машина». Играют в нее так: игрок делает ставку, компьютер наугад выбирает три фрукта. Ставка либо теряется, либо возвращается к игроку, увеличенная в несколько раз, в зависимости от того, какие именно фрукты были выбраны компьютером.

Игры «на ловкость» образуют практически бесконечную совокупность игр в придуманной, искусственной среде с заданными закономерностями. Игры на ловкость очень напоминают обычные подвижные игры, где надо «успеть», «догнать», «убежать», «поймать» и так далее — скорость в них основное. Примером служит игра «Не пересекай». По экрану движутся две ленты. Каждый из двух игроков управляет лентой своего цвета, которая в отсутствие управляющих сигналов движется в одном направлении и поворачивает направо или налево по указанию игрока. Игра заканчивается, когда один из участников «врезается» своей лентой в границу игрового поля или в ленту противника.

Другим примером может служить игра «Декатлон» — пример удачного моделирования спортивных реалий средствами персонального компьютера. На экране отображается обстановка, в которой проводится один из видов спортивного десятиборья. Несколько клавиш заменяют играющему, «спортсмену», части тела и мускульную силу. Привлекательно в игре то, что она хорошо отражает наиболее критические моменты состязания. Спринтерский бег — это попеременное (нет, не отталкивание ногами) нажатие на клавиши: правая, левая, правая, левая. Чем чаще, тем лучше. Перемещение условного бегуна на экране и дистанция подобраны так, чтобы даже самый резвый не побил мирового рекорда. Барьерный бег сложнее, сложнее и действия «бегуна» на клавиатуре. Кроме самого факта нажатия на клавишу учитывается и его длительность, что интерпретируется как величина усилия, влияющего на преодоление барьера. Еще сложнее прыжки с шестом: нужно быстро разбежаться (элемент спринта), вовремя поставить шест (особая клавиша), не переставая при этом бежать, наконец, оттолкнуть шест, перелетев через планку. Всего четыре клавиши, но нажимать их нужно согласованно, в нужные моменты времени, а в совокупности сложность манипуляций на клавиатуре, если не может сравниться с реальным прыжком, то все же дает хорошее представление о действиях шестовика.

Игры, тренирующие и обучающие, правильнее определить как обучение или упражнение в игровой обстановке. Тренирующие игры способствуют отработке каких-либо навыков, которыми игрок уже обладает. Примером тренирующих игр, или, как еще называют такие игры, игр-тренажеров, могут служить «Тренировка в счете» и «Мешанина». Первая из них использует вычислительные способности компьютера и имеет все основания заменить учителя-человека, для которого проверка арифметических знаний ученика является удручающе однообразной работой. Идея игры прозрачна. «Ученику» предлагается избрать одно из арифметических действий, в котором он будет практиковаться: сложение, вычитание, умножение или деление. Далее программа выбирает случайным образом числа из заданных обучаемым диапазонов и использует их как операнды в арифметическом действии. Обучаемый должен правильно назвать результат. Проверив правильность ответа, программа выдает соответствующее уведомление и повторяет упражнение.

Игра «Мешанина» заключается в том, что игрок разгадывает слово, буквы в котором переставлены случайным образом. На отгадывание отводится ограниченное время. Неверные попытки не учитываются. Играют поочередно два игрока. Побеждает тот, кто затратил на заданное количество слов меньшее время. Вариант игры, выполненный на материале английской лексики, может оказаться полезным при изучении английского языка. Игра развивает «лексическую интуицию», активизирует знание английских диграфов, помогает улучшить правописание. Вариант игры на русском лексическом материале полезен для младших школьников.

Обучающие игры помогают приобрести новые знания и навыки, полезные в практической деятельности. Например, игра «Анаконда» помогает изучать клавиатуру пишущей машинки. В этой игре змея, состоящая из алфавитно-цифровых символов, гонится за своим хвостом. На хвосте сидит человек, задача которого уничтожить змею, отщипывая от нее символ за символом. Если скорость, с которой человек нажимает на соответствующие клавиши, мала, то змея, постепенно удлиняясь, догоняет свой хвост и уничтожает человека.

Игра «От одного до десяти» помогает малышу запомнить, в каком порядке идут числа от одного до десяти. Вопрос, который в каждый ход предлагает программа, выглядит так: «Какое число идет за N?» (N — это любое число от 1 до 9). Даже если ребенок пока еще не умеет читать, сыграв с вашей помощью в эту игру несколько раз, он легко научится понимать, что у него спрашивают. Правильный ответ награждается призом — красочной картинкой, неправильный — влечет повторное задание того же самого вопроса.

Подклассом обучающих игр являются *развивающие игры*. Свое название они заслужили благодаря тому, что эффективно развивают наиболее важные и фундаментальные с точки зрения умственного развития способности и навыки, такие как сообразительность, умение сосредоточиться, умение мыслить логически, пространственное воображение, умение искать закономерности, память. Приведем лишь один пример. Одной из первых игр, ориентированных на реализацию способностей к поиску закономерностей, является игра «Приключение», сделанная в Массачусетсском технологическом институте (США) приблизительно в 1975 г. Авторы: Will Crowther и Don Woods. Цель игры — исследование пещеры и поиск ценностей, спрятанных там. «Пещера полна опасностей. Гигантская змея, тролль и дракон препятствуют игроку на его пути. Банда кровожадных тварей практикуется в метании ножа в его незащищенное тело. Пират-клептоман постоянно ворует у него найденные им драгоценности. Но хуже всего то, что его лампа выгорает по прошествии определенного времени, а исследование пещеры в темноте — самоубийство» [4]. В этой игре полный перечень и поведение игровых объектов не сообщаются игроку. Он сам в процессе игры должен выяснить: сколько их, как их найти и как с ними бороться.

Организация игрового процесса

Рассмотрим игры пошаговые и игры реального времени. *Пошаговые игры* состоят из последовательности дискретных шагов (ходов). Время между ходами никак не влияет на течение и исход игры. Примером пошаговых игр являются компьютерные варианты традиционных настольных игр. В играх *реального времени*, аналогом которых служат подвижные игры, отдельные ходы, как правило, не рассматриваются (вряд ли стоит говорить о сделанном ходе в игре «Декатлон» при попытке взять очередную высоту). В играх реального времени играющий управляет поведением объекта в игрушечной среде, постоянно оказывая на него управляющее воздействие (скачущая по ипподрому лошадь делает очередной прыжок только по специальному приказу, убегающий от врага — еще один шаг направо только после того, как нажата соответствующая клавиша и т. д.).

Различие между пошаговыми играми и играми реального времени проявляется более всего при реализации. В пошаговых играх особое внимание уделяется стратегии, которой должен придерживаться компьютер-соперник (если он сам играет, а не создает игровую ситуацию). В играх реального времени наибольшее внимание уделяется созданию разнообразной, увлекательной и красочной среды обитания игрушечного персонажа. Кроме того, большое значение для них имеет своевременная реакция компьютера на управляющие воздействия игрока, поскольку исход игры часто зависит от того, какое время проходит между двумя последовательными нажатиями на клавиши.

В ряде игр их сложность зависит от уровня подготовки ее участников: чем успешнее реализуется игра, тем труднее становится играть. Среди таких игр можно дополнительно различать игры с несколькими уровнями сложности, усложняющиеся и адаптирующиеся. В играх с несколькими уровнями сложности игрок сам может выбрать уровень, на котором он будет играть. Сыграв несколько партий, можно изменить уровень сложности на более подходящий. В усложняющихся играх игроку, успешно закончившему партию, дается возможность сыграть еще одну партию на более высоком уровне сложности. Таким образом, игрок проводит столько партий, сколько уровней он сумеет преодолеть. Такой способ изменения уровня сложности не является наилучшим, поскольку основную массу времени игрок вынужденно проводит в несложных для себя партиях, что в конце концов может вызвать у него раздражение.

По-иному обстоит дело в адаптирующихся играх. Здесь уровень сложности может не только увеличиваться, но и уменьшаться в зависимости от качества игры человека: игра всегда проходит на уровне сложности, примерно соответствующем уровню игрока, что создает у него положительный настрой, желание сыграть еще раз. Такая обстановка делает игру наиболее привлекательной.

Существуют игры конечные, в которых игрок может победить, и игры бесконечные, в которых победа никогда не наступает. Конечные игры обладают одним недостатком. Если в такой игре человек все время побеждает, то она становится для него неинтересной, слишком простой. Так же быстро надоедает игра, если человек все время проигрывает. Поэтому конечные игры проигрывают в конкуренции бесконечным. Побеждая на очередном этапе бесконечной игры, человек чувствует возросшее мастерство. С другой стороны, сколько бы он ни играл — окончательная победа невозможна, и всегда остается уверенность, что он может улучшить свое последнее достижение. Поэтому бесконечные усложняющиеся игры в течение длительного времени притягивают к себе игроков.

Еще один возможный способ деления компьютерных игр на группы — рассмотрение игр со случайными событиями и детерминированных игр. Для того чтобы игра стала действительно интересной, она должна обладать непредсказуемостью, неповторяемостью. В детерминированных же играх каждая игровая ситуация является единственным возможным следствием предшествующей ей ситуации и управляющего воздействия игрока. Игрок быстро изучает ведение игры, развитие игры перестает для него быть неожиданным и игра становится скучной. Это не страшно, только когда собственное поведение игрока (а значит, и общее течение игры) подвержено неизбежной случайности. Нельзя, например, повториться при игре в «не пересекай». Никогда пальцы не смогут выдержать жесткий ритм нажатия клавиш, точного соблюдения их последовательности и интервалов между ними. Конечно, общая стратегия может выдерживаться более-менее неизменной, и рисунок игры будет напоминать прежние, но все же картина каждый раз будет хоть чуть-чуть, а иной.

Во многих играх, однако, можно легко повторять свои ходы (этим, в частности, характеризуются пошаговые игры). В этих случаях важно, чтобы не повторялась программа-партнер. Разнообразие поведения игровой программы достигается, как правило, за счет случайного выбора последующей игровой ситуации из нескольких возможных (разумеется только в тех играх, в которых это возможно). Именно такие игры называются играми со случайными событиями. Примерами случайных шагов в действиях ЭВМ могут служить: выбор «поля действия» (точки на экране) посредством случайного задания его координат, случайная смена движения объекта на экране (вправо-влево-вверх-вниз), случайный выбор из нескольких заготовленных сбъектов (природа которых может быть самой разной: числа, рисунки, изображения предметов и т. п.), случайная смена цвета, неожиданный звуковой сигнал и т. п.

Разнообразие игровых ситуаций, к которому, как

правило, стремится большинство авторов компьютерных игр, не позволяет игре приедаться, однако оно не должно быть неуправляемым. Желательно, чтобы новые черты появлялись в игре при определенных условиях и в определенном порядке. Тогда у игрока будет возможность привыкнуть к новому и потренироваться в новой ситуации. Идеальным было бы возникновение по мере возрастания сложности игры совершенно новых объектов, живущих по новым законам и по-новому взаимодействующих с другими объектами и игроком. Однако это плохо увязывается с детерминированным характером программирования, которое может обеспечить лишь конечный набор объектов и ситуаций, не исключая и «случайное» (а фактически псевдослучайное) поведение объектов. На практике выходом из положения служит конечный набор ситуаций, расположенных по мере возрастания сложности (причем таким образом, чтобы выигрыш в последней ситуации был практически недостижим).

Заметим, что в настоящее время наиболее распространенным типом игр являются игры реального времени (бесконечные, усложняющиеся, недетерминированные). Почему именно такие игры создаются в расчете на широкого потребителя? Игры реального времени в большей степени захватывают внимание игроков, чем пошаговые, поскольку в них нельзя отвлекаться. Бесконечные игры более азартны, чем конечные, поскольку выигрыш, кажущийся достижимым, все время ускользает. Усложняющиеся — способны удовлетворить запросы более широкого класса игроков, чем игры с несколькими уровнями сложности, а реализуются они проще, чем адаптирующиеся. Недетерминированные игры, при прочих равных условиях, обладают бо́льшим разнообразием, чем детерминированные.

Как устроены игровые программы

В компьютерных играх зародилась, прошла обкатку, а затем проникла в серьезные программы идея «прямого управления обстановкой», получившая особенно широкое применение на персональных компьютерах. Традиционные способы игры с помощью компьютера основывались на языке команд и на текстовых описаниях обстановки игры. Это была как бы не сама игра, а имитация ее. Влиять на ситуацию игрок мог только через явно выраженного посредника, исполняющего его приказы.

Такая игра имела пошаговый и опосредованный характер, больше ориентировалась на аналитические способности игрока.

В современных игровых программах обстановка игры выглядит на экране как графическая, цветная, живая картинка, объекты которой перемещаются «по правилам игры», а некоторые из объектов поддаются прямому управлению с помощью небольшого числа клавиш. В первую очередь, это сам участник, точнее, объект (субъект), с которым он отождествляет себя на экране: он убегает, преследует, достигает, отступает и погибает. Это непрерывная игра в реальном времени. Игрок может уйти в нее с головой, принимая вымышленный мир за реальный, дисплей — за окно в этот мир, ощущая клавиши компьютера как продолжение своих пальцев.

Тот же принцип, пусть не в столь острой форме, кладется в основу серьезных программных продуктов. Например, чтобы удалить файл, можно просто указать на его условное изображение на экране, а затем «подтащить» его к изображению урны. Такие приемы облегчают общение с программой, а значит, увеличивают круг пользователей ЭВМ. В настоящее время принцип прямого наглядного управления широко используется в так называемых оффисных системах, предназначенных для автоматизации делопроизводства.

Строгих принципов написания игровых программ не существует, но некоторые общие соображения могут облегчить составление программы, а главное, ее дальнейшие изменения и усовершенствования.

Никакая игра не существует без правил. Это утверждение в равной (если не в большей) степени относится к играм компьютерным. В игровой программе правила носят даже более строгий и четкий характер. При этом сама программа следит за тем, как игрок (ученик, пользователь) их выполняет. Во всякой игре играющему предлагается определенная обстановка и правила поведения в рамках этой обстановки. Это и есть, собственно, то, что называют правилами игры и что составляет ее неотъемлемую внутреннюю характеристику. С точки зрения программы (точнее, программиста) правила — это набор процедур, определенных над структурами данных (наборами переменных). Процедуры и данные в совокупности воплощают в себе «внутренний мир» игровой программы. В процессе игры внутренний мир постоянно меняется: во-первых, по своим собственным законам, во-вторых, под воздействием игрока (игроков). Таким образом, в каждый момент времени игра находится в некотором состоянии, что игрок ощущает как текущую обстановку. Ей соответствует состояние внутренних переменных — наполнение структур данных программы.

Выделим следующие компоненты игровой программы: преамбула (начальная стадия), основной цикл, завершающие действия. Структуры данных определяются на начальной стадии. Иногда они очень просты (несколько целых чисел), в наиболее же интересных компьютерных играх они, как правило, имеют довольно сложную форму. На начальной стадии экран занимается начальной «позицией» игры (часто картинкой, но в простых случаях это могут быть те же числа). Начальная позиция, в сущности, представляет собой визуальный эквивалент начальных значений структур данных.

В преамбулу выносятся также приветствия программы игроку (при этом может сообщаться имя ее автора и дата создания), а также вопрос о том, нужны ли правила. При утвердительном ответе игрока на последний вопрос программа выдает на экран объяснительный текст. Если ответ отрицателен (обычно в случае, если игра ведется не первый раз и правила твердо усвоены), то игра переходит на следующую стадию.

Основной цикл — это та часть программы, где собственно и происходит игра. Здесь параллельно и согласованно изменяются внутренние структуры данных (по правилам игры) и перестраивается их отображение на экране. Главной задачей программы в основном цикле является поддержка диалога с игроком. В самом общем случае программа принимает от него управляющие воздействия, выражающие его поведение в игре. (Иногда эти воздействия можно трактовать как ходы.) Управление чаще всего происходит в форме нажатий на определенные клавиши клавиатуры. Другие устройства ввода (джойстик, мышь) используются реже как более экзотичные.

После того как программа восприняла управляющие воздействия игрока, необходимо произвести их анализ, изменить содержание внутренних структур данных (по правилам игры) и отобразить эти изменения на экране дисплея, чтобы игрок увидел результат своих действий в виде новой ситуации. Основной цикл (и все упомянутые действия программы) повторяется много раз до тех пор, пока не будет достигнуто условие окончания игры (также проверяемое внутри цикла). Таким условием может быть некоторая конечная позиция, определенная клавиша («конец игры»), ограничение на число повторений (шагов, или ходов игры) и др. Рассмотренная структура — лишь приблизительная и достаточно сильно варьируемая в деталях для конкретных программ.

(Окончание. Начало см. на с. 12)

Современные психологи считают, что игровая обстановка является наилучшей для обучения практически любому виду деятельности. Стремление добиться поставленной в игре цели стимулирует игрока напрячь свои силы, активизировать способности. Ранее говорилось об обучающих компьютерных играх. Их познавательный потенциал неоценим. Игры, возникшие как модель ситуации, взятой из реальной жизни, помогают изучать эту ситуацию и получать необходимые навыки. Те из них, которые воспроизводят (иной раз довольно неплохо) социальные, экономические, логические, биологические и другие закономерности, позволяют познавать их в игровой обстановке. Однако обучают не только игры, названные обучающими. Рассмотрим, например, качества, которые требуются от игрока для успешной игры в пошаговые игры и в игры реального времени. Игры реального времени требуют быстрой и правильной реакции на происходящие события, чувства времени, умения точно выдерживать заданные временные интервалы, способности следить за несколькими объектами одновременно. Пошаговые игры — знания алгоритма выигрыша или ведущих к нему эвристик, умения искать закономерности, способности к быстрому и максимально полному перебору основных вариантов, памяти на события, произошедшие в текущем и в прошлых сеансах игры. Как правило, все компьютерные игры, за исключением, пожалуй, немногочисленного класса игр «на везение», требуют от игрока умения сосредоточиться, способности интенсивно работать на протяжении иногда довольно длительного времени. Развитие у игрока всех упомянутых качеств является безусловным достоинством компьютерных игр.

Игрок должен хорошо владеть средствами управления ЭВМ, быстротой и точностью манипуляций. Играть с ЭВМ — значит неформально осваивать различные виды деятельности, связанные с использованием компьютеров, создавать предпосылки свободного владения ими. Однако компьютерные игры не только учат нажимать клавиши, но вводят в мир ЭВМ с его собственными законами и ограничениями, в мир, в котором «границы реальных возможностей человека впервые прямо соприкасаются с границами его фантазии» [2]. Широкое внедрение компьютерной техники почти во все сферы производственной деятельности требует, чтобы практически любой человек умел общаться с компьютером. С этой точки зрения игры являются незаменимым подспорьем. Один час увлекательной, правильно подобранной игры даст человеку гораздо больше информации о возможностях компьютера и способах взаимодействия с ними, чем недельное штудирование руководств и инструкций. Тем самым, компьютерные игры, вероятно, позволят наиболее безболезненным образом преодолеть психологический барьер, до сих пор отделяющий широкий круг потенциальных пользователей ЭВМ от необходимых им вычислительных ресурсов, будут содействовать массовой компьютеризации общества.

Телефон 135-13-40, Москва

ЛИТЕРАТУРА

1. Советский энциклопедический словарь.— М.: Советская энциклопедия, 1986.
2. Громов Г. Р. Игровая компонента персональной ЭВМ: стимулятор творчества, педагогический прием, жанр киноискусства // Микропроцессорные средства и системы.— 1987.— № 3.— С. 7—10.
3. Гнездилова Г. Г., Гончаров О. А., Сенин Г. В. Персональный компьютер в играх и задачах.— М.: Наука, 1988.— С. 219.
4. Johnson F. Fun and Games. In "IBM Personal Computer Handbook",—California Press: Berkley, 1983.

Статья поступила 1 марта 1988

За создание и внедрение автоматизированных комплексов была присвоена Государственная премия СССР за 1986 год группе специалистов РПО ВЭФ: Андерсону А. И., Глазову А. П., Дзенису Ю. Ж., Паберзу И. П., Кривченкову А. А., Пога И. К., Рекису Д. М., Червинскому Ю. К.

Микропроцессорные комплексы, разрабатываемые на РПО ВЭФ, используются пока только на промышленном производстве предприятия.

Адрес для запроса технической документации: 226039, Рига, РПО ВЭФ; тел. 36-32-36, Лещеева Елена Константиновна

ЛИТЕРАТУРА

1. Андерсон А. И., Вентиньш Я. Я., Виденискс П. О., Кривченков А. А., Лавров С. Н., Павловский В. В., Пекаревич В. Э., Червинский Ю. К. Микропроцессорные системы контроля и управления.—Рига: Зинатне, 1981.
2. Андерсон А. И., Глазов А. П., Кривченков А. А. Особенности организации АСУ промышленного предприятия с использованием микропроцессорных систем.— В кн.: Вычислительная техника социалистических стран.— М.: Финансы и статистика, 1984.— Вып. 15.— С. 154—160.
3. Виденискс П. О., Кривченков А. А., Червинский Ю. К. Персональные ЭВМ типа «ВЭФ-МИКРО».— В кн.: Тезисы докладов Всесоюзной научно-технической конференции по проблемам создания индивидуальных диалоговых систем на базе микроЭВМ.— Пущино: ОНТИ НЦБИ АН СССР, 1982.— С. 9—12.
4. Таланов В. А. Советский микропроцессорный комплекс: премия за «интеллект» // Советская панорама.— АПН.— 1984.— № 98.
5. Таланов В. А. Что могут микропроцессоры // ВДНХ СССР.— 1984.— № 12.

Статья поступила 4 октября 1986

Кооперативный Союз «Информатика»

30 октября 1988 г. в Московском доме научно-технической пропаганды им. Ф. Э. Дзержинского состоялся Первый съезд кооператоров страны, работающих в области информатики. Съезд учредил Кооперативный Союз «Информатика».

Основа деятельности Союза — содействие информатизации общества, создание и распространение средств вычислительной техники, сетей передачи данных, увеличение производства программного продукта и т. д.

Союз объединит творческие и производственные ресурсы в кооперативных и индивидуальных формах, сконцентрирует их на решении ключевых задач информатизации, выполнении государственных заказов и оказании услуг кооперативам, организациям и населению.

Союз обеспечит своих членов передовой техникой и технологией, поможет изучить состояние рынка информатики, организовать рекламу результатов своей деятельности, обмен опытом, продвижение своих товаров на внешний рынок, защиту авторских прав на программный продукт, подготовку кадров и т. д.

Основным источником средств Союза являются доходы от его хозяйственной деятельности. Членами Союза могут стать как коллективные, так и индивидуальные участники.

Первый съезд кооператоров принял Устав и выбрал руководящие органы Кооперативного Союза «Информатика». Президентом Союза избран **Корчагин Вячеслав Владимирович**, председателем правления Союза — **Карась Илья Зиновьевич.**

681.142.2

А. В. Погода

ПАКЕТ ПОДПРОГРАММ НА АССЕМБЛЕРЕ ДЛЯ АРИФМЕТИЧЕСКИХ ВЫЧИСЛЕНИЙ

Хотя подпрограммы (ПП) арифметических действий над числами в формате с плавающей запятой для микропроцессора КР580ИК80А известны и входят в состав почти всех языков высокого уровня, тем не менее, предлагаемый пакет, вероятно, заинтересует тех, кто разрабатывает устройства, функционирующие в реальном времени, и сталкивается с недостаточным быстродействием программы.

Дело в том, что обычно ПП арифметических действий написаны для чисел, имеющих 24-разрядную мантиссу и 8-разрядный порядок. Время их выполнения процессором КР580ИК80А не менее 2...4 мс при тактовой частоте 2 МГц. Для многих же практических задач такая точность представления чисел избыточна и ею можно пожертвовать в угоду быстродействию.

Предлагаемый пакет разработан для чисел, имеющих 16-разрядную мантиссу и 8-разрядный порядок. Диапазон представления чисел примерно $6\times10^{39}...3\times10^{38}$, относительная точность представления не хуже 3×10^{-5}. При создании ПП целью ставилось достижение максимального быстродействия, даже если это шло в ущерб объему программы.

В пакет входят 12 ПП (см. таблицу). Общий объем пакета 4492 байт.

Для ускорения вычислений квадратного корня, синуса, косинуса и арктангенса используются таблицы и линейная интерполяция. При этом относительная точность результатов не хуже 6×10^{-5}. При выполнении преобразований чисел (ПП EDH и DEH) также используются таблицы со значениями констант 10^N.

На основе предлагаемого пакета были разработаны программы для вычислителей, управляющих объектами в реальном времени. Для записи формул использованы макросредства языка ассемблера, при этом одно действие над операндами записывается одной строчкой и программа получается достаточно наглядной. Сравнение разработанных программ с эквивалентными, полученными с помощью компилятора языка БЕЙСИК, показало, что первые работают в 5...20 раз быстрее и требуют в 2...4 раза меньшего объема памяти. Кроме того, программист создает программу оптимальнее, чем компилятор.

Общий объем текста 12 ПП — 1760 строк, поэтому привести его полностью не представляется возможным. Для иллюстрации распечатаны две подпрограммы, входящие в состав пакета.

Подпрограмма RRH сдвигает вправо исходное 16-разрядное число, передаваемое в регистр HL. В регистр A передается число сдвигов (если A=0, то должен быть установлен флаг ZF). Результат записывается в HL. Флаг ZF равен нулю, только если HL равно нулю. ПП изменяет регистры A, C, F. Время выполнения ПП не более 219 тактов.

Подпрограмма APR используется для табличной аппроксимации функции. Входные данные для нее: DE — адрес таблицы 2-байт значений функции (ее объем не более 514 байт), HL — значение абсциссы. Выходное значение функции передается через регистры HL. ПП изменяет регистры D, E, A, F. Время выполнения ПП не более 506 тактов. ПП BMUL умножения содержимых регистров BC и A с помещением результата в регистры A, H, L приведена в работе [1]. Время выполнения ее не более 314 тактов. При разработке ПП MUL и DIV использовались работы [2, 3].

Характеристики ПП

Имя ПП	Максимальное время выполнения, такты	Объем ПП, байт	Объем таблицы, байт	Примечание
MUL — умножение чисел	905	204	—	
SUM — сложение	785	283	—	
SBT — вычитание	799	6	—	Обращается к ПП SUM
DIT — деление	1610	138	—	
SQR — вычисление квадратного корня	622	141	512	
SIN — синус	1166, $\lvert X\rvert<\pi/2$	176	918	Обращается к ПП SQR
COS — косинус	1923	19	—	Обращается к ПП SIN и SUM
ATN — арктангенс	1004, $\lvert X\rvert<1$ 3352, $\lvert X\rvert>1$	105	1032	Обращается к ПП SIN и SUM
CPX — сравнение чисел	125	29	—	
FIX — преобразование числа из формата с плавающей запятой в целое	474	54	—	
EDN — преобразование десятичной записи числа во внутреннее представление		335	258	
DEN — обратное преобразование		282		

```
RRH:                    ;ПП СДВИГА HL ВПРАВО НА A РАЗРЯДОВ
        RZ              ;ФЛАГ ZF ДОЛЖЕН БЫТЬ УСТАНОВЛЕН
RRH1:   CPI 0FH
        JNC     RRH15   ;ПРИ ЧИСЛЕ СДВИГОВ 15 И БОЛЕЕ
        SUI 8
        JNC     RRH14   ;ПРИ ЧИСЛЕ СДВИГОВ 8...14
        CPI 0FCH
        JNC     RRH7    ;ПРИ ЧИСЛЕ СДВИГОВ 4...7

        ADI 8           ;ЧИСЛО СДВИГОВ 1...3
        MOV C,A
RRHC3:  ORA A           ;ЦИКЛ НЕ БОЛЕЕ ТРЕХ РАЗ
        MOV A,H
        RAR
        MOV H,A
        MOV A,L
        RAR
        MOV L,A
        DCR C
        JNZ     RRHC3
        RET             ;HL - РЕЗУЛЬТАТ, ZF=1

RRH7:
        MOV C,A
        SUB A
RRHC4:  DAD H           ;ЦИКЛ НЕ БОЛЕЕ ЧЕТЫРЕХ РАЗ
        ADC A
        INR C
        JNZ     RRHC4
        MOV L,H
        MOV H,A
        RET             ;HL - РЕЗУЛЬТАТ, ZF=1

RRH14:
        MOV L,H
        MVI H,0
        RZ              ;ПРИ ЧИСЛЕ СДВИГОВ РАВНОМ 8
        MOV C,A
        MOV A,L
RRHC6:  ORA A           ;ЦИКЛ НЕ БОЛЕЕ 6 РАЗ
        RAR
        DCR C
        JNZ     RRHC6
        MOV L,A
        RET             ;HL - РЕЗУЛЬТАТ, ZF=1

RRH15:
        JNZ     RRH16   ;ПРИ ЧИСЛЕ СДВИГОВ БОЛЕЕ 15
        DAD H
        LXI H,1
        RC              ;ЕСЛИ СТАРШИЙ БИТ HL РАВЕН 1
        DCX H
        ORA A
        RET             ;HL=0, ZF=0

RRH16:  LXI H,0
        RET             ;HL=0, ZF=0
APR:                    ;ПП ТАБЛИЧНОЙ АППРОКСИМАЦИИ
        PUSH B          ;HL - АБСЦИССА
        MOV B,L         ;DE - АДРЕС ТАБЛИЦЫ
        MOV A,H
        ADD A
        MOV L,A         ;L=2*H
        MVI H,0
        DAD D           ;HL - АДРЕС В ТАБЛИЦЕ
        MOV E,M
        INX H
        MOV D,M         ;DE - ЗНАЧЕНИЕ ФУНКЦИИ ПРИ L=0
        PUSH D
        INX H
        MOV A,M
        SUB E
        MOV C,A
        INX H
        MOV A,M
        SBB D
        MOV E,B
        MOV D,A         ;DC - ПРИРАЩЕНИЕ ОРДИНАТЫ
        MOV A,E         ;A=L(ВХОДНОЕ)

        CALL    BMUL    ;AHL=BC*A

        MOV L,H
        MOV H,A         ;HL - ПОПРАВКА ОРДИНАТЫ
        POP B
        DAD B
        POP B           ;BC СОХРАНЯЕТСЯ, DE ИЗМЕНЯЕТСЯ
        RET             ;HL - ВЫХОДНОЕ ЗНАЧЕНИЕ
```

Те, кто заинтересуется пакетом, могут переписать на гибкий диск текст подпрограмм и пояснения к ним с примером. Общий объем текстовой документации — около 43 Кбайт, система — ОС 1800.

Адрес: 249020, Калужская обл., Обнинск, ул. Курчатова, 15, кв. 69; тел. 2-31-51

ЛИТЕРАТУРА

1. Глазов Л. Б., Костарев С. А., Сухапова Е. В. Эффективные программы умножения для микропроцессора КР580ИК80А // Микропроцессорные средства и системы.— 1986.— № 5.— С. 43—44.
2. Гудрич Джерри Л. Очень эффективная программа умножения и деления для микропроцессора 8080 // Электроника.— 1982.— Т. 55.— № 4.— С. 74—75.
3. Бернар Купе. Сверхэффективные программы умножения для микропроцессоров 8080 и Z80 // Электроника.— 1983.— Т. 56.— № 6.— С. 74—76.

Статья поступила 30 сентября 1987

УДК 681.3.06

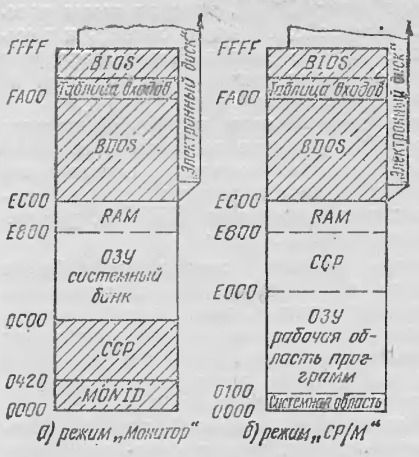
Д. А. Журавский, С. Н. Барков

ОПЕРАЦИОННАЯ СИСТЕМА CP/M В ПЗУ

Наиболее распространенная система для малых ЭВМ — система CP/M. Используемые версии ОС загружаются с магнитных дисков и должны иметь в качестве системного устройства носитель с прямым доступом. Однако во многих микроЭВМ такой носитель отсутствует. Для обеспечения удобной работы с такими компьютерами разработана данная версия ОС, размещаемая в ПЗУ объемом 8 Кбайт (типа К573РФ6, КМ558РР3 или 4 корпуса типа К573РФ2, К573РФ5). Эта ОС может также использоваться и с микроЭВМ, имеющими дисковые накопители. Так как наиболее ответственные операции по работе с внешними устройствами производятся через BIOS (секция ввода-вывода ОС), то помещение его в ПЗУ также повышает надежность работы ЭВМ в целом.

Возможности разработанной ОС: все возможности ОС CP/M версии 2.2; полноценная работа ПО, выполненного для CP/M 2.2 фирмами Микрософт, IBM и др.; работа с кассетным магнитофоном в качестве устройства, поддерживаемого системой; работа с телевизионным адаптером в качестве системной консоли с графическими возможностями; загрузка по линии связи и запуск программ; встроенный редактор командной строки; работа с пятью дисковыми устройствами («электронный диск», два 8-дюймовых и два 5-дюймовых накопителя); работа в режиме отладочного монитора, аналогичного мониторам таких ПЭВМ, как «Микроша», «РК-86» и др.

Область, занимаемая ПЗУ в адресном пространстве ЭВМ (см. рисунок), разбита на две части: нижняя область занимает 3 Кбайт и размещается по адресам ЭВМ от 0000H до 0BFFH,



Карта памяти

а верхняя — 5 Кбайт и размещается на адресах от 0EC00H до 0FFFFH. Каждая из двух частей в свою очередь разделена еще на две части: в верхней части находятся две резидентные компоненты ОС: BIOS (с 0FA00H по 0FFFFH — 1,5 Кбайт) и BDOS (с 0EC00H по 0F9FFH — 3,5 Кбайт), а в нижней — подключенные постоянно только в режиме монитора (см. рис. *б*): собственно монитор MON (с 0000 по 041FH — более 1 Кбайт) и процессор консольных команд, необходимый для работы в режиме ОС — CCP (с 0420H по 0BFFH — около 2 Кбайт), транслированный на адреса с 0E000H по 0E7FFH. Область памяти с 0E800H по 0EBFFH занята под ОЗУ, используемое в системных целях: буфера ввода-вывода и системные ячейки памяти. Таким образом, занимаемый объем памяти в режиме монитора — 8 Кбайт, а в режиме ОС — 5 Кбайт.

Еще одна особенность этой ОС: 5 Кбайт, постоянно занимаемые микросхемой ПЗУ, содержащей ОС, в адресном пространстве процессора, отключаются при операциях обращения с «электронным диском». Тем самым обеспечивается непрерывность ОЗУ по всему расширенному 20-разрядному обслуживаемому адресному пространству. После включения питания или сброса (рестарта) процессора устанавливается режим монитора, при этом производится определение объема подключенной памяти, назначение банка системной памяти и банков памяти для «электронного диска»; устанавливаются параметры «электронного диска» как системного устройства в соответствии с объемом подключенной памяти.

В режиме монитора имеются следующие возможности, поддерживаемые его командами:

D — просмотр области памяти,
F — заполнение области памяти константой,
S — просмотр и изменение содержимого памяти побайтно,
M — пересылка области памяти с возможностью пересечения исходной и результирующей областей,
C — сравнение областей памяти,
P — просмотр содержимого порта и запись в порт,
R — чтение файлов с магнитофона,
W — запись области памяти в файл на магнитофон,
Z — загрузка программ по линии связи,
U — чтение и запись физических секторов с дисковых устройств,
G — запуск программ по заданному адресу,
I — инициализация каталога дискового устройства,
L — запуск ОС с указанием логического имени системного диска (переход в режим ОС).

Встроенные команды в режиме ОС:

DIR — просмотр каталога дисковых устройств,
ERA — стирание файлов на дисковых устройствах,
SAVE — запись области памяти в файл на дисковые устройства,
TYPE — распечатка текстового файла с дисковых устройств,
READ — чтение файла с магнитофона и запись его в файл на дисковые устройства,
LD — загрузка файла по линии связи и его запись на дисковое устройство,
ST — повторный запуск программы, находящейся в памяти,
REN — переименование файлов на дисковых устройствах.

Описанные возможности разработанной системы являются расширенной версией, поставляемой с серийно выпускаемой микроЭВМ СМ1800 операционной системы ОС 1800 (1,2), загружаемой в нее с дискового носителя. Адаптация системы для работы в ПЗУ потребовала некоторого изменения распределения областей памяти среди системных компонент, составляющих систему, но со стороны пользователя интерфейс системы является стандартным и позволяет работать со всеми программами, созданными для CP/M 2.2 (ОС 1800).

Функциями дисковой части ОС не отличаются от стандартных, в терминальной части введены редактор командной строки (аналог SL в RT-11, частично имеется в CP/M 3.0), переназначение физических устройств в логические, увеличено число точек входа в подсистему ввода-вывода для обслуживания магнитофона и поддержки графических возможностей некоторых терминалов, а также для поддержки работы с прерываниями в реальном времени, введены дополнительные команды в процессор консольных команд (CCP), позволяющие работать в минимальной аппаратной конфигурации системы — без дисковых накопителей и расширенной памяти, сам же CCP подгружается и при «холодном», и при «горячем» старте системы не с диска, как в обычной системе, а переписыванием из теневого ПЗУ (нижней области), что существенно ускоряет работу. При работе с контроллерами, аналогичными разработанным авторами, настройка на конфигурацию не требуется, иначе необходима перетрансляция системы.

Разработанная система используется в различных конфигурациях микроЭВМ на основе одноплатного микроконтроллера [3] (различный объем памяти, набор внешних устройств, в том числе с различными дисковыми накопителями и без них) и показала бо́льшие преимущества по сравнению с системой, загружаемой с НГМД.

Телефон 466-17-09, Москва

ЛИТЕРАТУРА

1. Прохоров Н. Л., Ландау И. Я. МикроЭВМ СМ1800 и ее программное обеспечение//Микропроцессорные средства и системы. — 1984. — № 2. — С. 28—30.
2. Гиглавый А. В., Кабанов Н. Д., Прохоров Н. Л., Шкамарда А. Н. МикроЭВМ СМ1800: Архитектура, программирование, применение.— М.: Финансы и статистика, 1984.
3. Щелкунов Н. Н., Дианов А. П. Универсальный одноплатный микроконтроллер// Микропроцессорные средства и системы.—1986.— № 5.— С. 65—69.

Статья поступила 28 марта 1987

Совместные предприятия: принципы функционирования

Читатели интересуются, на каких принципах строятся совместные предприятия. В печати появляется много разнообразных сообщений на эту тему. Заметное внимание этим вопросам уделяет последнее время и весьма популярная среди более миллиона сотрудников столичных НИИ и КБ газета «Московский комсомолец». В начале года газета поместила обширный материал, посвященный одной из первых такого типа московских организаций — «ИНТЕРКВАДРО». Затем тема эта была продолжена (видимо, с учетом более глубокого изучения накопленного опыта) уже в разделе «тяжелого юмора» воскресного выпуска газеты:

НОВОСТИ ТЕХНИКИ. Япония и Арабская Республика Египет подписали договор о совместном производстве электронных часов со светящимся циферблатом: электроника будет японская, а цифры — арабские.

«Московский комсомолец»,
17 июля 1988 г.

УДК 681.326

А. В. Александров

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПО ВЫБОРУ ЭЛЕМЕНТНОЙ БАЗЫ ДЛЯ ТЕХНОЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ

Зачем и почему потребовалась система. Первая задача при разработке контроллеров для управления технологическими процессами и оборудованием — выбор элементной базы. При выборе элементной базы **основным критерием должна быть стоимость разработки.** Чем дешевле будет ваш контроллер, тем более вероятно, что он в новых условиях хозяйственного расчета найдет спрос. Однако совершенно ясно, что стоимость — не самоцель и **при минимальной стоимости контроллер должен решать вашу задачу.** Но и технические характеристики, как бы хороши они ни были, еще не гарантируют широкой применяемости и спроса на ваше изделие. Ваш контроллер должен иметь еще и высокую технологичность, с тем чтобы как можно ниже была его себестоимость. Помимо этого, нужно учесть, что условия могут быть таковы, что высокие технические и технологические качества ему не пригодятся, поскольку он не в состоянии выдерживать сильную тряску или сильный мороз.

Словом, выбор элементной базы контроллера превращается в нелегкую задачу, которая способна поставить в тупик и опытного разработчика, не говоря уже о начинающих.

Помимо всего прочего, эта задача еще и трудоемкая. Во всяком случае, когда в Головной консультационно-технический центр (ГКТЦ) приходит пользователь и просит помочь выбрать микропроцессор, то каждому консультанту известно, что это дело долгое.

Именно желание сэкономить время консультантов и посетителей ГКТЦ явилось побудительным стимулом к началу разработки системы. Значительно более важным фактором явилось желание достичь объективности и большего единообразия в оценках. В самом деле, системный анализ задачи производится консультантом прежде всего на основании своего опыта, и поскольку опыт — категория субъективная, то разные консультанты по-разному оценивают одни и те же факторы и, естественно, дают самые разные рекомендации. И если разнобой при оценке аппаратной части будущей микропроцессорной системы управления (МПСУ) все-таки ограничен существующей номенклатурой, то программное обеспечение (ПО) предоставляет значительно более широкие возможности для импровизации при оценке его характеристик.

Самый простой путь: обобщив опыт большого числа экспертов по данному вопросу, смоделировать деятельность некоего «среднего эксперта» на ЭВМ. Такая система была разработана и запущена в опытную эксплуатацию в ГКТЦ.

Разработка системы. Все, что описывается далее, было сделано в течение полугода двумя программистами (один из них еще и постановщик задачи).

Из всех имевшихся под рукой трансляторов только интерпретатор БЕЙСИКа располагал наиболее мощными средствами работы с текстовыми переменными, что позволило решить задачу в кратчайшие сроки. Поэтому при реализации экспертной системы выбран достаточно экзотический для этих целей язык — БЕЙСИК-МТ операционной системы РАФОС-2.

Работа была разбита на два почти самостоятельных направления: создание программы «обобщенный эксперт» (получившей название «Диалог») и создание СУБД и самой БД, в которой содержалась бы номенклатура микропроцессорных и сопутствующих БИС с краткими техническими характеристиками.

Нами была разработана СУБД, которая имеет простой программный интерфейс, большую емкость и высокое быстродействие. Все ее достоинства проявляются только при работе с базами данных, рассчитанных на чтение и обработку другой программой.

Методика выбора оптимального микропроцессорного комплекта. Оптимальное решение выбирается системой на основе последовательного анализа поставленной пользователем задачи, оценки технических данных микропроцессорной системы управления (МПСУ), удовлетворяющей требованиям потребителя.

Последовательность анализа задачи потребителя в общем одинакова для большинства консультантов ГКТЦ: анализ входных данных, типа задачи, условий работы; оценка длины программы и быстродействия; выбор микропроцессорного комплекта, периферийных устройств, отладочных средств; рекомендации по организационным вопросам проектирования МПСУ.

Для системы был принят следующий порядок: анализ данных по вводу-выводу, типа задачи, данных по быстродействию; рекомендации по выбору АЦП; оценка длины рабочей программы на основании данных пользователя, технических характеристик микропроцессора; выбор микропроцессора; анализ данных для выбора ОЗУ; оценка объема ОЗУ; выбор БИС ОЗУ; анализ данных для выбора ПЗУ; выбор ПЗУ; анализ требований по условиям применения, организационным вопросам, связанным с разработкой и внедрением будущей МПСУ; выбор отладочных средств.

Рассмотрим подробнее отдельные этапы методики.

Анализ данных по вводу-выводу. Микропроцессорная система управления, как это следует из ее определения, предназначена для реализации функций управления внешним объектом.

Наиболее важные параметры, характеризующие МПСУ в этой связи,— число каналов ввода-вывода дискретных, а также аналоговых сигналов; точность аналоговых сигналов.

С точки зрения обработки безразлично, выводится ли дискретный сигнал или вводится в микропроцессор (МП). По времени выполнения эти процессы примерно одинаковы: и дискретный ввод и вывод.

Что касается аналоговых параметров, то, как правило, время, затрачиваемое на вывод аналоговой величины, примерно в 2...6 раз меньше времени приема этой величины. В процессе считывания, особенно если параметры опрашиваются последовательно, в любой программе существует некий логический механизм, опрашивающий параметры и распределяющий считанные значения в памяти. Эта величина (40...120 байт) зависит от соотношения разрядности обрабатываемых данных и разрядности МП.

Исходя из этого, необходимо ввести два параметра: числа аналоговых входов и выходов.

Таким образом, по интерфейсу МПСУ определяются четыре параметра: максимальная разрядность считываемых данных; общее число дискретных входов и выходов, числа аналоговых входов-выходов.

Поскольку система дает оценку задачи, а не ее точное решение, можно не рассматривать ситуацию, при которой точность представления данных по различным каналам различна, и взять максимальную разрядность данных.

В ходе проведения консультаций в ГКТЦ удалось выделить наиболее частые типы задач. Классификация задач следующая: контроль параметров по заданным уставкам с сигнализацией об отклонениях от режима; то же, но с ведением протокола технологического процесса; то же, но с хранением в памяти промежуточных результатов; то же, но с управлением параметрами, управление технологическим процессом по заданным уставкам; управление техпроцессом по закону регулирования, вычисляемому без помощи трансцендентных функций (по «простому закону»); управление техпроцессом в реальном масштабе времени по циклограмме;

управление техпроцессом по «простому закону» в реальном масштабе времени; управление техпроцессом в реальном масштабе времени по закону, вычисляемому с помощью трансцендентных функций.

Практически любая пользовательская задача может быть с небольшими погрешностями сведена к одному из перечисленных выше типов.

Для чего вообще нужна подобная классификация? Наиболее трудоемкий этап в создании МПСУ — разработка ПО, а один из наиболее важных параметров, характеризующий МПСУ,— объем ПЗУ. Точно определить объем требуемого для хранения управляющей программы ПЗУ можно только после разработки. Однако даже опытным программистам трудно предсказать объем будущей программы. Обычно длина будущей программы оценивается по характеру обработки данных, требованиям к сигналам управления, количеству и типу входных данных, сложности задачи (многофункциональности программы).

Рассмотрим последний критерий. Оценивая объем еще не написанной программы, программист исходит только из своего опыта и сведений о длинах некоторых стандартных программ. Примерно прикидывая, что нужно будет использовать в данной программе, опытные разработчики способны оценить объем создаваемой программы с точностью 5...10 %. При этом ошибка тем больше, чем больше разрабатываемая программа, чем больше количество внутренних связей программы. Обычно программист это учитывает и делает поправку на сложность решаемой задачи. Система тоже делает поправку на сложность, именно поэтому возникла необходимость в классификации задач.

Поправка на сложность программы вычисляется так: $Z_2 = M_2 \cdot \ln(Z_1) + 1$, где Z_2 — коэффициент сложности программы; M_2 — коэффициент, учитывающий тип задачи и способный принимать значения из натурального ряда чисел 1...11. Численные значения M_2 находятся во взаимно-однозначном соответствии с рядом, характеризующим тип задачи пользователя; Z_1 — общее число всех входов и выходов МПСУ.

Логарифмическая зависимость между Z_2 и Z_1 выбрана из опытных данных и достаточно хорошо отражает реально существующие связи этих величин.

Анализ данных по критерию «быстродействие МП». Это один из важнейших критериев при выборе МП. Однако более рационально оценивать производительность МП, которая является функцией быстродействия и разрядности МП: $P = f(1/T_5 K)$, где P — производительность МП; $1/T_5$ — частота тактовых импульсов; K — разрядность МП.

Методика рассчитана на выбор «минимально-достаточного» МП. Под минимально-достаточным процессором (МДП) мы будем понимать некий идеальный МП, который в состоянии решить поставленную пользователем задачу. МДП обладает следующим свойством: его разрядность и быстродействие достаточны для решения задачи пользователя, но без запаса по быстродействию и разрядности. Исходя из этой концепции по данной методике необязательно, чтобы пользователь назвал желаемые быстродействие и разрядность. Эти значения вычисляются как функции многих переменных, значения которых в свою очередь являются граничными условиями задачи пользователя. В частности, следует рассмотреть частоту опроса параметров — F, минимально возможное быстродействие, минимально возможную разрядность, времена реакции системы управления на аварийные ситуации, а также на управляющие воздействия.

Значение частоты опроса параметров запрашивается у пользователя.

Минимально возможное быстродействие (50 кГц) эквивалентно среднему значению числа операций в секунду.

За минимально возможную разрядность принимается число четыре (поскольку методика распространяется на МП с фиксированной системой команд).

Времена реакции системы на аварийные ситуации и на управляющие воздействия оцениваются пользователем. Эти величины требуются для дальнейшей оценки быстродействия МП. В методике приняты три класса АЦП: серийно выпускаемые БИС АЦП; АЦП на операционных усилителях с двойным интегрированием (медленные АЦП); АЦП, требующие разработки (если требования заказчика не удовлетворяются вышеприведенными классами АЦП по быстродействию и разрядности).

Оценка длины управляющей программы важна как с точки зрения быстродействия системы, так и с точки зрения выбора ПЗУ. Длина программы оценивается по формуле $L = INT(\ln \cdot (L_n L_1)/100) \cdot 100$, где L — длина программы в адресах (оценка); L_1 — вычисленная длина программы; INT — функция целой части.

Видно, что оценка проводится с точностью до 100 адресов. В применении этой оценки к ОМЭВМ с объемом ПЗУ 1 Кбайт погрешность составляет 10 %, поскольку на практике объем встроенного ПЗУ используется практически полностью.

Длина программы L_1 определяется по формуле

$$L_1 = (Z_2 + M_6 \cdot 2 + 40) \cdot \sqrt{M/K} + M_7 \cdot 15 + M_5 \cdot 10 + L_5,$$

где Z_2 — коэффициент сложности программы; M_6 — число аналоговых входов; M — максимальная разрядность данных; K — разрядность МП; M_7 — число аналоговых выходов; M_5 — число дискретных входов и выходов; L_5 — суммарная длина вычислительных подпрограмм.

Первый член этого выражения характеризует длину программы в зависимости от числа аналоговых входов. Длина управляющей программы слабо зависит от числа опрашиваемых параметров потому, что если входные данные обрабатываются по единому алгоритму или алгоритм обработки группы данных является подмножеством от алгоритма обработки другой группы данных, то длина программы увеличивается лишь за счет увеличения числа команд ветвления к той или иной группе обрабатывающих алгоритмов. Ветвление осуществляется, как правило, одной командой — двухадресной или трехадресной. Отсюда коэффициент 2 при M_6. Константа 40 выбрана экспериментально и численно равна средней длине подпрограммы, принимающей и первично обрабатывающей поступающие данные. Корень квадратный из отношения M/K — это корректирующий коэффициент на соотношение разрядности обрабатываемых данных и разрядности МП. Коэффициенты 15 и 10 при M_7 и M_5 соответственно численно равны средним длинам программ финишной и первичной обработок аналоговых и дискретных данных.

Величина L_5 вычисляется по формуле

$$L_5 = \sqrt{18/K} \sum_{j=1}^{n} L_j X_j,$$

где L_j — длина j-подпрограммы в адресах; X_j — коэффициент «участия» j-подпрограммы (принимает значение 0 или 1); K — разрядность МП; n — число подпрограмм. Выражение $18/K$ — это поправочный коэффициент на разрядность МП. Точность оценки L_5 существенно зависит от числа n. При достаточно большой библиотеке одноименных программ по каждой из трех существующих разрядностей (4, 8 и 16) в перспективе можно отказаться от поправочного коэффициента.

Оценка технических характеристик МП, выбор МП. Технические характеристики требуемого МП оцениваются так: при заданной разрядности вычисляется отношение времен опроса параметров за один полный проход программы в соответствии с требованиями поль-

зователя к времени одного прохода программы, которое получается из расчетных данных:

$$C = T_3 INT(M/K + 0{,}99)/M_6 \cdot T_1 + (W \cdot 300\,000 + M_5 + M_7) \cdot 10) \cdot T_5),$$

где C — отношение требуемого времени прохода программы к реальному времени прохода; T_3 — время опроса одного параметра в соответствии с ТЗ пользователя; K — разрядность МП; M_6 — число аналоговых входов системы; T_1 — вычисленное время аналого-цифрового преобразования; W — суммарное время выполнения базовых подпрограмм; T_5 — быстродействие процессора (среднее время выполнения одной команды); M_7 — число аналоговых выходов; M_5 — число дискретных входов и выходов, INT $(M/K + 0{,}99)$ — корректирующий множитель на соотношение разрядности обрабатываемых данных и разрядности МП.

Вычислив отношение, анализируют его значение. Если «C» > 1, то расчетные технические данные микропроцессора удовлетворяют решению задачи пользователя. Если «C» < 1, то расчетная частота МП итеративно наращивается до тех пор, пока C не станет большим единицы (или равным ей). Однако ясно, что итерация не может происходить бесконечно, ограничение — предельная тактовая частота реального МП в данном классе по разрядности. Поэтому, если в ходе итеративного уменьшения периода T_5 достигнуто граничное для данного класса разрядности значение, то следует увеличить расчетную разрядность в два раза. После этого процесс повторяется. Если разрядность K = 16 и дальнейшее уменьшение T_5 становится невозможным, то в рамках системы задача не имеет решения. Граничным условием, помимо временны́х и разрядности, служит длина программы в адресах. В случае, если расчетная длина программ L превышает 1000 адресов при разрядности K = 4, то разрядность увеличивается до восьми. Такое положение методики в приложении к конкретно существующей номенклатуре означает, что если L > 1000, то ОМЭВМ типов К1814 и К1820 не могут быть применены в силу ограниченности адресного пространства. При выборе характеристик МП учитываются также соотношения между T_5 и временами реакции системы на управляющие воздействия и на аварийные ситуации. Если хотя бы одно из этих времен меньше расчетного значения T_5, то задача в рамках системы становится неразрешимой, когда T_5 достигло границы возможного изменения.

На основании технических характеристик расчетного МП можно подобрать реальный МП или ОМЭВМ.

Анализ данных для выбора ОЗУ. Поскольку БИС ОЗУ можно разделить по типам и соответствующим областям применения, пользователь уточняет задачу, а именно: требуется ли сохранение в ОЗУ промежуточных данных о технологическом процессе, и если требуется, то сколько байт займет хранимая информация. Если промежуточные результаты хранить не требуется, то можно подсчитать оценочный объем ПЗУ:

$$V_{ПЗУ} = M/8 \cdot M_6 \cdot (M_6 \cdot 6 + M_7 \cdot 3) \ln (M_6 + M_7) + M_5 + L/32,$$

где M — разрядность данных; M_6 — количество аналоговых входов; M_7 — количество аналоговых выходов; M_5 — число дискретных входов-выходов; L — длина программы в адресах.

Для округления полученного результата можно воспользоваться формулой $V_1 = INT(V_{ПЗУ}/10) \cdot 10 + 10$, где INT (X) — функция целой части числа.

В формуле для $V_{ПЗУ}$ частное $M/8$ — это корректирующий коэффициент для пересчета данных произвольной разрядности в байты. Логарифм от суммы $M_6 + M_7$ — коэффициент запаса, учитывающий неточность прогноза при увеличении числа входов и выходов. Частное L/32 учитывает неизбежный рост объема ОЗУ при увеличении длины управляющей программы (увеличение числа подпрограмм влечет за собой рост числа ветвлений и числа ячеек ОЗУ, необходимых для хранения промежуточной информации).

Если пользователю нужно хранить промежуточную информацию, следует уточнить два параметра: энерговооруженность управляемой системы и ее критичность к сбоям по питанию и, как следствие, к частичной или полной потере промежуточной информации о технологическом процессе. Если потребитель ограничен мощностью источника питания и его система критична к потере информации, следует рекомендовать применение ОЗУ, изготовленного по КМОП-технологии. Если же система имеет высокую энерговооруженность, то тип ОЗУ определяется, исходя из его объема. При объемах свыше 16 Кбайт рекомендуется ОЗУ динамического типа, при меньших объемах — статического. Критерием здесь служит число корпусов БИС и ИС, необходимых для реализации требований пользователя.

Объем ОЗУ оценивается по формуле

$$V_{ОЗУ} = M/8 \cdot (M_6 \cdot 6 + M_3 \cdot 3) \ln (M_6 + M_7) + M_5 + 1{,}2A + L/32.$$

Эта формула совпадает с формулой для $V_{ПЗУ}$, за исключением члена 1,2A, где A — объем ОЗУ в оценке пользователя; 1,2 — коэффициент запаса.

Номенклатура БИС ОЗУ весьма разнообразна, поэтому необходимо предлагать пользователю не одну серию или один тип микросхемы, а ряд типов, отсортированных по критерию «минимальное число корпусов БИС» (на каждом предприятии существуют ограничительные перечни на применение отдельных типов радиокомпонентов). Отбор наиболее подходящих БИС ОЗУ производится после предварительной сортировки по критерию «минимальное число корпусов БИС» ОЗУ. Данный критерий выбран из соображений надежности, минимальных габаритов и высокой технологичности будущей системы. Исходя из указанного критерия, следует отобрать ряд БИС в порядке возрастания количества корпусов, необходимых для реализации задачи пользователя.

Выбор ПЗУ. Для выбора подходящего ПЗУ анализируют два основных параметра: срок хранения информации в ПЗУ и возможность влияния на ПЗУ солнечной или иной радиации. Если срок хранения программы не превышает трех лет в условиях отсутствия активных излучений, предпочтение отдается БИС серии К573. В остальных случаях выбирается серия К556. Критерии и порядок подбора БИС ПЗУ совпадают с порядком подбора БИС ОЗУ.

Выбрав БИС ПЗУ, следует уточнить климатические факторы, в которых будет эксплуатироваться МПСУ. Если температурный диапазон превышает пределы, допустимые по ТУ для данного типа МП, потребитель должен предусмотреть меры конструктивной защиты МПСУ от перегрева или переохлаждения.

Методика также должна учитывать, что неподготовленные пользователи нуждаются в справочной информации по отдельным вопросам применения МП (в основном по выбору средств отладки). Выбор отладочных средств сводится к установлению взаимно-однозначного соответствия между типом МП и средством отладки.

Необходимо также оценить экономическую целесообразность автоматизации отдельных изделий. Поэтому в методике принято следующее: **если система автоматизации по стоимости превышает 20 % стоимости объекта автоматизации, то либо пользователь некорректно сформулировал задачу, либо изделие автоматизируется не с помощью МП.**

Телефон 468-81-75, Москва

Статья поступила 4 мая 1987

А. Е. Злачевский

КОМПАКТНОЕ УСТРОЙСТВО УПРАВЛЕНИЯ ПЕРФОЛЕНТОЧНЫМ ВВОДОМ И ВЫВОДОМ ДЛЯ МИКРОЭВМ С КАНАЛОМ МПИ

В настоящее время основное средство хранения и отладки программ для микроЭВМ — гибкие магнитные диски (ГМД). Однако иногда необходимо иметь и перфоленточный комплекс устройств (для обработки перфолент на различных типах ЭВМ, включая ЕС ЭВМ и СМ ЭВМ).

Промышленность выпускает отдельные устройства для микроЭВМ «Электроника 60»: устройство управления В3 15 ВВЛ-1000—003 для ввода информации с перфоленты в микроЭВМ с помощью фотосчитывателя FS-1501 или СП-3 и устройство управления В21 15 ВВЛ-150—001 для вывода данных из микроЭВМ на перфоленту с помощью перфоратора ПЛ-150М. Каждое из этих устройств выполнено на стандартной печатной плате одинарного типоразмера, т. е. 252×143 мм. Таким образом, чтобы с их помощью организовать перфоленточный комплекс для микроЭВМ с каналом МПИ, необходимо занять два места в канале. При этом затрачено 38 микросхем серий К155 и К559, а суммарное потребление по цепи +5 В составляет 1,4 А [1, 2].

Ни одна из известных модификаций серийно выпускаемых ДВК на базе одноплатных микроЭВМ МС 1201, МС 1201.01 и МС 1201.02 не комплектуется устройствами управления перфоленточным вводом и выводом, хотя интерпретатор языка БЕЙСИК, выполненный в кристалле (на БИС К1801РЕ1) и входящий в комплект поставки некоторых модификаций ДВК, нуждается в таком перфоленточном комплексе [3, 4].

Разработано и успешно эксплуатируется компактное устройство управления перфоленточным вводом и выводом для любой микроЭВМ с каналом МПИ. Данные в микроЭВМ вводятся с помощью фотосчитывателя FS-1501 (модификация А/М), выводятся на перфоленту с помощью перфоратора ПЛ-150М. Устройство выполнено на плате одинарного типоразмера, содержит 20 микросхем, а потребление по цепи +5 В составляет не более 0,7 А. Достоинство предлагаемого устройства — его идентичность, с точки зрения программиста, устройствам В3 и В21. Это позволяет пользоваться любой программой перфоленточной операционной системы ПЛОС, а также программой ввода абсолютного загрузчика с перфолент, «зашитой» в системном ПЗУ микроЭВМ.

Основа устройства — БИС К1801 ВП1-033 (в режиме контроллера байтового параллельного интерфейса) и БИС К1801ВП1—034 (в режиме устройства передачи информации [5]).

340086, Донецк, ул. Донецкая, 39, ПО «Донецкгорэлектротранс», отдел АСУ. Тел.: 91-04-33, 93-52-11, 93-52-31

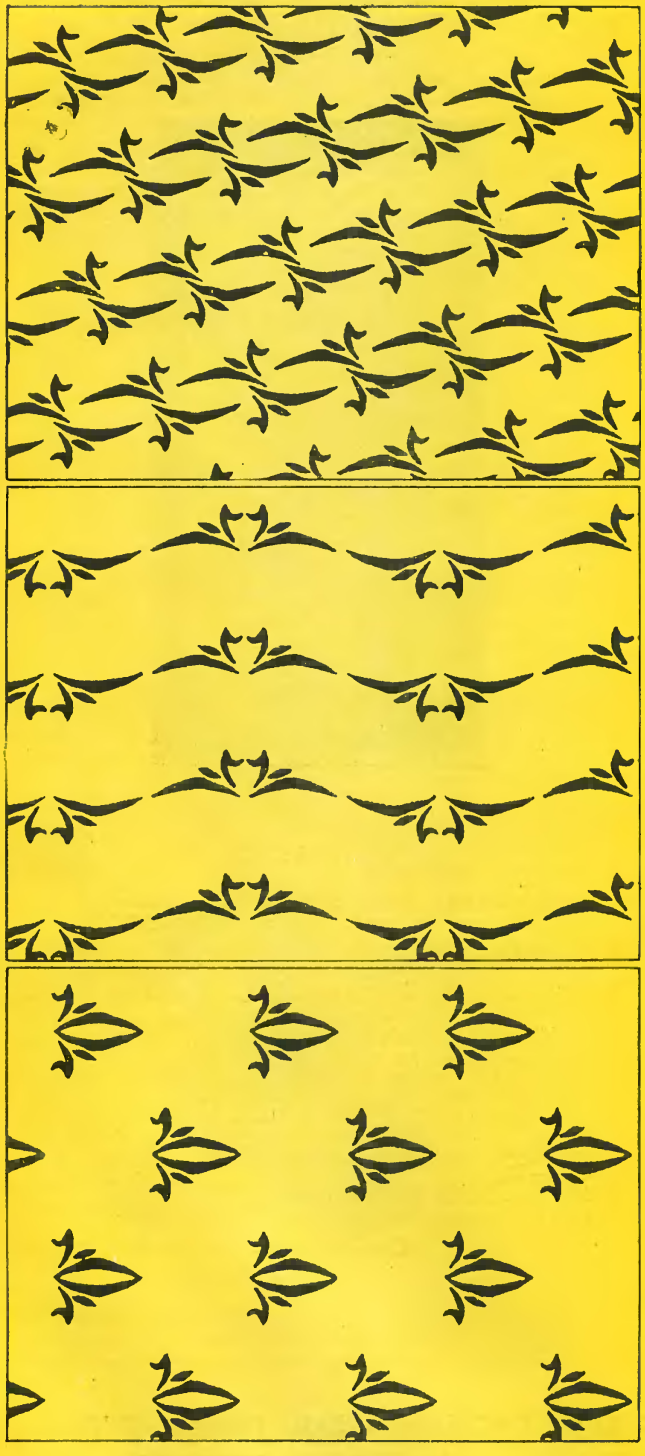


ЛИТЕРАТУРА

1. Устройство управления В3. Техническое описание и инструкция по эксплуатации. 3.858.379 ТО.
2. Устройство управления В21. Техническое описание и инструкция по эксплуатации. 3.858.367 ТО.
3. Попов А. А., Хохлов М. М., Глухман В. Л. Диалоговые вычислительные комплексы «Электроника НЦ 80—20» // Микропроцессорные средства и системы.— 1984.— № 4.— С. 61—64.
4. Дшхунян В. Л., Борщенко Ю. И., Отрохов Ю. Л., Шишарин С. А. Одноплатные микроЭВМ ряда «Электроника МС 1201» // Микропроцессорные средства и системы.— 1985.— № 2.— С. 8—13.
5. МикроЭВМ «Электроника НМС 11100.1» (МС 1201). Техническое описание. ШНО.305.019 ТО.

Сообщение поступило 6 августа 1987

ЖУРНАЛ «МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ» — ВАШ ПЕРВЫЙ ПУТЕВОДИТЕЛЬ В МИР МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ — МИР ТЕХНИКИ БУДУЩЕГО. НЕ ЗАБУДЬТЕ СВОЕВРЕМЕННО ОФОРМИТЬ ПОДПИСКУ НА 1989 ГОД!



М.В. Дараган, С.Т. Мишакова, А.Б. Ходулёв

КОМПЬЮТЕР И

«Искусство орнамента содержит в неявном виде наиболее древнюю часть известной нам высшей математики», – пишет в своей книге «Симметрия» известный немецкий математик Герман Вейль. Среди декоративных узоров древних египтян содержатся все возможные виды симметричного расположения на плоскости любых фигур. Существует всего 17 типов плоских орнаментов. Они были найдены почти одновременно в 1891 году русским ученым У.С. Федоровым и немецким А. Шенфлисом, причем один из них шел геометрическим путем, а другой использовал теорию групп. Особый интерес к орнаментам проявился у европейских ученых в связи с развитием кристаллографии.

Идея создания программы, рисующей орнаменты на экране дисплея, возникла из потребности обучения студентов художественно-графических факультетов педагогических институтов. Рисование производится средствами Графора [1]. Авторы программы воспользовались идеей плоского бордюра, распространяемого затем до плоской симметрии [2]. В конце заметки помещена подпрограмма рисования одного из типов орнаментов, соответствующая левому нижнему рисунку.

Различные способы расположения на плоскости одного и того же элемента дают совершенно неожиданный декоративный эффект. Трудность заключается в том, что оценить его возможно только после того, как нарисовано достаточно большое количество элементов. Рисование их с помощью шаблонов – весьма трудоемкий и неинтересный процесс. Творчество в нем проявляется в создании самого элемента симметрии и оценке полученных орнаментов. Рутинную часть работы вполне успешно выполняет персональная ЭВМ, снабженная графическим дисплеем. В заметке помещены примеры орнаментов 6 из 17 различных типов. Интересно, что применение двух цветов добавляет к 17 одноцветным орнаментам еще 46 типов [3].

Программа может быть полезна художникам-оформителям.

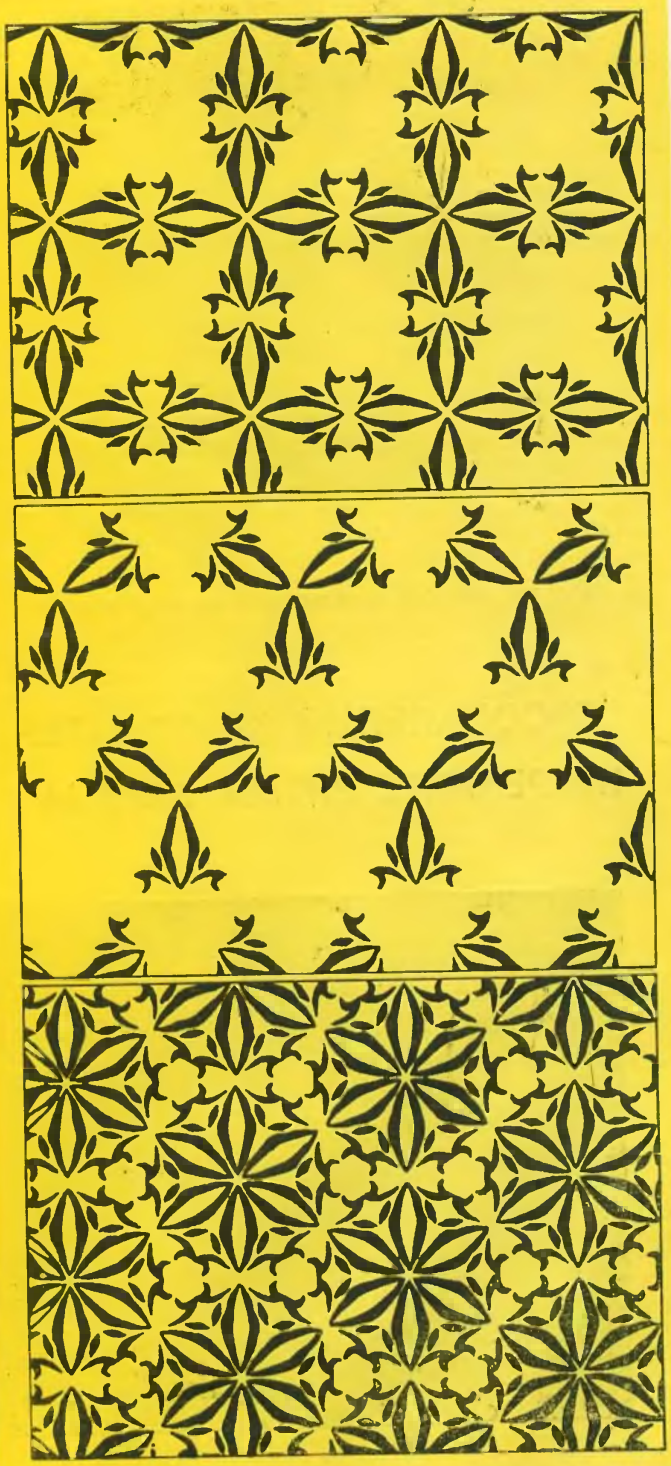
ОРНАМЕНТЫ

Литература

1. Баяковский Ю.М., Галактионов В.А., Михайлова Т.Н. Графор. Графическое расширение фортрана. Наука, М.; 1985, с. 288.
2. Шубников А.В., Копцик В.А. Симметрия в науке и искусстве. Наука, М., 1972, с. 339.
3. Левитин К.Е. Геометрическая рапсодия. Знание, М., 1984, с. 176.

Оригинал-макет этой заметки (вместе с иллюстрациями), целиком подготовлен с помощью персонального компьютера IBM PC. Текст напечатан литературной гарнитурой, заголовок – шрифтом «Эхо» художника Г. Кликушина.

```
      SUBROUTINE RIS3
      CALL PAGE(32.,20.,' ',1,1)
      CALL SETBGR(3)
      CALL SETPAL(0)
      DX=4.2
      DY=2.6
      TX=-DX
      DO 2  J=1,5
      CALL BEGLEV
      DO 3  K=1,2
      CALL BEGLEV
      DO 1  I=1,8
      CALL ELEM
      CALL TRANSL(2.*DX,0.)
1     CONTINUE
      CALL ENDLEV
      CALL LSCALE(0.,DY,DX,DY,-1.)
3     CONTINUE
      CALL ENDLEV
      TX=-TX
      CALL TRANSL(TX,2.*DY)
2     CONTINUE
      CALL ENDPG(' ')
      RETURN
      END
```



МИКРОЭВМ С МАРКОЙ ВЭФ



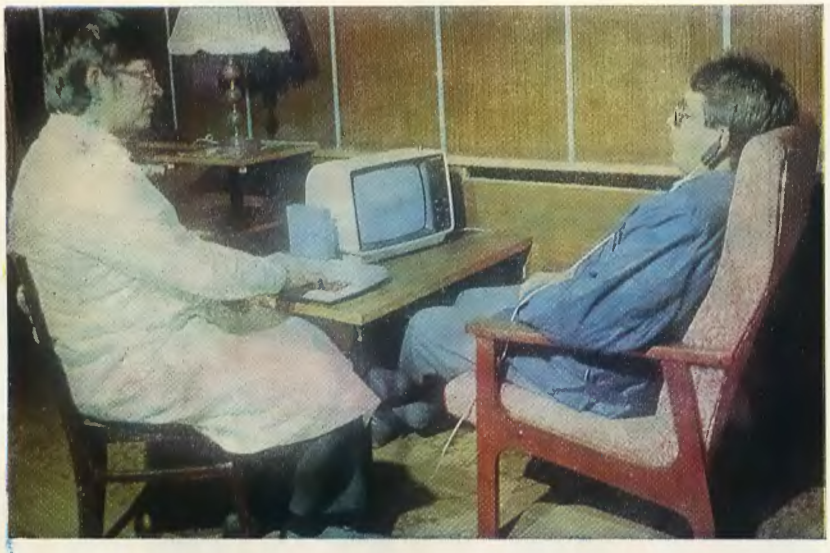
Стенд для контроля качества выпускаемой продукции

(К ст. Таланова В. А.)

На Рижском производственном объединении ВЭФ им. В. И. Ленина для выпуска квазиэлектронных АТС, новейших телефонных аппаратов, популярных «Спидол» используют микроЭВМ и микропроцессоры на всех стадиях изготовления — от разработки до контроля готовой продукции.

Большое внимание уделяется контролю качества выпускаемой продукции. С помощью системы контроля параметров радиоприемников «ВЭФ-ПАРАМЕТР» в условиях массового производства осуществляется многосторонняя проверка изготовляемых изделий (программа оценки записана жестко). Проверяются диапазоны проверяемых частот, чувствительность, ограниченная усилением и шумами, односигнальная избирательность по зеркальному и соседнему каналам.

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР В МАССОВОЙ ДИАГНОСТИКЕ И ТРЕНИНГЕ РИТМА СЕРДЦА



Экспериментальная установка

(См. ст. Водоватова Ф. Ф. в № 6, 88)

Возрастание степени эмоционального напряжения приводит к срыву высшей нервной деятельности и соответственно к рассогласованию механизма регуляции непроизвольных (автоматических) физиологических реакций, а в дальнейшем к необратимым патологическим явлениям и, в первую очередь, в сердечно-сосудистой системе.

Для исследования вопросов, связанных с саморегуляцией ритма сердца, была создана экспериментальная установка.

УДК 631.3

Э. М. Пройдаков

ДИСАССЕМБЛЕР, УПРАВЛЯЕМЫЙ ДАННЫМИ

При утрате исходного текста программы, при адаптации для новых типов микропроцессоров, ОЭВМ и сопроцессоров системного и инструментального, реже прикладного, программного обеспечения (ПО) и в ряде других случаев используются инструментальные средства следующих видов: отладчики, дисковые редакторы [1], редакторы объектных файлов, редакторы сообщений, программы «разваливания» библиотек на модули, интегрированные инструментальные средства [2], дисассемблеры (реассемблеры, деассемблеры и обратные ассемблеры). Единая терминология из-за немногочисленности публикаций устанавливается медленно, однако опрос разработчиков показал, что общепринятым является термин дисассемблер. Это инструмент, позволяющий восстановить из объектного (машинного) кода исходный текст программы, т. е. перевести ее в другое качественное состояние. Некоторые программисты рассматривают дисассемблеры только как инструмент для взлома чужих программ, поэтому проблемы, связанные с их разработкой, долго обходились молчанием. Автор считает, что, с одной стороны, вопросы этики относятся к уровню мира программистов, а не к инструментарию, а с другой,— у дисассемблеров имеются вполне благопристойные применения. Восстановленный исходный текст нельзя приравнять к настоящему исходному тексту программы, так как, в общем случае, оказываются утерянными имена идентификаторов (меток), комментарии, а также, возможно, макросы, псевдокоманды и директивы трансляции.

Если программа разрабатывалась на Макроассемблере и разработчиком не предпринимались специальные меры против дисассемблирования, то восстанавливаемый текст семантически адекватен исходной программе. При использовании языка высокого уровня (Си, ПЛ/М, Паскаль) восстановленная программа не адекватна исходной и ее читаемость тем хуже, чем выше уровень использованного исходного языка программирования. Существуют достаточно успешные попытки создания программ, восстанавливающих из объектного кода текст на языке высокого уровня (ЯВУ). В основе таких программ (1-я фаза) лежит высококачественное дисассемблирование, а затем (2-я фаза) выделение конструкций ЯВУ.

Основная проблема дисассемблирования программы — качественное разделение команд и данных (в статье рассматриваются дисассемблеры только для машин с архитектурой Дж. фон Неймана [3]. В ЭВМ с другими архитектурами этот процесс может иметь существенные отличия или вообще невозможен). При смешении данные воспринимаются как команды, что приводит к появлению в листинге дисассемблирования (называемого также псевдоассемблерным листингом) бессмысленных и искаженных участков текста, что часто позволяет визуально выделять такие участки. Длина сбойного участка тем больше, чем больше средняя длина команды у данного МП.

В компиляторах с ЯВУ разграничены области команд и данных, однако константы очень часто включаются непосредственно в кодовый сегмент.

По числу просмотров объектного кода дисассемблеры (по аналогии с трансляторами) делятся на одно-, двух- и многопроходные. Однопроходные, называемые также «дисассемблерами без меток», используются в основном в программных отладчиках и мониторах (команда L) и разрабатываются, если необходимо быстро получить псевдоассемблерный текст. Второе название отражает их существенную особенность — в псевдоассемблерном листинге отсутствуют метки в командах переходов и вызовах подпрограмм (стоят абсолютные шестнадцатеричные адреса). Поэтому после ассемблирования такого восстановленного текста мы получаем в общем случае абсолютную программу. За один проход (просмотр объектного кода от начала до конца) при ограниченном объеме ОЗУ нельзя собрать всю информацию о метках, поэтому теряются «ссылки назад».

Метки, создаваемые дисассемблерами, обычно имеют вид Lxxxx, где — первая буква английского слова Label (метка), aLxxxx — адрес перехода или подпрограммы. Метки данных имеют ту же структуру, но начинаются с буквы D. Такой вид меток наиболее удобен при разборе псевдолистинга. Программист с помощью редактора текстов может заменить эти метки на более информативные.

Наиболее распространены двухпроходные дисассемблеры, которые позволяют получать метки в листинге дисассемблирования, но, как правило, не решающие полностью проблемы разделения команд и данных. После них программист «вручную» подправляет сомнительные места.

Проблему разделения решают интерактивные и многопроходные дисассемблеры.

Интерактивный дисассемблер не разделяет команды и данные сам. Это делает программист, пользуясь командами просмотра объектного кода (дамп и однопроходное дисассемблирование). При этом выявляются участки данных и определяется тип данных. Информация фиксируется в таблицах дисассемблера, учитывается им при последующих дисассемблированиях и в итоге получается достаточно качественный восстановленный исходный текст. Интерактивный дисассемблер позволяет программисту вводить свои имена меток и комментарии, поэтому разбор текста программы совмещается с процессом дисассемблирования. Большие трудности появляются при попытке отличить друг от друга числа с плавающей точкой одинарной и двойной точности, визуально хорошо выявляются только символьные данные. Однако если программа выводит сообщения на 7-сегментные индикаторы (или другие устройства со специальным кодированием символов), то для выявления сообщений необходима специальная программа символьного дампа.

Многопроходные дисассемблеры, используя точку входа в программу, строят таблицы ее ветвей с помощью однопроходного дисассемблирования (запоминаются адреса из команд переходов и вызовов подпрограмм). Ветвь программы заканчивается командой: безусловного перехода, безусловного вызова подпрограммы, возврата из подпрограммы или перехода по адресу в регистре, так как после этих команд могут быть помещены данные. Пользуясь накопленной информацией, дисассемблер определяет границы (адрес начала и адрес конца) следующей ветви программы. Код, не попавший в эти границы, рассматривается как данные. Однако это могут быть, например, подпрограммы, обращение к которым осуществляется с помощью таблиц переходов. Поэтому процесс построения границ ветвей программы может быть значительно ускорен за счет интерактивного «консультирования» дисассемблера со стороны программиста.

Интересная особенность этого процесса — возможность выявления неиспользуемых участков программы, что позволяет создавать средства для оптимизации ее объема; это особенно важно, если программа затем записывается в ППЗУ/ПЗУ. Например, известно, что при компоновке программы, написанной на языке высокого уровня, к ней добавляется значительный объем неиспользуемых библиотечных модулей, особенно для языков типа Фортран.

Второй по значимости, но более глубокой является проблема разрешения противоречия между статической природой кода программы и ее динамическим выполнением. Решение этой проблемы трудно автоматизировать.

Примером являются самомодифицирующиеся программы.

На входе дисассемблера может быть файл с объектным кодом программы в виде: загрузочного модуля, «образа памяти», HEX-файла, объектного модуля, библиотеки объектных модулей. Часто входные данные передаются ему в некоторой области памяти, например при необходимости посмотреть содержимое микросхемы ППЗУ, объемом 2, 4, 8, 16 или 32 Кбайт. На инструментальной микроЭВМ, имеющей программатор, можно производить дисассемблирование программ с других микроЭВМ и различных микропроцессорных контроллеров. Последнее часто приходится делать при ремонте различного импортного оборудования и отсутствии документации.

Практика показала, что для этих целей достаточно иметь однопроходный дисассемблер, но его необходимо разработать за 3, 4 ч. Для этого автор предлагает следующую методику.

Как правило, в системе команд большинства МП в первом байте команды находится код операции. Таким образом, возможно кодирование до 256 различных кодов операций. В реальных МП набор команд значительно меньше, но для уменьшения длины команды (соответственно увеличения быстродействия МП) в ее первом байте вместе с кодом операции кодируются регистры и признаки. Если бы все команды МП были однобайтовыми, то мнемонику кода операции (команды) и ее операнды можно было бы занести в одну таблицу, расположив их по возрастанию значений соответствующих машинных кодов, и, индексируя эту таблицу содержимым очередного байта машинного кода дисассемблируемой программы, извлекать из таблицы почти готовый элемент получаемого псевдоассемблерного листинга. Однако в системе команд реальных МП имеются команды длиной в два и более байта (что зависит от архитектуры МП), кроме того, команды могут иметь поля с изменяемой информацией: адреса, непосредственные данные, признаки режимов адресации и т. п. Поэтому в описанную выше таблицу для таких команд заносятся специальные символы, позволяющие дисассемблеру определить, что нужно сделать со следующими байтами команды. Например, для МП КР580ИК80А таких символов нужно два: 〈〉#—следующий байт команды является байтом непосредственных данных (строка в таблице для команды MVI A, 〈байт〉 выглядит так: «MVI A, #»); @ — следующие два байта команды содержат адрес перехода или слово непосредственных данных (строка в таблице для команды безусловного перехода: «JMP @»).

Такие записи в таблице назовем шаблонами, а эти специальные символы — ключами шаблона. При разборе очередной команды дисассемблер по наличию ключей в шаблоне вызывает соответствующую подпрограмму. Набор ключей шаблона при необходимости может быть легко расширен. В дисассемблере для сопроцессора ввода-вывода 8089 используется, например, около 10 различных ключей шаблона.

Информация о длине команды неявным образом содержится в шаблоне. Несуществующие коды операций в псевдоассемблерном листинге отображаются байтовыми константами — DB 0XXH (XX — содержимое текущего байта дисассемблируемой программы). Работа по перенастройке дисассемблера на систему команд другого МП сводится к анализу его системы команд, подготовке и замене таблицы шаблонов, добавлению (или удалению) подпрограмм для работы с новыми (старыми) ключами шаблона.

Для облегчения ввода шаблонов с помощью редактора текстов набираются пары «код команды — шаблон» в произвольном порядке. После ввода таблицы дисассемблер производит их сортировку по возрастанию кодов. Управление данными заключается в том, что в любой момент времени порядок выполнения операций определяется текущим состоянием обрабатываемых данных.

Дисассемблер, управляемый данными, реализован на языках программирования БЕЙСИК-80/БЕЙСИК-86 и может работать в операционных системах CP/M-80, ISIS-II, МикроДОС, ДОС 1810, MS-DOS, CP/M-86 и совместимых с ними (мобилен на уровне переноса текста [4]).

Указанный подход оправдал себя при разработке дисассемблеров для 8-разрядных микроЭВМ на базе МП КР580ИК80А, Z-80, F-8, MC6800, семейства однокристальных микроЭВМ К1816ВЕ48, КР1816ВЕ51 и сопроцессора ввода-вывода 8089. Так, например, настройка дисассемблера на систему команд МП F-8 заняла 6 ч до получения работающего дисассемблера.

Структуры данных воспринимаются программистом лучше, чем структуры управления [5], поэтому отладка сводится к проверке корректности шаблонов.

Выделение с помощью ключа шаблона «@» адресов в командах переходов и создание таблицы адресов переходов позволяют построить однопроходный дисассемблер со ссылками вперед и двухпроходный, создающий псевдолистинг с метками.

Наличие меток не только облегчает разбор программы, но и позволяет превратить ее после некоторого редактирования из абсолютной в перемещаемую. На первом проходе в таблице меток собирается информация о них. В промежуточном файле метки (Lxxxx:) ставятся перед каждой командой, а перед вторым проходом таблица сортируется по возрастанию адресной части метки. Промежуточный файл считывается и лишние метки удаляются; этим упрощается реализация второго прохода дисассемблера и диагностируются ситуации, при которых метка указывает не на начало команды (проблема разделения команд и данных).

Для выделения текстов в объектном коде в области комментариев листинга дисассемблирования традиционно выдается символьное представление объектного кода. Это позволяет визуально выделять связанные тексты.

Описываемые дисассемблеры использовались в основном для восстановления управляющих программ и драйверов, записанных в ППЗУ и считываемых с помощью программатора в HEX-файл [6]. Поэтому не было необходимости в построении интерактивных дисассемблеров, однако метод это позволяет.

Следует отметить, что в составе документации многих ОС отсутствует документ, описывающий форматы загрузочного модуля, абсолютного и перемещаемого объектных модулей и внутренний формат библиотеки объектных модулей, знание которых облегчило бы не только создание дисассемблеров, но и ряда других инструментальных средств: отладчиков, программ дампа, контроля и редактирования объектных модулей, средств ведения проектов, оптимизаторов программ, компиляторов, программ обслуживания программаторов и эмуляторов, специальных загрузчиков и др. Разумно было бы ввести в документ «Руководство системного программиста» обязательный раздел «Структура объектного файла».

К сожалению, автору не удалось найти общую систему ключей шаблонов и вместо набора дисассемблеров, настроенных на конкретные МП, построить единую программу с подгружаемой таблицей шаблонов.

Симметричность таблицы шаблонов относительно входа и выхода позволяет применять указанный метод для быстрого построения конверторов из одной системы команд в другую (например, из КР1816ВЕ48 в КР1816ВЕ51), однопроходных ассемблеров и ассемблеров-дисассемблеров.

На этом примере можно отметить закон сохранения сложности программ: усложнение структуры данных упрощает код и наоборот.

129301, Москва, ул. Касаткина, д. 3, ВНИИМТ; тел. 283-97-64

ЛИТЕРАТУРА

1. Системные программы МикроДОС.— М.: МЦНТИ, 1986.
2. ISIS-II. Software Toolbox.— Intel Corp.— USA,— 1980.— N 121727.
3. Майерс Г. Архитектура современных ЭВМ.— М.: Мир,1985.— Т. 1.
4. Машинно-независимые операционные системы.— М.: МЦНТИ, 1987.
5. Кертис Б., Солоуэй Э. М. Брукс Р. Е. и др. Психология программных систем.— ТИИЭР.— 1986.— Т. 74.— № 8.— С. 42—60.
6. Микропроцессоры: системы программирования и отладки / Под ред. В. А. Мясникова, М. Б. Игнатьева.— М.: Энергоатомиздат, 1985.

Статья поступила 4 мая 1987

УДК 681.3.06

В. М. Арпаксыд, И. Б. Володарский, А. Е. Дорфман

МОБИЛЬНАЯ СУБД С ВКЛЮЧАЮЩИМ ЯЗЫКОМ СИ ДЛЯ ПЕРСОНАЛЬНЫХ И МИНИ-ЭВМ

Система управления базами данных сетевой организации СУБД ВЕСТА ориентирована на язык программирования Си, обладает высокой степенью мобильности и может использоваться как на больших и мини-ЭВМ (ЕС ряда 2, СМ1420), так и на персональных компьютерах («Электроника 85», ЕС1840, Искра 226) в среде операционных систем типа UNIX и MS DOS. Все программы СУБД написаны на языке Си с четкой локализацией системно- и машинно-зависимых частей. Они могут быть перенесены на любую ЭВМ (и операционную систему), где есть компилятор и стандартная библиотека языка Си.

СУБД ВЕСТА в значительной степени удовлетворяет требованиям КОДАСИЛ по организации баз данных. Характерными особенностями СУБД являются:

отсутствие ограничений на типы наборов. Допускаются многочленные наборы, циклы и петли. Запись может быть членом сразу нескольких наборов и одновременно владельцем нескольких других наборов;

поддержка хронологических, сортируемых и управляемых типов наборов с включением новых членов по дисциплине очереди или стека, по возрастанию или убыванию значений полей сортировки, перед или после текущего члена набора;

логическое упорядочение однотипных записей с помощью индексирования по многим ключевым полям. Допускаются как уникальные, так и дублируемые ключи. Для организации индексов используются алгоритмы растущего B*-дерева;

возможность манипулирования как целыми записями, так и отдельными полями, которыми могут быть данные любого допустимого в языке Си типа, массивы однотипных элементов, структуры и объединения разнотипных подполей и, наконец, массивы структур и объединений;

эффективное управление размещением записей фиксированной длины в странично-организованных файлах с возможностью повторного использования пространства страниц, освобождаемого после удаления записей;

ускорение обработки за счет программно-организованной кеш-памяти страниц, сокращающей число операций ввода-вывода.

В состав СУБД ВЕСТА входят компилятор языка описания данных, синтаксически подобного языку Си, набор утилит обслуживания и библиотека функций языка Си, образующих интерфейс манипулирования данными.

Ближайший аналог СУБД ВЕСТА — система db VISTA фирмы RAIMA Corp., функционирующая в среде MS DOS. Однако программной совместимости между системами нет, СУБД ВЕСТА обладает значительно более широкими возможностями, лучше согласована с требованиями КОДАСИЛ.

СУБД ВЕСТА предназначена для использования в приложениях, требующих высокой скорости обработки и эффективной организации постоянно обновляемой базы данных при относительной стабильности схемы базы и номенклатуры запросов к ней. К таким приложениям относятся задачи АСУ ТП, оперативно-диспетчерского управления, подсистемы первичного учета в АСУП, локальные задачи АРМ на базе персональных компьютеров.

252011, Киев, Печерский спуск, 19, Киевское отделение Украинского Государственного проектного и проектно-конструкторского института «Тяжпромавтоматика»; тел. 290-82-56.

Статья поступила 22 июня 1987

УДК 681.3.06 : 332.1

А. В. Бернов, В. Н. Поротов

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ПРОГРАММИРОВАНИЯ И ОТЛАДКИ КОНТРОЛЛЕРОВ НА ОСНОВЕ МИКРОПРОЦЕССОРОВ СЕРИИ К1801

Аппаратно-программный комплекс (АПК) разработан на основе ПЭВМ ДВК 2М для автономной и комплексной отладки программного обеспечения и аппаратных средств микроконтроллеров на основе БИС серии К1801.

На АПК можно преобразовать программу на макроассемблере в формат абсолютной загрузки (LDA), загрузить программу в формате LDA в контроллер, отладить ее в диалоговом (пультовом) режиме, организовать доступ контроллера к ресурсам ПЭВМ. На этапе

СРЕДСТВА ОТЛАДКИ МП СИСТЕМ

комплексной отладки обеспечивается проверка работоспособности внешних устройств контроллера, накапливается статистика их работы.

Аппаратные средства АПК включают кроме ПЭВМ модуль ввода-вывода для обмена информацией между контроллером и ПЭВМ. Основу модуля (рис. 1) составляет программируемый периферийный адаптер (ППА) КР580ВВ55. Двунаправленная шина D ППА подключается через шинный формирователь к младшим разрядам общей шины контроллера. Адаптер программируется на работу в режиме 1 — «ввод-вывод с квитанциями». Порт А ППА программируется на вывод, порт В — на ввод. Порт С является регистром управления вводом-выводом. Схема выборки БИС ППА состоит из дешифратора адреса на микросхемах D1, D2.1, D3 и регистра-защелки D5. Данной схеме соответствуют следующие адреса: портов А — 177506_8, В — 177504_8, С — 177502_8 и регистра управления — 177500_8. Повторители D8, D9 служат для усиления сигналов, поступающих от порта А ППА.

Связь контроллера с ПЭВМ осуществляется через

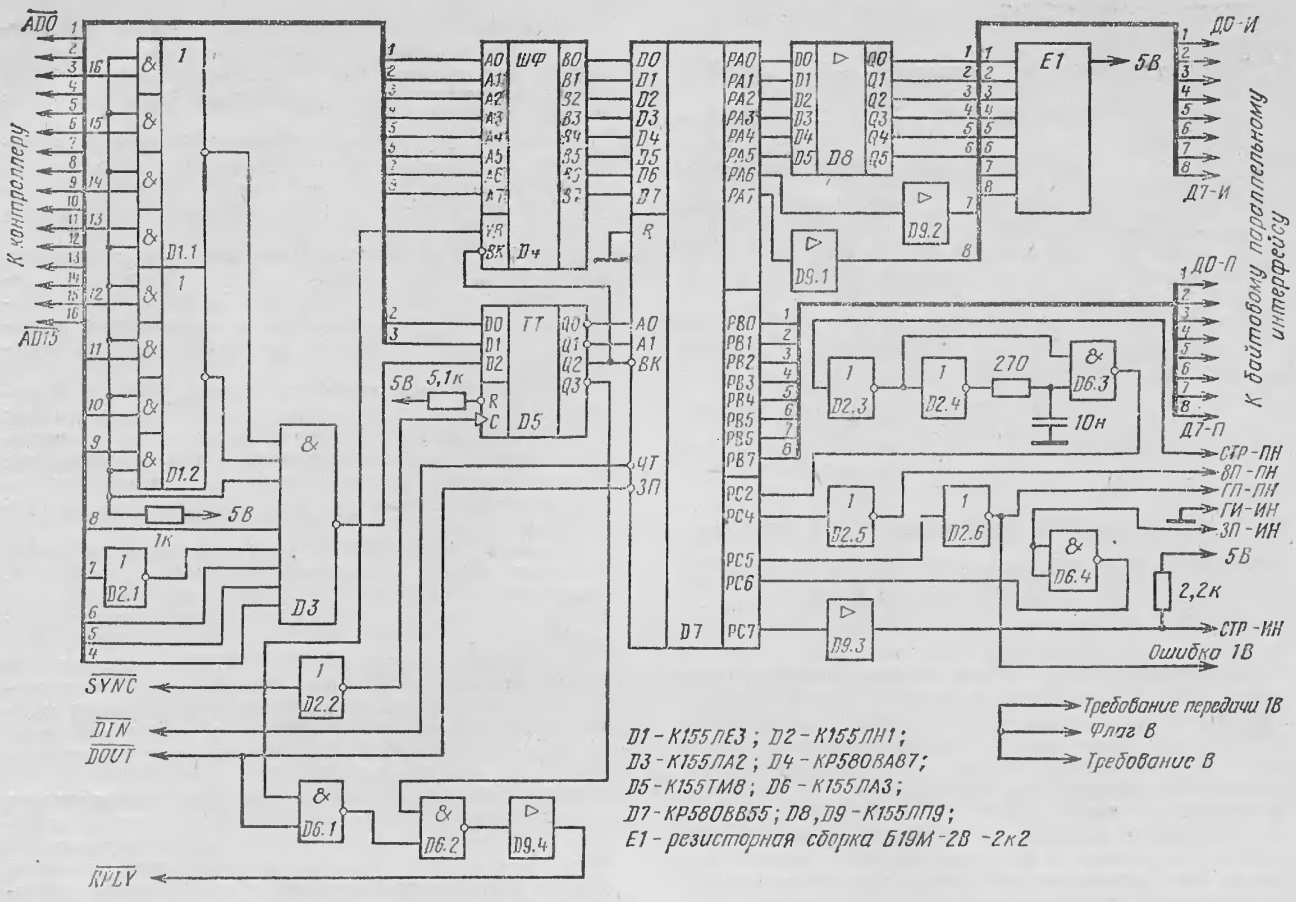


Рис. 1. Электрическая схема модуля ввода-вывода

устройство байтового параллельного интерфейса (входит в состав ДВК) [2].

Программные средства АПК представлены программой-монитором, состоящей из резидентной (объем 0,5 Кслов, зашивается в ППЗУ и выполняется контроллером) и кроссовой (выполняется ПЭВМ в ОС RT-11) частей. Их взаимодействие осуществляется с помощью подпрограмм ввода-вывода (рис. 2, 3), обеспечивающих обмен информацией по каналу ПЭВМ — контроллер. В каждый момент времени одна из частей монитора находится в активном состоянии, другая — в пассивном. Пассивная часть отрабатывает команды, выдаваемые активной частью. Резидентный монитор активен только тогда, когда в контроллере выполняется прикладная программа, которая может получить доступ к ресурсам ПЭВМ, вызывая подпрограммы резидентного монитора командами EMT (табл. 1) [3].

Монитор проверяет состояние канала ПЭВМ — контроллер. В случае обнаружения ошибки при проверке байта на четность или рассинхронизации частей монитора (обе части одновременно работают на ввод или вывод), а также при возникновении внутренних прерываний в контроллере кроссовый монитор выводит на терминал ПЭВМ сообщение об ошибке, переходит в активное состояние и инициирует перезапуск резидентного монитора. Аналогичные действия вызывает двойная команда оператора ПЭВМ CTRL/C. Резидентный и кроссовый мониторы написаны на языке макроассемблера. В связи с тем, что резидентный монитор обрабатывает внутренние прерывания контроллера с векторами 4_8, 10_8, 14_8, 30_8, ПЗУ удобнее всего расположить в области нулевых адресов адресного пространства контроллера. В противном случае для обеспечения нормальной работы резидентного монитора в этой области должно находиться ОЗУ.

Работа АПК. Прикладные программы на языке макроассемблера вводятся в ПЭВМ с помощью текстового редактора. Пример прикладной программы с использованием макрокоманд из макробиблиотеки для запроса соответствующих ресурсов ПЭВМ показан на рис. 4.

Пусть исходная программа на макроассемблере помещена в файл FIL.MAC. Получить объектный модуль и файл листинга программы, с которым работает монитор в пультовом режиме, можно по команде

RU MACRO FIL, MACLIB.SML/M FIL, FIL,

где MACLIB.SML — макробиблиотека.

Вызвав компоновщик, получают файл в формате абсолютной загрузки с расширением LDA:

RU LINK FIL/L/B : 20000 FIL.

Ключ B : 20000 сообщает адрес, с которого надо компоновать программу, т. е. адрес той области ОЗУ контроллера, в которой будет размещена прикладная программа. В данном примере этот адрес равен 20000_8.

Запуск АПК происходит следующим образом: схема начального пуска контроллера после включения питания передает управление на нулевой адрес ПЗУ с резидентным монитором (контроллер запускается только один раз после включения питания). Кроссовый монитор вызывается командой RU RA, печатает на терминале ПЭВМ символ «*» и ждет команду FIL1, FIL2/K, где FIL1 — имя файла в формате абсолютной загрузки

```
; ——— З А Г О Л О В О К ———
.PSECT WRRD
;          АССЕМБЛЕР MACRO-11/ RT11
; ***********************************************
; * WRBYTE -ПОДПРОГРАММА, ПЕРЕДАЮЩАЯ МЛАДШИЕ 7 БИТ R0  *
; *          В ЭВМ                                *
; * RDBYTE -ПОДПРОГРАММА, ПРИНИМАЮЩАЯ 7 БИТ В R0 ИЗ  *
; *          ЭВМ                                  *
; ***********************************************
; ——— О П Р Е Д Е Л Е Н И Я ———
        PORTA=177506    ;ПОРТ А
        PORTB=177504    ;ПОРТ В
        PORTC=177502    ;ПОРТ С
        REGUP=177500    ;РЕГИСТР УПРАВЛЕНИЯ
; ——— О С Н О В Н А Я   П Р О Г Р А М М А ———
        MOV     #247,@#REGUP    ;ПРОГРАММИРОВАНИЕ БИС
                                ;КР580ВВ55
        .GLOBL  PROW    ;ПОДПРОГРАММА ИЗМЕНЕНИЯ 7-ГО
                        ;БИТА R0 ДО ЧЕТНОГО ЧИСЛА
                        ;ЕДИНИЦ

WRBYTE::CALL    PROW            ;СДЕЛАТЬ В R0 ЧЕТНОЕ
                                ;ЧИСЛО ЕДИНИЦ
        COM     R0              ;ИНВЕРТИРОВАТЬ ДАННОЕ
        MOV     #17,@#REGUP     ;СТР-ИН=1
        MOV     R0,@#PORTA      ;ПЕРЕДАТЬ БАЙТ
WAIT:   TSTB    @#PORTC
        BPL     WAIT            ;БАЙТ НЕ ПЕРЕДАН.ОЖИДАТЬ
        MOV     #10,@#REGUP     ;ЗП-ПН=1
        RTS     PC

RDBYTE::MOV     R2,-(SP)        ;СОХРАНИТЬ R2
        MOV     #REGUP,R2       ;АДРЕС РЕГ. УПР. В R2
        MOV     #13,@R2         ;ГП-ПН=0
        MOV     #11,@R2         ;ЗП-ПН=0
WAIT2:  BIT     @#PORTC,#2      ;ПРОВЕРИТЬ ПЕРВЫЙ РАЗРЯД
        BEQ     WAIT2           ;БАЙТ НЕ ПРИНЯТ.ОЖИДАТЬ
        MOV     #10,@R2         ;ЗП-ПН=1
        MOV     @#PORTB,R0      ;ПРИНЯТЬ БАЙТ
        MOV     #12,@R2         ;ГП-ПН=1
        COM     R0              ;ИНВЕРТИРОВАТЬ ДАННОЕ
        MOV     R0,R2
        CALL    PROW            ;СДЕЛАТЬ В R0 ЧЕТНОЕ
                                ;ЧИСЛО ЕДИНИЦ
        CMPB    R0,R2
        BEQ     1$
        MOV     #300,R0         ;ОШИБКА ПО ЧЕТНОСТИ
        CALL    WRBYTE          ;СООБЩИТЬ КРОСС.МОНИТОРУ
        BR      ERROR
1$:     CMPB    #300,R0         ;ПРОВЕРКА НА КОМАНДУ
                                ;ПЕРЕЗАПУСКА
        BNE     NOER
        .GLOBL  START
ERROR:  JMP     START           ;ПЕРЕЗАПУСК РЕЗ.МОНИТОРА
NOER:   MOV     (SP)+,R2        ;ВОССТАНОВИТЬ R2
        RTS     PC

        .END
```

Рис. 2. Подпрограммы ввода-вывода резидентного монитора

```
; ——— З А Г О Л О В О К ———
.PSECT WRRDK
;          АССЕМБЛЕР MACRO-11/ RT11
; ***********************************************
; * RDKON -ПОДПРОГРАММА, ПРИНИМАЮЩАЯ МЛАДШИЕ 7 БИТ ИЗ  *
; *          КОНТРОЛЛЕРА В R0 ЭВМ                 *
; * WRKON -ПОДПРОГРАММА, ПЕРЕДАЮЩАЯ 7 БИТ ИЗ R0 ЭВМ  *
; *          В КОНТРОЛЛЕР                         *
; ***********************************************
; ——— О П Р Е Д Е Л Е Н И Я ———
        RSI=177550      ;РЕГИСТР СОСТОЯНИЯ ИСТОЧНИКА
        RP =177552      ;РЕГИСТР ПРИЕМНИКА
        RSP=177554      ;РЕГИСТР СОСТОЯНИЯ ПРИЕМНИКА
        RI =177556      ;РЕГИСТР ИСТОЧНИКА
; ——— О С Н О В Н А Я   П Р О Г Р А М М А ———
        .GLOBL  PROW    ;ПОДПРОГРАММА ИЗМЕНЕНИЯ 7-ГО
                        ;БИТА R0 ДО ЧЕТНОГО ЧИСЛА
                        ;ЕДИНИЦ

RDKON:: CLC             ;ОШИБОК НЕ БЫЛО
NE:     MOVB    @#RSI,R0
        BPL     RDK     ;ОЖИДАТЬ ДАННОЕ ОТ КОНТРОЛЛЕРА
        MOVB    @#RP,R0 ;ПРИНЯТЬ ДАННОЕ
        BCS     RDK     ;ЕСЛИ БЫЛА ОШИБКА,ТО ВВЕСТИ
                        ;ЕЩЕ ДАННОЕ
        MOV     R0,R1
        CALL    PROW    ;СДЕЛАТЬ В R0 ЧЕТНОЕ ЧИСЛО ЕДИНИЦ
        CMP     R0,R1
        BEQ     NOER    ;ОШИБКИ НЕТ
        SEC             ;ОШИБКА ПО ЧЕТНОСТИ
RDK:    MOVB    @#RSP,R0
        BPL     NE      ;ОШИБКИ РАССИНХРОНИЗАЦИИ НЕТ
        MOV     @#RSP,R0;ПРОВЕРИТЬ ГОТОВНОСТЬ РСП
        BMI     NE      ;ОШИБКИ РАССИНХРОНИЗАЦИИ НЕТ
        MOV     #^D10,R0;
1$:     NOP             ;ПОДОЖДАТЬ
        SOB     R0,1$
        MOV     @#RSP,R0;ПРОВЕРИТЬ ГОТОВНОСТЬ РСП
        BMI     NE      ;ОШИБКИ НЕТ
        MOV     #300,R0
        DALL    WRKON   ;ПЕРЕВЕСТИ РЕЗ.МОНИТОР В
                        ;ПАССИВНОЕ СОСТОЯНИЕ
        .GLOBL  START
        JMP     START   ;ПЕРЕЗАПУСК КРОССОВОГО МОНИТОРА
NOER:   RTS     PC

WRKON:: CALL    PROW    ;СДЕЛАТЬ В R0 ЧЕТНОЕ ЧИСЛО ЕДИНИЦ
WRK:    TST     @#RSP
        BMI     WAI     ;ГОТОВНОСТИ НЕТ
        TSTB    @#RSP
        BPL     WAI     ;RSP НЕ ГОТОВ
        MOVB    R0,@#RI ;ПЕРЕДАТЬ ДАННОЕ
        RTS     PC
WAI:    TSTB    @#RSI
        BPL     WRK
        SEC             ;ОШИБКА РАССИНХРОНИЗАЦИИ
        BR NE           ;ПРИНЯТЬ ДАННОЕ

        .END
```

Рис. 3. Подпрограммы ввода-вывода кроссового монитора

LDA; FIL2 — имя файла листинга исходной программы (расширение по умолчанию LST). Ключ K может принимать следующие значения: P — загрузить файл FIL1 в контроллер и перейти в пультовый режим, E — окончить работу.

Если значение ключа не указано, то программа из файла загружается в контроллер и выполняется с адреса, указанного после директивы макроассемблера .END. Спецификации файлов FIL1 и FIL2 не обязательны.

Монитор может проверить канал ПЭВМ — контроллер. Для этого оператор в ответ на «*» должен ввести ⟨ВК⟩. Признаком пультового режима является литера «>». Для того чтобы вернуться из пультового режима к вводу новой командной строки, необходимо ввести двойное CTRL/C. Система команд пультового режима (табл. 2) практически аналогична системе команд системного отладчика ODT [3] операционной системы RT-11 с некоторыми отличиями:

Таблица 1

Состав макробиблиотеки

Имя макрокоманды	Аргумент в команде EMT	Комментарии
.TOUT	0	Вывод литеры из R0 на терминал
.TINP	1	Ввод литеры в R0 с клавиатуры
.T08, .T010	2,3	Вывод содержимого R0 на терминал в виде восьмеричного и десятичного числа со знаком соответственно
.T18, .T110	4,5	Ввод данного в R0 в виде восьмеричного и десятичного числа со знаком с клавиатуры

Окончание табл. 1

Имя макрокоманды	Аргумент в команде ЕМК	Комментарии
.REG	6	Вывод содержимого регистров микропроцессора на терминал
.PRI	7	Печать сообщения, находящегося по адресу R0
.WR, .RD	10, 11	Открытие в ПЭВМ файла для записи и чтения соответственно. R0 указывает на блок из четырех слов со спецификацией файла в коде RAD50
.OUT	12	Запись значения R0 в файл
.INP	13	Чтение слова из файла в R0
.CLO, CLI	14, 15	Закрытие файла в ПЭВМ для записи и чтения соответственно
.EX	16	Окончание работы, перевод кроссового монитора в активное состояние (эквивалентно двойной команде оператора CTRL/C)
—	17, 20, 21, 22	Используются резидентным монитором

```
; ——————— З А Г О Л О В О К ———————
.PSECT PROG
;              АССЕМБЛЕР MACRO-11/ RT11
; ***********************************************************
; * ПРОГРАММА ВЫВОДА ДАННЫХ ВНЕШНЕГО УСТРОЙСТВА            *
; * КОНТРОЛЛЕРА НА ТЕРМИНАЛ ЭВМ  С ОДНОВРЕМЕННЫМ           *
; * СОХРАНЕНИЕМ В ФАЙЛЕ.                                   *
; ***********************************************************
; ——————— О П Р Е Д Е Л Е Н И Я ———————
        RS=177550       ;РЕГИСТР СОСТОЯНИЯ ВУ КОНТРОЛЛЕРА
        RD=177552       ;РЕГИСТР ДАННЫХ ВУ КОНТРОЛЛЕРА
; ——— О С Н О В Н А Я   П Р О Г Р А М М А ———

        .MCALL .PRI,.TINP,.TI10,.OUT   ;ВЫЗОВ МАКРОКОМАНД
        .MCALL .TOUT,.TO8,.TO10,.CLO   ;ИЗ БИБЛИОТЕКИ
        .MCALL .EX,.WR

BEG:    .PRI    #MNG1   ;ПЕЧАТЬ СООБЩЕНИЯ
        .TINP   R5      ;R5-УКАЗАТЕЛЬ РЕЖИМА ПЕЧАТИ
        .WR     #FIL    ;ОТКРЫТЬ ФАЙЛ ДЛЯ ЗАПИСИ В ЭВМ
        .PRI    #MNG2   ;ПЕЧАТЬ СООБЩЕНИЯ
        .TI10   R1      ;R1-СЧЕТЧИК СЛОВ
REP:    .TOUT   #12     ;ПЕРЕВЕСТИ СТРОКУ
        .TOUT   #15
WAI:    TSTB    @#RS
        BPL     WAI     ;ОЖИДАНИЕ ГОТОВНОСТИ ВУ
        MOV     @#RD,R0 ;ПРИНЯТЬ ДАННОЕ
        .OUT            ;СОХРАНИТЬ ДАННОЕ В ФАЙЛЕ
        CMPB    #'B,R5
        BNE     DES
        .TO8            ;ПЕЧАТЬ ДАННОГО В ВОСЬМЕРИЧНОМ
                        ;КОДЕ
        BR      WM
DES:    .TO10           ;ПЕЧАТЬ ДАННОГО В ДЕСЯТИЧНОМ КОДЕ
WM:     SOB     R1,REP  ;ВВОД СЛЕДУЮЩЕГО ДАННОГО
        .CLO            ;ЗАКРЫТЬ ФАЙЛ
        .EX             ;ЗАКОНЧИТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ

MNG1:   .ASCII <16>/ФОРМАТ ПЕЧАТИ ДАННЫХ:/<12><15>
        .ASCII /B -ВОСЬМЕРИЧНЫЙ/<12><15>
        .ASCIZ /D -ДЕСЯТИЧНЫЙ/<17>
MNG2:   .ASCII <16><12><15>/КОЛИЧЕСТВО СЛОВ ДАННЫХ/
        .ASCII <17><200>
        .EVEN
FIL:    .RAD50 /DK /
        .RAD50 /DANN
        .RAD50 /DAT/

        .END    BEG
```

Рис. 4. Пример прикладной программы

Таблица 2

Команды пультового режима

Команда	Комментарии
A/	Открывает слово по адресу A в виде шестизначного восьмеричного числа
AX	Открывает слово в коде
A\	Открывает байт, чтобы ввести новое значение байта (предварительно необходимо ввести литеру «=»)
<ВК>	Закрывает ячейку или регистр
<ПС>	Закрывает ячейку или регистр и открывает следующую ячейку или регистр
—	Закрывает ячейку или регистр и открывает предыдущую ячейку или регистр
—	Открывает ячейку по относительному адресу
@	Открывает ячейку по абсолютному адресу
>	Открывает ячейку по смещению (аналогично команде условного перехода процессора)
<	Восстанавливает адрес, который был до выполнения команд — —@>
0	Обеспечивает вычисление смещения для инструкций перехода
R0 ... R7	Обращение к регистрам процессора
RS	Обращение к регистру состояния процессора
AR0 ... AR7	Заносит A в соответствующий регистр смещения
B0 ... B7	Указывает адреса точек останова, заводит новые точки останова. Для удаления останова необходимо после B0 ... B7 ввести команду D
C0 ... C7	Обращение к счетчику прохождений соответствующих точек останова
AG	Выполняет программу с адреса A
KP	Продолжает выполнение программы, после достижения точки останова заносит в счетчик прохождения точки останова число K
<ЗБ>	Удаляет последнюю введенную литеру, выполняет K команд программы и переводит монитор в пультовый режим
A1	Выводит на терминал фрагмент листинга с адресом A
↑	Сдвигает листинг вниз
↓	Сдвигает листинг вверх
↓	Включает режим сопровождения адресов
↖	Выключает режим сопровождения

синтаксис команд изменен так, чтобы для ввода любой команды требовалось нажать всего одну клавишу;

имеется команда удаления последней введенной литеры <ЗБ>;

неверные команды монитор игнорирует;

введены команды работы с листингом отлаживаемой программы.

Адреса в пультовом режиме вводятся в таком же формате, как и в ODT: N, K, где N — номер регистра смещения (от 0 до 7); K — восьмеричное число с максимальным значением 177777.

Абсолютное значение адреса вычисляется как сумма K и содержимого регистра смещения. Если в командной строке указано имя файла листинга, то монитор, перейдя в пультовый режим, разделит экран ПЭВМ на две страницы. В верхней отображается листинг отлаживаемой программы, в нижней принимаются команды пультового режима.

Уменьшить время, затрачиваемое на отладку программ, можно, используя многозадачный FB-монитор [4] операционной системы RT-11. Если монитор разместить в оперативном разделе, а редактор текстов — в фоновом, то оператор, переходя из оперативного раздела в фоновый (из монитора в редактор), может сразу исправлять обнаруженные ошибки в исходном файле с программой на макроассемблере.

Кроссовый монитор АПК поставляется в файлах RA.SAV и RA.REL, резидентный монитор — в файлах MON.MAC, макробиблиотека — в файлах MACLIB.SML.

Телефон 534-62-05, Москва

ЛИТЕРАТУРА

1. Прангишвили И. В. Микропроцессоры и микроЭВМ.— М.: Энергия.— 1979.— 223 с.
2. Дшхунян В. Л., Борщенко Ю. И., Отрохов Ю. Л., Шишарин С. А. Одноплатные микроЭВМ ряда «Электроника МС1201». // Микропроцессорные средства и системы.— 1985.— № 2.— С. 8—13.
3. Сингер М. МиниЭВМ PDP-11: Программирование на языке ассемблера и организация машины / Пер. с англ.— М.: Мир, 1984.— 272 с.
4. Валикова Л. И., Вигдорчик Г. В., Воробьев А. Ю., Лукин А. А. / Под общей редакцией В. И. Семика. Операционная система СМ ЭВМ РАФОС: Справочник.— М.: Финансы и статистика.— 1984.— С. 207.

Статья поступила 22 мая 1987

УДК 681.325—181.5

П. И. Пальгин, С. М. Рождественский, И. И. Шагурин

МИКРОПРОГРАММИРУЕМЫЙ КОМПЛЕКС ДЛЯ ТЕСТИРОВАНИЯ И ОТЛАДКИ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Основу контрольно-отладочного комплекса составляют микроЭВМ ДВК3М2 и микропрограммируемый блок отладки и контроля систем (БОКС), связывающий канал микроЭВМ и отлаживаемую (тестируемую) микропроцессорную систему (МПС) (см. рисунок). Программные средства включают стандартную операционную систему РАФОС (ОС ДВК), монитор MONEML и пакет служебных программ для разработки микропрограмм, обеспечивающие работу БОКС.

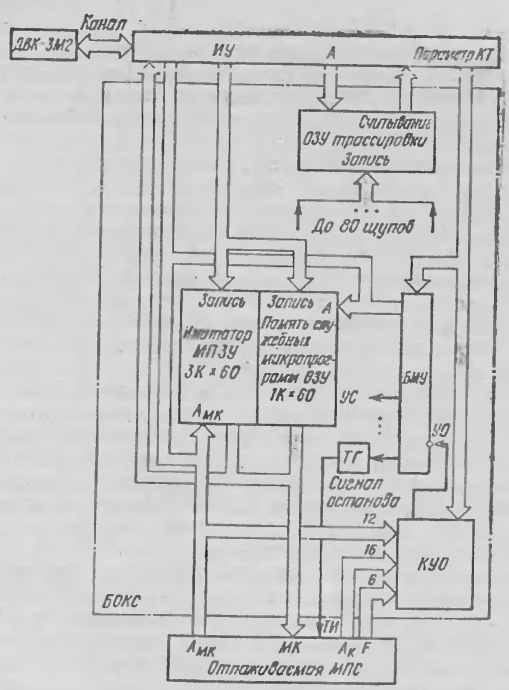
Комплекс позволяет в процессе отладки выполнять следующие функции:

трассировку программ и микропрограмм, выполняемых отлаживаемой МПС;

останов выполнения программ (микропрограмм) в заданных контрольных точках;

просмотр в режиме останова содержимого трассировочной памяти, регистров и ячеек памяти отлаживаемой МПС;

коррекцию микропрограмм отлаживаемой МПС, которые хранятся в имитаторе микропрограммного ЗУ (МПЗУ), входящего в состав БОКС.



Структурная схема микропрограммируемого комплекса

Пользователь, начиная работу, вводит в мпкроЭВМ с помощью монитора параметры точек останова (КТ), служебные и рабочие микропрограммы. По каналу микроЭВМ через интерфейсное устройство (ИУ) последовательно осуществляются запись текста рабочих микропрограмм в имитатор МПЗУ, запись микропрограмм в память служебных микропрограмм, считывание введенных микропрограмм с целью проверки правильности записи, ввода параметров первой КТ в компаратор условий останова (КУО) и блок микропрограммного управления (БМУ).

Под управлением монитора MONEML одновременно запускаются отлаживаемая МПС и БОКС. При выполнении условий останова, которые фиксируются в КУО, вырабатывается сигнал останова (УО) и БМУ разрешает запись состояния до 80 точек отлаживаемой МПС в ОЗУ трассировки. Кроме того, БМУ устанавливает адрес служебной микропрограммы на адресных входах памяти служебных микропрограмм. Выполняя эти микропрограммы, отлаживаемая МПС в течение ряда циклов выводит в ОЗУ состояние заданных ячеек памяти и регистров, которые доступны микропрограммно. Затем вся информация из ОЗУ через ИУ считывается в память микроЭВМ и выводится на экран дисплея. После анализа поступивших данных пользователь может продолжить отладку или внести изменения в текст рабочих микропрограмм.

Основные узлы БОКС имеют следующее назначение: ИУ обеспечивает взаимодействие между БОКС и микроЭВМ и выполняет ввод и вывод рабочих и служебных микропрограмм, параметров останова (адресов команд и микрокоманд, сочетаний флагов и числа повторений условий останова), задание адресов ОЗУ трассировки, вывод данных, ввод слова управления тактовым генератором (через БМУ). Оно выполнено на микросхемах параллельных портов типа КР580ИК55 и регистров типа К589ИР12, К1804ИР3. КУО сравнивает текущие значения флагов, адресов команд и микрокоманд с поступающими от микроЭВМ условиями останова и при совпадении формирует сигнал с УО для БМУ. КУО собран на микросхемах К531СП1П.

Имитатор МПЗУ, память служебных микропрограмм и ОЗУ трассировки расположены на отдельной плате. Все эти ЗУ имеют раздельные входы записи-считывания. Имитатор МПЗУ структурно аналогичен МПЗУ отлаживаемой МПС и выполняет функции памяти микропрограмм. Память служебных микропрограмм дополняет адресное пространство имитатора и имеет аналогичную архитектуру. Имитатор МПЗУ и память служебных микропрограмм реализованы на микросхемах К541РУ1. Для получения трассировки состояний отлаживаемой МПС в реальном масштабе времени быстродействующее ОЗУ трассировки «пропускает» всю информацию, поступающую на информационные входы. ОЗУ выполнено на микросхемах КР185РУ9, имеющих время выборки $t_в = 40$ нс, что обеспечивает тактовую частоту БОКС, равную 10 МГц.

БМУ управляет работой БОКС. Наличие блока позволило реализовать схему, быстро перестраиваемую под конкретные условия. Блок состоит из основной СУАМ (схемы управления адресом микрокоманды), ПЗУ микропрограмм, регистра микропрограмм и вспомогательной СУАМ. Основная СУАМ задает адрес выбираемой из ПЗУ микропрограммы, соответствующей определенному режиму останова.

Комплекс в целом реализует следующие фиксированные режимы останова: по адресу команды (16 разрядов), адресу микрокоманды (12 разрядов), комбинациям флагов (6 разрядов), адресу команды и флагам, адресу команды и микрокоманды, адресу микрокоманды и флагам. Число повторений условий останова — 4095. Если фиксированных условий останова недостаточно, пользователь может определить их самостоятельно, изменив микропрограммы работы БМУ.

Вспомогательная СУАМ формирует адреса микрокоманд для блока памяти служебных микропрограмм и запоминает адрес возврата в место останова. Основная и вспомогательная СУАМ собраны на микросхемах КР1804, микропрограммное ПЗУ — на микросхемах К556РТ5.

Тактовый генератор (ТГ) предназначен для согласования работы отлаживаемой МПС и БОКС и выполнен на микросхеме К1804ГГ1.

БОКС занимает две печатные платы (плата логики и плата памяти) в конструктиве «Электроника 60».

Напряжение питания БОКС — 5 В, потребляемая мощность — 20 В, объем ОЗУ трассировки — 64×80, число повторений КТ — 4096. Работа с комплексом проходит два этапа: настройку БОКС и контроль и отладку МПС.

На этапе 1 исходной информацией для настройки являются структура отлаживаемой МПС и загрузочный модуль программы (микропрограммы) ее функционирования. На этапе 2 пользователь определяет режимы останова МПС, и если имеющихся в комплексе фиксированных режимов останова недостаточно, то программирует БОКС на нужные режимы с помощью транслятора TRANSL, программно-логической модели EMULAT и прошивки ПЗУ PRZU. Программы написаны на языке Паскаль и занимают соответственно, 29, 29 и 17 блоков оперативной памяти. Разрядность микрокоманды микроассемблера для управления БОКС в программе TRANSL составляет 24 бита, разделенных на 11 полей, отвечающих за управление отдельными функциональными узлами. Имеется возможность ввода комментариев. Трансляция микропрограммы осуществляется построчно. По окончании трансляции на экран микроЭВМ выводится файл, содержащий сообщения об ошибках. Программа EMULAT предназначена для отладки оттранслированной микропрограммы управления БОКС и работает в диалоге в режимах настройки (определения внешних и внутренних связей) и отладки (пошаговом и автоматическом).

Результатом отладки микропрограммы управления БОКС на программно-логической модели служит карта прошивки ПЗУ микропрограмм для БМУ.

После определения режимов останова пользователь с помощью монитора MONEML задает конкретные условия останова выполнения рабочей программы (микропрограммы) МПС. Система команд монитора следующая: SCOM — останов по адресу команды; SMCOM — останов по адресу микрокоманды; SFLAG — останов по комбинации внешних признаков (флагов); OPT — определение имен полей ОЗУ трассировки БОКС и способа их представления; DIR — просмотр содержимого ОЗУ трассировки: MPZUCR — задание новых контрольных точек в программе монитора.

Возможно применение комбинаций команд SCOM/SMCOM; SCOM/SFLAG; SMCOM/SFLAG. При обращении к командам SCOM, SMCOM, SFLAG задается адрес команды, микрокоманды и комбинации флагов останова. Диапазон адресов команд 0..65535, адресов микрокоманд 0..4095, комбинаций флагов — 6 флагов. Далее пользователь вводит число повторений заданного условия в диапазоне 0...4095. При обращении к команде OPT необходимо указать форму представления результата (двоичная, десятичная, восьмеричная) и задать распределение выводимой информации по разрядам ОЗУ трассировки. Команда DIR определяет способ просмотра содержимого ОЗУ трассировки (экран, печать), глубину вывода (до 64 шагов) и выводимые поля ОЗУ трассировки, заданные командой OPT. При использовании команды MPZUCR выделяются КТ для монитора MONEML. Предусмотрена возможность программного задания задержки останова (в тактах) ТГ после нахождения точки останова. Задаваемые пользователем КТ записываются в файл контрольных адресов.

Пример последовательности задания останова по адресу команды и микрокоманды. После запуска программы MONEML на экране появится сообщение: «Файл контрольных адресов введен? (Y/N)». После набора Y программа предлагает символом * ввести комбинацию команд режима останова. Пользователь набирает SCOM/SMCOM. На экране появится сообщение: «Введите адрес команды останова». Пользователь вводит адрес останова и нажимает клавишу ⟨ВК⟩. На экране появится сообщение: «Введите адрес микрокоманды останова». Пользователь вводит адрес останова и нажимает ⟨ВК⟩. На экране появится сообщение: «Введите число повторений заданного условия». Пользователь вводит число повторений и нажимает клавишу ⟨ВК⟩. На экране снова появится сообщение: «Введите число повторений заданного условия», относящееся к заданию адреса микрокоманды. Пользователь вводит число повторений и нажимает клавишу ⟨ВК⟩. Программа отыскивает нужную точку, выдает сообщение: «Точка останова найдена» и предлагает ввести следующую комбинацию команд.

Таким образом, на этапе 1 пользователь составляет файл КТ, в котором задаются конкретные условия останова и вид представления выводимой информации.

На этапе 2 пользователь подключает щупы БОКС к интересующим его точкам МПС, вводит тексты рабочих и служебных микропрограмм в ЗУ. Затем запускается программа MONEML. После останова в заданном месте на экран дисплея выводится информация из ОЗУ трассировки о состоянии МПС. Процесс отладки может осуществляться в автоматическом режиме. Конечным результатом отладки являются макет и программа (микропрограмма) функционирования МПС.

Комплекс сочетает функции генератора слов, логического анализатора и эмулятора, что позволяет активно воздействовать на отлаживаемую МПС и выводить на экран дисплея состояние микропрограммно доступных ячеек памяти, регистров, стеков; тестировать МПС с помощью микропрограмм, записанных в память служебных микропрограмм. Микропрограммная архитектура БОКС обеспечивает гибкость при настройке на различные МПС и высокое быстродействие, необходимое для отладки и контроля в реальном времени. Аппарат-

ные и программные средства комплекса могут быть использованы как для отладки микропрограммируемых МПС, так и МПС на основе микропроцессора с фиксированным набором команд в пределах ограничений по разрядности шины адреса и быстродействию.

Описанный комплекс может быть использован как автономное устройство или в составе универсальных отладочных систем, построенных на основе микроЭВМ типа ДВК3М2 [1, 2], например аппаратно-программный комплекс МИКРОСОТ [3].

Телефон 324-91-55, Москва, Рождественский С. М.

ЛИТЕРАТУРА

1. Средства отладки, лабораторный практикум и задачник. / Под ред. Л. Н. Преснухина.— М.: Высшая школа, 1986.— Т. 3.
2. Дж. Макленд. Персональный компьютер, расширяющий возможности контрольно-измерительных и испытательных систем // Электроника.— 1986.— № 6.— С. 52—68.
3. Иванов Е. Л., Муренко Л. Л., Широков Ю. Ф. Универсальные отладочные средства автоматизации проектирования микропроцессорных устройств // Микропроцессорные средства и системы.— 1984.— № 3.— С. 53—57.

Статья поступила 17 июня 1987

УДК 681.321.06

Д. Ю. Бухвостов, Ю. П. Жиляев, А. И. Иванов, В. Ю. Казанцев, А. Н. Печенкин

ДВК-2 — «ЩЕЛЬ» — ПРОСТОЙ ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ОТЛАДКИ ВСТРАИВАЕМЫХ УПРАВЛЯЮЩИХ МИКРОЭВМ НА БАЗЕ МПК БИС СЕРИИ К588

Наиболее трудна для разработчиков встраиваемых управляющих микроЭВМ совместная отладка программного обеспечения (ПО) и аппаратуры, так как средств для общения с управляющей микроЭВМ и управления ею недостаточно.

В этом случае можно использовать специализированные отладочные комплексы, например «Электроника НЦ-803» или АРМ 2.05.01 * для эмулирования аппаратуры управляющей микроЭВМ. Однако применение таких комплексов для отладки взаимодействия ПО и аппаратуры управляемой системы неоправданно дорого.

Другой путь — использовать для отладки процессор управляющей микроЭВМ (заведомо исправный). В этом случае можно организовать канал общения управляющей микроЭВМ с программно-совместимой серийной микроЭВМ, оснащенной всеми необходимыми средствами хранения и отображения информации. Для организации такого канала общения процессор управляющей микроЭВМ должен иметь систему прерываний и прямого доступа к памяти (ПДП).

Отладочный комплекс ДВК-2 — «ЩЕЛЬ» (в качестве инструментальной используется микроЭВМ ДВК-2) разработан для отладки аппаратуры и ПО встраиваемых управляющих микроЭВМ на базе МПК БИС К588.

Аппаратурная часть комплекса (рис. 1): инструментальная микроЭВМ ДВК-2; плата «Щель», встраиваемая в инструментальную микроЭВМ; отлаживаемая микроЭВМ и переходная плата для подключения процессора отлаживаемой микроЭВМ.

Программная часть комплекса — это операционная система ОС ДВК; монитор «Щель», размещаемый в памяти инструментальной микроЭВМ; программа START — STOP, размещаемая в памяти отлаживаемой микроЭВМ.

Идея комплекса — использовать возможности ОС ДВК для создания ПО встраиваемой управляющей микроЭВМ и для отладки ее аппаратуры, а также автоматизировать совместную отладку ПО и аппаратуры. В комплексе со стороны инструментальной микроЭВМ

* Мильнер А. Д. Автоматизированные рабочие места для проектирования встроенных микропроцессорных устройств // УСиМ. — 1985. — № 5. — С. 51—58.

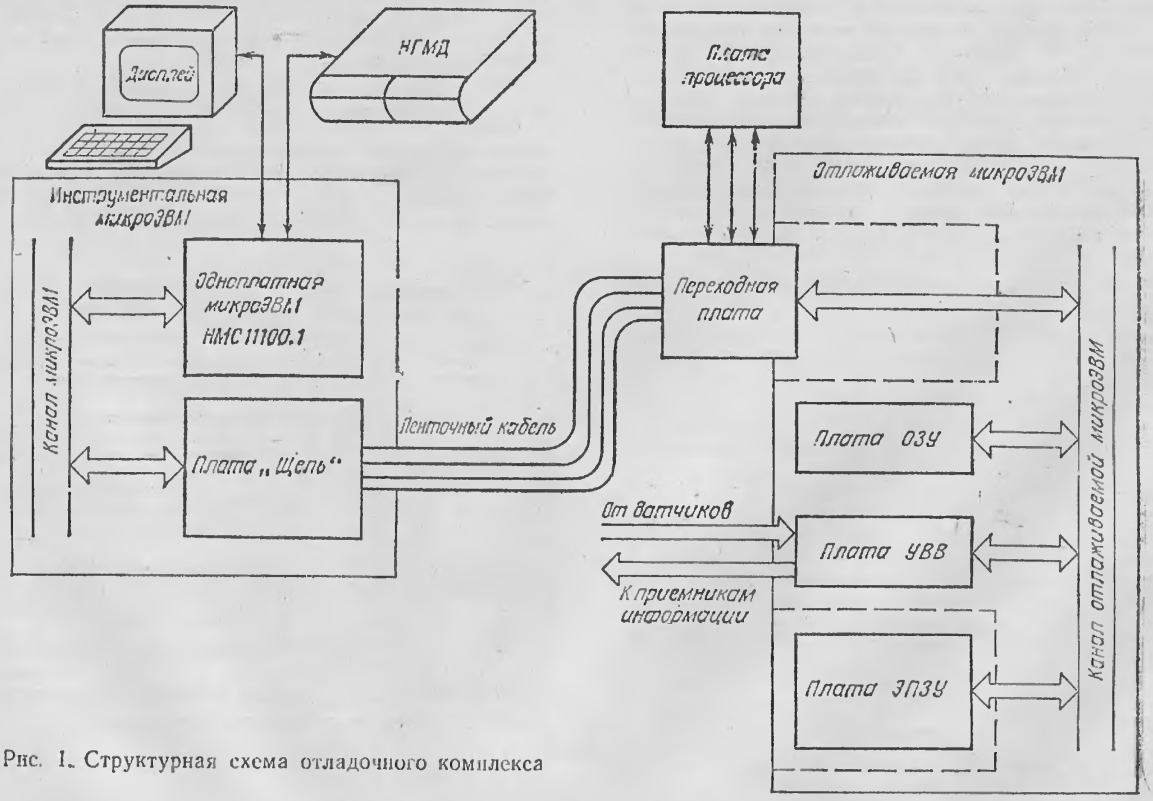


Рис. 1. Структурная схема отладочного комплекса

реализован программный доступ к адресному пространству отлаживаемой микроЭВМ в режиме ПДП и программное управление режимами работы (запуск, останов, останов в заданной точке) процессора отлаживаемой микроЭВМ. Это позволяет реализовать: пуск с заданного адреса; установку и удаление точки останова; прерывание; продолжение выполнения программ с точки прерывания; отображение и изменение содержимого ЗУ, регистров внешних устройств, регистров общего назначения и слова состояния программы на дисплее инструментальной микроЭВМ; проведение тестово-диагностических процедур с ЗУ и каналом отлаживаемой микроЭВМ; преобразование десятичных чисел во внутренний код отлаживаемой микроЭВМ и обратно.

Технические характеристики комплекса ДВК-2 — „Щель"

Разрядность канала, бит	16
Система команд	ОСТ 11 305.909—82
Интерфейс канала	ОСТ 11 305.903—80
Число функций	25
Объем нестандартного (140×230 мм) оборудования, плат	2
Требуемая память для ПО	
инструментальной микроЭВМ, слов	6000
отлаживаемой микроЭВМ, слов	76
Уровень сигналов	ТТЛ
Напряжение источника питания:	
плата «Щель», В	+5±10%
переходная плата, В	+5±10%
Потребляемая мощность:	
плата «Щель», Вт	6
переходная плата, Вт	5

Плата «Щель» подключается к каналу инструментальной микроЭВМ и входит в ее конструктив. Переходная плата вставляется в конструктив отлаживаемой микроЭВМ на место платы процессора, который на время отладки подключается к специальному разъему переходной платы. По окончании работ переходная плата удаляется, и процессор отлаженной микроЭВМ устанавливается на свое место. Плата ПЗУ отлаживаемой микроЭВМ на время отладки заменяется платой эмулятора ПЗУ (ЭПЗУ). Плата ЭПЗУ — это ОЗУ, в ячейки которого со стороны инструментальной микроЭВМ разрешены запись и считывание, а со стороны отлаживаемой микроЭВМ — только считывание.

Аппаратура. Со стороны инструментальной микроЭВМ плата «Щель» (рис. 2) содержит четыре программно-доступных регистра, адреса которых находятся в зоне внешних устройств: регистр управления (РУ), регистр состояния (РС), регистр данных (РД) и регистр адреса (РА). Четырехразрядный РУ предназначен для управления режимом работы отлаживаемой микроЭВМ посредством запуска процессора (Пуск Н); требования прямого доступа к каналу (ТПД Н); сброса доступа к каналу (Сброс ПДП Н); требования останова (Т Ост Н). РС предназначен для сообщения инструментальной микроЭВМ о состоянии отлаживаемой микроЭВМ посредством сигналов предоставления доступа к каналу (ПВ Н); выполнения программы останова (Ост Н).

Режимом работы отлаживаемой микроЭВМ управляют так. Перед обменом информацией инструментальная микроЭВМ устанавливает сигнал ТПД Н. Получив через РС сигнал ПВ Н, инструментальная микроЭВМ получает доступ к каналу отлаживаемой микроЭВМ. Работа с каналом заканчивается сигналом Сброс ПДП Н. Сигнал Т Ост Н, поступающий в отлаживаемую микроЭВМ, воздействует на вход требования прерывания процессора, который выполняет программу обслуживания прерывания, заканчивающуюся командой установки флага Ост Н. Сигнал Ост Н, поступая через РС в инструментальную микроЭВМ, сигнализирует о выполнении программы останова отлаживаемой микроЭВМ.

Доступ к адресному пространству отлаживаемой микроЭВМ осуществляется с помощью РА, в который программно заносится необходимый адрес, и РД, через который программно осуществляется обмен данными. Запись информации в РА сопровождается выработкой в устройстве управления сигнала СИА Н, а запись в РД (считывание из РД) — сигналов Вывод Н (Ввод Н). Эти три сигнала через буфер шины управления (ШУ) и ленточный кабель передаются на переходную плату.

Аппаратно РА и РД объединены и выполнены на двунаправленных шинных формирователях. Так же объединены РУ и РС. Работой шинных формирователей (ШФ) управляет устройство управления, открывающее ШФ РД и РА на выдачу информации при передаче адреса (или данных) в отлаживаемую микроЭВМ и на прием информации при вводе данных от отлаживаемой микроЭВМ. ШФ РУ и РС открывается на выдачу информации при обращении к РУ и на прием при обращении к РС. Поскольку обмен информацией происходит в цикле ПДП, РД и РА блокируются при неактивном значении сигнала ПВ Н, поступающего на плату «Щель» от переходной платы. Устройство управления также формирует сигнал СИП Н при обращении инструментальной микроЭВМ к одному из четырех регистров платы «Щель».

Переходная плата (рис. 3) согласует канал отлаживаемой микроЭВМ с сигналами платы «Щель», а также формирует циклы прерывания и прямого доступа к каналу отлаживаемой микроЭВМ и сигнализацию о выполнении этих операций. Буфер шины данных-адреса (ШДА) и буфер ШУ переходной платы согласуют сигналы длинной линии (ленточного кабеля)

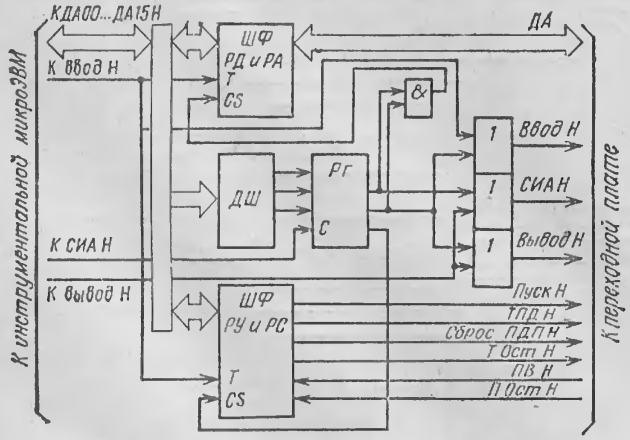


Рис. 2. Функциональная схема платы «Щель»

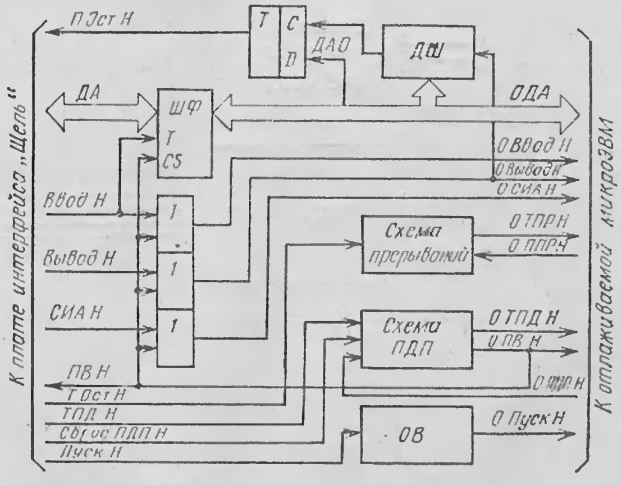


Рис. 3. Функциональная схема переходной платы

и сигналы канала отлаживаемой микроЭВМ (сигналы отлаживаемой микроЭВМ имеют индекс «0»). Схема ПДП формирует цикл ПДП и вырабатывает сигнал ПВ Н, поступающий в отлаживаемую микроЭВМ и в плату «Щель». Схема индикации останова — это одноразрядный порт вывода, настроенный на адрес 160000_8, программно доступный отлаживаемой и инструментальной микроЭВМ. Для запуска отлаживаемой микроЭВМ в переходной плате с помощью одновибратора вырабатывается импульс запуска, переводящий процессор отлаживаемой микроЭВМ на микропрограмму начального пуска.

Программное обеспечение комплекса состоит из монитора «Щель», загружаемого в память инструментальной микроЭВМ, и программы START — STOP, загружаемой в память отлаживаемой микроЭВМ.

Монитор «Щель» — это программа, в диалоговом режиме выполняющая следующие команды управления и отображения состояния отлаживаемой микроЭВМ:

отображение на дисплее и изменение содержимого любого слова ОЗУ и ЭПЗУ;

отображение на дисплее и изменение содержимого всех регистров общего назначения (РОН) и слова состояния процессора (ССП). Выполнение команды реализуется посредством доступа к области ОЗУ с именем REGIST;

загрузку заданного массива из памяти инструментальной микроЭВМ в память отлаживаемой микроЭВМ или обратно. При помощи этой команды отлаживаемая целевая программа загружается в ЭПЗУ;

пуск с заданного адреса. Выполнение команды реализуется записью заданного адреса пуска в седьмой РОН (счетчик команд СК), с последующей подачей сигнала запуска;

останов по команде с клавиатуры инструментальной микроЭВМ. Выполнение команды реализуется подачей сигнала требования прерывания отлаживаемой микроЭВМ с последующим тестированием бита обратной связи в РС. После установки бита обратной связи в РС, что свидетельствует об останове отлаживаемой микроЭВМ, в режиме ПДП считывается и отображается на дисплее содержимое всех РОН и ССП;

установка и отмена точки останова в программе отлаживаемой микроЭВМ. Выполнение команды реализуется подачей сигнала запуска. Поскольку СК после останова настроен на следующую за точкой останова команду, продолжение произойдет автоматически.

Программа START — STOP (рис. 4) располагается в вершине ЭПЗУ и состоит из трех модулей: START, STOP и STOPB. Все три модуля работают с областью памяти из девяти слов, расположенной в вершине ОЗУ и имеющей название REGIST. Эта область служит для хранения содержимого всех РОН и ССП перед остановом и загрузки их перед запуском. Программа START загружает РОН и ССП из области REGIST. Она получает управление через вектор 24. Загрузка седьмого РОН, являющегося СК, и означает пуск с заданного адреса. Программа STOP выгружает содержимое РОН и ССП в область REGIST, подает сигнал обратной связи (ОСТ Н) и переходит к зацикливанию. Останов процессора имитируется зацикливанием процессора (до подачи сигнала запуска). Поскольку программа STOP получает управление по прерыванию, значения СК и ССП выгружаются в REGIST из стека. Программа STOPB работает аналогично программе STOP, однако она получает управление по командному прерыванию ВРТ и перед сохранением содержимого РОН и ССП выполняет команду, являющуюся точкой останова. Таким образом, небольшая часть памяти отлаживаемой микроЭВМ (66 слов в вершине ЭПЗУ и 9 слов в вершине ОЗУ), а также ячейка с адресом 160000 (порт вывода сигнала обратной связи) задействованы под нужды комплекса «Щель» и не могут использоваться целевой программой отлаживаемой микроЭВМ. Впрочем, для последней это несущественное ограничение.

```
;          ПРОГРАММЫ  START,STOP И STOPB
;           АССЕМБЛЕР MACRO-11 / ОС ДВК .
;   ***********************************************
;   * START - ПРОГРАММА ЗАПУСКА                   *
;   * STOP  - ПРОГРАММА ОСТАНОВА ПО ВНЕШНЕМУ СИГНАЛУ *
;   * STOPB - ПРОГРАММА ОСТАНОВА ПО КОМАНДНОМУ ПРЕ- *
;   *         РЫВАНИЮ ДЛЯ ОТЛАДКИ (ВРТ)           *
;   * REGIST- ОБЛАСТЬ ОЗУ ДЛЯ СОХРАНЕНИЯ СОДЕРЖИМОГО *
;   *         РОН И ССП                           *
;   ***********************************************

START:  MOV     REGIST,   R0     ;ЗАГРУЗКА РОН
        MOV     REGIST+2, R1
        MOV     REGIST+4, R2
        MOV     REGIST+6, R3
        MOV     REGIST+10,R4
        MOV     REGIST+12,R5
        MOV     REGIST+14,SP
        MTPS    REGIST+20        ;ЗАГРУЗКА ССП
        JMP     @REGIST+16       ;ЗАГРУЗКА СК

STOPB:  NOP                      ;МЕСТО КОМАНДЫ
                                 ;ТОЧКИ ОСТАНОВА
        MFPS    REGIST+20        ;ВЫГРУЗКА ССП
        MOV     R0,REGIST        ;ВЫГРУЗКА РОН
        MOV     R1,REGIST+2
        MOV     R2,REGIST+4
        MOV     R3,REGIST+6
        MOV     R4,REGIST+10
        MOV     R5,REGIST+12
        MOV     (SP)+,REGIST+16 ;ВЫГРУЗКА СК
        TST     (SP)+
        MOV     SP,REGIST+14
        MOV     #177777,@#160000;СИГНАЛ ОБРАТНОЙ СВЯЗИ
ST1:    BR      ST1              ;ЗАЦИКЛИВАНИЕ

STOP:   MOV     R0,REGIST        ;ВЫГРУЗКА РОН
        MOV     R1,REGIST+2
        MOV     R2,REGIST+4
        MOV     R3,REGIST+6
        MOV     R4,REGIST+10
        MOV     R5,REGIST+12
        MOV     (SP)+,REGIST+16 ;ВЫГРУЗКА СК
        MOV     (SP)+,REGIST+20 ;ВЫГРУЗКА ССП
        MOV     SP,REGIST+14
        MOV     #177777,@#160000;СИГНАЛ ОБРАТНОЙ СВЯЗИ
ST2:    BR      ST2              ;ЗАЦИКЛИВАНИЕ
                ;ВЕРШИНА ЭПЗУ

REGIST: 0       ;R0
        0       ;R1
        0       ;R2
        0       ;R3
        0       ;R4
        0       ;R5
        0       ;SP
        0       ;PC
        0       ;PSW
                ;ВЕРШИНА ОЗУ
```

Рис. 4. Программа «START—STOP»

Стандартный способ применения объектного модуля (6 Кбайт) программы монитора «Щель» заключается в следующем. Объектный модуль разработанной на инструментальной микроЭВМ целевой программы компонуется с объектным модулем монитора «Щель» в единый загрузочный модуль с помощью программы LINK ОС ДВК. После включения этого модуля в работу монитор «Щель», выполнив предварительные действия (в частности, загрузку в память отлаживаемой микроЭВМ программы START — STOP), переходит в режим диалога. Командой загрузки массива целевая программа загружается в ЭПЗУ отлаживаемой микроЭВМ, и можно приступать к ее отладке.

450000, Уфа-центр, ул. К. Маркса, 12, УАИ, кафедра ПЭ, Иванову А. И.; тел. 23-78-36

Статья поступила 17 ноября 1987

УДК 681.326

В. В. Конько, А. Н. Штурман, В. А. Недорубан

АППАРАТНО-ПРОГРАММНЫЙ ОТЛАДОЧНЫЙ КОМПЛЕКС

МикроЭВМ «Электроника К1-20», выполненная на базе МПК БИС серии КР580, широко используется в качестве контроллера технологического, испытательного, контрольно-измерительного оборудования. Диалог с микроЭВМ осуществляется с помощью программы Монитор через пульт управления с шестнадцатеричной клавиатурой и девятиразрядный семисегментный светодиодный дисплей. Программа, под управлением которой микроЭВМ работает как контроллер, хранится в ППЗУ пользователя. Новые программы для микроЭВМ создаются с учетом приведенной выше конфигурации технических средств (рис. 1).

Для выполнения этапов 1...4 целесообразно применить программно совместимую с «Электроникой К1-20» ЭВМ с развитыми аппаратными и программными средствами, а именно вычислительный комплекс на базе микроЭВМ «Электроника МС 0401». Он используется в качестве рабочих мест отладчика программных средств для аппаратуры на базе БИС серии КР580.

Система команд, представление информации определяются архитектурой МП КР580ИК80А:

Формат данных	Двоичные числа с фиксированной запятой, отрицательные — в дополнительном коде со знаком, 8 бит
Способы адресации	Прямая, косвенная, непосредственная, неявная
Число команд	78 (111 операций)
Разрядность, бит	
адреса	16
данных	8; 16
Емкость, Кбайт	
ПЗУ	16
ОЗУ	64
внешней памяти (накопители на магнитных дисках), Мбайт	до 1
Система прерываний	Программная
число линий аппаратного прерывания	8
число маскируемых уровней	8
глубина прерываний	8
время реакции на прерывание, мкс	18
Интерфейс	
Разрядность, бит	
данных	8 или 1
адреса	8
Скорость обмена	
по параллельным каналам по типу интерфейса ИРПР-72, Кбайт/с	50
по последовательному каналу интерфейса типа ИРПС, бит/с	56 000
Количество 16-разрядных программируемых таймеров для отладки программ	2
Внешние устройства	Термопечать, видеотерминал, накопитель на магнитных дисках, ленточный перфоратор, фотосчитывающее устройство

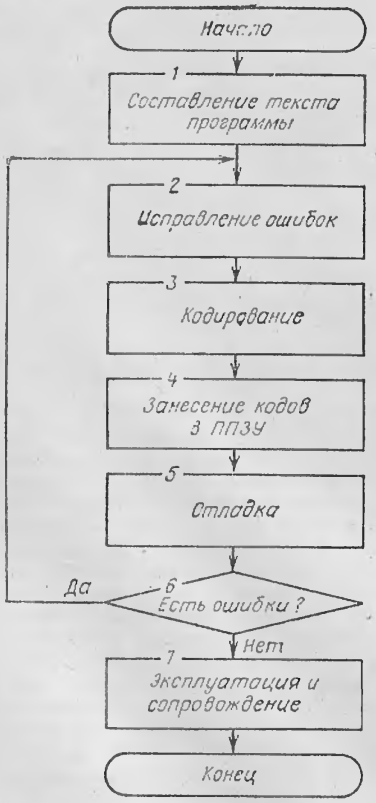


Рис. 1. Алгоритм создания программ пользователя для микроЭВМ «Электроника К1-20»

В базовом варианте микроЭВМ работает с программами Редактор, Ассемблер, Загрузчик, Отладчик, позволяющими автоматизировать этап 3-го алгоритма и существенно ускорить этапы 1 и 2. Использовав для эмуляции ППЗУ пользователя ОЗУ «Электроники МС 0401» (до 64 Кбайт), этап 4-го алгоритма можно исключить из тела цикла, занося в ППЗУ окончательный вариант отлаженной программы. Этапы 5 и 6 проводятся в составе разрабатываемых на базе микроЭВМ «Электроника К1-20» систем.

Для сопряжения микроЭВМ «Электроника К1-20» и «Электроника МС 0401» (рис. 2) подвергнута небольшой доработке плата «Электроники К1-20». Новые элементы на ней — резисторы R1, R2, конденсатор C1, микросхема D1. D-триггер (D73.2) управляет входом готовности RDYIN тактового генератора КР580ГФ24, который перед этим отсоединяется от контакта А31 разъема XC1. На освободившийся контакт подается сигнал ГТ с платы устройства ввода-вывода микроЭВМ «Электроника МС 0401».

Цепочка R1C1 и элементы D1.1 и D1.2 служат для установки триггера в исходное состояние при включении питания. Триггер переводит микропроцессор «Электроники К1-20» в состояние ожидания при обращении к памяти с адресом 5000Н и выше (это начальный адрес ППЗУ пользователя). Состояние ожидания длится, пока «Электроника МС 0401» не сформирует на шине $\overline{МД0}$...$\overline{МД7}$ данные, расположенные по адресу, указанному «Электроникой К1-20» на шине $\overline{МА0}$...$\overline{МА15}$. После этого на триггер D73.2 подается сигнал ГТ, «Электроника К1-20» считывает данные и продолжает выполнение программы.

Программа эмуляции ППЗУ (рис. 3) функционирует на микроЭВМ «Электроника МС 0401», опрашивая разряд PC0 микросхемы интерфейса D7. Активный уровень сигнала $\overline{ВМ}$ указывает на обращение «Электроники К1-20» к ППЗУ пользователя. При этом с портов PA и PB интерфейса D7 считывается адрес ячейки ППЗУ и соответствующие данные выдаются через порт PB интерфейса D8, где они статистически фиксируются. По линии PC4 микросхемы D7 формируется отрицательный импульс, который возвращает триггер D73.2 в единичное состояние.

Программа эмуляции не накладывает никаких ограничений на функциональные возможности микроЭВМ «Электроника К1-20», позволяя отлаживать разрабатываемые программы в пошаговом и трассировочном режимах на реальном макете разрабатываемой системы. Важен только порядок запуска программ: сначала на «Электронике МС 0401» запускается программа эмуляции, за-

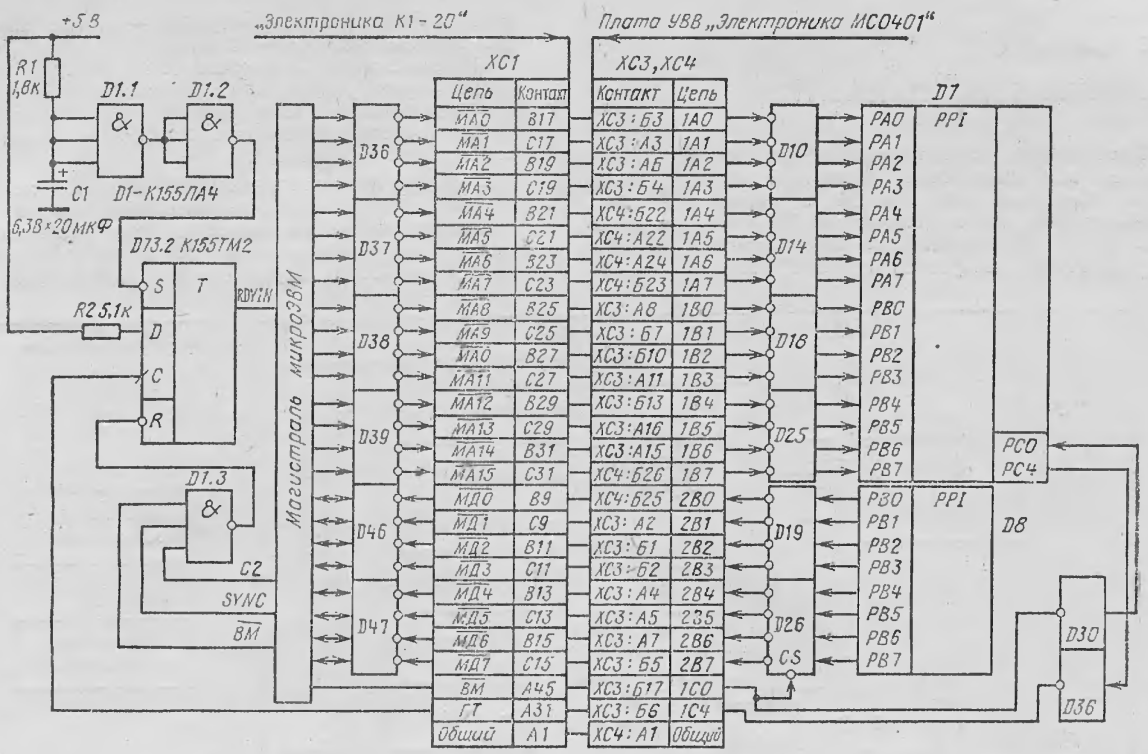


Рис. 2. Схема сопряжения микроЭВМ «Электроника К1-20» и «Электроника МС 0401»

```
;АССЕМБЛЕР "ЭЛЕКТРОНИКА К1-10" ВЕРС. 1.1
; ********************************************
; *  EMUL-ПРОГРАММА ЭМУЛЯЦИИ ППЗУ ПОЛЬЗОВАТЕЛЯ  *
; *  ДЛЯ МИКРОЭВМ "ЭЛЕКТРОНИКА К1-20"          *
; ********************************************
; ИСПОЛЬЗУЕМЫЕ РЕГИСТРЫ: A,H,L
; ПОДПРОГРАММЫ          : НЕТ
        ORG     2000H
;
;
LAD     EQU     14H     ;МЛАДШИЙ БАЙТ АДРЕСА
HAD     EQU     15H     ;СТАРШИЙ БАЙТ АДРЕСА
CON     EQU     16H     ;РЕГИСТР СИГНАЛОВ ОБМЕНА
PORT1   EQU     17H     ;РУС D7
DATA    EQU     19H     ;РЕГИСТР ДАННЫХ
PORT2   EQU     1BH     ;РУС D8
;
;
EMUL:   MVI     A,93H   ;D7: A-ВВОД,B-ВВОД
        OUT     PORT1   ;C(0-3)-ВВОД,C(4-7)-ВЫВОД
        MVI     A,80H   ;D8: B-ВЫВОД
        OUT     PORT2
        MVI     A,0FFH  ;ГТ=0
        OUT     CON
;
INP:    IN      CON     ;ЗАПРОС НА ЭМУЛЯЦИЮ?
        ANI     01H
        JZ      INP     ;НЕТ
        IN      LAD     ;ВВЕСТИ МЛАДШИЙ
        MOV     L,A     ;БАЙТ АДРЕСА
        IN      HAD     ;СТАРШИЙ БАЙТ
        MOV     H,A
        MOV     A,M     ;СЧИТАТЬ ДАННЫЕ
        OUT     DATA    ;ВЫВОД В РЕГИСТР
        XRA     A
        OUT     CON     ;ГТ=1
        MVI     A,10H
        OUT     CON     ;ГТ=0
        JMP     INP
        END
```

Рис. 3. Программа эмуляции ППЗУ

тем с шестнадцатеричного пульта микроЭВМ «Электроника К1-20» начинают отладку программы.

Организация программного обмена данными между микроЭВМ «Электроника К1-20» и «Электроника МС 0401» позволила создать при минимуме схемных и конструктивных изменений аппаратно-программный отладочный комплекс ограниченной мощности. В отличие от других, у него расширены возможности отладки технических средств вычислительных систем на базе микроЭВМ «Электроника К1-20» *

Технические характеристики аппаратно-программного отладочного комплекса

Объем адресуемой памяти, Кбайт	64
Число аппаратных линий прерываний	8
Число параллельных каналов обмена информацией	6
Число последовательных каналов ввода-вывода	1/1
Внешние носители информации	Гибкие магнитные диски, перфолента

Программы пользователя могут быть написаны на языках ассемблер, ПЛ/М, БЕЙСИК, Фортран (при наличии соответствующих трансляторов). МикроЭВМ «Электроника МС 0401», входящая в комплекс, предоставляет широкие возможности по написанию и предварительной отладке программ пользователя. При окончательной отладке на реальной аппаратуре используются сервисные возможности программы Монитор микроЭВМ «Электроника К1-20»: работа в автоматическом и пошаговом режимах, остановы в заданных точках, просмотр и изменение содержимого ячеек памяти и внутренних регистров МП и т. д.

Замедление выполнения программ пользователя не играет роли при отладке, за исключением случаев, когда временные интервалы формируются программно.

При небольшой доработке программы эмуляции комплекса можно использовать для отладки любых МП систем на базе МПК БИС КР580.

Телефон 3-53-53, Ровно, (д.), Конько В. В.

Статья поступила 24 сентября 1987

* Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.— С. 269.

УДК 681.32

А. Н. Кривцов

СИГНАТУРНЫЙ АНАЛИЗАТОР

Предлагаемый сигнатурный анализатор (СА) предназначен для диагностики микропроцессорных (МП) систем. Методика работы с СА, а также требования к МП-системам для использования СА изложены в работах [1, 2]. Быстродействие анализатора ограничено быстродействием микросхем серии К155: минимальная длительность тактовых импульсов — 01 мкс; максимальная тактовая частота — 10 МГц.

Входная логика СА рассчитана на подключение к контролирующим устройствам, допускающим дополнительное подключение в контролируемой точке одного ТТЛ-входа. В местах, где такое подключение недопустимо, необходимо согласование.

В состав анализатора входят (рис. 1): модуль выделения сигнатуры, модуль индикации, модуль питания. В модуле выделения сигнатуры преобразуются циклические последовательности (ограниченные сигналами START и STOP), вводимые от диагностируемых уст-

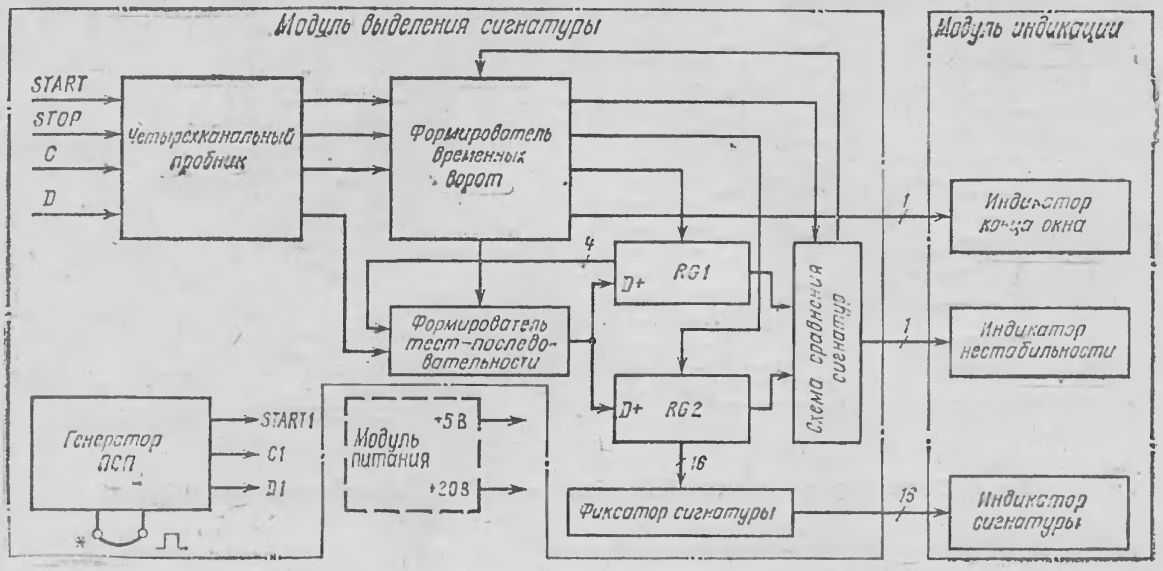


Рис. 1. Структурная схема сигнатурного анализатора

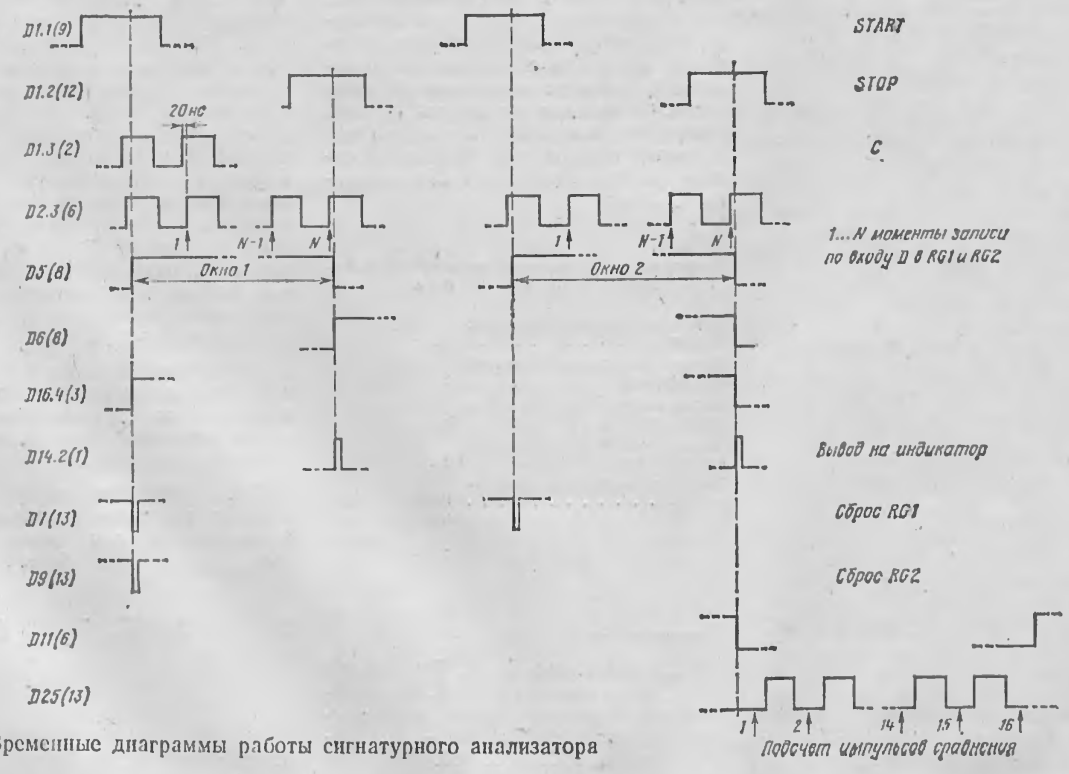


Рис. 2. Временные диаграммы работы сигнатурного анализатора

ройств. Результат — 16-разрядный код. Модуль индикации преобразует этот код в четырехзначную сигнатуру.

В состав модуля выделения сигнатуры входят (рис. 1, 3): 4-канальный пробник (D1, D2) с гистерезисной характеристикой, формирователь временны́х ворот-окон (D4, D5, D6, D16), два 16-разрядных регистра сдвига — RG1 (D7, D8) и RG2 (D9, D10), схема сравнения сигнатур (D17.1, D13.3, D17.2, D11, D13.4), фиксатор двоичного кода сигнатуры (D18, D19).

Через пробник D1, D2 принимаются сигналы от испытуемого устройства: STOP, START, тактовые сигналы C, информационные сигналы D с контролируемых точек. С помощью переключателей S2...S4 задаются активные уровни сигналов START, STOP, C, D, при которых запускается анализатор. В каждом конкретном случае выбирается свой оптимальный режим записи сигнатур, позволяющий получить максимально полную и достоверную информацию от тестируемого устройства.

При приходе сигнала START по фронту тактового сигнала C задается фронт окна (рис. 2), по фронту сигнала STOP фронтом C — срез окна. Формирователь временны́х ворот разрешает запись информации (через входы D+), поступающей с формирователя тест-последовательности в RG1 и RG2 (во время первого окна — в оба регистра, а во время второго — только в RG1). В это время в RG2 хранится код, полученный в первом окне. По срезу окна 2 запускается схема сравнения кодов сигнатур, полученных в первом и во втором окнах. При этом на входы D+ RG1 и RG2 подается «Лог. 0». Содержимое регистров последовательно выталкивается для сравнения микросхемой D17.1. Как только D11 насчитает 16 тактовых импульсов, сравнение заканчивается: модуль выделения сигнатуры готов к приему следующей последовательности. При несовпадении содержимого RG1 и RG2 запускается одновибратор D12.2, управляющий индикатором нестабильности. По срезу каждого окна выдается информация из RG1 в фиксатор кода сигнатуры и запускается индикатор конца окна. Моменты занесения битов данных в регистры сдвинуты от фронтов C на время τ, позволяющее исключить запись информации в анализатор во время переходных процессов на шинах проверяемых устройств.

Режим работы анализатора задается и с помощью переключателей S5 и S6. При данном положении переключателя S5 (рис. 3) фронт и срез окон формируется фронтами START. Этот режим используется, когда нет возможности получить сигнал STOP или в нем нет необходимости. При противоположном положении S5 фронт окна задается сигналом START, а срез окна — сигналом STOP. Переключателем S6 выбирают однократный или многократный запуск анализатора.

Для диагностики СА в модуль выделения сигнатуры включен генератор псевдослучайной последовательности (ГПСП), формирующий сигналы для тестирования — START1, C1, D1. Длина контрольной последовательности — 216 бит. Частота C1 — около 200 кГц. Вход ✳ используется тогда, когда необходимо оценить работу СА от ГПСП, тактируемого рабочей частотой проверяе-

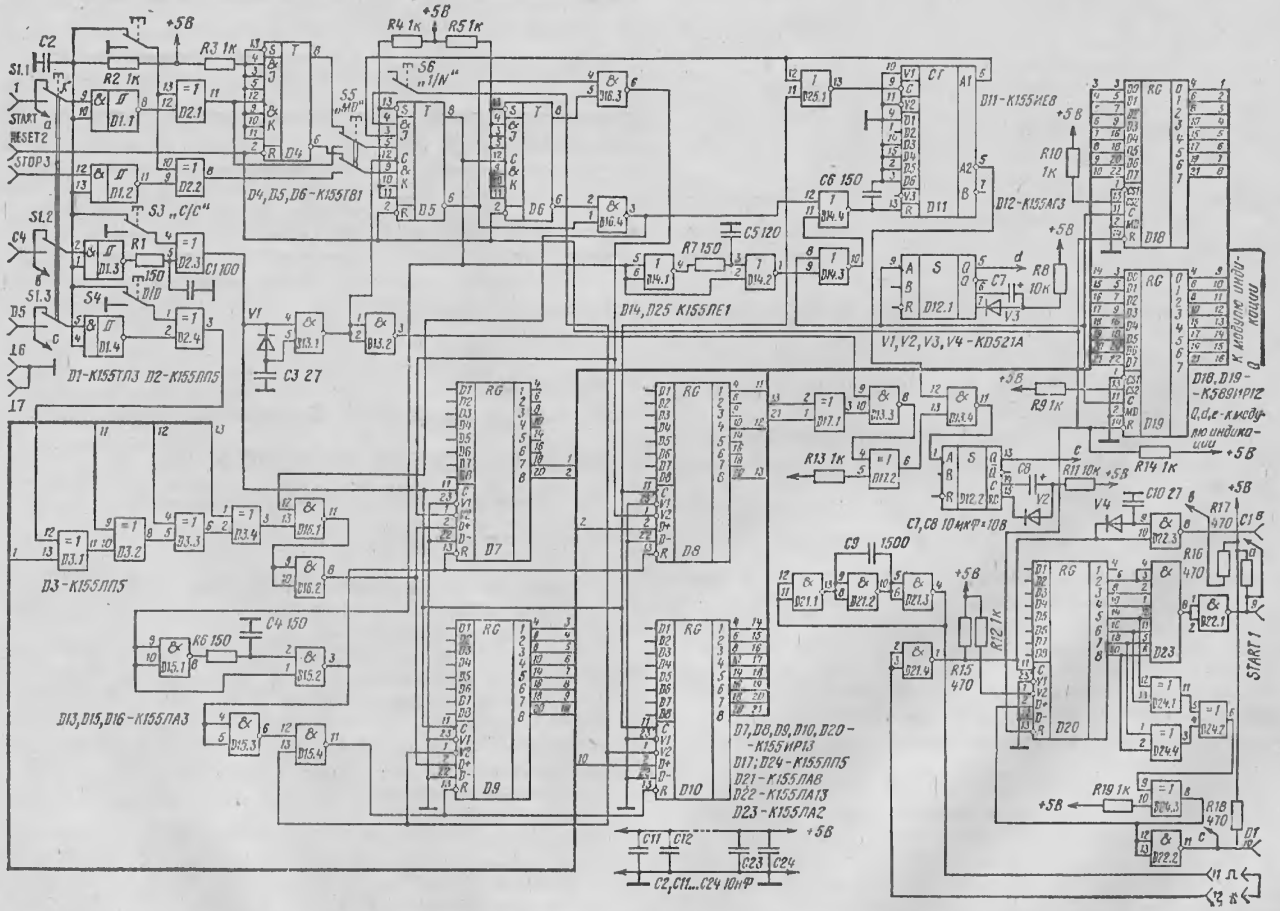


Рис. 3. Принципиальная схема модуля выделения сигнатуры и ГПСП

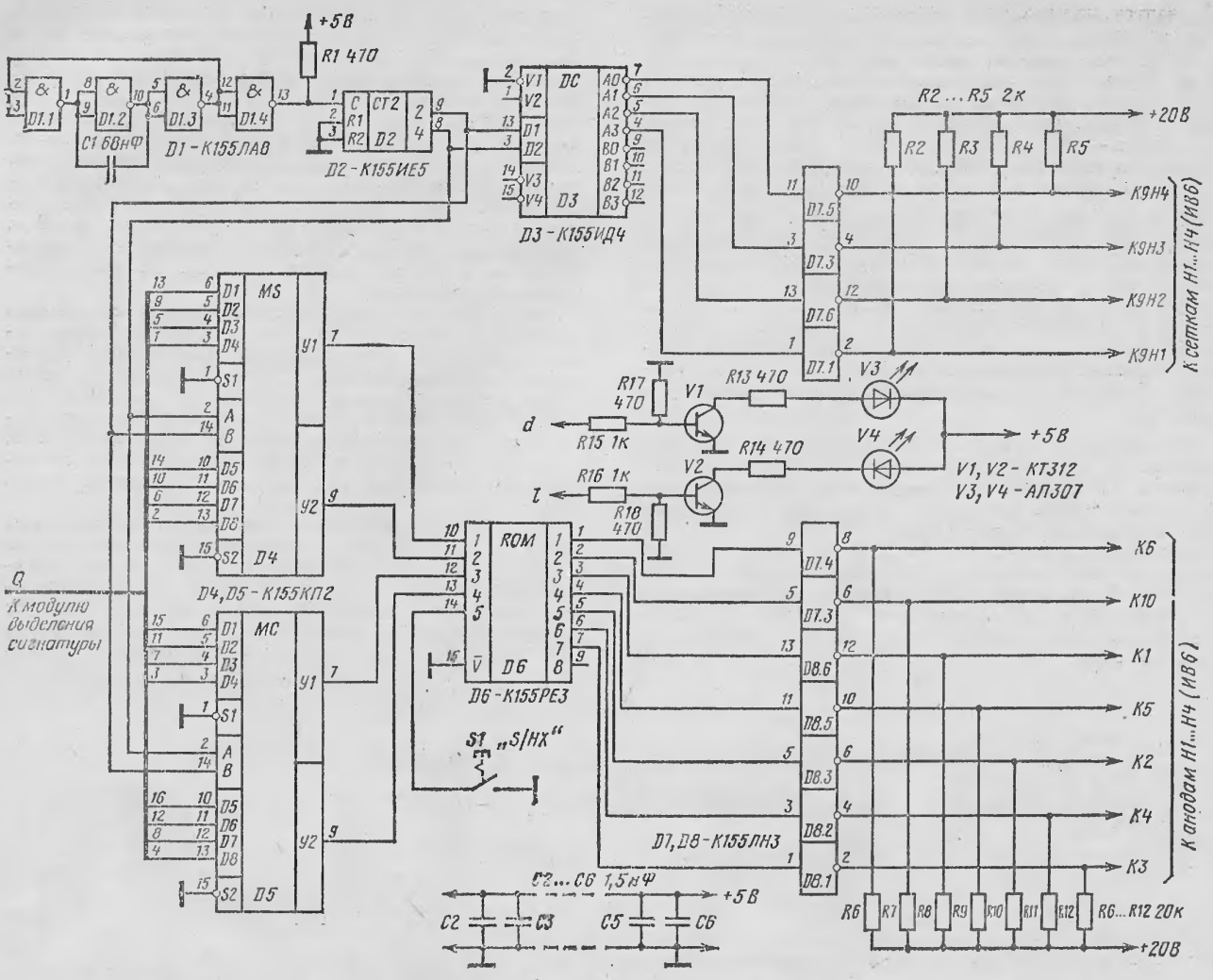


Рис. 4. Принципиальная схема модуля индикации

мого устройства. Перемычка между ✳ и ⎍ в этом случае убирается.

Модуль индикации (рис. 4) позволяет индицировать как обычный шестнадцатеричный, так и видоизмененный код:

0 1 2 3 4 5 6 7 8 9 A B C D E F
0 1 2 3 4 5 6 7 8 9 A C F H P U

Вид индикации выбирается с помощью S1. В качестве индикаторов сигнатуры используются ИВ6.

Модуль питания обеспечивает +5 В (2 А), +20 В (200 мА). Потребление тока от источника +5 В — 1,3 А.

Сигнатурный анализатор использовался при поиске неисправностей в микропроцессоре Z80

Телефон 74-40-59, Караганда, Капинос А. Н.

ЛИТЕРАТУРА

1. Мирский Г. Я. Микропроцессоры в измерительных приборах.— М.: Радио и связь, 1984.
2. Микропроцессоры. Системы программирования и отладки / Под ред. В. А. Мясникова, М. Б. Игнатьева.— М.: Энергоатомиздат, 1985.

Статья поступила 14 октября 1987

УДК 681.326

М. А. Ордынский, И. Г. Багреева, Ю. З. Нехамкин

РЕЗИДЕНТНЫЙ МОНИТОР ДЛЯ ОТЛАДКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Один из путей совершенствования систем ЧПУ — создание систем с элементами сетевой архитектуры, мультипроцессорных УЧПУ [1, 2].

Предлагаемый монитор разработан для УЧПУ типа ЗС 150 и предназначен для отладки программного обеспечения контроллеров на базе МП КР580ИК80А. Работа монитора поддерживается прямым доступом в память МП и программным доступом к управлению тремя сигналами RESET, HOLD и INT.

Фукции монитора:

выбор контроллера;

загрузка программы с устройства ввода перфоленты (или из специально организованного в ОЗУ микроЭВМ буфера) в выбранный контроллер; произвольного фрагмента программы из ОЗУ МП в буфер ОЗУ микроЭВМ;

работа с регистрами МП, с ячейками ОЗУ МП или буфера ОЗУ микроЭВМ;

запуск программы с заданного начального адреса с остановом по этому адресу;

принудительный останов программы и ее выполнение по шагам.

В состав монитора входят две асинхронно работающие программы: MOD1 (4 Кбайта) размещена в памяти микроЭВМ и загружает MOD2 (256 байт) в ОЗУ выбранного контроллера и обращается к подпрограммам, реализующим функции монитора. С помощью программы MOD2 содержимое регистров, указателя стека, счетчика команд и регистра флагов пересылается в фиксированную область ОЗУ МП и восстанавливается непосредственно перед запуском обслуживаемой программы или отдельной команды в пошаговом режиме. Взаимодействие программ организовано с помощью двух семафоров.

Оригинальный способ реализации пошагового режима позволил решить проблему переменного формата команд (за исключением группы команд обращения к подпрограммам) без их интерпретации.

Изложенный подход допускает обобщение на системы с числом иерархических уровней больше двух.

Телефон 130-67-96, Ленинград

ЛИТЕРАТУРА

1. Флейтер Е. Г. Организация межпроцессорного обмена в УЧПУ с подчиненными контроллерами // Микропроцессорные средства и системы.— 1987.— № 2.— С. 43—48.
2. Найденов Л. В., Туманов Л. А., Романенков В. А. Пошаговый режим при разработке и отладке программ для микропроцессорных средств на базе БИС КР580ИК80А // Микропроцессорные средства и системы.— 1984.— № 3.— С. 85—86.

Статья поступила 15 июня 1987

УДК 681.3.06

В. Б. Бродин, И. И. Шагурин

МИКРОПРОГРАММИРУЕМЫЙ СХЕМНЫЙ ЭМУЛЯТОР ДЛЯ ОТЛАДКИ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Основное средство комплексной отладки микропроцессорных (МП) систем, их программного обеспечения в реальном масштабе времени — схемные эмуляторы. Традиционно их построение на основе БИС той же серии, на которой строится отлаживаемая система [1, 2]. Однако при таком подходе не обеспечивается универсальность схемного эмулятора. Кроме того, при эмуляции микропроцессора серии К1810, однокристальных микроЭВМ (ОМЭВМ) типа К1816ВЕ51 и др. данный подход встречает значительные трудности. В высокопроизводительных МП этого класса реализован конвейерный принцип выборки и выполнения команд, организована очередь команд, поэтому состояние МП определяется не текущей выполняемой командой, а сегментом программы и идентифицировать его весьма трудно. Схемный эмулятор оказывается сложным в разработке, отладке и контроле при эксплуатации. Трудности эмуляции еще более возрастают для ОМЭВМ, у которых на кристалле, кроме АЛУ, интегрирована память программ и данных, блок обработки прерываний, жестко синхронизируемые блоки последовательного порта и счетчиков-таймеров [3]. Выводы микросхемы имеют альтернативные функции, режим останова отсутствует, выборка из внутренней памяти программы ОМЭВМ совмещена с обменом через порты ввода-вывода.

При традиционном подходе к эмуляции прерывания высшего приоритета используются для организации режима останова и выполнения подпрограмм эмуляционного обмена, счетчики-таймеры и последовательный порт изменяют свое состояние в ходе выполнения этих подпрограмм, половина портов ввода-вывода теряют альтернативные функции и служат только для адресации внешней памяти программ.

В целом, возможности эмуляции при этом ограничены и не могут удовлетворить пользователей, максимально реализующих ресурсы БИС микропроцессоров и ОМЭВМ.

Построение схемного эмулятора на основе микропрограммируемого имитирующего процессора — один из методов преодоления указанных трудностей. Разработать такой имитатор достаточно сложно, однако он дает возможность создать модульную, иерархическую структуру, адаптируемую для целого класса МП и ОМЭВМ, контролепригодную на этапах производства и эксплуатации. Значительные аппаратные затраты на реализацию имитирующего процессора компенсируются простотой создания собственно эмулятора на его основе (причем с высокими эксплуатационными характеристиками).

Разработанный на кафедре микроэлектроники МИФИ схемный эмулятор отладочного комплекса «КОМПАС» основан на имитационном подходе и состоит из двух основных блоков — имитирующего процессора и блока адаптации и эмуляции (рис. 1). Основная функция первого блока — физическое моделирование прототипной БИС МП или ОМЭВМ, второй блок обеспечивает настройку имитатора на функции прототипа, осуществляет управление в режимах эмуляции и диагностики.

Имитирующий процессор должен выполнять операции в соответствии с системой команд и логико-временными диаграммами прототипа в реальном масштабе времени.

Имитатор состоит из набора функциональных модулей (на которых моделируется прототипная БИС) и модулей микропрограммного управления. Модульная структура обеспечивает модифицируемость, наращиваемость и контролепригодность эмулятора на стадиях производства и эксплуатации.

Анализ структурных решений БИС 8-разрядных МП и ОМЭВМ показал, что **микропрограммируемый имитирующий процессор ценой небольших дополнительных затрат может быть выполнен универсальным, настраиваемым на функции МП К580, К1821, ОМЭВМ К1816ВЕ48, К1816ВЕ51.** Исходя из этого он включает в себя 8-разрядное АЛУ с аппаратной реализацией операций типа умножения, ОЗУ данных (256 байт), модуль счетчиков-таймеров, модули: обработки прерываний, последовательного порта, портов ввода-вывода общего назначения, служебных регистров.

Имитатор должен иметь избыток быстродействия схем управления, чтобы использовать его при настройке на временные диаграммы прототипа. Поэтому микропрограммное устройство управления реализовано в виде двух модулей MPCU1 и MPCU2, работающих параллельно. Каждый из них реализует конвейерный метод выборки микрокоманд. MPCU1— ведущий, управляет общей последовательностью выборки и выполнения команд прототипной БИС. MPCU2 — ведомый, управляет обменом через порты ввода-вывода и счетчиком команд. Оба модуля управления имеют схему формирования следующего адреса, память микрокоманд и регистр микрокоманд MPCU1 реализован на основе микросхемы К1804ВУ4, имеет память микрокоманд 1К×80 разрядов (формат микрокоманды приведен на рис. 2,*а*). Он содержит также схемы инициализации и управления режимами эмуляции. MPCU2 осуществляет только последовательную выборку с помощью счетчика микрокоманд, имеет память микрокоманд 256×12 разрядов (формат микрокоманды приведен на рис. 2, *б*). Стартовый адрес микроподпрограммы сообщает ему модуль MPCU1. На функции конк-

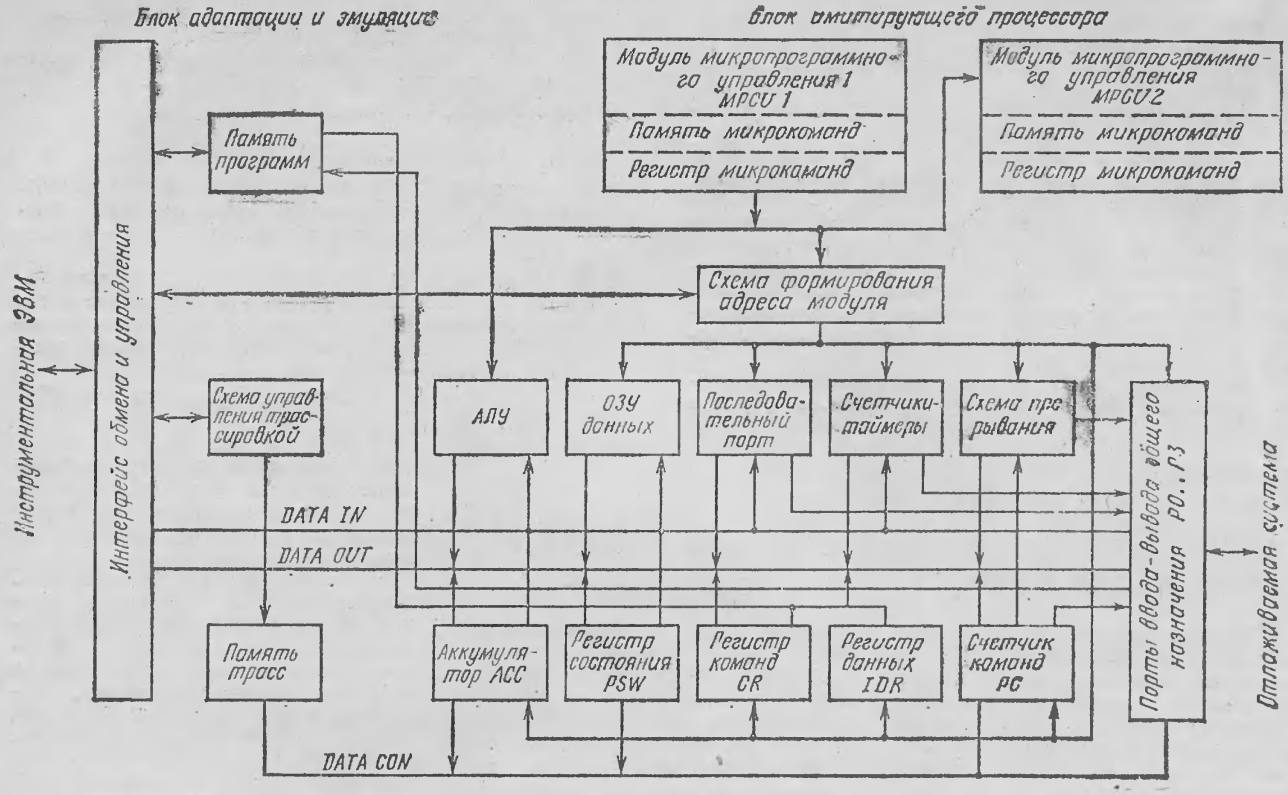
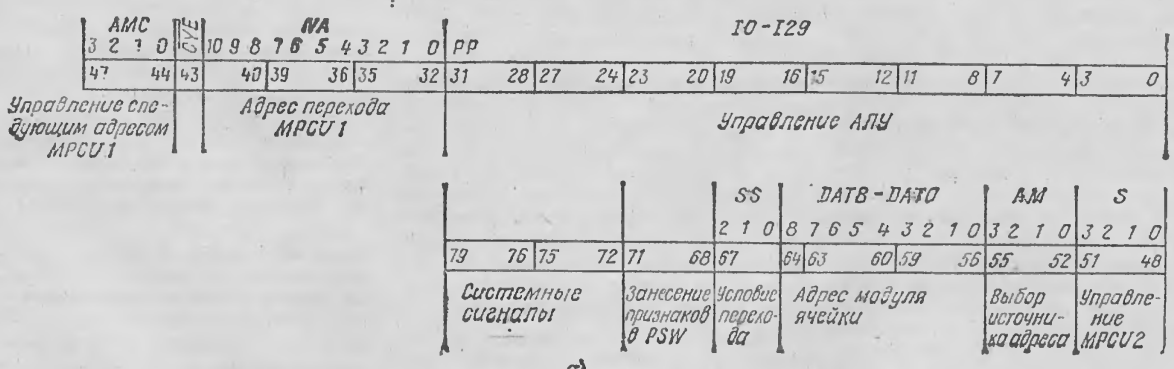


Рис. 1. Структура схемного эмулятора с микропрограммным управлением



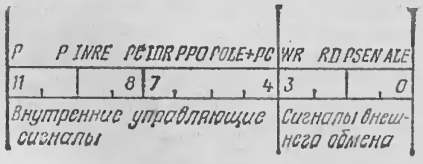


Рис. 2. Формат микрокоманды модуля управления MPCU1 (*а*) и модуля MPCU2 (*б*)

ретной микропроцессорной БИС имитатор настраивается занесением в память микрокоманд соответствующей микропрограммы-интерпретатора с языка команд прототипа. В качестве примера приведена модификация схемного эмулятора, настроенная на функции ОМЭВМ К1816ВЕ51. Специализация эмулятора на данный тип ОМЭВМ реализована определением символических имен регистров и распределением их адресов (рис. 3), а также назначением соответствующих функций внешних выводов (рис. 4).

MPCU1 и MPCU2 управляют работой функциональных модулей имитирующего процессора, большинство которых выбираются посредством схемы формирования адреса модуля. Адрес формируется исходя из кода команды прототипа и поля адреса текущей микрокоманды. Основная часть программно-доступных регистров прототипа моделируются ячейками ОЗУ данных, причем, есть доступ к целому байту и к отдельным битам. В виде отдельных модулей служебных регистров реализованы программно-недоступные регистры прототипа (CR — регистр команд, IDR — регистр байта данных команды) и часть программно-доступных, но выбираемых в системе команд одновременно с остальными (ACC — аккумулятор, PSW — регистр состояния, PCH, PCL — старший и младший байты счетчика команды PC и т. д.). Адресное пространство ОЗУ данных и всех функциональных модулей общее.

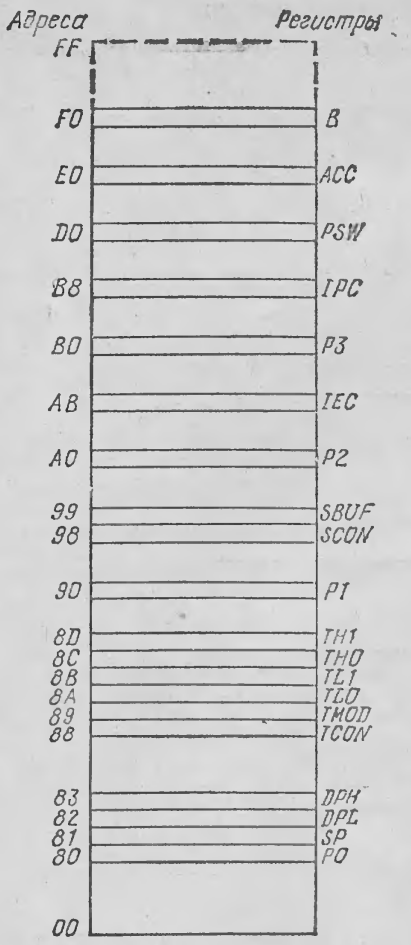


Рис. 3. Распределение адресного пространства при эмуляции ОМЭВМ К1816ВЕ51

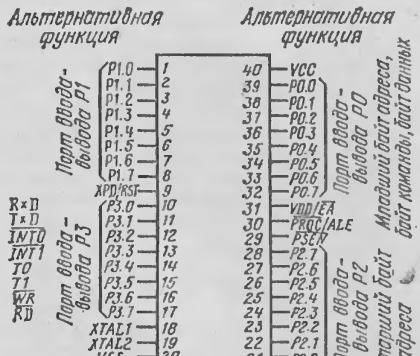


Рис. 4. Функции выводов общего назначения при эмуляции ОМЭВМ К1816ВЕ51

Обмен между функциональными модулями имитатора и блоком адаптации и эмуляции осуществляется по шинам DATA IN и DATA OUT.

Третья шина DATA CON служит для доступа к служебным регистрам в реальном времени и занесения их состояния в память трасс эмулятора. В целом такая структура позволяет совмещать внутренний и внешний обмены с одновременным контролем состояния при сложных логико-временных диаграммах.

Из функциональных модулей наиболее сложны АЛУ, последовательный порт, счетчики-таймеры, схема обработки прерываний и модуль портов ввода-вывода общего назначения.

Модуль АЛУ (рис. 5) построен на основе двух микросхем К1804ВС1 и микросхем умножителя К1802ВР2, управляется отдельным полем микрокоманды MPCU1 (разряды I0...I31). Особенности АЛУ — выполнение операции типа умножения отдельной микросхемой умножителя, управляемой разрядами I20...I23 микрокоманды, и операций сдвигов и перестановок над содержимым обрабатываемого байта посредством мультиплексоров, управляемых разрядами I11...I12. Микрокоманды адреса (16 разрядов) обрабатываются в два цикла. В целом АЛУ позволяет выполнять все операции систем команд прототипов.

Модуль последовательного порта обеспечивает дуплексную передачу информации через выводы RxD, TxD портов общего назначения. Имеются четыре режима работы: режим 0 — расширенный синхронный ввод-вывод с помощью регистров сдвига, режим 1 — асинхронный приемопередатчик с 10-разрядным кадром и переменной скоростью передачи: режим 2 — асинхронный приемопередатчик с 11-разрядным кадром и фиксированной скоростью передачи; режим 3 — асинхронный приемопередатчик с 11-разрядным кадром и переменной скоростью передачи. Данные хранятся в буферном регистре SBUF, а управляющее слово — в регистре SCON (рис. 3).

Модуль таймеров-счетчиков содержит два 16-разрядных таймера-счетчика, которые управляются регистрами TCON и TMOD. У каждого из них четыре режима: режим 0—8-разрядный таймер-счетчик с делением на 32; режим 1—16-разрядный таймер-счетчик; режим 2 — автоматически перезагружаемый 8-разрядный таймер-счетчик; режим 3 — два таймера и счетчик.

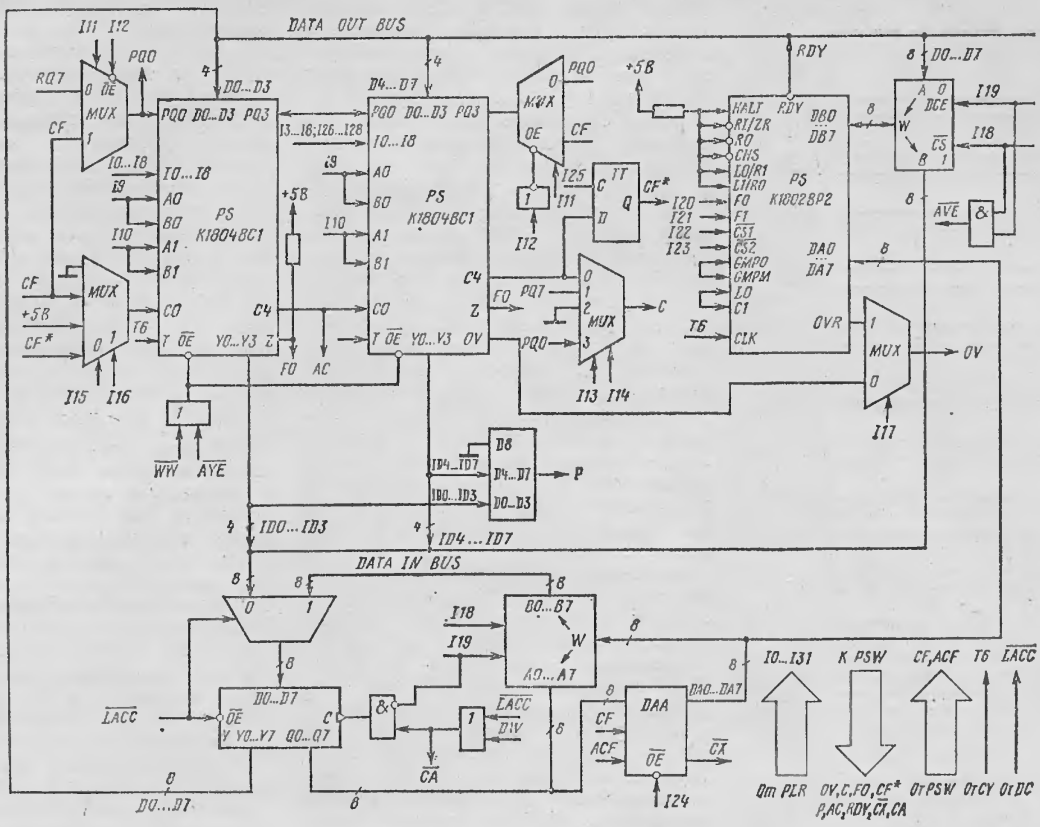
Модуль обработки прерываний способен воспринимать запросы от трех внутренних и двух внешних источников. Внешние запросы поступают на входы портов общего назначения. Пользователь может программно назначать источникам прерываний приоритеты, разрешать и запрещать отдельные запросы. Регистры IPC, IEC — управляющие.

В виде модуля портов ввода-вывода общего назначения выполнены четыре параллельных порта. Это означает, что их роль может изменяться в зависимости от типа имитируемой БИС и после настройки функции отдельных выводов — в зависимости от режима работы. В частности, в ОМЭВМ К1816ВЕ51 (рис. 4) при работе с внутренней памятью программ (4 Кбайт) все порты P0...P3 могут использоваться для обмена данными, а при обращении к внешней памяти программ (до 64 Кбайт) через порты P0 и P2 выводится адрес. Порт P3 также имеет альтернативные функции, связанные с работой последовательного порта (RxD, TxD), обработкой прерываний ($\overline{INT0}$, $\overline{INT1}$), работой счетчиков-таймеров (T0, T1), выдачей сигналов записи и чтения ($\overline{WR}$, $\overline{RD}$). Все порты имитатора имеют поразрядное управление для работы на ввод или вывод с возможностью установки в третье состояние. Конкретная функция каждого порта реализуется подключением соответствующего функционального модуля к шинам DATA IN, DATA OUT в качестве приемника или источника информации. Если через порты P0 и P2 выдается адрес, то он поступает непосредственно из счетчика команд PC.

Блок адаптации и эмуляции осуществляет связь с инструментальной микроЭВМ типа «Электроника 60» и позволяет реализовать настройку на функции прототипной БИС; инициализацию имитирующего процессора; загрузку и редактирование рабочей программы; эмуляцию отлаживаемой системы; контроль и диагностику эмулятора. Структура эмулятора для реализации этих этапов адаптируется управляющей программой, с которой пользователь работает в режиме меню и диалога. Блок содержит модули интерфейса обмена и управления; эмуляционной памяти программ; управления трассировкой; памяти трасс. Модулям этого блока соответствуют адреса памяти инструментальной ЭВМ.

Модуль интерфейса позволяет адаптировать структуру эмулятора в соответствии с этапом работы, режимом и уровнем обмена. Адаптация — процесс создания из модулей эмулятора определенной совокупности аппаратных и программных средств, реализующих указанные этапы работы эмулятора и позволяющих пользователю получить информацию требуемого объема и глубины.

Каждый этап может быть выполнен в различных режимах. В частности, эмуляция возможна в пошаговом режиме, режиме с остановами в контрольных точках и режиме прогона рабочей программы прототипной БИС. На каждом этапе информацию можно получать и интерпретировать на уровне отдельных микрокоманд (в том числе их полей),


DATA OUT BUS
DATA IN BUS
MUX
PS K1804BC1
PS K1804BC1
MUX
MUX
MUX
K1002BP2
DAA
DGE
MUX
&
&
1
1
1

случае это не существенно. Принципиально то, что при использовании микропрограммируемого имитирующего процессора появляется возможность его полной остановки, включая счетчик-таймер и последовательный порт. При традиционном методе эмуляции эти блоки меняют свое состояние при выполнении подпрограмм чтения состояния процессора, поскольку на них продолжают поступать тактовые сигналы. Важно и то, что благодаря имитирующему процессору пользователь может применять прерывания высшего уровня (при традиционном подходе занять под служебные функции эмулятора).

Этап контроля собственно эмулятора также включает в себя несколько режимов для доступа к модулям и тестирования каждого из них. Управление ведется при этом на уровне микрокоманд и макрокоманд тестирующей программы эмулятора. Предусмотрено сравнение с эталонным образцом прототипной БИС.

Схемный эмулятор позволяет отладить систему пользователя в реальном времени с точностью до такта синхросигнала прототипной БИС. Функционирование эмулятора, т. е. соответствие системы команд и диаграмм обмена, проверено с помощью функциональных тестов, а также при совместной работе с макетом отлаживаемой МП-системы.

Элементная база схемного эмулятора — микросхемы серий К1804, К531, К556, К541. Конструктивно он выполнен в виде четырех плат (около 250 микросхем) в крейте ЭВМ «Электроника 60». Общая потребляемая мощность — 80 ВА. Программное обеспечение — около 1,0 тыс. операторов языка Паскаль.

Телефон 324-91-55, Москва

ЛИТЕРАТУРА

1. 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.

2. Иванов В. И., Лобанов В. И., Митрофанов А. В. Отладочные средства для малоразрядных однокристальных микроЭВМ//Микропроцессорные средства и системы.—1984.—№ 2.— С. 42—45.

3. Мун Дж. Микрокомпьютер для эмуляции // Электроника.— 1980.— Т. 53.— № 16.— С. 48—53.

Статья поступила 15 июня 1987

УДК 681.32

Л. Г. Ерухимов, Р. Р. Вахитов

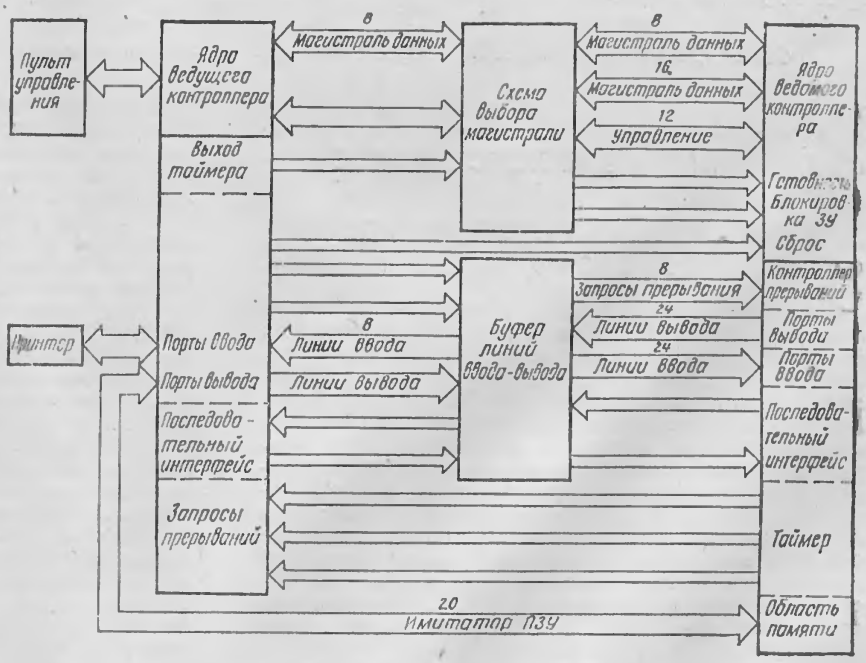
СТЕНД «МИКРОТЕСТ» ДЛЯ ДИАГНОСТИКИ ОДНОПЛАТНОГО МИКРОКОНТРОЛЛЕРА «ЭЛЕКТРОНИКА МС 2702»

Основные неисправности, возникающие в процессе работы одноплатных микроконтроллеров «Электроника МС 2702»,— повреждение металлизированных покрытий, нарушение контактов в местах пайки и выход из строя шинных формирователей от перегрева.

В связи с массовой эксплуатацией контроллеров остро необходима простая и удобная в обращении система диагностики. За ее основу взят идентичный контроллер «Электроника МС 2702» (назовем его ведущим). Из структурной схемы стенда (см. рисунок) виден алгоритм диагностики исследуемого контроллера (ведомого). Диагностирование заключается в анализе состояния всех линий, имеющих физический выход на разъемы. Реализация осуществляется программно-аппаратными средствами, схемой выбора магистрали и подачей специального разрешающего сигнала на вход «Готовность» ведомого контроллера. Этот сигнал формируется таймером 1 ведущего контроллера в режиме одиночного аппаратно-формируемого строба. Длительность сигнала достаточна для обработки ведомым контроллером одного байта команды, код которой задается ведущим. Таким образом, анализируются все состояния контроллера: Сброс (исходное состояние), Ожидание, Чтение ЗУ, Запись ЗУ, Чтение ВВ, Запись ВЫВ., Останов, Прерывание. Затем, сравнивая состояния линий управления адресов и данных с истинными, выявляют неисправные.

Схема выбора контролируемой магистрали состоит из двух БИС параллельных интерфейсов КР580ВВ55А, работающих в обычном программируемом режиме. Буфер линий ввода-вывода — это также два параллельных интерфейса КР580ВВ55А (один из них работает только в режиме чтения, другой — только в режиме записи, что позволило эффективно опросить все параллельные каналы ввода-вывода); согласующий буфер оптронов К155ЛИ1; буфер последовательной передачи К155ЛН1.

Перед началом проверки неисправная плата вставляется в разъем стенда. Из панелек ППЗУ извлекаются все имеющиеся в нем микросхемы памяти. В гнездо с абсолютным адресом 0000 вставляется вилка съемного имитатора ППЗУ, контролирующего состояние шин адреса, выбор кри-



Структурная схема стенда диагностики

сталла ППЗУ и имитирующего программу. После включения питания или нажатия кнопки «Сброс» наличие неисправности распечатывается на бумаге. По инструкции пользователя определяется методика устранения неисправности. После ее устранения или при отсутствии таковой имитатор ППЗУ поочередно вставляется в оставшиеся панели ППЗУ с вызовом соответствующей программы проверки через пульт управления. После всех указанных процедур и устранения обнаруженных неполадок одноплатный микроконтроллер можно признать годным к эксплуатации (сигнализируется соответствующим сообщением).

От редакции:

Неисправности, с которыми столкнулись авторы статьи, очень типичны. Возникает вопрос к Министерству, разработавшему устройство: что сделано для того, чтобы повысить качество контроллеров и чтобы Ваша продукция не заставляла людей отвлекать силы на разработки подобного рода?

702100, Чирчик Ташкентской обл., ул. Осипенко, д. 3, кв. 32, Ерухимову Л. Г. Тел. сл. 5-12-69

Статья поступила 18 мая 1987

УДК 681.3.06

В. Л. Динес, В. А. Мартюхин

ДИАЛОГОВЫЙ ОТЛАДЧИК ДЛЯ МИКРОПРОЦЕССОРА КР580ИК80

Отладчик объемом 3 Кбайт, реализованный на микропроцессорной системе КТС ЛИУС-2, позволяет выполнять трассировку отлаживаемой программы, дизассемблирование (ретрансляцию) программы и вспомогательные операции.

Обращение к отладчику происходит по специальной команде, содержащей адрес, с которого начинается отслеживание программы. Во время останова на дисплей выдается кадр, включающий (см. рисунок) адрес останова, код и мнемонику команды; следующий исполняемый адрес; содержимое регистров (аккумулятор и регистр флагов дополнительно представлены в двоичном формате); адреса, коды и мнемонику нескольких следующих исполняемых команд; текущее время таймера.

Диалоговая трассировка отлаживаемой программы допускает покомандное выполнение, повтор выдачи кадра, обычное выполнение группы команд с остановом по указанному адресу, запуск программы в обычном режиме, перезапуск трассировки с другого адреса. Наряду с этими выполняются такие вспомогательные операции, как ввод-вывод для указанного порта, чтение области памяти, модификация байта и регистров. Дисассемблер обеспечивает вывод команд в символьном представлении (совместно с адресами и кодами команд на консоль) и ретранслированной программы на перфоленту в символьном виде с абсолютной адресацией (без адресов и кодов).

```
*ТРАССИРОВКА* TIME- 11:26.14

0063 21 9301  LXI H,00193H

NEXT ADRES=0066
                        SZ0A 0P1C
A=00=0000 0000  F=0000 0010
B=00  C=00
D=00  E=00
H=00  L=00
SP=7E00


0066 CD F80F  CALL 00FF0H
0069 21 0000  LXI H,00000H
006C 39       DAD  SP
006D 22 D302  SHLD 002D3H
0070 3E FF    MVI A,0FFH

WHAT (ПС,ВК,E,G,I,O,X,D,S,V,K)?
```

Заполнение кадра дисплея во время останова монитора

ПС — клавиша ПС (перевод строки) — выполнение команды и переход на следующую с выдачей кадра; ВК — клавиша ВК (возврат каретки) — повтор выдачи кадра; E — выход монитора; G — запуск программы; I — ввод из указанного порта; O — вывод в указанный порт; X — распечатка или модификация регистров; D — распечатка области памяти; S — чтение или модификация байтов памяти; V — трассировка с указанного адреса; K — дисассемблирование части программы

Отладчик содержит также вектор команд переходов на стандартные программы-драйверы внешних устройств, типы и конфигурация которых определяются монитором.

За справками обращаться в ВЦ Рубежанского филиала Днепропетровского химико-технологического института. Тел.: 5-21-71, 7-30-12.

Сообщение поступило 24 июля 1987

УДК 681.32

Л. Л. Муренко

ОТЛАДОЧНЫЙ КОМПЛЕКС НА БАЗЕ ПЭВМ ТИПА ДВК ДЛЯ ОДНОКРИСТАЛЬНЫХ МИКРОЭВМ СЕРИЙ КМ1814, КМ1820, КМ1816

Однокристальные микроЭВМ (ОМЭВМ)—основа массовых встроенных микропроцессорных (МП) устройств нижнего уровня гибких автоматизированных производств, управления различным оборудованием и массовой бытовой аппаратурой. Все больше разработчиков МП-устройств используют массовые ОМЭВМ. Для инструментальной поддержки разработок таких устройств требуются отладочные средства, обладающие широкими функциональными возможностями, высокой производительностью и высоким техническим уровнем. При проектировании и отладке МП-устройств на базе ОМЭВМ используются следующие инструментальные средства: автономные отладочные устройства или автономные схемные эмуляторы [1, 2]; кросс-системы программирования ОМЭВМ [3, 4]; отладочные комплексы [5].

Автономные отладочные устройства — это самые распространенные отладочные средства МП-устройств на базе малоразрядных ОМЭВМ. Они обычно содержат микроЭВМ, память, эмулятор, клавиатуру, подобную микрокалькуляторной, и дисплей однострочный (8...20 символов). Отладочные устройства позволяют вводить программу в машинных кодах с клавиатуры или от внешней ЭВМ и комплексно отлаживать аппаратную и программную части МП-устройства на реальной аппаратуре его прототипа. Отладочные устройства недороги, в автономном режиме используются для проектирования и отладки простейших МП-устройств. Основной недостаток — отсутствие средств для автоматизации программирования МП-устройств, т. е. они не оказывают разработчику никакой помощи в создании программных средств. Несмотря на функциональные ограничения в программировании (малую производительность) автономных отладочных устройств простота изготовления и низкая стоимость определяют их широкое применение.

Кросс-системы представляют собой комплексы программных средств. В составе базовой ЭВМ они формируют исходные тексты и преобразуют их в машинные коды загрузочного модуля. Типовая кросс-система содержит в себе редактор текста, кроссассемблер, кроссредактор связей, программно-логическую модель целевой ОМЭВМ. Кросс-системы — эффективное средство автоматизации программирования МП-устройств.

Недостаток программных кросс-средств — нельзя выполнить интеграцию аппаратных и программных средств и их комплексную отладку на реальной аппаратуре прототипа МП-устройств. В результате единый циклический процесс проектирования и отладки МП-устройства разрывается (повышаются затраты времени на работу в целом).

Отладочные комплексы (ЭВМ, кросс-система программирования МП-устройств и схемные эмуляторы) обеспечивают выполнение всех необходимых этапов проектирования и отладки МП-устройств. Аппаратно-программные средства отладочных комплексов имеют большие функциональные возможности и высокие характеристики. Недостатки отладочных комплексов — повышенная стоимость и ограниченность применения для массовых малоразрядных ОМЭВМ.

Многопользовательский отладочный комплекс с распределенными функциями комплексной отладки (рис. 1) объединяет преимущества указанных инструментальных средств при оптимальной стоимости.

Многопользовательский отладочный комплекс обеспечивает аппаратно-программную поддержку проектирования МП-устройств на базе ОМЭВМ КМ1816ВЕ48, КМ1820ВЕ2, КМ1814ВЕ1.

В качестве основной ЭВМ в отладочном комплексе используются ДВК, «Электроника НЦ 80-20/2 и /3». Распределенные рабочие станции представляют собой автономные отладочные устройства (автономные схемные эмуляторы — АСЭ) [1, 2, 5]. Одно рабочее место — модульный программатор.

Устройство параллельно-последовательного обмена (УППО) данными между ПЭВМ и АСЭ имеет четыре последовательных радиальных канала ИРПС и параллельный магистральный канал (8 шин — двунаправленные шины данных, 4 шины — адресные, 4 шины — синхронизации и управления). Для умощнения шин и подключения к магистрали при длине магистрального кабеля более 10 м используется одно или несколько устройств УПМ. Параллельный и последовательный каналы применены в соответствии с типом канала в АСЭ. К ПЭВМ УППО подсоединяется через стандартную магистраль — МПИ.

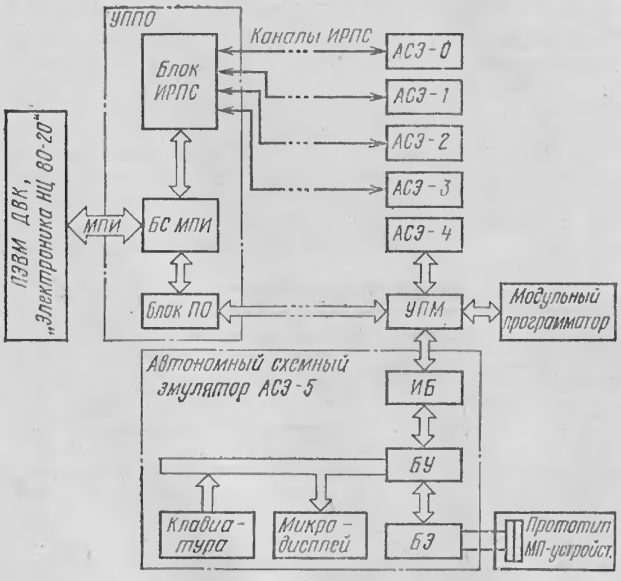


Рис. 1. Структурная схема многопользовательского отладочного комплекса для однокристальной микроЭВМ

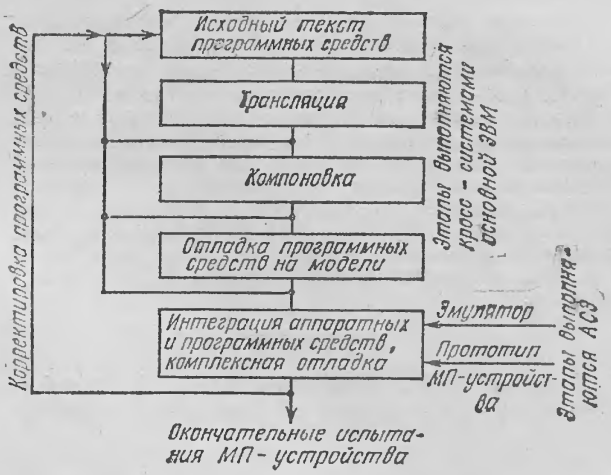


Рис. 2. Этапы проектирования и отладки МП-устройства с помощью МОК однокристальной микроЭВМ

Функции комплексной отладки распределяются по рабочим станциям — схемным эмуляторам. АСЭ обладают большой автономностью благодаря наличию в их составе блока управления с микроЭВМ, клавиатурой и микродисплеем. Распределение функций отладки по АСЭ позволяет использовать для одновременной работы 4...6 АСЭ, ориентированных на различные типы ОМЭВМ. Вместо одного из АСЭ с параллельным каналом обмена устанавливается модульный программатор ППЗУ (внешних и входящих в состав ОМЭВМ).

Программные средства многопользовательского отладочного комплекса (МОК): кросс-средства программирования в основном типовом составе для каждого типа ОМЭВМ, на которые ориентирован МОК; основная управляющая программа МОК для обращения и обмена данными между ПЭВМ и АСЭ; программа, управляющая модульным программатором; системные программы и средства программирования, входящие в состав ПЭВМ; внутренние управляющие программы АСЭ. К основной ПЭВМ АСЭ подключаются с помощью кабеля (до 10 м). На большее расстояние от ПЭВМ АСЭ устанавливается и подключается через дополнительное устройство УПМ. Проектирование и отладка МПУ с помощью комплекса показана на рис. 2. Формирование исходного текста, трансляция, компоновка и отладка программных средств МПУ осуществляются всеми пользователями по очереди на основной ПЭВМ с применением соответствующих кросс-средств.

Аппаратные и программные средства МПУ интегрируются распределенно на каждом автономном схемном эмуляторе в отдельности. Для ОМЭВМ (как правило, с небольшим размером программ) затраты времени на комплексную отладку МПУ в несколько раз больше, чем на формирование исходной программы и ее преобразование. Это позволяет использовать одну ПЭВМ 4...6 пользователям по очереди, практически не мешая друг другу. Применение ПЭВМ и кросс-средств автоматизации программирования резко повышает производительность труда разработчика МПУ: исключаются ошибки при многократном прохождении цикла проектирования, снижаются общие затраты времени. АСЭ имеет все необходимые аппаратно-программные средства для комплексной отладки МПУ, обращается к основной ПЭВМ только для записи на ГМД отлаженных модулей целевой программы и загрузки их в АСЭ на последующих этапах контроля и отладки.

Стоимость МОК в расчете на одного пользователя несколько превышает стоимость одного АСЭ на доступную величину $C_{ПЭВМ}/N$, где $C_{ПЭВМ}$ — стоимость ПЭВМ, а N — число пользователей комплекса.

Многопользовательский отладочный комплекс с распределенными функциями комплексной отладки функционально полностью завершен для инструментальной поддержки всех этапов проектирования и отладки МПУ. Он обладает более высокой производительностью, чем входящие в МОК средства по отдельности. Комплекс можно развивать по мере появления новых типов ОМЭВМ, наращивая соответствующие АСЭ и кросс-средства.

ЛИТЕРАТУРА

1. Иванов В. И., Лобанов В. И., Митрофанов А. В. Отладочные средства для малоразрядных однокристальных микроЭВМ // Микропроцессорные средства и системы.— 1984.— № 2.— С. 42.
2. Лобанов В. И., Мальков В. А. Аппаратное отладочное устройство для малоразрядной микроЭВМ // Электронная промышленность.— 1986.— № 2.— С. 52.
3. Белов А. М., Иванов Е. А., Муренко Л. Л. Комплекс кросс-программ «Электроника МИКРОСС»// Электронная промышленность.— 1986.— № 2.— С. 53.
4. Гукетлев Ю. Х., Квят В. Е., Потанин В. Ю. Кросс-система разработки математического обеспечения однокристальной микроЭВМ // Электронная промышленность.— 1986.— № 2.— С. 53.
5. Иванов Е. А, Муренко Л. Л., Широков Ю. Ф. Универсальная отладочная система автоматизации проектирования микропроцессорных устройств // Микропроцессорные средства и системы.— 1984.— № 3.— С. 53.

Статья поступила 19 апреля 1987

УДК 681.326

В. В. Тарасов

ОРГАНИЗАЦИЯ ПЕРЕДАЧИ УПРАВЛЕНИЯ МОНИТОРУ И ПРИКЛАДНЫМ ПРОГРАММАМ В ОТЛАДОЧНОМ УСТРОЙСТВЕ

Архитектура отладочного устройства (ОУ), в которой один микропроцессор (МП) поочередно выполняет функции отладочного монитора (системной программы) и физического моделирования (прикладной программы) микропроцессора проектируемой системы *, компактна и обладает большой надежностью. Подобный подход позволяет построить с минимальными затратами отладочную систему на базе серийно выпускаемых микроЭВМ, однако для обеспечения функции физического моделирования в реальном масштабе времени МП микроЭВМ должен быть идентичен МП проектируемой системы.

При совмещении МП функций контроля правильности хода выполнения прикладной программы (эмуляции) и его изменения используются директивы отладочного монитора, с помощью которых просматривается и модифицируется память и регистры МП; передается управление прикладной программе, аппаратно организованные остановы по циклам (пошаговый режим) и адресу контрольной точки. При этом желательно иметь в ОУ или отладочной системе технические средства, обеспечи-

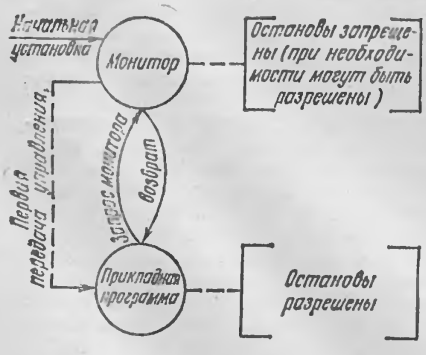


Рис. 1. Граф механизма передачи управления

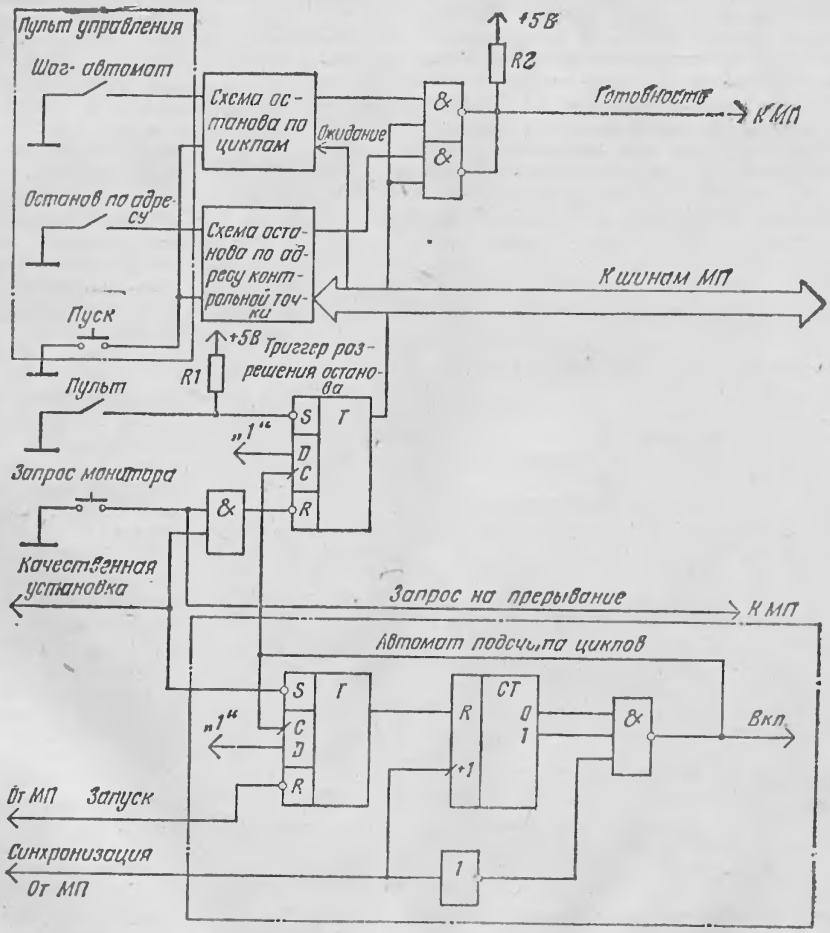


Рис. 2. Структурная схема устройства разрешения останова

* Лобанов В. И. Архитектура отладочных средств для микроконтроллера // Микропроцессорные средства и системы. — 1986. — № 3. — С. 30—31.

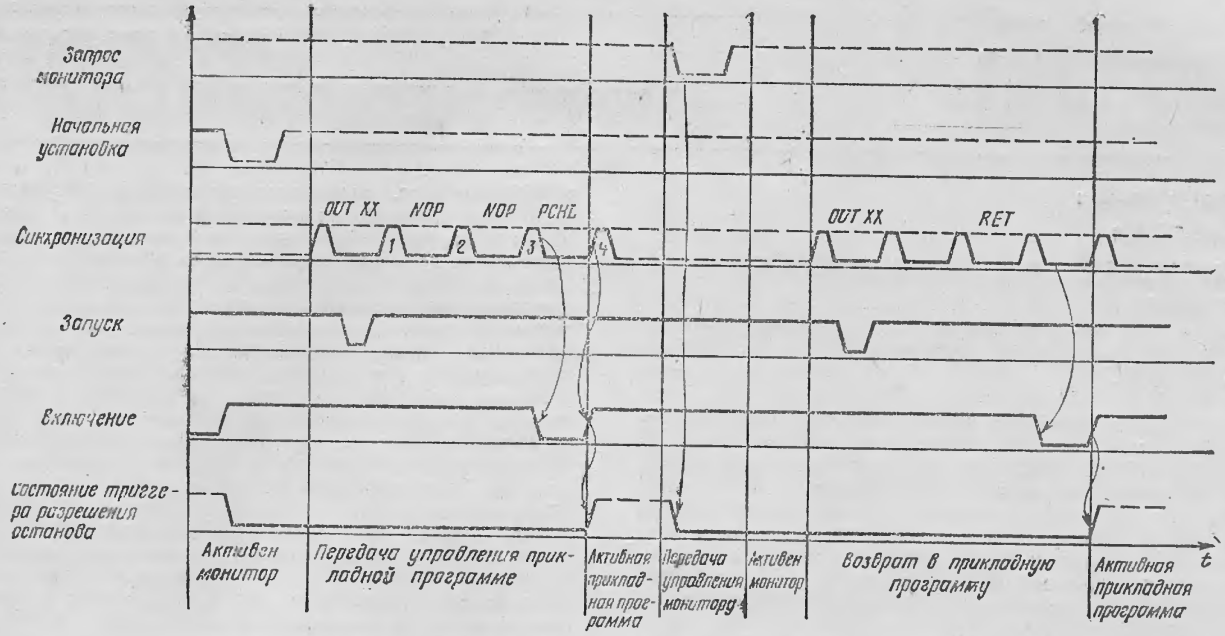


Рис. 3. Временна́я диаграмма устройства разрешения останова

вающие оперативную передачу управления между монитором и прикладной программой (без включения в последнюю дополнительных команд, рис. 1), и автоматическое разрешение остановов только при выполнении прикладной (отлаживаемой) программы или, при необходимости, и системных.

Рассмотрим аппаратно-программную реализацию механизма, обеспечивающего гибкое взаимодействие монитора с прикладной программой в ОУ, предназначенного для проектирования систем на базе микропроцессорного комплекта серии 580 (рис. 2).

С помощью триггера разрешения останова включаются и выключаются пошаговый режим и режим останова по адресу контрольной точки (своевременное включение триггера при передаче управления прикладной программе обеспечивает автомат подсчета циклов).

Подачей сигнала начальной установки (переключатель «Пульт» открыт) останов процессора запрещен, автомат подсчета циклов приводится в исходное состояние (рис. 3); МП подключается к системной шине ОУ и выполняет программу монитора.

Процедура передачи управления прикладной программе заканчивается последовательностью команд: OUT XX, NOP, NOP, PCHL, где XX — адрес автомата подсчета циклов. По обращению к автомату формируется сигнал «Запуск», по которому начинается подсчет импульсов цикла «Синхронизация» процессора (рис. 4).

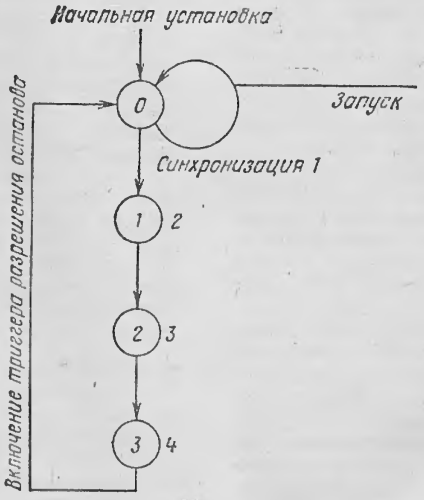


Рис. 4. Граф автомата подсчета циклов

Команда RET, выполняемая МП за три машинных цикла, используется для возврата в прикладную программу. Для разрешения останова в первом цикле команды прикладной программы (при первой передаче управления или при продолжении эмуляции) триггер должен быть включен по четвертому импульсу синхронизации. В этом случае при первой передаче управления прикладной программе и использовании для этой цели одноцикловой команды PCHL необходимо выполнение перед PCHL двух одноцикловых команд NOP.

При передаче управления прикладной программе МП разрешается останов; и он должен подключиться к системной шине проектируемой системы. Контроль правильности выполнения прикладной программы может производиться путем останова эмуляции по циклам или по адресу контрольной точки. Для более детального анализа хода эмуляции (просмотра и изменения содержимого памяти, регистров МП) необходимо передать управление монитору (клавиша «Запрос монитора» нажата). Процессору запрещается останов; он подключается к системной шине ОУ, формируется сигнал запроса на прерывание — управление передается монитору. Воспользовавшись директивами монитора («Возврат в прикладную программу»), можно продолжить эмуляцию. Выполнение этой процедуры аналогично передаче управления прикладной программе. Для наладки собственно ОУ достаточно включить переключатель «Пульт». Это позволит применить пошаговый режим и режим останова по адресу контрольной точки при выполнении системных программ. Таким образом, для осуществления рассмотренного протокола передачи управления в ОУ или серийных микроЭВМ достаточно простейшей аппаратной поддержки и внесения несложных изменений в соответствующие директивы монитора.

Телефон 3-39-18, Махачкала

Статья поступила 29 мая 1987

УДК 681.326—181.4

А. Ю. Сасов

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

В связи с переходом отечественной промышленности на выпуск микроЭВМ, совместимых с микроЭВМ серии IBM PC/XT (ЕС 1840, ЕС 1841, Искра 1030 и др.), разработаны программное (ПО) и аппаратное обеспечения системы цифровой обработки изображений. Система предназначена для решения задач коррекции, преобразования, анализа и реконструкции двумерных и трехмерных изображений в таких областях, как оптическая микроскопия, электронная микроскопия, дефектоскопия, дистанционное зондирование, и т. д.

1. Принципы построения системы цифровой обработки изображений. Основной принцип построения данной системы цифровой обработки изображений: получение предельно широких возможностей за счет развитого и оптимизованного ПО при минимальных сложности и, соответственно, стоимости аппаратной части. Основа комплекса — микроЭВМ, совместимая с IBM PC/XT. Для реализации полного объема ПО аппаратная часть должна содержать ОЗУ (640 Кбайт) и цветной дисплей. Желательно наличие матричного принтера, винчестерского диска и RAM-диска. Минимальный объем ОЗУ для реализации обработки изображений — 256 Кбайт.

Система цифровой обработки изображений позволяет вводить изображения с телекамеры (макросъемка, анализ фотоизображений, аэро- и космических снимков, с оптических и рентгенотелевизионных микроскопов, с просвечивающих электронных микроскопов и т. д.) и с растровых электронных микроскопов любого типа (отечественных и зарубежных).

Практически все производимые в настоящее время системы цифровой обработки изображений базируются на различных идеологиях, накладывающих определенные ограничения на круг решаемых задач. Например, в анализаторах изображений типа Leitz-TAS, Morphopericolor, Metapericolor информация обрабатывается методами математической морфологии. Эти методы позволяют эффективно преобразовывать и корректировать изображения, но их метрологический аппарат основан на статистическом анализе изображений и не позволяет получить конкретные метрологические параметры отдельных элементов структуры. В других существующих в настоящее время анализаторах изображений (IBAS, Magiscan) используется идеология индивидуального анализа отдельных элементов: пор, частиц и т. д. Это позволяет проводить всю гамму метрологических исследований, но в сотни раз увеличивает время получения среднестатистических характеристик структуры. Кроме того, существуют системы, предназначенные исключительно для коррекции изображений, спектральные анализаторы данных и многие другие.

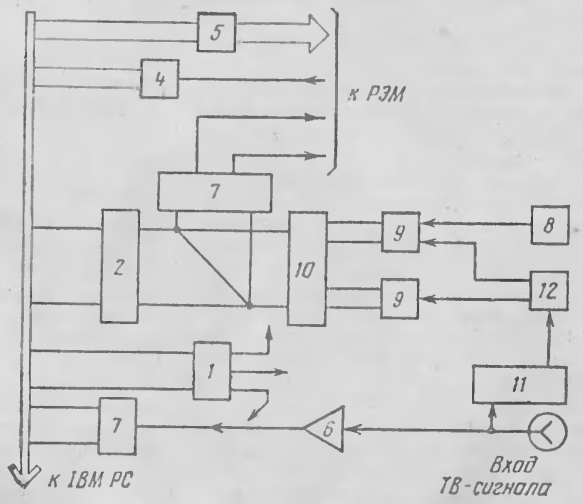
Исследователь обычно не может абсолютно точно ограничить круг задач для решения на приобретаемой аппаратуре, поэтому базирование анализаторов изображений на определенной идеологии рано или поздно лимитирует их использование.

При разработке данного анализатора изображений на базе микроЭВМ программно реализованы все наиболее распространенные подходы к обработке изображений. Кроме того, организован обмен изображениями между обрабатывающими программами так, что, например, изображение, эффективно откорректированное методами математической морфологии, может быть достоверно проанализировано программой индивидуального морфологического анализа. Это предельно расширяет возможности системы обработки изображений и в то же время исключает дублирование аппаратной части при необходимости реализации нескольких разных подходов.

Рассматриваемый анализатор изображений позволяет производить исследования, не имеющие аналогов в серийных приборах: быстрый анализ ориентации и степени анизотропии структуры на основе градиентных измерений, количественный анализ периодических и псевдослучайных структур по их одномерным и двумерным Фурье-спектрам с возможностью получения метрологических характеристик исследуемых объектов. На основе спектрального анализа реализован алгоритм детального исследования формы отдельных частиц, также не имеющий аналогов в анализаторах изображений. В отдельный пакет объединены уникальные программы для реализации микротомографии — метода неразрушающего исследования внутренней структуры микрообъектов.

Все программное обеспечение построено по принципу выбора пользователем режима по меню. Это позволяет работать с системой персоналу без предварительной подготовки.

2. Аппаратную часть системы цифровой обработки изображений составляет микроЭВМ, совместимая с IBM PC. Для ввода изображений используется одноплатный интерфейс, устанавливаемый в одно из гнезд канала ввода-вывода ЭВМ. Интерфейс содержит систему ввода изображений, поступающих в виде стандартного ТВ-сигнала, и систему ввода изображений с устройств с медленной разверткой (с растровых электронных микроскопов и т. п.). Формат вводимого изображения — 256×256 точек. При вводе изображений с систем с медленной разверткой яркость каждой точки оцифровывается восемью разрядами (256 градаций), при вводе ТВ-изображения — шестью разрядами (64 гра-



Блок-схема интерфейса ввода изображений

дации). Для считывания изображения с систем с медленным сканированием в интерфейсе формируются цифровые программно-управляемые развертки с регулируемой в широких пределах амплитудой и постоянной составляющей. Это позволяет адаптировать систему обработки изображений практически под любые растровые электронные микроскопы (РЭМ) и другие источники изображений. При вводе изображения на вход интерфейса должен поступать стандартный ТВ-сигнал (амплитуда 1 В, 625 строк, 50 Гц).

Блок-схема интерфейса (см. рисунок) содержит дешифратор адресов и команд (*1*), реверсивные счетчики курсора с возможностью установки от ЭВМ произвольного положения (*2*), цифро-аналоговые преобразователи для формирования медленных разверток РЭМ (*3*), аналого-цифровой преобразователь видеосигнала РЭМ (*4*), регистр программного управления режимом работы внешних устройств (*5*), видеоусилитель (*6*) и аналого-цифровой преобразователь ТВ-видеосигнала (*7*), кварцевый генератор тактовой частоты ТВ-синхронизации (*8*), счетчики точек и ТВ-строк (*9*), цифровой компаратор точки считывания сигнала (*10*), селектор синхроимпульсов из ТВ-сигнала (*11*), схемы задержки синхросигналов (*12*). Время аналого-цифрового преобразования для сигналов РЭМ — не более 30 мкс, для ТВ-сигнала — 125 нс. Интерфейс содержит всего 32 ИС широко распространенных серий. Это обеспечивает низкую стоимость и высокую надежность аппаратной части системы цифровой обработки изображений.

3. Программная часть системы цифровой обработки изображений. Для обеспечения низкой стоимости при предельно широких возможностях системы обработки изображений основная нагрузка переложена с аппаратных на программные средства. ПО состоит из пяти практически независимых библиотек программ, каждая из которых реализует определенный подход к обработке изображений. Каждая библиотека программ управляется выбором режимов по меню с возможностью передачи изображения из одной библиотеки программ в другую. Рассмотрим основные возможности каждой библиотеки программ.

Программа **«Коррекция и преобразование изображений»** позволяет вводить полутоновое изображение с телекамеры, с растрового микроскопа; считывать с диска или из буферной зоны RAM-диска (при хранении там изображения, сформированного другой программой); устанавливать оптимальную яркость, контрастность и фокусировку изображения с телекамеры или с растрового электронного микроскопа; корректировать яркость и контрастность, определять локальные оптические характеристики объекта: оптическую плотность, коэффициенты отражения или пропускания с возможностью калибровки по эталонным участкам, калибровку и анализ изображения в режиме регистрации отраженных электронов растрового электронного микроскопа с определением локального атомного номера вещества образца; подчеркивать границы и уменьшать шумы на исходном полутоновом изображении; выделять контуры объектов вычислением градиента сигнала; проводить преобразования Лапласа и Собела; корректировать неравномерность освещенности с возможностью ввода, хранения и преобразования опорного поля освещенности, арифметические операции между несколькими изображениями; записывать изображения на диск и в буферную зону RAM-диска; выводить полутоновое изображение на принтер.

Программа **«Обработка и анализ методами математической морфологии»** позволяет вводить полутоновое изображение с телекамеры, с растрового электронного микроскопа, с диска и из буферной зоны RAM-диска; преобразовывать полутоновое изображение в бинарное по одному порогу или в яркостном окне; проводить геометрические преобразования первого уровня для бинарного изображения: эрозию, дилатацию, открытие, закрытие, преобразования второго уровня: деагломерацию частиц, восстановление границ зерен, скелетизацию объектов, удаление частиц, пересекающих край изображения; проводить анализ морфологии: определение полной и относительной площади частиц (пор); полного периметра, числа частиц (пор), средней площади, периметра, размера, коэффициента формы одной частицы (поры), средних длины и толщины при исследовании нитевидных объектов, значений проекций частиц (пор) на горизонтальную, вертикальную и наклонные под углом 45° оси; записывать полутоновое или бинарное изображение на диск и в буфер на RAM-диске; выводить бинарное или полутоновое изображение на принтер.

Программа **«Индивидуальный морфологический анализ структуры»** позволяет вводить полутоновое изображение с телекамеры, с растрового электронного микроскопа, с диска или из буферной зоны RAM-диска; корректировать неравномерность освещенности без анализа опорного поля освещенности; автоматически преобразовывать полутоновое изображение в бинарное при исследовании изображений с двумя преимущественными градациями яркости; проводить диалоговое преобразование в бинарное по одному пороговому уровню яркости или в яркостном окне, цветокодирование полутонового изображения, вычисление гистограммы яркости полутонового изображения, анализ направления ориентации и степени анизотропии исследуемой структуры, полный морфологический анализ бинарного изображения на основе индивидуального измерения параметров каждой частицы (поры) с определением полной и относительной площадей, полного периметра, числа частиц (пор), среднего размера, площади, периметра, коэффициента формы одной частицы (поры), построение гистограмм распределения по размерам, площадям, коэффициенту формы, суммарным площадям в интервалах размеров пор (частиц), вычисление матриц распределения одновременно по двум параметрам — форме и размерам; выводить полутоновое изображение на принтер; записывать на диск и в буферную зону RAM-диска; калибровать увеличение для правильного получения результатов в размерных единицах; устанавливать диапазон размеров частиц (пор), включаемых в анализ, и других параметров анализа.

Программа **«Фурье-анализ изображений»** позволяет вводить полутоновое изображение с телекамеры, растрового электронного микроскопа, с диска и из буферной зоны RAM-диска; проводить одномерное преобразование Фурье заданной строки изображения и метрологический анализ периодичности исследуемой структуры по ее спектру, быстрое двумерное преобразование Фурье от полутонового изображения, метрологический анализ пространственной периодичности исследуемой структуры на основе двумерного Фурье-спектра; получать численные параметры анизотропии и гранулометрии псевдослучайных структур по двумерным Фурье-спектрам, проводить спектральный анализ формы частиц; выводить изображения и двумерные Фурье-спектры на принтер и записывать на диск и в буферную зону RAM-диска.

Программа **«Микротомография»** позволяет считывать двумерные теневые изображения с рентгенотелевизионного микроскопа и с рентгеновского микроанализатора в просвечивающем режиме; вводить изображения с диска и из буфера на RAM-диске; объективно устанавливать оптимальную яркость, контрастность и фокусировку изображения; юстировать аппаратную часть микросканера; реконструировать внутреннюю трехмерную структуру микрообъекта по теневым двумерным проекциям с программным управлением микросканером; проводить сбор проекционных данных с записью на винчестерский диск и последующей «off-line» реконструкцией; просматривать изображения заданного сечения объекта в виде полутонового или цветокодированного изображения; получать псевдотрехмерное изображение внутренней реконструированной микроструктуры объекта; выводить изображения теневых проек-

ций и реконструированных сечений на принтер и записывать их на диск или в буфер на RAM-диске.

Как видно из краткого обзора возможностей ПО, **данная система обработки изображений позволяет проводить практически все работы, реализуемые на серийных системах, и имеет множество оригинальных режимов, не имеющих в настоящее время аналогов.** Все разнообразие возможностей реализовано исключительно программными средствами без изменения аппаратной части ЭВМ типа IBM PC/XT. **Все программы рассчитаны на пользователя, не имеющего предварительной подготовки в области программирования и использования ЭВМ.** Выбор конкретных операций производится по меню, где представлены все возможные режимы для каждого комплекта программ, а в диалоговых режимах работы возможные действия пользователя указываются на экране ЭВМ. Каждая библиотека программ имеет варианты меню и вывода результатов на принтер на русском и английском языках.

4. Примеры использования системы обработки изображений. Применение данного комплекса обработки изображений позволяет поднять исследования в области оптической и растровой электронной микроскопии на принципиально новый уровень. Ввод информации с телекамеры открывает широкие возможности по анализу фотографических изображений, в том числе аэро- и космической информации.

Пример спектрального анализа изображения зоны разрушения образца из алюминиевого сплава при динамических нагрузках показан на рисунках *а—г* (см. 2-ю стр. обложки). Полутоновое изображение объекта (рис. *а*) получено с помощью растрового электронного микроскопа. Фурье-спектр, представленный над изображением, имеет характеристический пик. При совмещении курсора с этим пиком на экран выдается размер периода исходной структуры объекта. Более полную информацию можно получить из двумерного Фурье-спектра. Результат двумерного быстрого преобразования Фурье от полутонового изображения показан на рис. *б*. При метрологическом анализе по двумерному Фурье-спектру (рис. *в*) при совмещении курсора с рефлексом на спектре на экран выдается информация о характерном размере периода структуры объекта. Пример анализа ориентационных и гранулометрических характеристик данного объекта по двумерному Фурье-спектру показан на рис. *г*.

Использование рассмотренной системы целесообразно в физике, химии, геологии, биологии, при решении задач материаловедения, дефектоскопии и т. д. Огромную пользу может принести данная аппаратура в электронной промышленности, порошковой металлургии, синтезе композиционных материалов, полимеров, сплавов, во многих других областях.

Телефон 939-35-87, Москва

Статья поступила 20 мая 1988

УДК 681.322

А. В. Бабаянц, Ю. М. Горьковец, А. С. Вартанов, Г. Е. Фрезе

МИКРОПРОЦЕССОРНЫЙ ПУЛЬТ ТЕХНОЛОГА-ОПЕРАТОРА

Микропроцессорные системы АСУТП разработаны и внедрены на ряде биотехнологических производств Минмедбиопрома, продукцией которых являются кормовые дрожжи, микробиологические средства защиты растений и пенициллин медицинского назначения [1—3]. Системы построены с использованием специализированных микропроцессорных комплексов (МПК) Биоцикл МП-01-01, Абикорд МП-01-01, Аскорд МП-01-01 для децентрализованного управления технологическими агрегатами и микропроцессорных пультов технолога-оператора Биоцикл МП-01-06, Биоцикл МП-01-07, Биоцикл МП-01-08 для централизованного представления информации.

Персонал станции работает в основном за пультом технолога-оператора (ПТО), где на символьных дисплеях, цветных мониторах и АЦПУ представлена вся информация о процессе. Изменить уставки регулятором можно с ПТО и МПК.

Связь ПТО с МПК (1...21), непосредственно управляющими технологическими агрегатами, осуществляется с помощью бит-последовательных контроллеров магистрального типа КС 52.21. МПК могут быть удалены от магистрали на расстояние до 5 км. Пульт представляет собой многомикропроцессорное устройство, состоящее из трех блоков (рис. 1): одного ведущего Б1 и двух ведомых Б2-1, Б2-2. Обмен между блоками осуществляется по радиальной бит-последовательной линии.

Система обеспечивает представление информации в виде мнемосхем, барграфов и трендов на экранах цветных дисплеев, отображение текущей и обобщенной информаций в виде таблиц на символьных дисплеях, протоколирование хода технологического процесса, связь персонала с системой, текстовую поддержку оператора, диагностические функции, связь с другими системами.

Смена кадров на экранах дисплеев, а также ввод управляющих воздействий осуществляются по инициативе пользователя с клавиатуры. Технические характеристики ПТО приведены ниже.

Число МПК	до 21
Число микропроцессоров	3
Максимальная скорость обмена информацией с МПК, Кбит/с	12
Объем памяти, Кбайт	
ведущий блок ОЗУ (РПЗУ)	32 (32)
ведомый блок ОЗУ (РПЗУ)	32 (32)
Периодичность обновления информации на терминалах, с	2

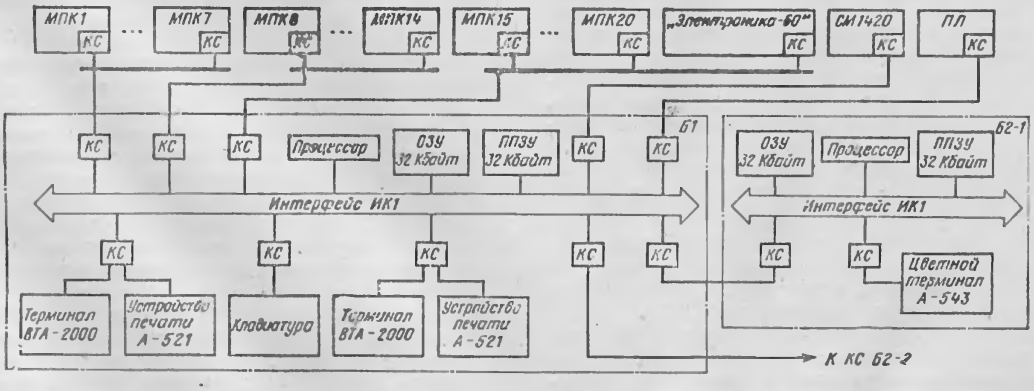


Рис. 1. Структура взаимосвязей элементов пульта, МПК и ЭВМ

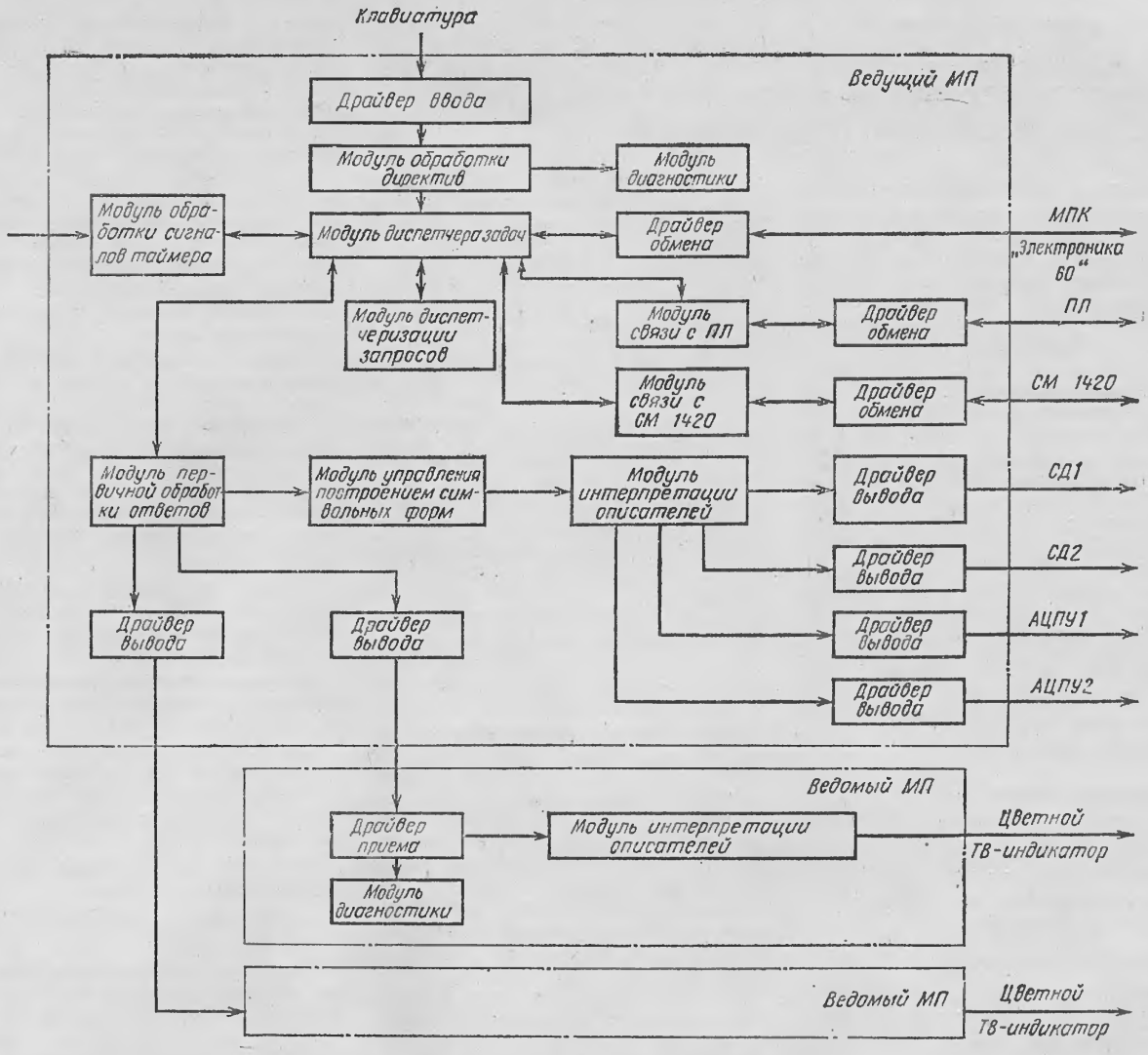


Рис. 2. Структура программного обеспечения ПТО

Число символьных дисплеев ВТА-2000	2
Число цветных ТВ-индикаторов А-543-14	2
Число устройств печати А-521-4	2
Число таблиц на символьных дисплеях для одного агрегата	до 30
Число видеограмм на цветных ТВ-индикаторах для одного агрегата	до 30
Число печатных документов	до 30

Программное обеспечение (ПО) в соответствии с архитектурой пульта технолога-оператора содержит программы для ведущего и ведомого МП. Каждая из этих программ включает библиотеки стандартных и специальных подпрограмм и управляющую программу (рис. 2).

Программы ПО ПТО написаны на языке макроассемблера для КТС ЛИУС-2. Объем ПО вместе с диагностическими функциями занимает 80 Кбайт. При создании ПО использованы специальные языки генерации символьных и графических форм представления информации, которые позволяют настроить его на конкретный объект. Программные средства являются типовыми и могут использоваться для различных модификаций ПТО. Для удобства разработки конкретных вариантов ПТО предусматривается создание системы генерации всех настроечных массивов.

364060, Грозный, пр. Кирова, 2, НПО «Промавтоматика»; тел. 7-13-40

ЛИТЕРАТУРА

1. Опыт разработки и внедрения АСУ ТП Мозырского завода кормовых дрожжей // А. А. Опришко, А. В. Бабаянц, В. П. Давыдов и др. // Приборы, средства автоматизации и системы управления.— М.: ЦНИИТЭИ приборостроения.—1986.—Вып. 3.—С. 35.
2. Микропроцессорная дисплейная АСУ ТП Мозырского завода кормовых дрожжей / А. А. Опришко, А. В. Бабаянц, В. П. Давыдов и др. В кн.: Опыт работы Грозненского НПО «Промавтоматика» по автоматизации объектов агропромышленного комплекса.— М.: ЦНИИТЭИ приборостроения.— 1986.— С. 5—9.
3. Система управления периодическим процессом ферментации с использованием микропроцессорного комплекса «Биоцикл» / В. П. Давыдов, Ф. Н. Чегодаев, М. М. Гадильшин и др. // Механизация и автоматизация управления.— 1985.— № 1.— С. 21—24.

Статья поступила 14 августа 1987

УДК [681.325.5—181.4+681.174] : 658.381

А. А. Жуков, В. Н. Кузин, М. Ю. Панов, А. В. Сомов, А. М. Хмелевский

МИКРОПРОЦЕССОРНАЯ АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА РАБОЧЕГО ВРЕМЕНИ

Одной из форм организации труда на производстве, основанной на объективном контроле и учете рабочего времени *, является так называемый гибкий рабочий день (ГРД). Предлагаемая автоматизированная система учета рабочего времени (АСУРВ) регистрирует входы-выходы работников через контрольно-пропускные пункты и обрабатывает накопленные данные. АСУРВ полностью исключает субъективный фактор при определении момента входа-выхода и баланса рабочего времени.

АСУРВ имеет двухуровневую структуру с одной центральной станцией (ЦС) и несколькими (максимум 32) периферийными (ПС), объединенными каналами связи. Удаленность ПС от ЦС может достигать 1000 м. Станции построены по магистрально-модульному принципу. Размер модулей 100×160 мм, элементная база — БИС серий КР580, К555, К537. Техническая характеристика АСУРВ приведена ниже.

Центральная станция

Центральный процессор . .	КР580ВМ80
Объем ОЗУ, Кбайт	64
Число каналов связи с ПС	32
Тип канала связи	Токовая петля
Напряжение питания, В (Гц)	220 (50)
Потребляемая мощность, Вт	850

Периферийная станция

Центральный процессор	КР580ВМ80
Объем ОЗУ, Кбайт	8
Объем ПЗУ, Кбайт	8
Число считывающих устройств	2
Потребляемая мощность, Вт	100

Исходные данные, справочники, промежуточные результаты хранятся на магнитных дисках. Конструктивно ЦС выполнена в виде стола, в тумбу которого встроены цифровой блок и НГМД «Электроника ГМД 7012». На столе размещаются дисплей 15ИЭ-00-013 и печатающее устройство УВВПЧ-30-004. Модули ВМ80-2 (центральный процессор), ВН59-1 (контроллер прерываний), ВИ53-1 (таймер), ОЗУ 64К-1 (ОЗУ 64 Кбайт), СК-2 (системный контроллер) образуют ядро, обеспечивающее функционирование операционной системы и прикладного программного обеспечения (рис. 1). Модули ВВ51-8 предназначены для организации двунаправленного обмена данными между ЦС и ПС. Модуль ВВ51-1 обеспечивает связь ЦС с дисплеем, модуль ВВ55-1 — с печатающим устройством (например, УВВПЧ-30-004), модуль КНГМД (контроллер накопителя на гибких магнитных дисках) с НГМД «Электроника ГМД 7012».

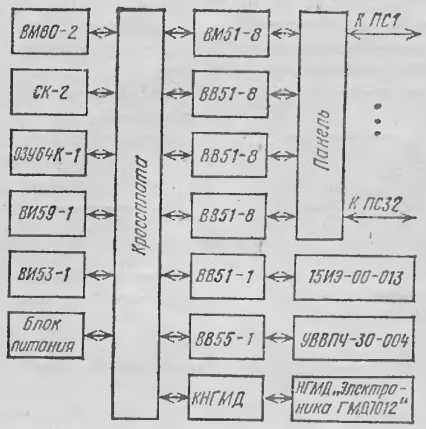


Рис. 1. Структурная схема центральной станции

Периферийная станция располагается на контрольно-пропускном пункте (КПП) и состоит из цифрового блока и двух считывающих устройств (рис. 2). Станция считывает и обрабатывает код с пропуска сотрудника, накапливает информацию о входах-выходах, выдает информацию по запросу на ЦС.

Назначение модулей ВМ80-2, ВН59-1, ВИ53-1 то же, что на ЦС. Модуль ЗУ 16К-1 (модуль памяти) включает в себя ПЗУ объемом 8 Кбайт и ОЗУ с сохранением информации объемом 8 Кбайт. Он предназначен для хранения управляющей программы ПС и текущей информации. Модуль ВИ1-1 — энергонезависимый счетчик времени с ценой отсчета 1 мин. Модули ВВ55-1 предназначены для связи с фотосчитывателями ФС-1 по параллельному каналу связи длиной до 5 м. Через модуль ВВ51-1 осуществляется связь с ЦС по последовательному каналу. Общий алгоритм функционирования системы следующий.

Каждому сотруднику выдается пропуск, имеющий поле кодовых отверстий, в котором кодируется определенный только ему принадлежащий номер. При проходе через КПП сотрудник должен вставить пропуск в приемный карман считывающего устройства, расположенный первым по направлению прохода. Тем самым определяется направление — вход или выход. Номер на пропуске кодируется избыточным кодом, что позволяет исправлять одиночные ошибки и выявлять двойные, тройные. Считанный с пропуска код декодируется и, если ошибок не обнаружено, то сотруднику дается разрешение на проход (зажигается соответствующий транспарант), а информация о проходе (номер сотрудника, время, направление прохода) записывается в буферную память.

Данные, накопленные на ПС, передаются на ЦС по запросу и записываются на ГМД. На ЦС формируется справочная база, основными компонентами которой являются графики режимов рабочего дня, список подразделений, данные об отработанном времени и причинах отклонения от нормального режима на месяц (табель).

Совместная обработка данных, получаемых с ПС, и справочных данных позволяет иметь следующие выходные документы:

список нарушений установленного графика (опоздание, преждевременный уход, отсутствие и т. д.);

время, отработанное каждым сотрудником за день, и отклонение от суммарной номинальной продолжительности рабочего времени с начала периода;

суммарное отработанное время за период гибкого графика;

статистические данные;

табель.

Обработка информации производится по программам, хранящимся на ГМД. Работой ЦС управляет оператор, вводя с клавиатуры дисплея команды и данные. Функционирование ПС происходит по программе, записанной в ПЗУ. Система слежения за питающими напряжениями вырабатывает сигнал «авария питания» при выходе напряжений за допустимые пределы, обрабатывает этот сигнал, текущие данные и передает

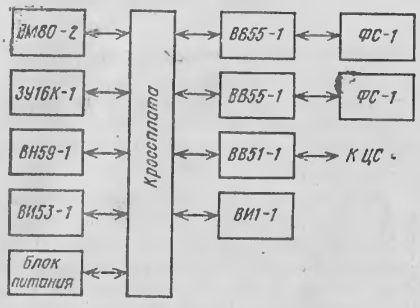


Рис. 2. Структурная схема периферийной станции

* Шубин А. А., Нольский Л. В., Коляновский Ю. В. Автоматизированная система учета рабочего времени на машиностроительном предприятии. — М.: Машиностроение, 1975.

накопленную информацию на запись в энергонезависимое ОЗУ. При появлении питающих напряжений ПС возобновляет работу в режиме «продолжение».

В процессе работы процессор ПС постоянно контролирует состояние пар светодиод-фотодиод для фиксации отказов считывателей информации. Информацию о состоянии считывающих устройств в любой момент времени можно запросить с ЦС и, в случае необходимости, определить место неисправности с точностью до пары элементов.

Система АСУРВ функционирует в течение нескольких лет с высокой надежностью при небольших эксплуатационных затратах.

Телефон 534-94-59, Москва

Статья поступила 7 сентября 1987

УДК 681.326

А. Б. Слюсар, Ю. В. Мороз, В. Г. Васильев

ЦИФРОВОЙ РЕГУЛЯТОР СКОРОСТИ НА ОСНОВЕ ОДНОКРИСТАЛЬНОЙ МИКРОЭВМ КР1816ВЕ35

Для стабилизации скорости вращения вала электродвигателя постоянного тока с частотным датчиком скорости разработан регулятор, функционирующий по методу дискретной фазовой коррекции с помощью импульсных цепей [1]. Данный метод позволяет без преобразования длительности импульсов в напряжение непосредственно вводить корректирующие непрерывные сигналы в импульсные последовательности. Управляющий сигнал формируется введением регулируемой задержки (РЗ) в импульсную последовательность задающей частоты $f_з$ перед сравнением ее на фазовом дискриминаторе (ФД) с последовательностью импульсов регулируемой частоты $f_р$ от частотного датчика скорости двигателя. Регулируемая задержка сдвигает фазу сигналов $f_з$ относительно $f_р$ пропорционально сумме корректирующих сигналов (рис. 1). Возникающая в результате модуляция фазы вызывает соответствующее изменение управляющего широтно-импульсного сигнала на выходе ФД.

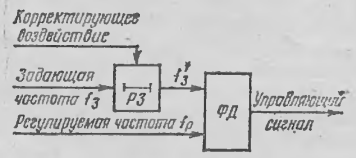


Рис. 1. Схема дискретной фазовой коррекции

Основной элемент регулятора скорости (рис. 2) — однокристальная микроЭВМ (ОМЭВМ) КР1816ВЕ35— аналог К1816ВЕ48, но без внутреннего ПЗУ команд [2]. ОМЭВМ функционирует по программе, зашитой во внешнем ПЗУ команд. Адрес команды фиксируется сигналом ALE в буфере адреса, собранном на регистрах К555ТМ8. Код команды считывается из ПЗУ по сигналу РМЕ. ОМЭВМ обменивается данными с каналами

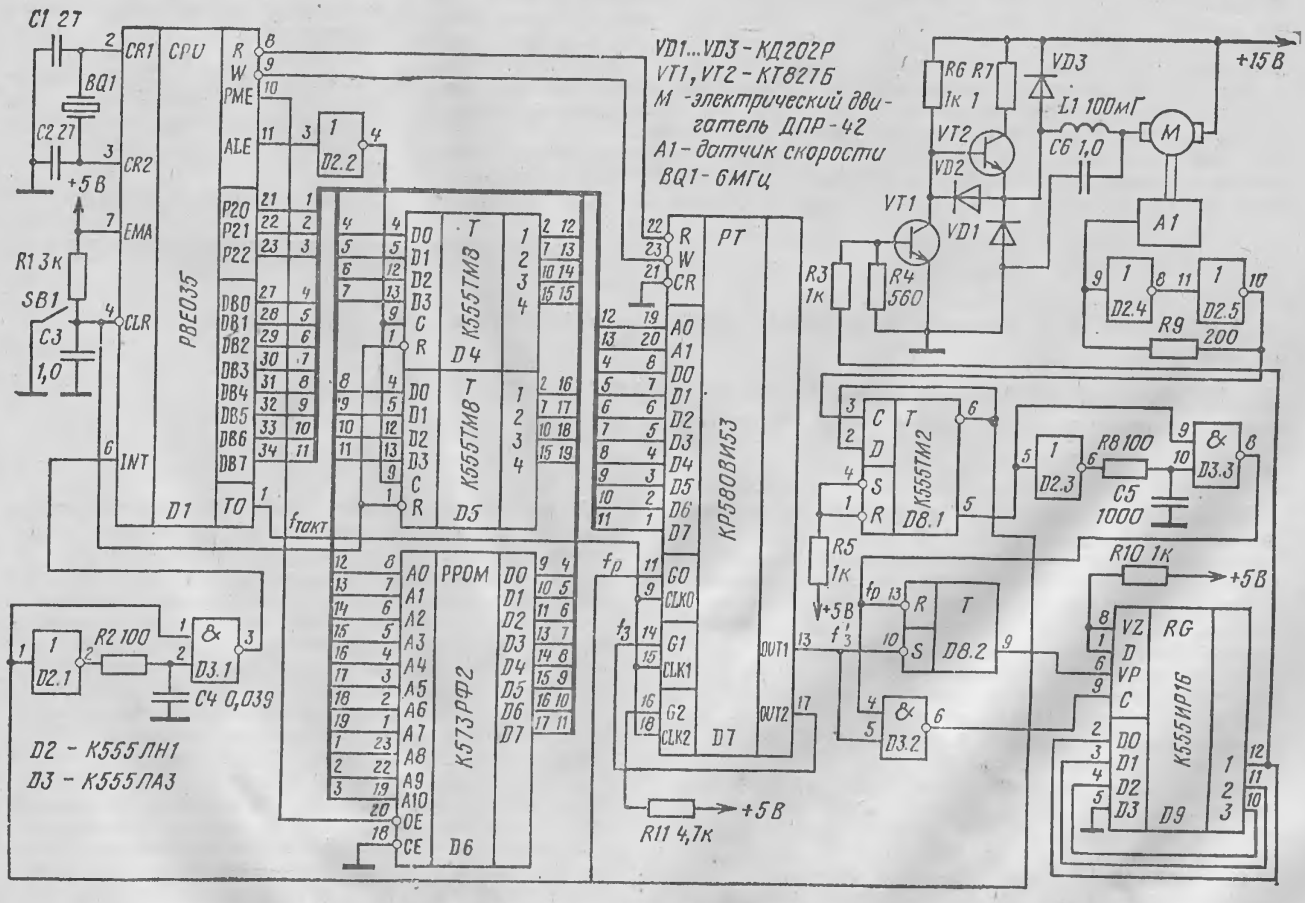


Рис. 2. Принципиальная электрическая схема цифрового регулятора скорости

программируемого таймера (ПТ) КР580ВИ53 [3] с помощью сигналов R (чтение) и W (запись), как с ячейками внешнего ОЗУ данных.

На канале «0» реализован измеритель длительности положительного полупериода f_p, получаемой делением на два частоты датчика скорости двигателя (А1). Дискретность отсчета определяется тактовой частотой $f_{такт}$ ОМЭВМ. Счет инициируется фронтом f_p и прекращается по срезу. По срезу происходит и прерывание ОМЭВМ после каждого измерительного цикла и вычисляется код коррекции, зависящий от величины отклонения длительности полупериода f_p от заданного значения. Канал «2» ПТ использован для получения $f_з$ делением $f_{такт}$. На канале «1» реализована регулируемая задержка (временной интервал от прихода фронта $f_з$ на вход канала «1» до появления его на выходе), определяемая 2-байтовым кодом, записанным в счетчик канала. Задержанные импульсы частоты $f_з$ сравниваются на ФД, выполненном на сдвиговом регистре К555ИР16, с импульсами f_p. С выхода ФД широтно-импульсный сигнал управления через усилитель мощности (транзисторы VT1, VT2) подается на якорь электродвигателя (М). Управляющая программа хранится в ППЗУ К573РФ2 и содержит блок инициализации и блок вычисления коррекции. В блоке инициализации выполняется начальная настройка аппаратуры регулятора: программируются режимы работы каналов ПТ и заносятся исходные значения в счетчики. Заканчивается блок инициализации циклом ожидания прерывания. В момент прерывания в счетчике канала «0» ПТ находится числовой код, соответствующий длительности полупериода f_p. По прерыванию управление передается блоку вычисления коррекции. По коду, прочитанному из канала «0», определяется отклонение τ_i измеренного полупериода от заданного значения и определяется разность $\Delta\tau$ между τ (отклонением в текущем периоде) и τ_{i-1} (отклонением в предыдущем периоде). Величина $\Delta\tau$ используется как приближенное значение производной отклонения по времени. Текущее значение τ_i запоминается и будет использовано в следующем периоде в качестве τ_{i-1}. Коррекция $t_к$ (числовой код задержки) вычисляется в соответствии с формулой $t_к = K1 \cdot (\tau_i + K2 \cdot \Delta\tau)$. Коэффициенты К1 и К2 подбираются экспериментально при настройке регулятора на конкретный тип двигателя. Разностная составляющая $K2 \cdot \Delta\tau$ обеспечивает апериодичность переходных процессов. Вычисленное значение коррекции записывается в счетчик канала «1» ПТ и ОМЭВМ возвращается в состояние ожидания. Все арифметические операции в программе выполняются, как с 2-байтовыми целыми числами без знака. Численные значения коэффициентов К1 и К2 выбираются равными целым степеням числа 2 (1, 2, 4...). Это позволяет заменить операции умножения операциями сдвига влево и, таким образом, значительно сократить время вычислений.

Настройка регулятора на различные типы электродвигателей постоянного тока и разные скорости вращения заключается в выборе коэффициентов К1 и К2 в формуле и в изменении коэффициента деления в канале задающей частоты (канал «2» ПТ). Аппаратурная часть регулятора остается без изменений.

При испытаниях регулятора с электродвигателем постоянного тока типа ДПР с индукционным датчиком скорости вращения (число зубьев на окружности Z=130; скорость вращения 11 об./с) коэффициент колебаний скорости составил 0,15...0,18 % (высокий показатель для данного класса электродвигателей). Исследования с другими типами двигателей постоянного тока на различных скоростях подтверждают перспективность применения ОМЭВМ в системах автоматики. Регулятор можно применять в системах, требующих плавного изменения скорости по какому-либо закону.

Телефон 229-24-07, Киев, Слюсар А. Б.

ЛИТЕРАТУРА

1. Трахтенберг Р. М. Импульсные астатические системы электропривода с дискретным управлением.— М.: Энергоиздат, 1982.— 169 с.
2. Кобылинский А. В., Липовецкий Г. П. Однокристальные микроЭВМ серии К1816//Микропроцессорные средства и системы.—1986.—№ 1.— С. 10—19.
3. Торгов Ю. И. Программируемый таймер КР580ВИ53 и его применение // Микропроцессорные средства и системы.—1984.—№ 1.— С. 77—84.

Статья поступила 3 февраля 1987

МНОГОМАШИННЫЕ КОМПЛЕКСЫ

УДК 681.322

В. А. Богатырев, Л. Д. Данилова, Л. С. Иванов, А. В. Каменев, А. Ю. Куконин, А. В. Охапкин

УСТРОЙСТВО ДЛЯ ПОСТРОЕНИЯ МНОГОМАШИННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Устройство сопрягается с микроЭВМ «Электроника 60» по интерфейсу МПИ и позволяет организовать вычислительную систему с конфигурацией типа «Общая магистраль» (ОМ). Обмен между микроЭВМ осуществляется 16-разрядными словами, сгруппированными в пакеты. Адрес ЭВМ-приемника указывается в заголовке пакета. Второе слово определяет число слов в пакете.

Устройство передает данные по адресу одной ЭВМ-приемника, всем машинам системы (широковещательная передача) и указанным машинам (групповая). Возможное число адресуемых микроЭВМ — 64.

Общая магистраль имеет три линии:

данных (Д) — для последовательной передачи адреса, данных, сигналов подтверждения установки канала связи с адресуемой ЭВМ и сигналов подтверждения приема каждого передаваемого слова;

сопровождения передаваемого бита (ТИ);

арбитража — для определения занятости ОМ.

Основные узлы (рис. 1).

Узел сопряжения (УС) обеспечивает связь с каналом микроЭВМ, имеет два уровня запроса прерывания (ТРБ1, ТРБ2) и программно-доступный флаг запроса на захват ОМ. Реализуется на БИС К1801ВП1-033 в режиме программируемого параллельного интерфейса.

Регистр данных (РД) обеспечивает параллельный прием, последовательную выдачу, последовательный прием и параллельную выдачу данных. Основу РД может составить, например, 8-разрядный регистр К155ИР13 или 4-разрядный регистр К555ИР16. В предлагаемом устройстве используется 8-разрядный регистр К155ИР13.

Регистр признаков (РП) — набор триггеров (программно доступных по чтению со стороны микроЭВМ), используемый для анализа состояния устройства. Для организации межмашинного обмена используются четыре признака: захват ОМ данным устройством (F1), чтение слова ЭВМ-приемником (F2), прием ЭВМ-приемником слова с ОМ (F3), адресуемость данного устройства ОМ (F4).

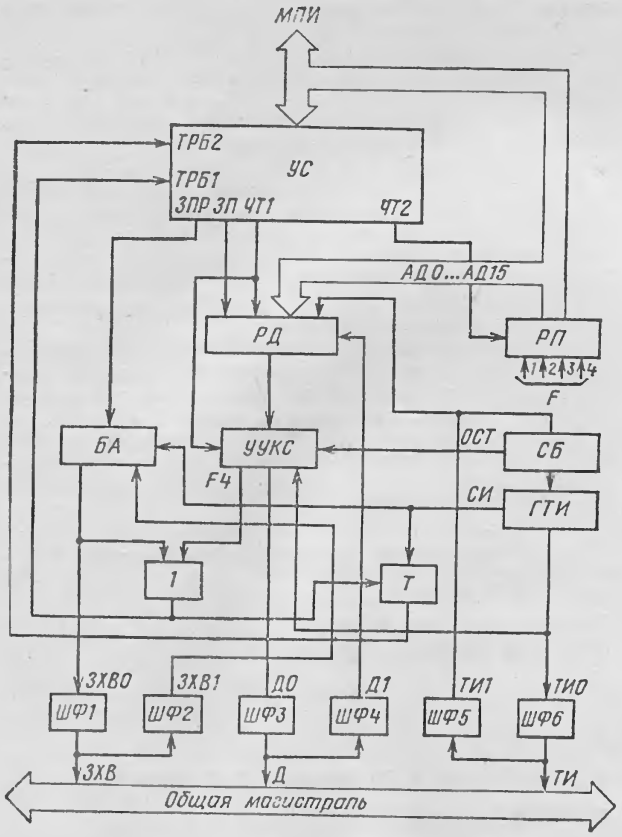


Рис. 1. Структурная схема устройства:

УС — узел сопряжения; РД — регистр данных; РП — регистр признаков; БА — блок арбитража; УУКС — узел управления каналом связи; СБ — счетчик битов; ГТИ — генератор тактовых импульсов; Т — таймер; ШФ1...ШФ6 — шинные формирователи

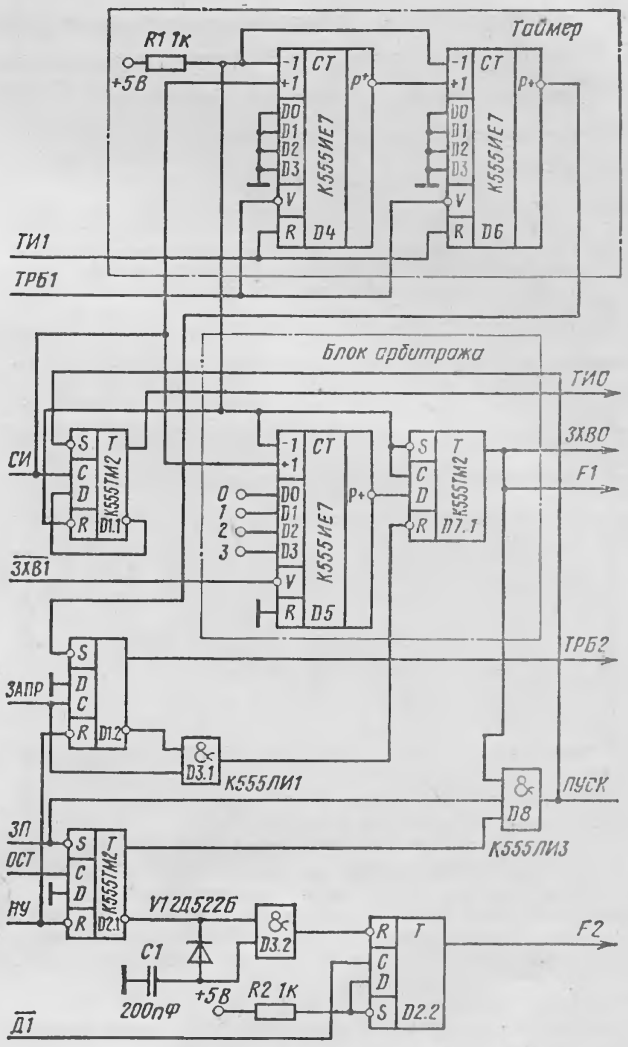


Рис. 2. Блок арбитража и таймер

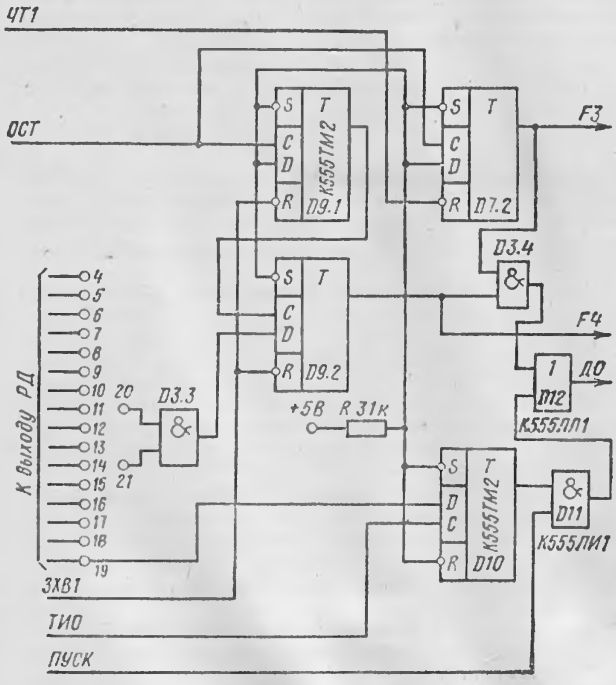


Рис. 3. Узел управления каналом связи

Блок арбитража (БА) реализует приоритетное обслуживание запросов на захват ОМ (рис. 2). На входах счетчика DБ (0, 1, 2, 3) задается приоритет устройств по наибольшему двоичному значению. Счетчик находится в режиме начальной установки, если ОМ занята (Лог.0 — на входе $\overline{ЗХВ1}$); при освобождении ОМ счетчики БА всех устройств переходят в режим счета. Использование счетчика К555ИЕ7 в БА позволяет организовать систему, состоящую из 16 микроЭВМ (при расширении системы разрядность счетчика БА должна быть увеличена).

Узел управления каналом связи (УУКС) (рис. 3) осуществляет адресацию устройств с ОМ 16-разрядным словом, широковещательную и групповую связи при асинхронном обмене. Адрес устройств на ОМ определяется соединением соответствующих выходов РД со входами элемента D3.3. Состояние триггера D9.2 является значением признака F4. После передачи каждого слова пакета все адресуемые ЭВМ выставляют на линию Д единичный уровень, снимаемый по чтению переданного слова всеми ЭВМ-приемниками. Признак F2 определяется по переходу уровня на линии Д (воспринимается триггером D2.2) Чтение переданного слова ЭВМ-приемниками реализуется при установлении признака F3.

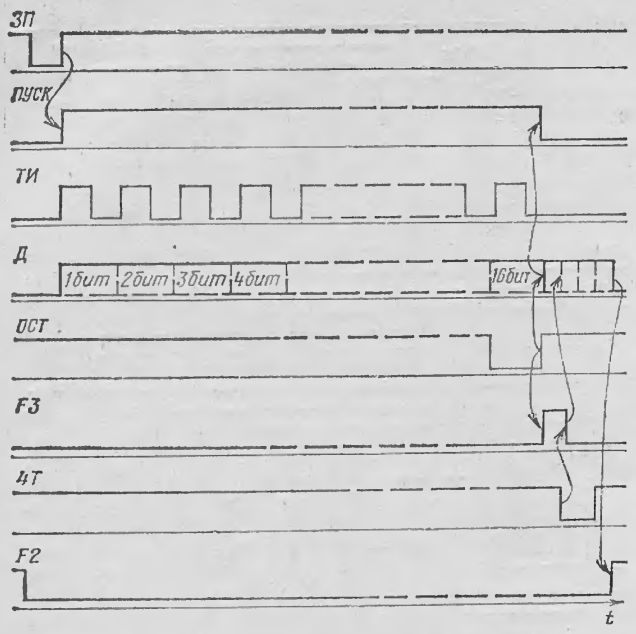


Рис. 4. Временная диаграмма передачи данных по общей магистрали.

Генератор тактовых импульсов (ГТИ) обеспечивает работу БА и таймера, вырабатывает последовательность импульсов сопровождения битов передаваемого слова в устройстве, захватившем ОМ.

Таймер (Т) формирует сигнал зависания при отсутствии передачи данных по захваченной ОМ за определенное время, фиксирует в ЭВМ-источнике факт отсутствия установки канала связи с адресуемой ЭВМ, вызывая требование прерывания ТРБ2. Сигнал зависания сбрасывает триггер запроса на захват ОМ (D1.2).

Этапы межмашинного обмена:

выдача требования захвата ОМ (ЗПР);

ожидание захвата ОМ (при этом ЭВМ может решать другие задачи);

выдача адреса ЭВМ-приемника (приемников) информации;

прерывание работы ЭВМ-приемников с переводом в режим приема пакета данных с ОМ;

передача пакета из ЭВМ-источника в ЭВМ-приемники;

освобождение ОМ.

Работа системы по передаче слова показана на рис. 4.

ЭВМ-источник начинает передачу пакета по прерыванию от БА, а ЭВМ-приемник осуществляет прием по прерыванию от УУКС, вырабатываемому при адресации устройства.

Передача данных через ОМ возможна в двух режимах: в синхронном — сигналы подтверждения установки канала связи с адресуемыми ЭВМ, в асинхронном дополнительно передаются и анализируются сигналы правильности приема каждого слова. ЭВМ-приемник может выдавать пакет-квитанцию о правильности передачи по адресу, указанному в информационном пакете. Для выдачи пакета-квитанции требуется захват ОМ. Для считывания информации одной ЭВМ из другой последней передается пакет-запрос с указанием имени и адреса передачи.

Введение режима передачи пакета данных через ОМ по собственному адресу («режим шлейфа») упрощает отладку и диагностику системы.

Устройство выдает два запроса на прерывание (ТРБ1, ТРБ2) и два вектора прерывания. Прерывание по ТРБ1 идентифицируется ЭВМ программно по признакам F1 и F4. Требования прерывания могут быть программно замаскированы.

Использование в каждой микроЭВМ двух предлагаемых устройств позволяет дублировать ОМ. При этом возможно разрешение работы ЭВМ по одной или двум ОМ в режиме разделения времени (по двум ОМ — при адресации к ней с разных магистралей или с одной из ОМ к ЭВМ, передающей данные по другой ОМ).

Конструктивно устройство занимает половину платы микроЭВМ «Электроника 60», содержит одну БИС К1801ВП1-033 и 23 микросхемы малой и средней степени интеграции. Имеется вариант конструктивного выполнения устройства в виде платы размерами 110×170 мм и массой 0,15 кг. Разработаны программы поддержки межмашинного обмена.

Экспериментальные исследования показали стабильную работу многомашинных систем при удалении абонентов на расстояние 500 м (среда распространения сигналов — витая пара, скорость обмена 1 Мбит/с).

Предложенное устройство* может найти широкое применение при построении многомашинных вычислительных систем для децентрализованного управления, например, электронными АТС.

* А. с. 1339576 СССР. Устройство для сопряжения ЭВМ с общей магистралью / В. А. Богатырев, Л. С. Иванов. — Опубл. 1987, Б. И. № 35.

Телефон 555-24-40, Ленинград (после 19 ч.),
Богатырев Владимир Анатольевич

Статья поступила 9 апреля 1987

УДК 681.324

В. А. Богатырев, А. Ю. Куконин, Л. С. Иванов

КОНТРОЛЛЕР МНОГОМАШИННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ С ОБЩЕЙ МАГИСТРАЛЬЮ

Высокой надежностью, ремонтопригодностью и возможностью наращивания при простоте реализации обладают многомашинные вычислительные системы с конфигурацией типа «Общая магистраль». Производительность таких систем определяется пропускной способностью, повысить которую удается, используя буферизацию данных, передаваемых при межмашинном обмене. Передача информации между ОЗУ абонентов системы осуществляется через блоки буферной памяти, находящиеся в каждом контроллере системы.

Такой способ позволяет свести до минимума простои общей магистрали и делает эффективной работу системы с двумя и более магистралями при использовании каждым абонентом соответствующего числа контроллеров [1].

Предлагаемый контроллер предназначен для объединения в многомашинную вычислительную систему до 16 микроЭВМ «Электроника МС 1201.01» при обмене 8-разрядными словами, сгруппированными в пакеты. Длина пакета может быть произвольной в пределах, задаваемых объемами блоков буферной памяти. После передачи каждого пакета контроллер освобождает магистраль, предоставляя возможность передачи другим абонентам системы.

Инициатором обмена является ЭВМ-источник, которым может быть любая микроЭВМ системы. Адреса ЭВМ-приемника и источника информации передаются по общей магистрали в заголовке каждого пакета. В последнем слове находятся данные об объеме передаваемой информации.

Контроллер (рис. 1) позволяет осуществлять передачу по общей магистрали пакетов объемом до 16 байт одной ЭВМ, группе машин или всем машинам системы одновременно.

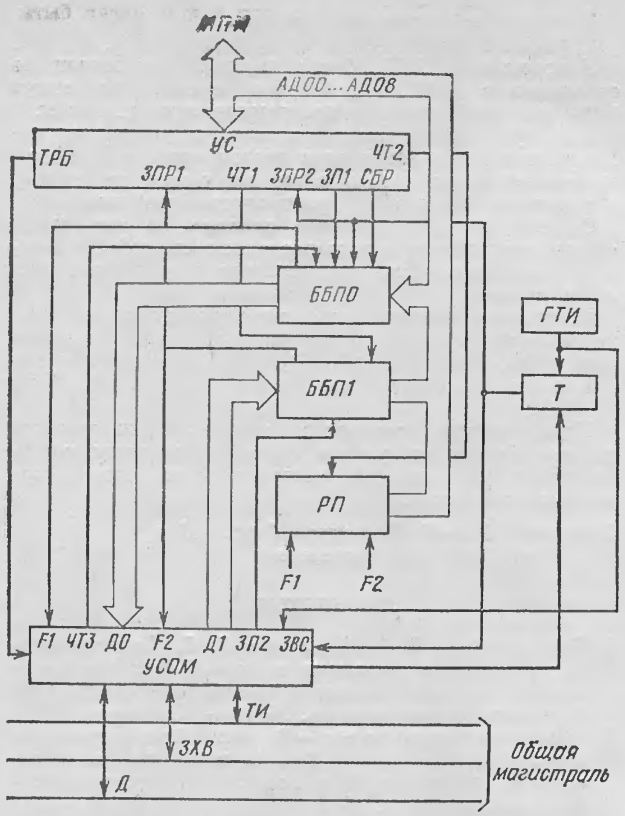


Рис. 1. Структурная схема контроллера:

УС — узел сопряжения; ББП0, ББП1 — блоки буферной памяти передатчика и приемника; РП — регистр признаков; УСОМ — устройство сопряжения с общей магистралью; ГТИ — генератор тактовых импульсов; Т — таймер

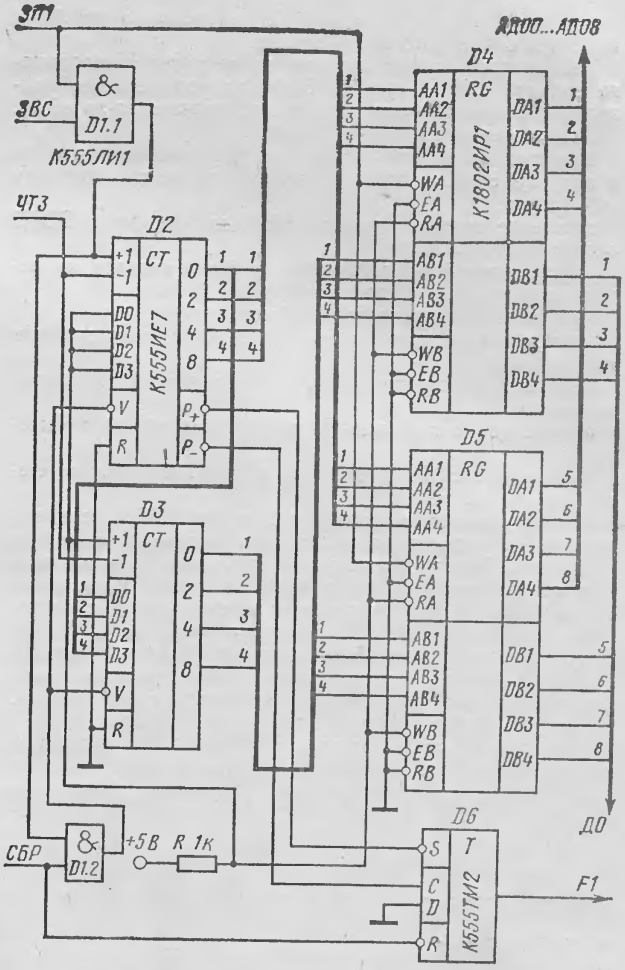


Рис. 2. Блок буферной памяти передатчика

Общая магистраль состоит из трех линий. Линия данных (Д) обеспечивает последовательную передачу слов пакета и сигналов подтверждения каждого слова в блок буферной памяти контроллера, принимающего пакет. При широковещательной и групповой передаче данных сигнал подтверждения формируется по «монтажному ИЛИ» при чтении всеми адресуемыми контроллерами. Линия синхронизации (ТИ) передает тактовые сигналы, вырабатываемые контроллером, захватившим общую магистраль (при передаче каждого слова пакета); линия арбитража (ЗХВ) определяет занятость общей магистрали.

Узел сопряжения обеспечивает подключение к каналу микроЭВМ до четырех контроллеров по интерфейсу МПИ; выполнен на БИС К1801ВП1-033, работающей в режиме программируемого параллельного интерфейса (ППИ).

Блоки буферной памяти построены по принципу «первый пришел, последний обслужен» на базе быстродействующих двухканальных СОЗУ К1802ИР1 [2], содержат счетчики адреса и триггеры, определяющие занятость блоков буферной памяти (рис. 2).

Запись в ББП0 происходит с канала А микросхем D4, D5 по адресу, определяемому содержимым счетчика D2. Данные для передачи через общую магистраль считываются с канала В микросхем D4, D5 по адресу на выходах счетчика D3. Триггер D6 характеризует состояние памяти ББП0.

Регистр признаков обеспечивает программный анализ ЭВМ состояния блоков буферной памяти своего контроллера (признаки F1 и F2).

Устройство сопряжения с общей магистралью (УСОМ) предназначено для захвата общей магистрали, передачи и приема данных, адресации контроллера с общей магистралью; включает в себя регистр данных, блок децентрализованного арбитража, дешифратор адреса, счетчик битов, формирователи импульсов, шинные формирователи.

Регистр данных построен на основе 8-разрядного регистра сдвига К155ИР13, обеспечивающего преобразование параллельного кода данных в последовательный и наоборот.

При адресации ОМ четыре младших разряда адресного слова определяют позиционно номера контроллеров-приемников внутри группы, также указанной позиционно в четырех старших разрядах адресного слова. Такой дешифратор адреса позволяет адресовать 8-разрядным словом 16 абонентов (4 группы по 4 абонента) при индивидуальной, широковещательной и групповой связях.

Формирователи импульсов, выполненные на основе микросхемы К155АГ3, вырабатывают импульс чтения информации из ББП0 (ЧТ3) и записи в ББП1 (ЗП2).

В качестве шинных формирователей используются передатчики КР559ИП4 и приемники КР559ИП5. Импульсы синхронизации от генератора тактовых импульсов подаются на вход счетчиков блока арбитража и тай-

мера постоянно, а на линии ТИ — только в моменты передачи очередного слова пакета.

Таймер служит для защиты от непрограммированного захвата ОМ и выхода ЭВМ из подпрограмм поддержки межмашинного обмена, возможного из-за сбоев.

Этапы межмашинного обмена:

проверка занятости ББП0 (по программному чтению и анализу ЭВМ-источника соответствующего разряда РП (F1));

занесение пакета данных ЭВМ-источником в ББП0 своего контроллера;

выставление ЭВМ-источником сигнала запроса на передачу пакета по общей магистрали (ТРБ);

ожидание предоставления общей магистрали;

передача адресного слова контроллеров-приемников информации;

передача всего пакета в случае незанятости ББП1 приемников;

аппаратное освобождение источником общей магистрали;

прерывание ЭВМ-приемников и перевод в режим чтения пакета из ББП1.

При адресации приемнику с занятым ББП1 передачи данных не происходит и источник переходит в режим зависания. Через определенный интервал времени таймер контроллера-источника вырабатывает сигнал, освобождающий общую магистраль; восстанавливает состояние ББП0 и выдает прерывание по сигналу ЗПР2. Для передачи пакета ЭВМ-источник должен повторно выставить сигнал запроса (ТРБ); ЭВМ-приемник, получив прерывание по чтению, считывает слово пакета, хранящего информацию о длине этого пакета.

При чтении остальных слов ЭВМ-приемник программно уменьшает на единицу содержимое этого слова и осуществляет соответствующий контроль. Окончание приема пакета в ОЗУ можно определять по признаку F2.

Контроль правильности приема данных может быть реализован путем подсчета контрольных сумм.

У прерывания ЗПР1 более высокий приоритет, по сравнению с ЗПР2. Требования прерывания могут быть при необходимости программно замаскированы.

Для обеспечения самодиагностики и упрощения отладки системы в контроллер введен «режим плейфа», позволяющий передавать пакет по общей магистрали и принимать его в ББП1 контроллера-источника.

Система может эффективно работать по нескольким общим магистралям с использованием каждой ЭВМ соответствующего числа контроллеров в режиме разделения времени.

Контроллер конструктивно выполнен на плате микро ЭВМ «Электроника 60» с использованием 38 микросхем (БИС К1801ВП1-033, К1802ИР1 и микросхем малой и средней степени интеграции серий К555, К155, КР559).

Предложенный контроллер может найти широкое применение для построения быстродействующих многомашинных управляющих вычислительных систем различного назначения.

Телефон 555-24-40, Ленинград (после 19 ч.),
Богатырев Владимир Анатольевич

ЛИТЕРАТУРА

1. Богатырев В. А., Иванов Л. С. Децентрализация управления системами связи.— В сб.: Автоматизированные системы декаметровой радиосвязи. Сер. Радиоэлектроника и связь, № 12.— М.: Знание, 1986.
2. Микропроцессоры.— В кн.: Архитектура и проектирование микроЭВМ. Организация вычислительных процессов / Под ред. Л. Н. Преснухина.— М.: Высшая школа, 1986.

Статья поступила 11 июня 1987

681.325.5+681.326

П. В. Бух-Винер

СЕТЬ МИКРОЭВМ В РЕЖИМАХ РЕАЛЬНОГО ВРЕМЕНИ

Необходимость сети ЭВМ для решения задач реального времени

При разработке и внедрении каждого комплекса из нескольких микроЭВМ для управления процессами реального времени (РВ) оттачивается и выходит на алгоритмический уровень концепция управления данным объектом РВ. Независимо от этого процесса и параллельно с ним на более мощных многопользовательских мини-ЭВМ накапливаются в виде баз данных и пакетов прикладных программ научные, технические и другие сведения для оценки и прогноза поведения изучаемых объектов РВ.

Внедренные микроЭВМ РВ успешно управляют объектом, но не имеют возможностей для использования накопленных в мини-ЭВМ сведений. Возникает задача объединения их в сеть [1, 2].

Основные свойства традиционной сети ЭВМ в режиме разделения времени

Многопользовательские мини-ЭВМ (МПЭВМ) работают в режиме разделения времени и ресурсов (РВР) между пользователями терминалов. Замена дисплея на ПК [3] приводит к образованию сети ЭВМ (рис. 1) типа «звезда» [4]. Самый распространенный интерфейс в такой сети — телеграфный стык «Токовая петля» на 9,6 Кбод, так как он обеспечивает простоту монтажа и достаточное расстояние между главной МПЭВМ и удаленными микроЭВМ. Для такого способа соединения в состав системного программного обеспечения (ПО) обеих ЭВМ введены программы под-

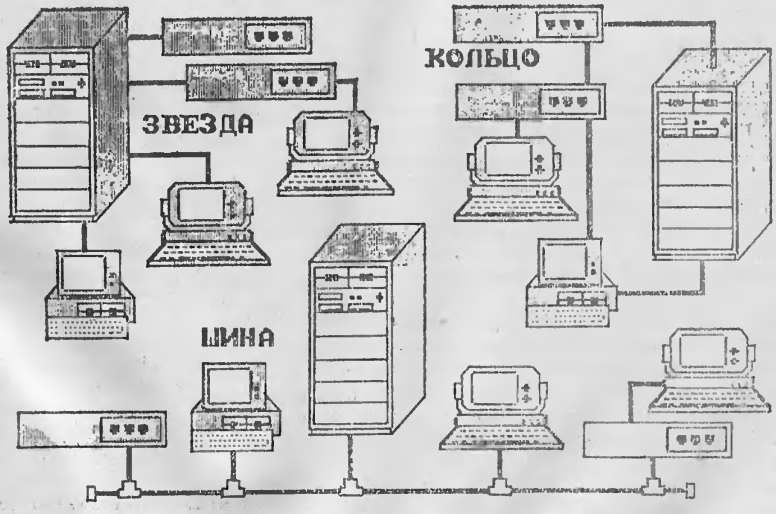


Рис. 1. Возможные соединения ЭВМ в сеть

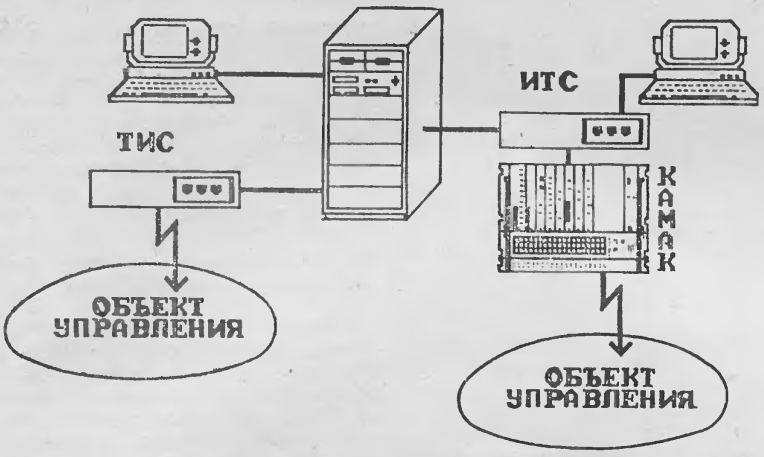


Рис. 2. ТИС и ИТС — управляющие микроЭВМ реального времени

держания протокола обмена по линии связи.

Кроме того, ПО ПК должно обеспечивать режим прозрачности и эмуляцию терминала главной ЭВМ. Терминал иногда эмулируется и аппаратно (ДВК-1). В обоих случаях оператор сетевого интеллектуального терминала должен при эксперименте на своем ПК создать файл для МПЭВМ, затем вызвать в своем ПК программу, поддерживающую протокол обмена по сети, и передать файл в МПЭВМ для продолжения обработки экспериментальных данных по программам МПЭВМ. Электронная почта и многооконный экран дополняют режим прозрачности [5 ... 7].

Как видно, для сетевого ПО в основном развиваются средства файлового обмена и диалога между операторами ПК сети. Проблема же автоматической синхронной работы программ в различных микроЭВМ и МПЭВМ остается в тени, хотя для задач РВ наиболее важно исключить оператора из стадии обмена по сети.

Особенности сопряжения с объектом реального времени

Попытки объединения в одной ЭВМ задач многопользования и РВ нередко терпят неудачу. Например, присоединение к СМ-4 аппаратуры КАМАК сводит на нет быстродействие последней либо запрещает коллективное пользование МПЭВМ. В описаниях мощных, класса VAX, мини-ЭВМ пользователя РВ предостерегают от попыток применять ЭВМ в качестве концентратора данных, а на платах связи с объектом РВ устанавливают перемычки запрета ПДП [8]. Сегодня, как правило, в МПЭВМ с режимом РВ коллективное пользование запрещено. Скорость, основной показатель РВ, существенно зависит от блочного обмена по каналу ПДП: запрета обработки стандартных аппаратных прерываний; запрета динамического распределения памяти; обеспечения наивысшего аппаратного и программного приоритета задач обслуживания аппаратуры РВ.

Понятно, что такие системы сильно специфичны и поэтому дороги. Их заменяют на микроЭВМ РВ. Однако ПО конкретной микроЭВМ РВ очень зависит от конфигурации и жестко ограничено размером памяти и скоростью конкретной микроЭВМ.

«Электроника 60». ДВК-1, КАМАК

Широкий набор плат в стандарте «Электроника 60» и модулей КАМАК позволяет собрать различные вычислительные установки РВ. На их основе реализованы терминальный исполнитель сети (ТИС) и интеллектуальный терминал сети (ИТС), разработанные в ИВНДиНФ АН СССР, (рис. 2).

Терминальный исполнитель сети—это минимальная конфигурация «Электроники 60» и модулей КАМАК. В ее состав кроме ЦП микроЭВМ и 54К памяти на ПЗ входят контроллер КАМАК и устройство последовательного обмена УПО1. При этом УПО1 адресуется, как консоль, и включена вместо дисплея в терминальный порт МПЭВМ. Желательно иметь перепрограммируемую постоянную память ПП2. В части КАМАК можно использовать АЦП712 на 40 кГц, таймер СТ/1, ЦАП10, счетчики, коммутатор аналоговых сигналов, мультиплексор и т. д. При наличии более совершенных плат «Электроники 60» (АЦП, ЦАП, БПФ и т. п.) использование дорогостоящего оборудования КАМАК не обязательно.

Интеллектуальный терминал сети—это ТИС с системной консолью. При этом УПО1 адресуется, как дополнительный порт для обмена по сети. Иначе, ИТС — это ПК с выходом в сеть и на объект РВ. Пользуясь последним определением, удобно применить для построения ИТС однoплатную микроЭВМ «Электроника НМС» из ДВК-1, имеющую два (консоль, сеть) необходимых порта, полную память и гнездо для микросхемы К573РФ3 (ППЗУ). Очень важное для РВ свойство ДВК-1 (разнесенная регенерация памяти) позволяет достоверно оцифровывать быстрые

```
@LSI                                  -вызов кроссассемблера в терминальной сессии МПЭВМ
 HOST-EL60-ASSEMBLER    VERSION 4/19.12.84
LDN(1,48,49,50,54):1                  -системный номер ТИС/ИТС в МПЭВМ
$H                                    -подсказка команд кроссассемблера
 A-ASSEMBLY ANY EL60 PROGRAM
 L-LOAD TO EL60
 !-SWAPPER: EL60-TERMINAL
 E-EXIT
 H-HELP
$A                                    -ассемблировать исходный текст для ТИС/ИТС
INPUT-FILE:(-F:SYS)-FIG-3
LIST-FILE:

  1> 000000: 000000 000000 000000      .=1700 ; стартовый адрес и голова модуля
  2> 001700: 000240 000000 000000  CHANL: NOP; резерв входного параметра
  3> 001702: 000240 000000 000000  CODE:  NOP; резерв параметра результата
  4> 001704: 013737 001700 160070  START: MOV   @#CHANL,@#160070;задать канал
  5> 001712: 022737 002000 160074  ADCTST:CMP   #2000,@#160074  ;конец прео-
  6> 001720: 001374 000000 000000         BNE   ADCTST          ;бразования?
  7> 001722: 013737 160072 001702         MOV   @#160072,@#CODE ;снять число
  8> 001730: 005767 177746 000000         TST   CODE            ;протирка знака
  9> 001734: 100005 000000 000000         BPL   PLUS
 10> 001736: 042737 100000 001702         BIC   #100000,@#CODE
 11> 001744: 005437 001702 000000         NEG   @#CODE
 12> 001750: 000000 000000 000000  PLUS:  HALT
SYMBOLS TABLE:
     2> 001700: CHANL        3> 001702: CODE         4> 001704: START
     5> 001712: ADCTST      12> 001750: PLUS
OCTAL SIZE: 001700-001750
 DEC.CELLS=    20
$LOAD ADDR.=001700B, SIZE=  40 BYTES      -загрузка модуля в ТИС/ИТС
$E                                        -выход из кроссассемблера
PRODUCED BY FROLOV'S LABORATORY 31.12.1983
@
```

Рис. 3. Фрагмент терминальной сессии МПЭВМ при работе с кроссассемблером. Пример программы ТИС РВ опроса АЦП

```
@FTN
 FORTRAN COMPILER FTN-2090G
$COM (-F:SYS)-FIG-4,1,100
    1*          SUBROUTINE TEST(LDN,CHANL,CODE)
    2*          INTEGER OUT,START,CHANL,CODE
    3*          IN=1700B; OUT=1702B; START=1704B  -согласование адресов
    4*          CALL CELL(LDN,IN,ICHAR('/'))      -открыть ячейку
    5*          CALL CELL(LDN,CHANL,13)           -поместить CHANL
    6*          CALL CELL(LDN,START,ICHAR('G'))   -исполнить с адреса START
    7*          CALL CELL(LDN,CODE,ICHAR('/'))    -взять результат АЦП в CODE
    8*          CALL CELL(LDN,0,13); END          -закрыть ячейку

8 LINES COMPILED . OCTAL SIZE=    166 .
CPU-TIME USED IS 0.3 SEC.
$EX
@
```

Рис. 4. Фрагмент терминальной сессии МПЭВМ при работе с текстом управляющей программы на языке Фортран-77 для программы на рис. 3

(до 32 кГц) аналоговые сигналы [9 ... 10].

Программирование и работа исполнителя сети

Конструкция ТИС позволяет пользователю любого терминала сети зарезервировать один (несколько) ТИС, поместить в них программы РВ в виде абсолютных модулей и управлять из РВР-программы МПЭВМ этими модулями. Программа в МПЭВМ — головная, написана на языке Фортран-77. Для обеспечения такого стиля программирования написан кроссассемблер языка «Электроники 60». Предлагаемую версию использовать просто: она не зависит от конкретной МПЭВМ (поскольку написана на языке Фортран-77). Подготовленный в текстовом редакторе МПЭВМ текст программы для ТИС — это входные данные кроссассемблера. Ассемблирование выполняется в один проход.

Листинг полученного абсолютного модуля выдается на любое устройство МПЭВМ с диагностикой ошибок и таблицей меток. Затем по команде пользователя кроссассемблер загружает полученный модуль в ТИС. Не выходя из кроссассемблера, пользователь может устанавливать свой терминал в режим консоли ТИС и отлаживать программу ТИС автономно. Каждая программа ТИС должна заканчиваться командой HALT (рис. 3).

Для обмена с программой ТИС предлагается программа на языке Фортран-77. Она позволяет на данный логический адрес ТИС послать целое число и управляющий символ языка пультового терминала микроЭВМ «Электроника 60» (рис. 4). Тогда головная программа пользователя для управления ТИС должна состоять из последовательности директив пульта ТИС, передаваемых в эту программу. Программа CELL устроена по принципу «вопрос — ответ», т. е. после передачи очередной директивы в ТИС эта программа ждет ответа ТИС. Таким образом, синхронизируется работа двух программ в разных ЭВМ. Ответ ТИС помещается на место второго входного параметра этой программы. Кстати, сам кроссассемблер пользуется этой программой. Вид программы зависит от конкретной МПЭВМ и от ее системного ПО. Если данный ТИС работает все время по одному набору программ, то целесообразно записать эти программы в ППЗУ ТИС. При использовании ППЗУ типа ПП2 это можно сделать во время загрузки кроссассемблером абсолютного модуля. Именно так записывается системное ПО в ПП2 для ИТС. Такая технология во многом подобна использованию внутрисхемных эмуляторов для отладки записываемых в микропроцессоры контролирующих программ.

Программирование и работа интеллектуального терминала сети

Основная трудность этого режима — необходимость сочетать работу двух программ разных ЭВМ с обслуживанием пультового терминала ИТС. Для этого создана особая операционная система (для ИТС). Сегодня она существует в минимальном виде (не поддерживает стандартных для ПК дисков и принтера) и занимает менее 1К ПП2. Поэтому назовем ее монитором.

Монитор осуществляет все режимы ТИС. Кроме того, он практически не требует специальных навыков от опе-

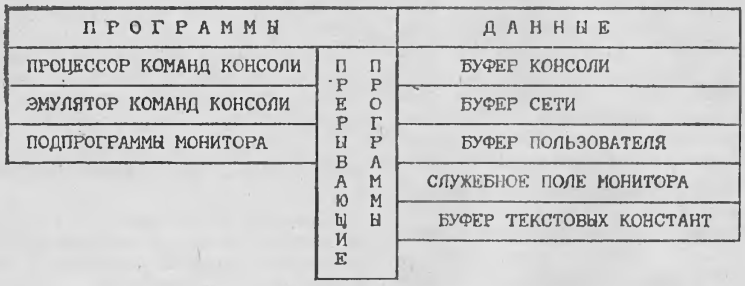


Рис. 5. Основные блоки монитора ИТС

```
      .=140000; стартовый адрес монитора
       .GLOBL ; объявление системных меток
   CMAX=100   ; размер консольного буфера
   NMAX=100   ; размер буфера обмена с сетью
  STACK=1546  ; стек обслуживания прерываний монитора
  LENTH=1550  ; длина реплики оператора без  ВК
 ANSWER=1552  ; указатель реплики оператора
   LAST=1700  ; конец буфера пользовательских данных
   DATA=137000; указатель буфера пользовательских данных
  STORE=137002; указатель копий общих регистров при прерывании.
 CBUFST=137022; указатель буфера консоли
 NBUFST=137124; указатель буфера обмена с сетью
  START=137720; стартовый адрес загруженной программы пользователя
BRANCH=137722; адрес исполняемой программы
 TOTAL=137724; контрольная сумма входного блока данных
CFYLLE=137726; указатель заполнения          буфера консоли
CFETCH=137730; указатель выбора символов из буфера консоли
NFYLLE=137732; указатель заполнения          буфера консоли
NFETCH=137734; указатель выбора      данных из буфера обмена с сетью
 LOADR=137736; указатель выбора прог.кодов из буфера обмена с сетью
  HOLD=137740; указатель выбора  символов из буфера обмена с сетью
 ADRES=137742; адрес загрузки программы пользователя
  CELL=137744; число из буфера обмена с сетью
  CHAR=137746; управляющий символ из буфера обмена с сетью
 IHEAD=137750;       указатель входного блока данных
 ISIZE=137752;       размер   входного блока данных
 ISTEP=137754; шаг заполнения входного блока данных
 OHEAD=137756;       указатель выходного блока данных
 OSIZE=137760;       размер   выходного блока данных
 OSTEP=137762; шаг заполнения выходного блока данных
 POINT=137764; число измерений по всем каналам
 CHANL=137766; количество каналов
 SINCH=137770; битовая маска синхроимпульсов по каналам
 DELAY=137772; шаг между двумя измерениями
SYMBOL=137774; код последней нажатой клавиши
  HOST=137776; статус монитора
```

Рис. 6. Служебное поле и распределение памяти монитора ИТС

ратора ИТС. Программа монитора убыстряет исполнение задач по сравнению с ТИС. Монитор использует систему прерываний микроЭВМ, без изменений работает с ДВК-1 НМС и внешне реализует только режим прозрачности. Он состоит из командного процессора (рис. 5), эмулятора команд консоли, системных подпрограмм и двух прерывающих программ (драйверов) консоли и сети. Для каждого драйвера выделяется буфер обмена, хранящий данные во время исполнения других программ ИТС. Эмулятор команд консоли обеспечивает исполнение команд пультового терминала микроЭВМ «Электроника 60», поступающих из интерфейса, не являющегося консолью (из сети). Кроме того, он исполняет дополнительные команды блочного обмена по линии связи. Монитор поддерживает выдачу на консоль ИТС текстовых констант по команде программного прерывания ввода-вывода IOT. Поскольку пользователь имеет полный текст монитора, то можно дописать свои программы или видоизменить имеющиеся. Кроссассемблер помнит все метки монитора как глобальные, поэтому пользователь может ссылаться на них в своей программе ИТС по наименованию (рис. 6). При сборке ИТС монитор надо занить в ППЗУ ПП2 или К583РФ3 с адреса 140000, пользуясь кроссассемблером и схемой соединения ТИС. После этого монитор готов к работе.

Пример. Рассмотрим пакет программ автоматизации научных исследований с помощью ИТС. Пусть программа на Фортране МПЭВМ должна анализировать данные, получаемые от различных программ РВ в диалоге с оператором данного ИТС. В пользовательском буфере 1700—137000 рекомендуется размещать загружаемый модуль с адреса 1700 в сторону увеличения адресов, а данные с объекта РВ — с адреса 137000 в сторону убывания адресов. При загрузке абсолютного модуля монитор вычислит конечный адрес этого модуля и поместит его в служебное слово ADRES = 137742 (см. рис. 6).

При написании подпрограмм для ИТС сначала резервируют память под передаваемые параметры, затем перечисляют безусловные переходы к модулям, а далее располагают тексты самих модулей. Получив листинг кроссассемблера по такой схеме, удобнее искать восьмеричные адреса для ссылок в программе МПЭВМ (рис. 7). Все программы абсолютного модуля считаются подпрограммами монитора, поэтому необходимо заканчивать их командой возврата из подпрограммы RTS. Если при исполнении программа использует регистры R6 и R7, то их надо запомнить и потом восстановить. Для этого в мониторе выделено служебное поле, начиная с адреса STORE = 137002

```
 1> 000002: 000000 000000 000000          .=1700
 4> 001700: 000240 000000 000000  NCHNL: NOP; количество каналов
 5> 001702: 000240 000000 000000  FIRST: NOP; номер первого канала
 6> 001704: 000167 000010 000000         JMP CLEAR; инициализация КАМАК
 7> 001710: 000167 000014 000000         JMP  SHOW; просмотр на осциллографе
 8> 001714: 000167 000026 000000         JMP START; сканирующая программа
 9> 001720: 012737 001000 166000  CLEAR: MOV #1000,@#166000; init CAMAC
10> 001726: 000207 000000 000000         RTS R7; возврат в монитор
11> 001730: 012737 000020 166000   SHOW: MOV #20,@#166000;      присоединение
12> 001736: 013737 001700 166640         MOV @#NCHNL,@#166640; нужного канала
13> 001744: 000207 000000 000000         RTS R7; возврат в монитор
14> 001746: 106427 000340 000000  START: MTPS #340          ;запрет прерываний
15> 001752: 010637 137016 000000         MOV R6,@$STORE+14;указатель стека
16> 001756: 012700 002700 000000         MOV #DATA,R0;голова пользоват.буфера
17> 001762: 000000 000000 000000  ; текст сканирующей программы.....
18> 001762: 013706 137016 000000         MOV @$STORE+14,R6;указатель стека
19> 001766: 106427 000000 000000         MTPS #0       ;разрешение прерываний
20> 001772: 000207 000000 000000         RTS R7; возврат в монитор
SYMBOLS TABLE:
    2> 002700= DATA          3> 100000= COUNT         4> 001700: NCHNL
    5> 001702: FIRST         9> 001720: CLEAR        11> 001730: SHOW
   14> 001746: START
OCTAL SIZE: 001700-001772
DEC.CELLS=   29
```

Рис. 7. Фрагмент пакета программ ИТС для обслуживания крейта КАМАК

(рис. 6). Особенно быстрые измерения требуют запрета прерываний при опросе крейта КАМАК. После завершения такой программы прерывания необходимо вновь разрешить. Во время работы без прерываний оба драйвера монитора не работают.

Программирование МПЭВМ для ИТС почти не отличается от варианта ТИС, но за счет наличия монитора получается дополнительная возможность блочного обмена с ИТС. Предлагаемая программа (рис. 8) содержит вызов стандартной программы BLOCK, позволяющей послать одномерный целый массив Фортрана в память ИТС и обратно. При этом надо указать начальный адрес, размер и шаг между соседними числами в памяти ИТС.

Практическая реализация

Рассмотренная конфигурация сети создана в 1983 г. в Институте высшей нервной деятельности и нейрофизиологии АН СССР для автоматизации научных исследований. За прошедшее время для ИТС отлажен и постоянно действует пакет прикладных программ анализа электроэнцефалограмм, вызванных потенциалов мозга и нейронной активности. ИТС позволяет анализировать аналоговые сигналы с частотой опроса вплоть до 32 кГц. ТИС используется для испытания вновь создаваемых уникальных

```
SUBROUTINE CAMAC(NCHNL,NFIRST,IDATA)
      INTEGER IDATA(1),START;  START=1726B  -согласование параметров
      J=1700B; CALL CELL(1,J,ICHAR('/'))    -открыть ячейку NCHNL в ИТС
      J=NCHNL; CALL CELL(1,J,10)            -передать первый параметр
      J=NFIRST;CALL CELL(1,J,13)            -передать второй параметр
               CALL CELL(1,START,ICHAR('G'))-запустить сканирование АЦП
      CALL BLOCK(0,2700B,100,NCHNL,IDATA)   -получить снятый временной ряд
       END

SUBROUTINE OSCILLOGRAPH(CHANNEL)
      INTEGER CHANNEL,SHOW; SHOW=1722B
      J=1700B;  CALL CELL(1,J,ICHAR('/'))
      J=CHANNEL;CALL CELL(1,J,13)
                CALL CELL(1,SHOW,ICHAR('G')); END

SUBROUTINE INITIALIZATION
      INTEGER CLEAR;    CLEAR=1716B
           CALL CELL(1,CLEAR,ICHAR('G')); END

******** блочный обмен IO=0-ввод/1-вывод *********
      SUBROUTINE BLOCK(IO,ADRES,SIZE,STEP,MAS)
      INTEGER ADRES,SIZE,STEP,MAS(SIZE),IANS(8)
      L=ADRES     -восьмеричный адрес начала массива в ИТС
      N=SIZE*2    -количество обмениваемых ячеек (слов)
      K=STEP*2    -шаг между ячейками в ИТС (0-подряд)
  IF(IO.EQ.0)THEN;J=137756B
  ELSEIF(IO.EQ.1)THEN;J=137760B;ELSE;RETURN;ENDIF
CALL CELL(1,J,ICHAR('/'));CALL CELL(1,L,10);CALL CELL(1,N,10)
      IF(IO.EQ.0)THEN; CALL CELL(1,K,ICHAR('B'))
DO FOR I=1,SIZE;K=INCH(1);J=INCH(1);MAS(I)=IOR(ISHFT(J,8),K);ENDDO
      ELSE; CALL CELL(1,K,ICHAR('L'))
DO FOR I=1,SIZE;CALL OUTCH(1,MAS(I));CALL OUTCH(1,ISHFT(MAS(I),-8))
      ENDDO; ENDIF; END
```

Рис. 8. Фрагмент пакета программ ИТС для обслуживания крейта КАМАК

плат, моделирующих некоторые процессы высшей нервной деятельности на МЭВМ РВ.

Описанный в статье метод автоматического обмена использован при включении в общеинститутскую вычислительную сеть ПК типа IBM PC.

117865, Москва, ул. Бутлерова, д. 5А, ИВНДиНФ АН СССР; тел. 338-77-00

ЛИТЕРАТУРА

1. Громов Г. Р., Ширшиков Н. В., Литвиненко Л. А. Микромашинный комплекс для управления биотехнологическими объектами // Микропроцессорные средства и системы.— 1984.— № 2.— С. 54—59.
2. Соломенцев Ю. М. Возможности систем управления в гибком автоматизированном производстве // Микропроцессорные средства и системы.— 1984.— № 2.— С. 73—74.
3. Sniger P. Terminal emulation software links micros to mainframes.— Mini-Micro-Systems.— 1984. — Vol. 17.— N 13.— P. 183—192.
4. Estrin J. Hybrid Technologies rewrite the rules for local area networks.— // Mini-Micro-Systems. — 1985.— Vol. 18.— N 1.— P. 195—204.
5. Cruz S., Catchings B. KERMIT: A file transfer protocol for universities, part 1: Design considerations and specifications//BYTE.— 1984.— June.— P. 255—278.
6. Cruz S., Catchings B. KERMIT: A file transfer protocol for universities, part 2: States and transitions, heuristic rules and examples // BYTE.— 1984.— July.— P. 143—145, 400—403.
7. Zintgraff S., Barringer D D. Bremner F. J. Data transfer concepts and strategies in a psychological research environment//Behavior Research Methods, Instruments, & Computers.— 1986, Apr.— Vol. 18.— N 2.— P. 236—239.
8. CAMAC—DMA—ND100 General Information: ND12.004.1.-- 1974.— June.
9. Центральный процессор М2. ТО и инструкция по эксплуатации.— ЦНИИ «Электроника», 1979.
10. Набор КАМАК-2. ТО и инструкция по эксплуатации.— ЦНИИ «Электроника», 1980.

Статья поступила 17 ноября 1987

УДК 681.326

В. И. Жданов, В. Н. Бобылев, Н. Ф. Гринь, Т. Г. Уткина

АВТОНОМНЫЕ ОБУЧАЮЩИЕ УСТРОЙСТВА ДЛЯ ОМЭВМ СЕРИИ К1816

Однокристальные микроЭВМ (ОМЭВМ) серии К1816 [1, 2] все более широко применяются при создании недорогих микроконтроллеров 2 ... 4 Кбайт).

Для отладки программ и аппаратной части микроконтроллеров авторами разработаны простые автономные отладочные устройства, называемые часто «оценочными модулями» (ОМ). Они весьма эффективны 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, разрабатывать и отлаживать в реальном масштабе времени узлы РЭА и программное обеспечение для них. При проектировании ОМ авторы придерживались следующих принципов:

1. ОМ должны выполняться как однопроцессорные отладочные устройства, в которых попеременно выполняются программы монитора и пользователя. Это позволяет создавать недорогие отладочные устройства с минимальными для пользователя ограничениями на применение ресурсов ОМЭВМ.

2. Информация о состоянии внутренних регистров ОМЭВМ и памяти программ пользователя вводится (выводится) в машинных кодах с помощью клавиатуры и однострочного дисплея. Клавиатура должна иметь 16 клавиш для ввода шестнадцатеричных кодов команд и не более восьми функциональных клавиш для управления ОМ.

Однострочный дисплей обычно имеет 3—4 сегментных индикатора для адреса и два для индикации данных. Кодированная информация об ошибках оператора и неисправностях ОМ при необходимости выводится на индикаторы.

3. ОМ должны иметь каналы для обмена информацией с ПЭВМ и для подключения стандартных периферийных устройств (бытового магнитофона, печати и т. п.).

Периферийные устройства подключаются через специализированные платы-модемы. Наличие каналов позволяет использовать для разработки ПО кросс-средства пользователя.

4. ОМ должны выполняться в виде одноплатных конструкций с открытым корпусом, без встроенного источника питания (при этом резко уменьшаются габаритные размеры и масса устройств). Панель корпуса из цветного оргстекла служит светофильтром для индикаторов дисплея и раскрывает пользователю внутреннюю структуру.

Автономное учебно-отладочное устройство для ОМЭВМ К1816ВЕ48 модели «Электроника ОУ-48» [3] создано согласно этим четырем принципам.

Отладочное устройство «Электроника ОУ-49» используется для разработки, отладки и тестирования аппаратуры на основе ОМЭВМ КМ1816ВЕ39 (48 и 49) с последую-

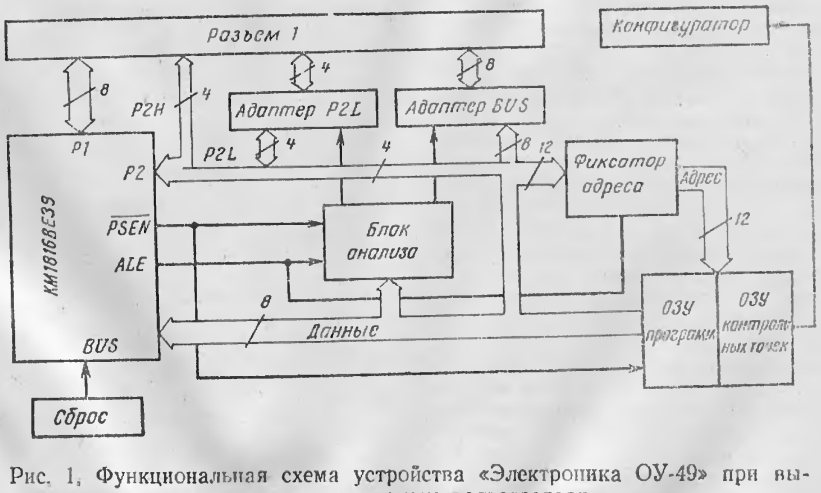


Рис. 1. Функциональная схема устройства «Электроника ОУ-49» при выполнении программы пользователя

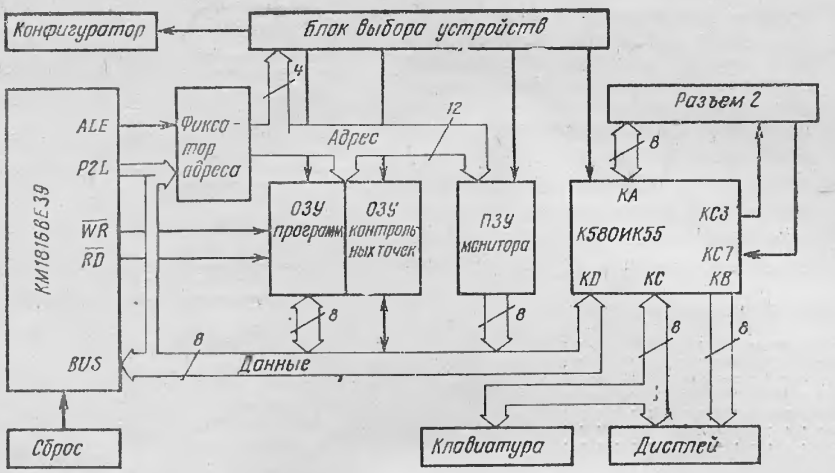


Рис. 2. Функциональная схема устройства «Электроника ОУ-49» при выполнении программы монитора

щей записью отлаженных программ во внутреннее СППЗУ ОМЭВМ или в СППЗУ серий К573.

В отладочном устройстве применена ОМЭВМ КМ1816ВЕ39, попеременно выполняющая программы пользователя и монитора (рис. 1 и 2).

ОЗУ устройства связано с ОМЭВМ шинами данных и управления, а через фиксатор адреса — с адресной шиной. С этой же шиной соединен параллельный интерфейс. Он обслуживает встроенную клавиатуру, дисплей, а также разъем параллельного канала связи с периферийными устройствами. ОЗУ контрольных точек предназначено для записи признака останова в адресах, где пользователь желает остановить отлаживаемую программу. Необходимые ОЗУ или интерфейс выбираются блоком выбора устройства в соответствии с сигналами, поступающими на него от логики управления.

Выводы ОМЭВМ непосредственно или через шинный формирователь соединены с разъемом эмуляции, посредством которого «Электроника ОУ-49» подключается к проектируемой системе вместо ОМЭВМ КМ1816ВЕ39 (49, 48) или к плате модуля программатора.

Устройство позволяет:

контролировать и изменять содержимое ОЗУ программ и внутренних регистров микроЭВМ;

выполнять программу пользователя в реальных адресах в пошаговом режиме;

осуществлять прогон программы пользователя в реальном масштабе времени с возможностью останова в контрольных точках;

эмулировать работу ОМЭВМ КМ1816ВЕ49 (48);

вводить информацию (в шестнадцатеричном коде) в ОЗУ программ (и выводить из него) из вычислительного комплекса типа «Электроника 60».

Устройство может записать отлаженную программу во внутреннее СППЗУ ОМЭВМ или ИМС К573РФ2 (РФ5) при подключении к разъему эмуляции модуля программатора.

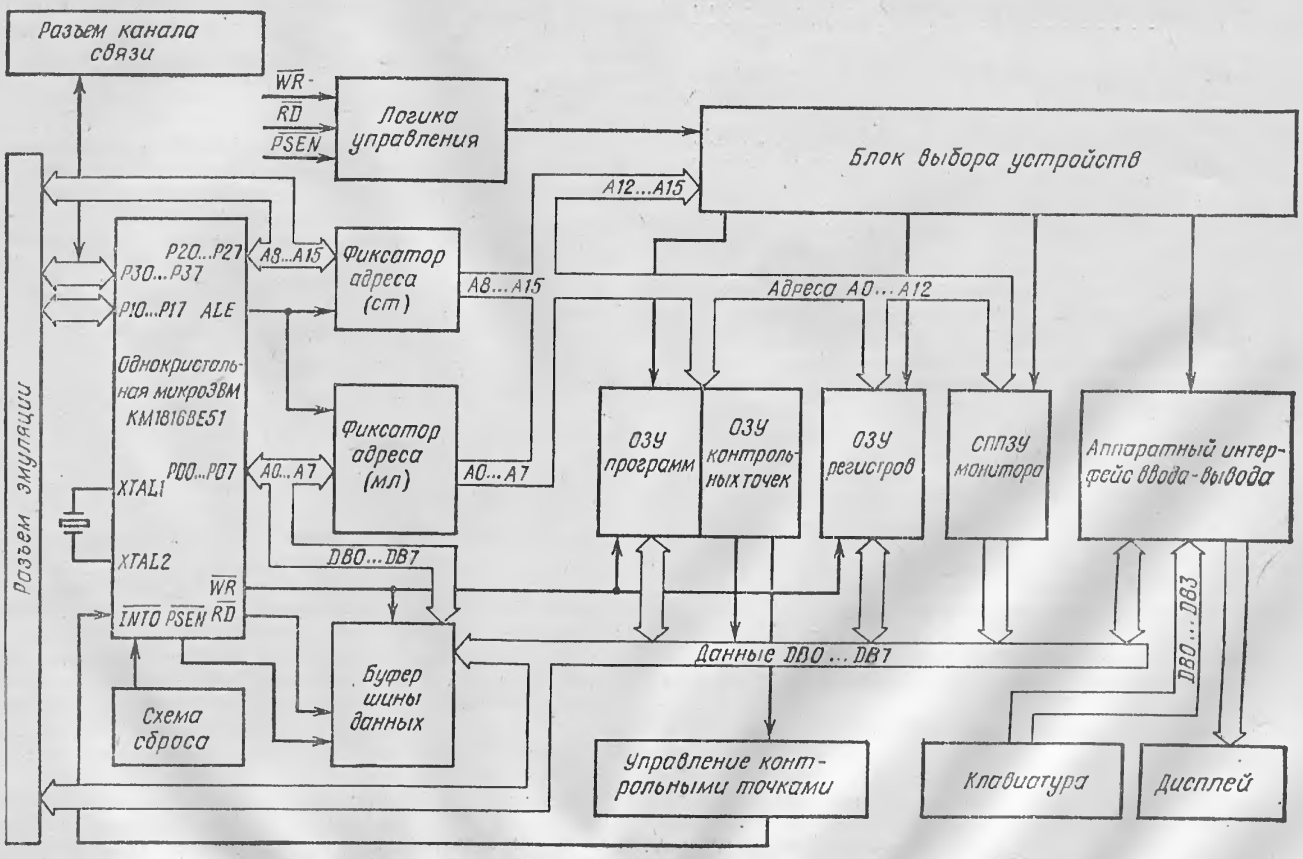


Рис. 3. Структурная схема оценочного модуля «Электроника ОМ-51»

«ОУ-49» выполнено в виде однопалатной конструкции со встроенной клавиатурой и светодиодным дисплеем. Никель-цинковый аккумулятор обеспечивает сохранность программ пользователя при отключенном питании.

Для организации параллельного асинхронного 8-разрядного канала связи с вычислительным комплексом служит 31-контактный разъем типа ГРПМ.

Технические характеристики „Электроники ОУ-49"

Разрядность	8
Объем ОЗУ, Кбайт	4
Объем СППЗУ монитора, Кбайт	2
Напряжение питания, В	5±3%
Потребляемый ток, А, не более	2
Габаритные размеры, мм	304×214×35
Масса, кг, не более	1,8

Оценочный модуль «Электроника ОМ-51» (ОМ-51) — автономное учебно-отладочное средство для МП-устройств. Основные функции ОМ-51 (рис. 3) реализуются с помощью резидентной программы-монитора, хранимой в ПЗУ (рис. 4).

«Электроника ОМ-51» предоставляет пользователю ручной ввод программы пользователя в виде машинных кодов в ОЗУ программ; индикацию адреса, данных и признаков контрольных точек на однострочном дисплее (6 знакомест); выполнение программы пользователя в реальном масштабе времени с остановом в контрольных точках и пошаговом режиме; ввод и запоминание параметров контрольных точек; вывод на дисплей содержимого внутренних регистров ОМЭВМ и возможность его модификации; контроль содержимого ЗУ по заданному адресу с возможностью коррекции содержимого ОЗУ; хранение программ пользователя на кассете бытового магнитофона; автоматический контроль и индикацию наличия ошибок при записи и считывании информации с магнитофона.

Технические характеристики „Электроники ОМ-51"

Разрядность	8
Объем ОЗУ, Кбайт	4
Объем СППЗУ монитора, Кбайт	2
Объем ОЗУ контрольных точек, Кбит	4
Дисплей	Буквенно-цифровой, однострочный, 6 знакомест
Скорость обмена информацией с внешним накопителем (магнитофоном), Бод	110/300
Напряжение питания, В	5±5%
Потребляемый ток, А, не более	2,5
Габаритные размеры, мм	304×214×35
Масса, кг, не более	1,5

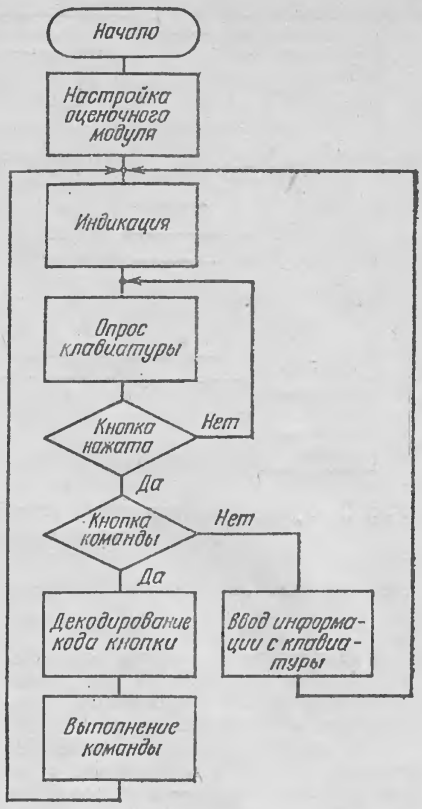


Рис. 4. Схема алгоритма программы монитора

Дополнительное ПО включает в себя программы-утилиты передачи и приема данных с контролем, проверки ОЗУ программ, ОЗУ регистров и ПЗУ монитора, переписи массива данных из одной области ОЗУ программ в другую область ОЗУ.

Телефон 468-13-70, Москва

Сообщение поступило 10 ноября 1987

ЛИТЕРАТУРА

1. Кобылинский А. В., Липовецкий Г. П. Однокристальные микроЭВМ серии К1816 // Микропроцессорные средства и системы.— 1986.— № 1.— С. 10—19.
2. Крылов Е. И. Однокристальные микроЭВМ серии К1814, К1820, К1816 // Микропроцессорные средства и системы.— 1985.— № 2.— с. 3—7.
3. Гутовец Н. И., Жданов В. И., Колодяжный В. Н. и др. Автономные обучающие устройства для отладки микроконтроллеров // Электронная промышленность.— 1986.— С. 8—10.

681.326.2

Б. И. Беляев

ДОПОЛНИТЕЛЬНЫЕ 7 КБАЙТ ОПЕРАТИВНОЙ ПАМЯТИ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА МС 1201.01»

МикроЭВМ «Электроника МС 1201.01» имеет ОЗУ емкостью 64 Кбайт, однако для программ пользователя доступно лишь 56 Кбайт, из-за того что 8 Кбайт адресного пространства отведены под встроенное ПЗУ, системное ОЗУ (СОЗУ) и регистры внешних устройств. Вместе с тем объем оперативной памяти, доступной для программ пользователя, можно увеличить до 63 Кбайт, оставив 1 Кбайт для СОЗУ и регистров внешних устройств. Область ПЗУ с программой ПУЛЬТ, резидентными тестами и загрузчиком с гибких магнитных дисков — программно-отключаемая, причем программа ПУЛЬТ остается работоспособной.

Для расширения оперативной памяти требуются два корпуса микросхем типа К155ТМ2 и К555ЛЕ4, которые устанавливаются в незадействованные посадочные места на плате микроЭВМ данного типа, и проводные соединения, выполненные по схеме, изображенной на рисунке.

Усовершенствование заключается в преобразовании сигнала LOCK, формируемого контроллером памяти К1801ВП1-030. Этим сигналом определяется источник читаемой информации (LOCK=0 — обращение к ОЗУ

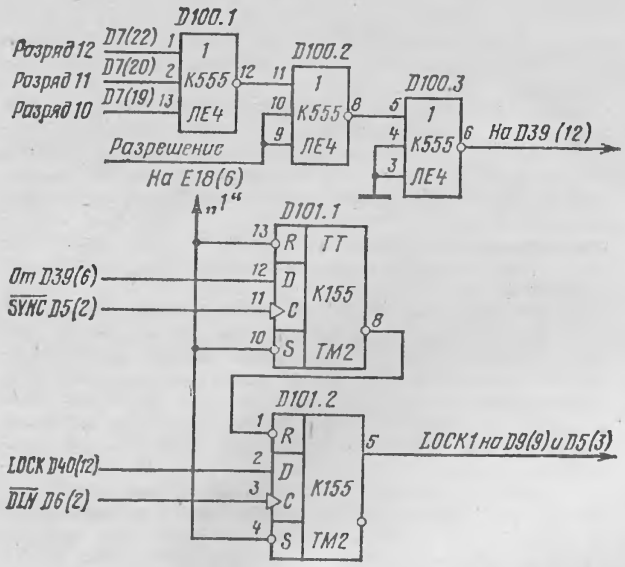


Схема введенного узла

п регистрам внешних устройств; LOCK=1 — обращение к встроенному ПЗУ). Введенный узел состоит из **управляемого дешифратора адреса**, выполненного на основе микросхемы К555ЛЕ4, учитывающей нагрузочную способность выходов микропроцессора, и **формирователя сигнала** LOCK1 на триггере К155ТМ2.

Управляемый дешифратор адреса подключается к мультиплексору выборки банков ОЗУ D39 (12) вместо резистора (матрица резисторов E18(6)), задающего в стандартном варианте «Лог. 1» на данном входе. Такая конструкция позволяет на выходе D39(6) данного мультиплексора формировать «Лог. 1» при обращении к адресам 160000 ... 175776 при условии, что вход управления адресным дешифратором РАЗРЕШЕНИЕ=0. Если РАЗРЕШЕНИЕ = 1, то реакция мультиплексора на эту группу адресов будет, как в непеределанном варианте. На остальные адреса он реагирует в зависимости от состояния переключателей управления банками ОЗУ. Такого включения достаточно, чтобы обращаться к скрытой области ОЗУ с адресами 170000 ... 172776 и 174000 ... 175776. Доступность ОЗУ с адресами 160000 ... 167776 зависит от состояния регистра начального пуска (РНП — 177716) процессора, а на адресах 173000 ... 173776 размещается загрузчик с гибких магнитных дисков, сосредоточенный во встроенном ПЗУ и всегда доступный для чтения.

Желательно, чтобы область ОЗУ, доступная для программного обращения, была непрерывной в адресном пространстве 160000 ... 175776. Для этого вместо сигнала LOCK формируется сигнал LOCK1, который принимает значение «Лог. 0» при обращении к этой группе адресов.

Формирователь сигнала LOCK1 состоит из двух триггеров. Первый из них тактируется умощненным сигналом $\overline{\text{SYNC}}$ процессора с D5(2) и фиксирует состояние мультиплексора выборки банков ОЗУ D39 (6), он же управляет сбросовым входом второго триггера, который синхронно с тактирующим умощненным сигналом $\overline{\text{DIN}}$ D6 (2) фиксирует состояние сигнала LOCK D40 (12). С выхода второго триггера снимается сигнал LOCK1, управляющий схемой формирования сигна-

Зависимость между адресами и состоянием переключателей

Положение SA 1.4	Положение SA 1.5	Адрес РКС	Адрес РД
1	1	177170	177172
0	1	177174	177176
1	0	177200	177202

лов ВВОД ПЗУ (D9.3) и ВВОД ОЗУ (D9.4), заложенной в плате микроЭВМ.

Сигнал РАЗРЕШЕНИЕ для управляемого дешифратора адреса формируется с помощью устройства интерфейса накопителя на гибких магнитных дисках (УИГМД) на основе БИС К1801ВП1-033, установленной на плате микроЭВМ. УИГМД можно использовать, если в данной микромашинной системе оно не было задействовано. Функции сигнала РАЗРЕШЕНИЕ выполняет сигнал ПУСК Н (D38 (6)) в ИГМД. Для работы необходимо в том же интерфейсе на вход ЗАВЕРШЕНО 2 Н подать «Лог. 0», т. е. вход D31 (3) соединить с шиной ОБЩИЙ.

Адреса регистров УИГМД в пространстве микроЭВМ зависят от положения переключателей SA 1.4, SA 1.5 на плате (см. таблицу). Состояние переключателя с крайним положением движка по стрелке в таблице обозначается 1, обратное — 0, причем состояние SA 1.4=0, SA 1.5=0 использовать не рекомендуется.

Предположим, что SA 1.4=1, SA 1.5=1, тогда обращение вида

MOV #1, @#177170; включение ОЗУ 160000 ... 175776, ... (текст программы)

MOV #40000, @#177170; отключение ОЗУ 160000 ... 175776; LOCK1=LOCK

вначале программы сбрасывает сигнал ПУСК Н (РАЗРЕШЕНИЕ = 0), а в конце устанавливает его (РАЗРЕШЕНИЕ = 1).

Можно добавить и аппаратное выключение режима работы с ОЗУ 160000 ... 175776, если, например, вместо сбросового сигнала на входе INIT данного УИГМД использовать сигнал останова D2 (6). Тогда при аппаратном останове можно работать с программой ПУЛЬТ. Следует отметить, что если в момент аппаратного останова шло обращение процессора к адресам 160000 ... 175776, то возможно прерывание по вектору 4, поскольку приоритет данного прерывания выше, чем у прерывания по останову. Но если вектор 4 инициализировать, то ничего фатального не произойдет. Такая ситуация редка, но вероятность ее появления отлична от нуля.

Приведенный способ формирования сигнала РАЗРЕШЕНИЕ не единственный. В статье рассмотрен лишь вариант управления частью скрытой области ОЗУ, 1 Кбайт адресов (176000..177776) которой отводится все же под адреса регистров внешних устройств и СОЗУ.

Предлагаемое усовершенствование можно применить и в микроЭВМ типа «Электроника Н МС 11100.1» и «Электроника МС 1201.02» с учетом логики их работы. Основное различие будет лишь в том, что на печатных платах упомянутых микроЭВМ отсутствуют незадействованные посадочные места для микросхем.

Телефон 196-97-53, Москва

Статья поступила 24 августа 1987

УДК 681.326—181.4

Я. П. Тененбаум, А. И. Туркот

ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В УПРАВЛЯЮЩЕМ ВЫЧИСЛИТЕЛЬНОМ КОМПЛЕКСЕ НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

При современном развитии энергосистем, росте мощностей, передаваемых по межсистемным линиям электропередач, чрезвычайно актуально обеспечить их надежную работу с помощью автоматических устройств на базе промышленных микроЭВМ. Подобные устройства отличаются бо́льшей надежностью, селективностью и точностью по сравнению с традиционными средствами противоаварийной автоматики.

При выборе технических средств подобного устройства следует учитывать, что устройство эксплуатируется в условиях вибрации, мощных электромагнитных полей; объем входной и выходной информаций (дискретной и аналоговой) — большой; время реакции устройства на внешние возмущения — 5...10 мс. Эти факторы определяют работу устройства в качестве программируемого контроллера без внешней памяти типа магнитных дисков и магнитных лент. По производительности, архитектуре, наличию устройств связи с объектом (УСО) микроЭВМ «Электроника 60» обеспечивает пользователей необходимыми вычислительными ресурсами.

Ряд особенностей микроЭВМ «Электроника 60» затрудняют использование в подобном режиме стандартного программного обеспечения (перфоленточной ОС, ФОДОС). Применить перфоленточную ОС невозможно, так как даже кратковременное нарушение энергоснабжения приводит к отказу устройства, а восстановление его работоспособности требует значительных затрат времени и вмешательства обслуживающего персонала.

Применение ФОДОС (или RM-монитора ОС RT-11 [1]) и ОС RSX-11S возможно, но эти ОС не учитывают особенностей распределения памяти «Электроники 60». Процессор типа М2 позволяет использовать расположенный на нем банк ОЗУ в двух режимах: либо с адреса 0, либо с адреса 20000. Однако с адреса 0 архитектурно расположены адреса векторов прерываний, что определяет использование для нулевого банка памяти X (адреса с 0 по 17776) ППЗУ. Таким образом, для эффективного использования возможностей процессора М2 принята «слоистая» структура памяти: ППЗУ — ОЗУ — ППЗУ и т. д. Применение же RT-11, RSX-11S не позволяет компоновать ПО, эффективно используя неоднородную структуру памяти.

В связи с этим разработано ПО для управляющего вычислительного комплекса (УВК) применительно к противоаварийной автоматике отключения нагрузки (САОН) энергосистем [2, 3].

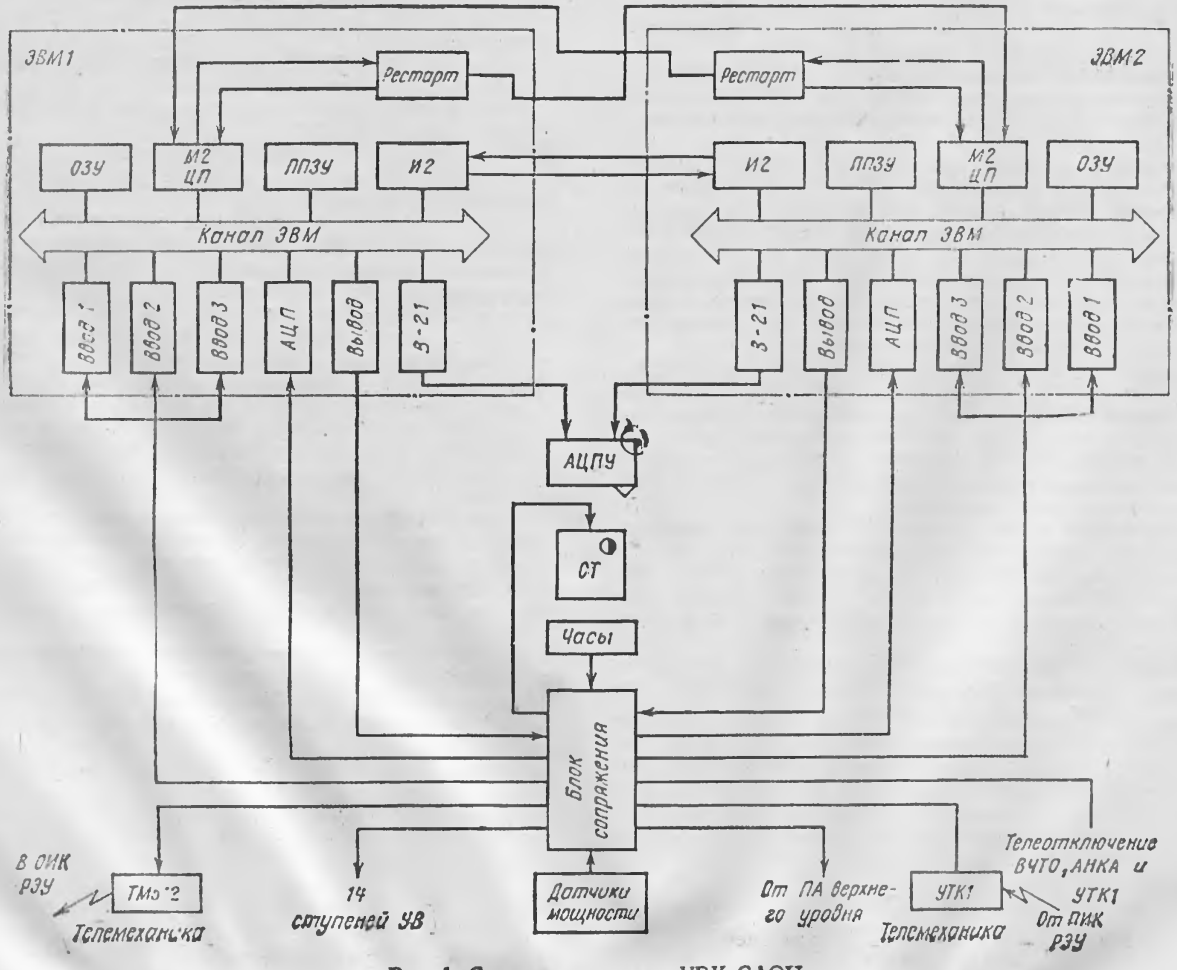


Рис. 1. Структурная схема УВК САОН

Для повышения надежности функционирования в состав УВК включены две микроЭВМ (рис. 1).

Программная поддержка работы УВК осуществляется разработанной и апробированной ОС ДКС-2 (двухкомпьютерная система версии 2). ОС разрабатывалась для функционирования в реальном времени при размещении в перепрограммируемом ПЗУ (ППЗУ). Все программы ОС написаны на языке МАКРО-11 мини-ЭВМ семейства СМ-3, СМ-4, СМ-1420 в ДОС СМ и ОС РВ.

Написание программ ОС на языках высокого уровня (Си, Фортран, Паскаль...) значительно ускорило бы разработку ПО, однако по сравнению с ассемблером это увеличило бы в 2—3 раза объем занимаемой памяти и, соответственно, ухудшило бы временные показатели системы в целом [4].

Характеристики ОС УВК САОН

Объем входной информации, сигнал	
дискретной	74
аналоговой	16
Объем выходной информации:	
дискретных управляющих воздействий (УВ), сигнал	14
сигналов, выполняющих вспомогательные функции	8
документов, выдаваемых на печать, вид	9

В системе функционируют 13 периодических задач с циклами от 20 мс до 1 мин и 20 апериодических задач, запускаемых на выполнение по инициативе других задач. Время реакции УВК от момента подачи импульса на вход до срабатывания выходных управляющих органов — 35...50 мс. ОС УВК САОН — однопроцессорная мультипрограммная система реального времени с произвольным числом задач двух приоритетов. Программируемый интервал времени запуска задач — 20 мс...20 мин с дискретностью 20 мс.

Пользователю предоставляется возможность организовать: выполнение периодических задач с заданным интервалом времени; постановку в очередь задач, готовых к выполнению, а также задач, ожидающих выполнения через заданный интервал времени; снятие задач из очереди.

Кроме того, в ОС организованы: буферирование выходных сообщений; автоматическое включение и отключение печатающего устройства; развитая диагностика как системных сбоев, так и всех устройств УВК; инициация системы (старт и рестарт) при включении питания УВК; системная библиотека стандартных подпрограмм и библиотека пользователя.

Состав ОС — ядро ОС, организующее инициацию системы, запуск задач в соответствии с их очередностью и приоритетом, обслуживание программных прерываний, диагностику системных ошибок, контроль за временем выполнения задач, обработку внешних прерываний; монитор ОС, организующий обмен информацией двух микроЭВМ УВК и реконфигурацию УВК при отказе какого-либо устройства; драйверы внешних устройств; блок диагностики и индикации неисправностей устройств комплекса технических средств УВК.

Подробное описание функционирования всех составных частей [1] ОС заняло бы слишком много места.

Монитор ОС выполняет нестандартные функции. В рассматриваемом УВК принята циклическая параллельная обработка данных в двух микроЭВМ с взаимной сверкой результатов в каждом цикле по принципу асинхронного дуплекса. Монитор ОС программно поддерживает межмашинный обмен (ММО) с ожиданием готовности «партнера» к обмену не более трех таймерных прерываний (60 мс). ММО организован (рис. 2) по прерываниям устройства параллельного обмена И2 с двумя физическими векторами прерываний (А и Б). Каждый вектор имеет свою программу обра-

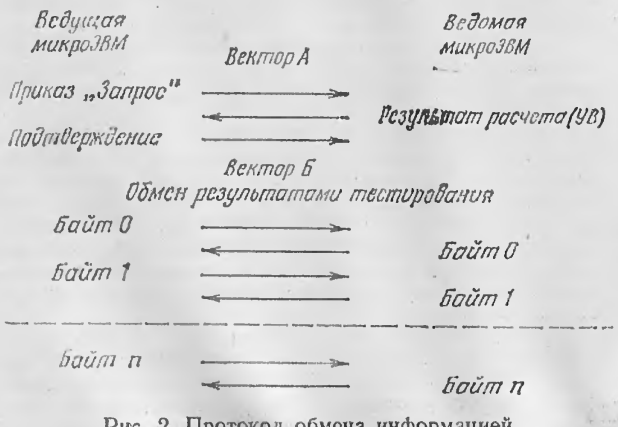


Рис. 2. Протокол обмена информацией между микроЭВМ

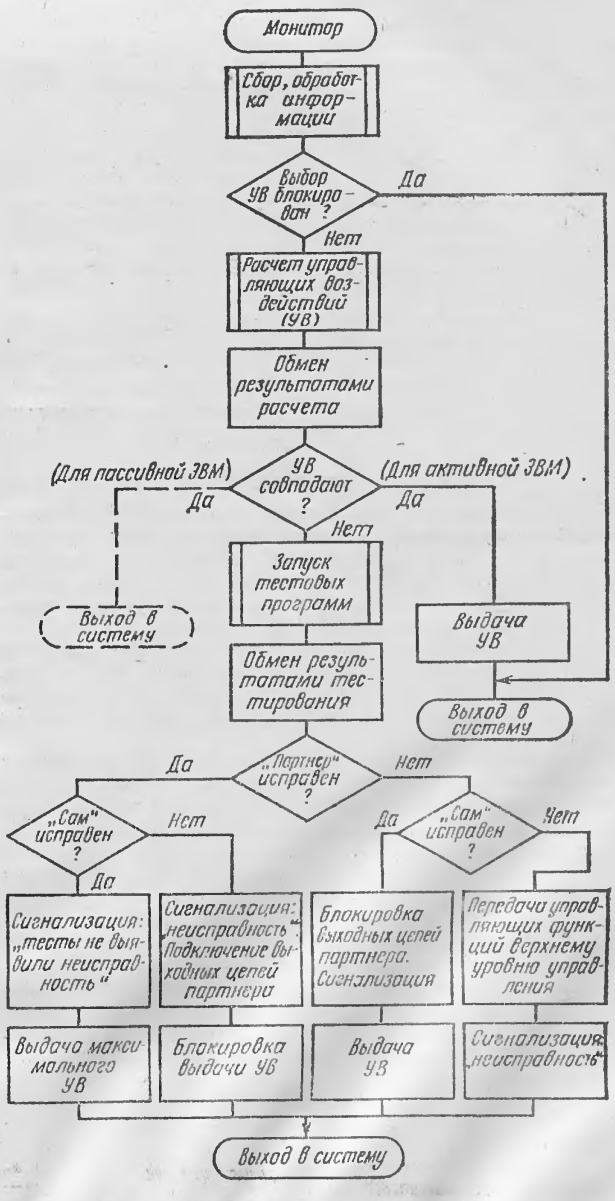


Рис. 3. Функциональная схема монитора

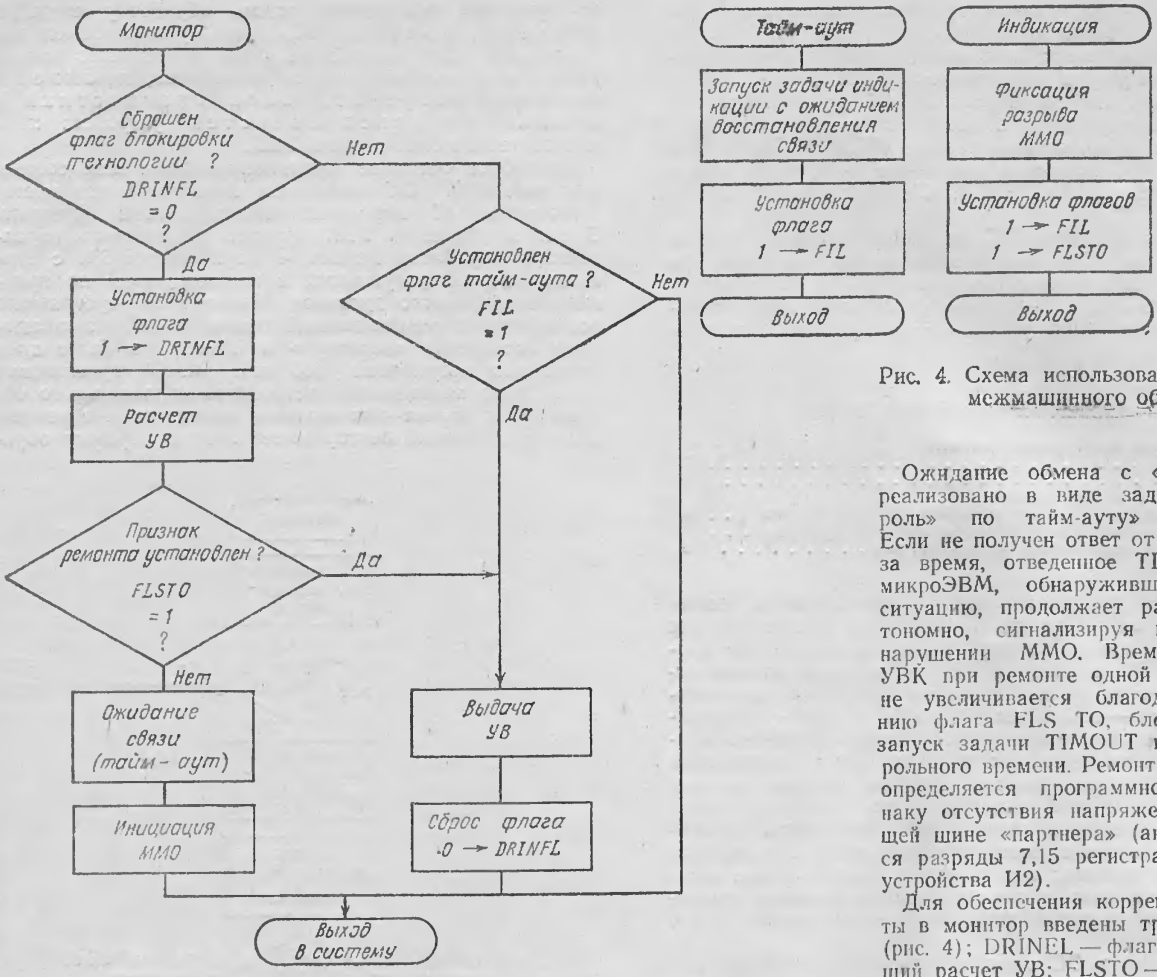


Рис. 4. Схема использования флагов межмашинного обмена

ботки прерывания. По вектору А микроЭВМ обмениваются результатами расчета. В случае совпадения результатов обмен прекращается. Если результаты не совпали, то обмен продолжается на втором уровне прерываний — по вектору Б происходит обмен результатами работы диагностических программ.

На основании полученной информации формируются признаки наличия фатальных неисправностей «свой», «чужой». После анализа в обеих микроЭВМ этих признаков выбирается соответствующий вариант работы УВК (рис. 3).

Различные конкретные действия УВК при ветвлениях отражают специфические особенности контролируемой технологии. В описываемом УВК выдача большего из двух рассчитанных УВ обеспечивает устойчивую работу энергосистем.

Межмашинный обмен основывается на следующих положениях:

1. Рассчитав УВ, необходимо инициировать ММО с ожиданием связи с «партнером» через время Т.
2. В течение времени Т2 блокировать расчет УВ.
3. При подтверждении разрыва ММО производить выдачу УВ без последующей инициации ММО.
4. Индикация разрыва ММО должна быть разовой. Повторную инициацию производить после подтверждения разрыва.
5. При нормальном функционировании двухмашинного УВК выдачу УВ производить в подпрограмме обработки вектора А.
6. При случайном разрыве (сбое) ММО ввести задержку индикации разрыва ММО на время Т2, предоставляя возможность его восстановления.

Ожидание обмена с «партнером» реализовано в виде задачи «Контроль» по тайм-ауту» (TIMOUT). Если не получен ответ от «партнера» за время, отведенное TIMOUT, то микроЭВМ, обнаружившая данную ситуацию, продолжает работать автономно, сигнализируя при этом о нарушении ММО. Время реакции УВК при ремонте одной микроЭВМ не увеличивается благодаря введению флага FLS TO, блокирующего запуск задачи TIMOUT после контрольного времени. Ремонт «партнера» определяется программно по признаку отсутствия напряжения на общей шине «партнера» (анализируются разряды 7,15 регистра состояния устройства И2).

Для обеспечения корректной работы в монитор введены три признака (рис. 4); DRINEL — флаг, блокирующий расчет УВ; FLSTO — флаг, блокирующий запуск задачи TIMOUT при ремонте одной микроЭВМ; FIL — флаг, блокирующий запуск задачи инициации ММО.

Опыт эксплуатации. ОС располагается в ППЗУ на микросхемах типа К573РФ. Объем занимаемой памяти — 14 Кбайт, из которых 4 Кбайт занимает ядро ОС. Времена ожидания Т1 и Т2 равны соответственно 60 и 120 мс. Управляющий осциллятор каждого процессора УВК настроен на частоту 8±0,1 МГц.

Средняя загрузка процессора при основном цикле опроса УСО и расчета УВ, равного 20 мс,— 90 %, остальное время процессор находится в динамическом останове, ожидая внешних событий (прерываний).

Более чем трехлетний опыт эксплуатации УВК на конкретном объекте выявил ряд схемных и конструктивных недоработок некоторых устройств микроЭВМ «Электроника 60».

Так, в процессоре М2 отсутствует блок обработки приоритета прерываний [5], и при подаче питания процессором осуществляется либо пуск по вектору 24 (устанавливается пользователем), либо обработка таймерного прерывания. Если не предпринимать анализа ситуации, то произойдет некорректный пуск системы. Для устранения этого недостатка в программу обработки таймерных прерываний введен контроль флага старта системы.

Аналого-цифровой преобразователь (АЦП) типа 15КЛ-60/8—010 непригоден для функционирования в режиме рестарт (кратковременное исчезновение питания). В связи с этим необходимо подключать входные сигналы к АЦП с задержкой не менее 10 мс после включения питания микроЭВМ.

При опросе устройств (типа 15КВ-60/32—001) ввода дискретных сигналов наблюдается сброс информации, выданной перед этим на регистр устройства 15КВВ-60—30—001 вывода дискретных сигналов. Поэтому пользователю необходимо сохранять выходную информацию перед опросом устройств ввода, а затем ее восстанавливать.

700096, Ташкент, Мукими, 182, предприятие Средазтехэнерго, электроцех; тел. 77-94-44

ЛИТЕРАТУРА

1. Вигдорчик Л. И. и др. Операционная система СМ ЭВМ РАФОС.— М.: Финансы и статистика, 1984.
2. Орнов В. Г., Мерпорт Э. И., Тененбаум Я. П. и др. Опыт разработки и внедрения САОН на базе микроЭВМ // Электрические станции.— 1983.— № 9.— С. 52—53.
3. Тененбаум Я. П., Кремер Э. Н., Мерпорт Э. И. и др.— Противоаварийная автоматика энергосистемы на базе двухмашинного комплекса микроЭВМ «Электроника 60» // Электрические станции.— 1986.— № 9.— С. 53.
4. Прангишвили И. В. Микропроцессоры и локальные сети микроЭВМ в распределенных системах управления.— М.: Энергоатомиздат, 1985.
5. Кормин Е. Г. Повышение отказоустойчивости систем автоматизации на основе микроЭВМ «Электроника 60» // Микропроцессорные средства и системы.— 1986.— № 3.— С. 40.

Статья поступила 28 сентября 1987

УДК 681.325.58

С. Е. Вдовин, В. Н. Волынчук, И. Н. Зибров, В. Т. Ковальчук

ПРИМЕНЕНИЕ БИС К1802ВР2 ДЛЯ ПОСТРОЕНИЯ МНОГОРАЗРЯДНЫХ УСТРОЙСТВ ДЕЛЕНИЯ

Устройство деления в интегральном исполнении (БИС К1802ВР2) обладает небольшой разрядностью (8 разрядов), а устройства, построенные на ИМС малой и средней степени интеграции, имеют невысокое быстродействие, громоздки и малонадежны. Предлагаемый метод построения многоразрядных устройств деления на малоразрядных функционально законченных делителях позволяет осуществлять деление чисел теоретически произвольной разрядности с достаточно высокими быстродействием и надежностью.

В качестве примера рассмотрено 16-разрядное устройство деления на базе 8-разрядной БИС К1802ВР2. В основу предлагаемого метода положено представление двоичных чисел в виде конечного полинома [1]:

$$A = A_n 2^{nk} + A_{n-1} 2^{(n-1)k} + \ldots + A_0 2^0,$$

где A — представляемое (n+1) k-разрядное число; A_i — k-разрядные весовые коэффициенты представления числа; k — разрядность функционально законченного делителя; (n+1) — число членов разложения.

Для представления 16-разрядного числа A в виде набора 8-разрядных компонент достаточно двух членов полинома:

$$A = A_1 2^8 + A_0 2^0.$$

В таком случае отношение 16-разрядных чисел A и B представляется в виде

$$\frac{A}{B} = \frac{A_1 2^8 + A_0 2^0}{B_1 2^8 + B_0 2^0}.$$

Последнее отношение необходимо представить линейной комбинацией отношений только 8-разрядных чисел. Здесь возможны два варианта. В первом число B требуется выразить как произведение 8-разрядных сомножителей, число которых должно быть равно трем [1]. При этом аппаратурное разложение числа на множители весьма затруднительно. Второй путь представляет собой вариант выполнения приближенных вычислений через отношение полиномов [1]. В результате выполнения операции деления полинома A на B получен бесконечный убывающий ряд:

$$\frac{A}{B} = \frac{A_1}{B_1} 2^0 + \left(\frac{A^0}{B_1} - \frac{A_1 B_0}{B_1 B_1}\right) 2^{-8} - \left(\frac{A_0}{B_1} - \frac{A_1 B_0}{B_1 B_1}\right) \frac{B_0}{B_1} 2^{-16} + \ldots,$$

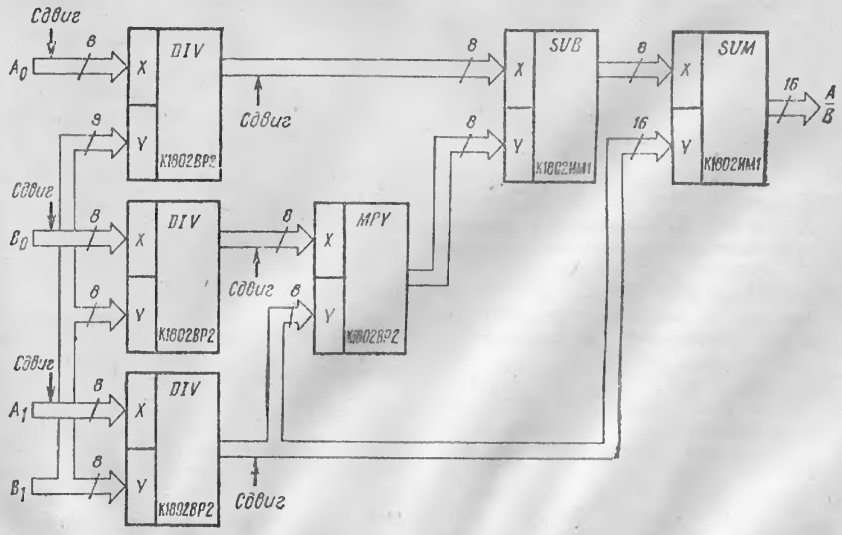
члены которого (начиная со второго) представляют собой геометрическую прогрессию со знаменателем $\frac{B_0}{B_1} 2^{-8}$.

Для осуществления операции деления над 16-разрядными числами вполне достаточно ограничиться первыми двумя членами ряда ввиду того, что при делении 16-разрядных чисел в частном никогда не будет верных значащих цифр больше, чем у операнда с меньшим количеством таковых [2], а абсолютная погрешность вычислений, представляющая собой сумму членов ряда, следующих за вторым, не превышает величины младшего разряда 16-разрядного числа, получаемого в результате вычислений.

Таким образом, результат деления двух 16-разрядных чисел A и B можно записать таким образом:

$$\frac{A}{B} \approx \frac{A_1}{B_1} 2^0 = \left(\frac{A_0}{B_1} - \frac{A_1 B_0}{B_1 B_1}\right) 2^{-8}.$$

Такого приближения вполне достаточно для вычисления отношения с точностью до 16 разрядов. Для сохранения большего количества верных значащих цифр в результате вычисления отношения A к B, т. е.



Функциональная схема устройства деления 16-разрядных чисел

для уменьшения погрешности операции деления, необходимо, чтобы числа A и B были нормализованными. В этом случае A_1 и B_1 никогда не будут меньше половины своих максимально возможных значений.

При реализации операции деления на БИС К1802ВР2 делимое должно быть не больше делителя [3]. Возникает необходимость в процедуре сдвига на один разряд вправо величин A_1, A_0 и B_0. Окончательное выражение приобретает вид

$$\frac{A}{B} \approx \frac{A_1 2^{-1}}{B_1} 2^{12^0} + \left(\frac{A_0 2^{-1}}{B_1} 2^1 + \frac{B_0 2^{-1}}{B_1} 2^1 \frac{A_1 2^{-1}}{B_1}\right) 2^{-8}.$$

Анализ этого выражения показывает, что для реализации операции деления над 16-разрядными числами A и B требуется три 8-разрядных делителя, один 8-разрядный вычислитель и один 16-разрядный сумматор. Для реализации сдвигов величин A_1, A_0, B_0 и последующего учета этих сдвигов при получении результата не требуется никаких аппаратурных затрат, так как сдвиги являются фиксированными и осуществляются коммутацией соединительных шин ко входам и выходам перечисленных устройств. В качестве делителей и умножителя в разработанном устройстве применены БИС К1802ВР2, а в качестве вычитателя и сумматора — БИС К1802ИМ1 (возможно применение произвольных АЛУ ТТЛ-логики). Функциональная схема устройства деления 16-разрядных чисел на базе 8-разрядных БИС К1802ВР2 представлена на рисунке. Устройство имеет следующие параметры: время вычисления отношения 16-разрядных чисел — порядка 2,5 мкс, относительная погрешность вычислений — не более 0,0015 %, потребляемая мощность — около 8,5 Вт.

262009, г. Житомир, ул. Циолковского, 4, б, кв. 56, Вдовину Сергею Евгеньевичу

ЛИТЕРАТУРА

1. Бронштейн И. Н., Семендяев К. А. Справочник по математике для инженеров и учащихся втузов.— М.: Наука, 1980.
2. Рабинович З. Л., Рамануаскас В. А. Типовые операции в вычислительных машинах.— Киев: Техника, 1980.
3. Данилов Р. В., Ельцова С. И., Иванов Ю. П. Применение интегральных микросхем в электронной вычислительной технике: Справочник/Под ред. Б. Н. Файзулаева, Р. В. Тарабрина.— М.: Радио и связь, 1987.

Статья поступила 7 июля 1987

УДК 681.32

А. В. Бокарев, А. И. Кабанов

АДАПТЕР-ПРЕОБРАЗОВАТЕЛЬ АДРЕСА

Функциональные возможности и структура систем, создаваемых на основе микроЭВМ, зависят от характеристик процессора, ОЗУ и от наличия внешних запоминающих устройств (ВЗУ) большой емкости. Серийные ВЗУ семейства СМ ЭВМ работают в 18-разрядном адресном пространстве; процессор диалогового вычислительного комплекса (ДВК) «Электроника МС 0502.09» — в режимах 16-разрядного виртуального адресного пространства, а также преобразования (перемещения) в 18-разрядное и 22-разрядное физические адресные пространства. В режиме 16-разрядного виртуального адресного пространства адресуются 64 Кбайт памяти. Виртуальный адрес непосредственно переходит в физический адрес ОЗУ. Только адреса 160000—177777 передаются странице ввода-вывода, расположенной в области физических адресов 17760000—17777777 (рис. 1, *а*).

В режиме преобразования в 18-разрядное физическое адресное пространство адресуются 256 Кбайт памяти. Виртуальный адрес перемещается в 18-разрядном физическом адресном пространстве. При этом диспетчер памяти отображает 248 Кбайт ОЗУ и 8 Кбайт страницы ввода-вывода (рис. 1, *б*). В данном режиме функция управления памятью ДВК совместима с серийными процессорами СМ ЭВМ.

В режиме преобразования в 22-разрядное адресное физическое пространство адресуются 4096 Кбайт памяти. Виртуальный адрес перемещается в 22-разрядном физическом адресном пространстве. При этом диспетчер памяти отображает адресное пространство полностью (рис. 1, *в*).

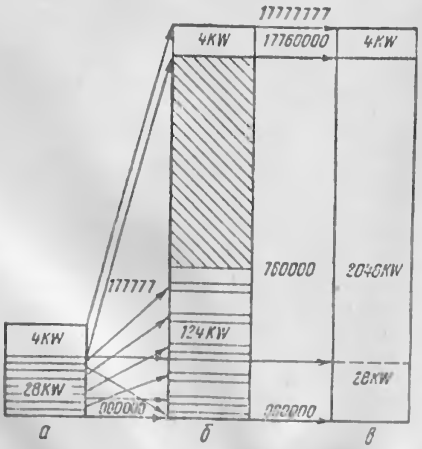


Рис. 1. Режимы работы: 16-битовый (*а*), 18-битовый (*б*), 22-битовый (*в*).

Подключение ВЗУ семейства СМ ЭВМ с 18-разрядным адресом к магистрали ДВК с 22-разрядным адресом осуществляется через адаптер-преобразователь адреса (АПА). Это позволяет максимально использовать возможности ОС RT-11, RSX-11M, TSX, UNIX.

Функции АПА (рис. 2): взаимное преобразование интерфейса «Общая шина» (ОШ) и межмодульного параллельного интерфейса (МПИ); преобразование 18-разрядного адреса магистрали ОШ в 22-разрядный адрес магистрали МПИ. АПА состоит из схемы преобразования интерфейсов К1801ВП1—054 (СПИ), буферного регистра адреса (РБА), схемы приоритета прерываний (СПП), двунаправленных усилителей, генератора тактовых импульсов (ГИ), арифметическо-логического устройства (АЛУ), регистров адреса базы (РАБ), регистра последнего отображаемого адреса (РПА) и сопутствующих логических схем.

Схема СПИ взаимно преобразует сигналы управления магистралей МПИ и ОШ, а также управляет другими узлами АПА. Регистр РБА хранит адрес на время передачи с магистрали МПИ на магистраль ОШ (программно недоступен).

21-битовым регистрам РАБ предоставлены 64 адреса страницы ввода-вывода (рис. 3). Заметим, что последний регистр (адреса 17770374 и 17770376) можно использовать для чтения и записи, но не для перемещения адресов магистрали ОШ, так как последний регистр отображает адреса 17760000—17777777, занятые страницей ввода-вывода.

Схема СПП передает запросы и разрешает прерывание между магистралями МПИ и ОШ, а также транслирует разрешения на прерывание в магистраль МПИ при отсутствии запросов на прерывание в магистрали ОШ (четыре уровня прерываний).

Двунаправленные усилители предназначены для электрофизического сопряжения внутренних линий АПА с магистралями МПИ и ОШ. Генератор ГИ синхронизирует работу СПИ. АЛУ формирует физический адрес в магистраль МПИ, суммируя базовый адрес, хранящийся в регистре РБА, с битами (12...01) адреса, поступающего из магистрали ОШ.

Регистр РПА состоит из двух слов, расположенных по адресу 17777734 (младшее слово) и 17777736 (старшее слово), и содер-

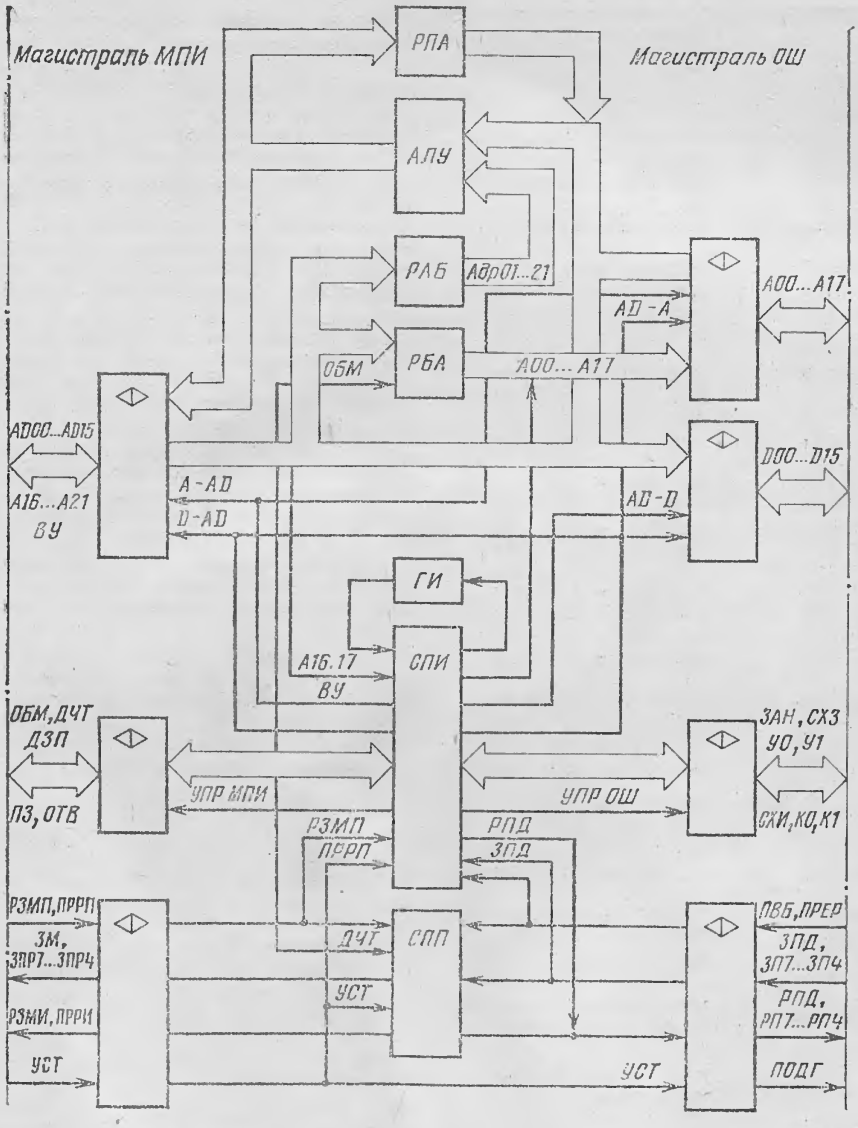


Рис. 2. Структурная блок-схема адаптера-преобразователя адреса

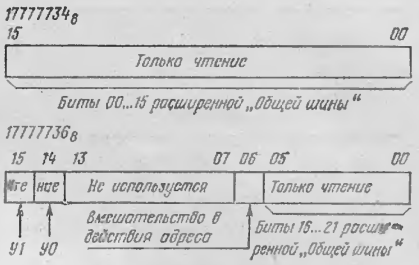


Рис. 4. Регистр РПА

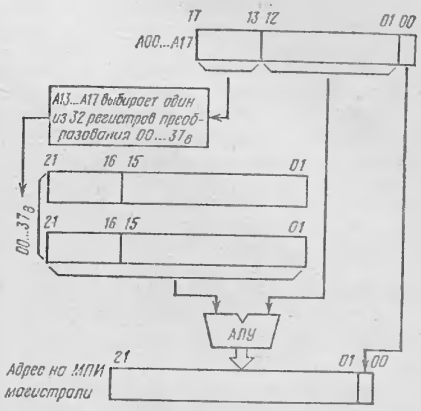


Рис. 5. Построение физического адреса

Функция преобразования адреса в АПА очень похожа на функцию, выполняемую диспетчером памяти.

Конструктивно АПА выполнен на двусторонней печатной плате, устанавливаемой в каркас ДВК. Для подключения магистрали ОШ на плате установлены два разъема СНО53—60—2—В. Питание АПА осуществляется от блока питания ДВК.

Телефон 532-84-22, Москва

ЛИТЕРАТУРА

1. Глушкова Г. Г., Иванов Е. А. МикроЭВМ семейства «Электроника» // Микропроцессорные средства и системы.—1986.—№ 4.— С. 7.

жит 22-битовый адрес памяти магистрали МПИ, соответствующий последнему отображаемому в память адресу. Этот регистр используется для целей технического обслуживания и содержит также управляющие линии С0, С1 и БИТ вмешательства в действие переключателей (рис. 4). АПА работает без преобразования адресов и с преобразованием адресов. В режиме без преобразования адресов 18-разрядный адрес магистрали ОШ передается в магистраль МПИ без изменения.

Адреса магистрали ОШ преобразуются при выдаче сигнала UMAP (из диспетчера памяти процессора, низкий уровень), соответствующего биту 05 регистра состояния SR3 (адрес 17772516).

В режиме преобразования адресов при поступлении из магистрали ОШ адреса автоматически выбирается соответствующий регистр РАБ и считывается его содержимое. Этот 21-разрядный адрес (21...01) РАБ суммируется с 12-разрядным смещением в адресе магистрали ОШ (12...01 разряды) для формирования физического адреса (рис. 5). Разряд 00 передается из магистрали ОШ в магистраль МПИ без преобразования.

Технические характеристики

Режим работы	Преобразование адресов, без преобразования адресов
Потребляемый ток, А, не более	4,0
Напряжение питания, В	+5±0,25
Конструктив	Плата с двусторонним печатным монтажом
Габаритные размеры, мм	280×239×12
Масса, кг, не более	0,5

2. Кокорин В. С., Кридинер Л. С., Попов А. А., Хохлов М. М. Тенденция развития диалоговых вычислительных комплексов//Там же.— С. 11.
3. Мини- и микроЭВМ семейства «Электроника» М61./Б. Л. Толстых, И. Л. Талов, В. Г. Талов, В. Г. Цывинский и др.— М.: Радио и связь, 1987.— 296 с.: ил. (с. 162...165).

РЕГИСТР	ФИЗИЧЕСКИЙ АДРЕС ДЛЯ ПРЯМОГО ЧТЕНИЯ ИЛИ ЗАПИСИ		АДРЕСА "ОБЩЕЙ ШИНЫ", ОТОБРАЖАЕМЫЕ ПОСРЕДСТВОМ РЕГИСТРОВ
	МЛАДШИЙ	СТАРШИЙ	
0	17 770 200	17 770 202	000 000 – 017 777
1	17 770 204	17 770 206	020 000 – 037 777
2	17 770 210	17 770 212	040 000 – 057 777
3	17 770 214	17 770 216	060 000 – 077 777
4	17 770 220	17 770 222	100 000 – 117 777
5	17 770 224	17 770 226	120 000 – 137 777
6	17 770 230	17 770 232	140 000 – 157 777
7	17 770 234	17 770 236	160 000 – 177 777
10	17 770 240	17 770 242	200 000 – 217 777
11	17 770 244	17 770 246	220 000 – 237 777
12	17 770 250	17 770 252	240 000 – 257 777
13	17 770 254	17 770 256	260 000 – 277 777
14	17 770 260	17 770 262	300 000 – 317 777
15	17 770 264	17 770 266	320 000 – 337 777
16	17 770 270	17 770 272	340 000 – 357 777
17	17 770 274	17 770 276	360 000 – 377 777
20	17 770 300	17 770 302	400 000 – 417 777
21	17 770 304	17 770 306	420 000 – 437 777
22	17 770 310	17 770 312	440 000 – 457 777
23	17 770 314	17 770 316	460 000 – 477 777
24	17 770 320	17 770 322	500 000 – 517 777
25	17 770 324	17 770 326	520 000 – 537 777
26	17 770 330	17 770 332	540 000 – 557 777
27	17 770 334	17 770 336	560 000 – 577 777
30	17 770 340	17 770 342	600 000 – 617 777
31	17 770 344	17 770 346	620 000 – 637 777
32	17 770 350	17 770 352	640 000 – 657 777
33	17 770 354	17 770 356	660 000 – 677 777
34	17 770 360	17 770 362	700 000 – 717 777
35	17 770 364	17 770 366	720 000 – 737 777
36	17 770 370	17 770 372	740 000 – 757 777
37(*)	17 770 374	17 770 376	

* МОЖЕТ БЫТЬ ИСПОЛЬЗОВАН ДЛЯ ЧТЕНИЯ ИЛИ ЗАПИСИ, НО НЕ ИСПОЛЬЗУЕТСЯ ПРИ ПЕРЕМЕЩЕНИИ

Рис. 3. Регистр последнего отображаемого адреса

4. Бокарев А. В., Гаморин М. Ю., Кабанов А. И. БИС адаптера магистралей СМ ЭВМ и микроЭВМ // Микропроцессорные средства и системы.—1987.—№ 3.— С. 3.

5. Бокарев А. В., Гаморин М. Ю., Кабанов А. И. Адаптер магистралей МПИ—ОШ//Микропроцессорные средства и системы.—1987.—№ 3.— С. 6.

Статья поступила 26 августа 1987

УСТРОЙСТВА СВЯЗИ С ОБЪЕКТОМ

УДК 681.327.8—181.4

И. Н. Жукинский, И. В. Логвиненко, И. Ю. Скрипник, Л. В. Цымбурская

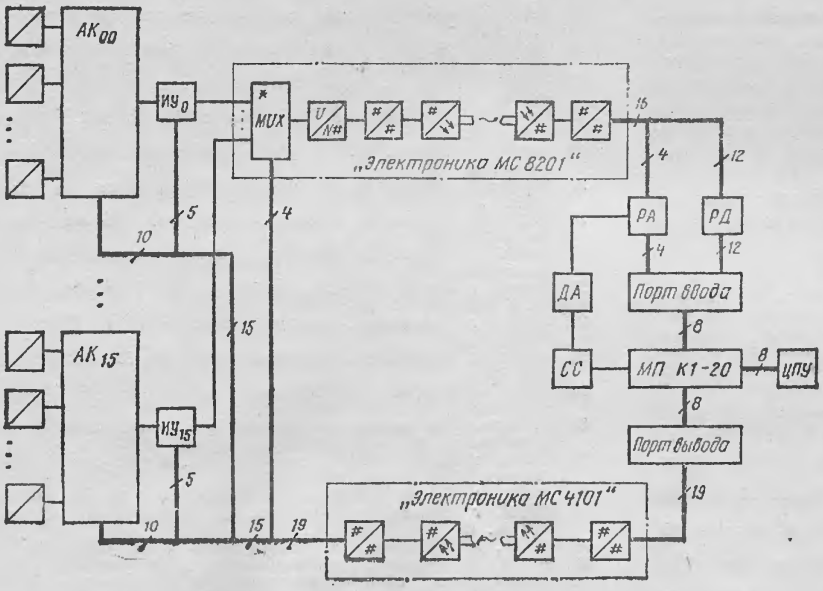
ВОЛОКОННО-ОПТИЧЕСКИЙ ИНТЕРФЕЙС В АДАПТИВНОЙ СИСТЕМЕ ДИАГНОСТИКИ ЭЛЕКТРОУСТАНОВОК

Применение микроЭВМ для управления системами диагностики действующих электроустановок связано с определенными трудностями: сильные электромагнитные поля, вибрации, необходимость обеспечения гальванических развязок между цепями съема информации и собственно микроЭВМ. Эти проблемы в основном решаются при использовании волоконно-оптических линий связи (ВОЛС). Применение ВОЛС обеспечивает высокую информационную плотность передающего канала, что весьма актуально при создании многоканальных диагностических систем.

Для построения системы можно использовать волоконно-оптическую систему сбора аналоговой информации «Электроника МС8201», которая позволяет циклически опрашивать 16 датчиков аналоговых сигналов. В реальных энергетических установках необходимо получать информацию от значительно бо́льшего числа датчиков. Например, диагностическая система промышленных турбогенераторов электрической энергии содержит более 500 датчиков. Учитывая, что уровни выходных сигналов этих датчиков составляют единицы милливольт, необходимо применять специальные аналоговые коммутаторы и измерительные усилители. Наилучшими метрологическими характеристиками для коммутации слабых аналоговых сигналов обладают контактные переключатели на герконовых реле, время переключения которых составляет 1...5 мс. Тогда время циклического опроса большого числа датчиков составит несколько секунд, что ограничивает возможности своевременной регистрации возникновения предаварийной ситуации.

Поскольку выходные сигналы от разных датчиков не равнозначны по своей информативности, целесообразно некоторые группы датчиков опрашивать более часто. В результате экспресс-анализа этих сигналов с помощью микроЭВМ определяется последовательность опроса всех остальных датчиков. Таким образом, возникает необходимость дистанционно управлять коммутаторами по заданному алгоритму. Для этой цели встречный канал управления легко реализуется с использованием волоконно-оптической системы передачи цифровой информации типа «Электроника МС4101». Этот вспомогательный канал позволит переключать коэффициенты усиления измерительных усилителей во всем динамическом диапазоне.

Функциональная схема системы диагностики электроустановок (см. рисунок) содержит 16 аналоговых измерительных коммутаторов ($АК_{00}$... ...$АК_{15}$) типа Ф799/1, имеющих 100 входных каналов для подключения к системе 1600 аналоговых датчиков. Специальными мерами, применяемыми при подключении датчиков (подключение витыми парами, экранирование и т. п.), можно без помех коммутировать слабые входные сигналы на уровне 1...5 мВ с погрешностью $\delta \leqslant 0{,}01$ %. В дальнейшем эти сигналы нормализуются одним из 16 программно управляемых измерительных усилителей $ИУ_{00}$...$ИУ_{15}$ и поступают на один из входов волоконно-оптической системы сбора аналоговых данных. «Электроника МС8201» содержит циклический



Функциональная схема системы диагностики электроустановок

мультиплексор аналоговых каналов (16×1), АЦП, преобразователь кодов (параллельного в последовательный и в код Манчестер II), приемно-передающий модуль с волоконно-оптическим кабелем длиной до 300 м, второй преобразователь кодов (последовательного в параллельный). Частота циклического опроса входных каналов определяется временем преобразования АЦП с $\tau=50$ мкс.

Код Манчестер II содержит 20 бит информации в одной посылке, которые передаются по волоконно-оптическому кабелю с частотой $f=8$ МГц. Для повышения помехоустойчивости каждая кодовая посылка сравнивается с предыдущей и при совпадении преобразуется в параллельный код и записывается в выходной регистр. Весь процесс передачи информации длится 5 мкс, т. е. за время преобразования информация передается более 10 раз.

Для согласования канала передачи с каналом микропроцессорной системы используется асинхронный приемный интерфейс, состоящий из регистров данных (РД) и адреса (РА), дешифратора адреса (ДА), схемы синхронизации (СС) и порта ввода. В простейшем случае СС фиксирует совпадение кода адреса с программно задаваемым кодом и формирует сигнал ввода информации с буферных регистров РД и РА через порт ввода на шину данных микропроцессора.

Микропроцессорный контроллер «Электроника МС2702» анализирует полученную информацию и вырабатывает код управляющего слова, который через порт вывода и волоконно-оптическую систему передачи цифровых данных «Электроника МС4101» поступает для управления аналоговыми коммутаторами $АК_{00}$... ...$АК_{15}$, перестройки измерительных усилителей $ИУ_{00}$...$ИУ_{15}$ и установления нового положения мультиплексора. Возможность внешнего управления мультиплексором обеспечивается путем доработки системы «Электроника МС8201». Это позволяет расширить возможности системы за счет адаптации к реальному объекту.

Адаптивность системы заключается в том, что в зависимости от результатов измерения сигналов контрольных датчиков и в соответствии с записанной программой контроллер определяет наиболее целесообразную очередность опроса всех датчиков, одновременно устанавливая оптимальные значения коэффициентов усиления измерительных усилителей.

Контроллер обращается к портам ввода и вывода, реализованным на микросхемах КР580ВВ55, как к внешним устройствам, анализируя результаты измерений, и выводит значения сигналов каждого (или группы) датчика на одно из внешних устройств: дисплей «Символ-К», цифровое отсчетное или печатающее устройство. Программное обеспечение системы позволяет вычислять статические параметры сигналов каждого датчика или группы датчиков и выводить их на внешние устройства.

252680, Киев, Проспект Победы, 56, Институт электродинамики, Жукинскому И. Н.; тел. 441-25-60

Статья поступила 15 июня 1987

ПИСЬМО В РЕДАКЦИЮ

Любительское объединение по компьютерной технике «Ц-8290» при МТО Свердловского района г. Перми (что за обаятельное и короткое название, не правда ли?) организовано 13 февраля 1987 г. одним из первых в стране. С самого начала своей деятельности мы ориентировались на полный хозрасчет и самоокупаемость, предполагая широкую хоздоговорную деятельность. Однако первый договор дался нелегко — он был заключен только 10 июня 1987 г. (на пусконаладочные работы). Уже к концу 1987 г. нам удалось набрать и выполнить объем в 100000 руб.: поставка двух компьютерных классов, доработка АРМов и тому подобные аппаратные и программные работы.

В настоящее время мы твердо стоим на ногах (смета на 1988 финансовый год составлена из расчета 2 млн. руб. оборота). Основными направлениями деятельности наконец-то смогут стать досуг молодежи (заработаны деньги на технику) и разработка учебных и прикладных программ для ЭВМ массового пользования (если удастся такие ЭВМ раздобыть).

Объединение «Ц-8290» было организовано на основе Положения о любительском объединении, клубе по интересам, которое было незначительно доработано. Главными трудностями в работе были и остаются отсутствие своей техники и своего помещения, отчего и приходится выступать в роли посреднической фирмы. Однако эти трудности, кажется, преодолеваются.

Кроме того, вызывает недоумение ориентация большинства предприятий и организаций на аппаратуру, технику в ущерб программному обеспечению (по нему у нас так и не было договоров!). Надеемся, что с массовой компьютеризацией положение изменится.

В настоящее время в нашей стране работает (даже по нашим скромным данным) более десятка хозрасчетных компьютерных клубов и любительских объединений. Однако обмен информацией между ними весьма ограничен несмотря на здравую идею Харьковского КЛП (см. прилагаемый каталог). Практически отсутствует система в работе всех этих разрозненных объединений. На очереди развертывание сети компьютерных центров. Боимся, что и с ними — разрозненность и неразбериха, как и вообще у нас с программным обеспечением.

Просим «МП» продолжить начатое в № 4 за прошлый год полезное дело — давать регулярную информацию о любительских объединениях и их достижениях. В каком-то смысле взять на себя роль координатора-

информатора и любителей тоже (профессионалы не могут пожаловаться на отсутствие полезной информации в журнале). Печатая буквально страничку с адресами, тематикой и имеющимися разработками объединений по компьютерной технике, журнал поможет всем любителям найти нужные им сведения, организовать междугородные связи, поделиться достижениями и проблемами.

По поручению Совета «Ц-8290» С. Щеглов (отв. за переписку, зам. председателя Совета)

АДРЕСНЫЙ КАТАЛОГ

Держатель информации — Харьковский клуб любителей компьютеров (КЛЛ). При обращении к любителям, перечисленным ниже, просьба ссылаться на нас: 310038, Харьков, ул. Маршала Батицкого, д. 20 а, общ. 6, к. 614, Пустовойтов С. В.

I. Адреса радиолюбителей, занимающихся реализацией или эксплуатацией компьютеров на базе МП КР580ИК80А

1. Днепропетровская обл., г. Орджоникидзе, ул. Калинина, д. 79, к. 92. Цвик А. В.
2. 324088, г. Кривой Рог, М. К. Юбилейный, д. 15, кв. 45. Назарчук Вадим («Ириша»).
3. 454128, г. Челябинск, Комсомольский просп., д. 91/а, кв. 29. Хмелевский Р. М.
4. 317900, Кировоградская обл., г. Александрия, ул. Калинина, д. 67, кв. 508. Сидоров Ю. А.
5. 624070, Свердловская обл., г. Березовский, ул. Мира, д. 1, к. 107. Жевак Александр.
6. 490035, г. Семипалатинск, ул. Крестьянская, д. 3, к. 505-Б. Василенко Ю. Г.
7. 349940, Ворошиловградская обл., г. Северодонецк, ул. Дзержинского, 14а, кв. 45. Лапин А. Ф.
8. 665841, г. Ангарск, 17-й М. К. Район, д. 1, кв. 157. Васильев Сергей.

II. Адреса лиц, работающих на микроЭВМ «Электроника Д3-28»

1. 422520, г. Зеленодольск, ул. Карла Маркса, д. 60, кв. 39. Кудосова Наталья.
2. 452620, БАССР, г. Октябрьский, Просп. Ленина, д. 14, кв. 47. Головацкий Сергей.
3. 460007, г. Оренбург, ул. Ходакова, д. 5. Кузнецов Д.
4. 410082, г. Саратов, СВВКИУРВ-«0». Тюрин Андрей.
5. 432013, г. Ульяновск, УВВКУС-Г. Николаев А. М.
6. 413800, Саратовская обл., г. Балаково, ул. Шевченко, д. 44, кв. 37. Полищук Ирина.
7. 160000, г. Вологда, Наб. Кедрова, 66, кв. 24. Кизарев Илья.
8. 314004, г. Полтава, ул. Баленко, 7, кв. 49. Савченко Ю. В.
9. 274022, г. Черновцы, ул. Стасюка, 3а, кв. 4. Бабийчук Светлана.
10. 644099, г. Омск, ул. Кароя Лигети, 8, кв. 15. Мартынов Ю. В.
11. 420015, г. Казань, ул. Галактионова, д. 2, кв. 5 (двор). Фукс А. М. (работает не только на «Д3-28»).
12. 423200, ТАССР, г. Бугульма, ул. Гоголя, д. 75, кв. 17. Закиров И.
13. 422520, ТАССР, г. Зеленодольск, а/я 161. Фомин С. В. (Клуб друзей компьютеров).
14. 163051, г. Архангельск, ул. Энгельса, д. 101, кв. 23. Воробьев И. В.

III. Адреса лиц, программирующих на микроЭВМ «Электроника БК-0010»

1. 692523, г. Уссурийск, пер. Тихий, д. 7, кв. 77. Предит В. Г.
2. 634048, г. Томск, пр. Фрунзе, д. 226, кв. 25. Лабутин Глеб.
3. 119618, г. Москва, Г-618, Солнцевский просп., д. 24, кв. 77, Квасов А.
4. 194156, г. Ленинград, пр. Энгельса, д. 23, ОВТ, к. 31-В, Сергушко Алексей, Крылов Кирилл, Коваленок Владимир.
5. 686920, Магаданская обл., Провиденский р-н, п. Уэлики, пер. Связистов, д. 2, кв. 8. Погаевский А. В.
6. 220094, г. Минск, пр. Рокоссовского, 25, кор. 2, кв. 43. Цекунов В. И.
7. 163051, г. Архангельск, ул. Энгельса, д. 101, кв. 23. Воробьев И. В.
8. 214007, г. Смоленск, ул. Октябрьской Революции, д. 40, кв. 11. Петров Игорь, тел. 5-58-30.
9. 117342, г. Москва, ул. Островитянова, д. 39, кв. 273. Седулин К. Ю.
10. 141980, Московская обл., г. Дубна, ул. Жуковского, д. 1, кв. 1. Лавренко А. Н.
11. 109472, г. Москва, ул. Ташкентская, д. 25/1, кв. 1. Вихров А.
12. 220131, г. Минск, ул. Кольцова, д. 32, кв. 160. Винокур В. Б.
13. 613020, Кировск. обл., г. Кирово-Чепецк, Первомайская, д. 4/4, кв. 64. Лимонов В. Л.
14. 195279, г. Ленинград, пр. Наставников, д. 42, кв. 112. Максеев Александр.
15. 142121, Московская обл., г. Подольск, ул. Ленинградская, д. 16, кв. 19. Ветров С. Г.
16. 192283, г. Ленинград, ул. Олеко Дундича, д. 19, кор. 3, кв. 20. Белоусов А. Ю.
17. 750053, г. Фрунзе, ул. Актюбинская, д. 69, кв. 13. Руднев Ю. С.
18. 121165, г. Москва, ул. Киевская, д. 20, кв. 217. Тараканов О. П.
19. 241014, г. Брянск, ул. Н. Советская, д. 144, кв. 19. Морозов И. В.
20. 198096, г. Ленинград, ул. Строителей, д. 9, кв. 129. Панаев Ю. Л.
21. 334320, г. Евпатория Крымской обл., ул. 9-го мая, д. 80, кв. 26. Ткаченко А. Я.
22. 248016, г. Калуга, ул. Ленина, д. 24/33. Лаврухина Е. Клуб школьников.
23. 624440, г. Серов Свердловской обл., ул. Парковая, д. 10, кв. 7. Моисеев С. Н.
24. 700081, г. Ташкент, квартал Маккашлык, д. 13, кв. 46. Меламед Я.
25. 420091, г. Казань, ул. Химиков, д. 51, кв. 20. Исламов Радик Г.
26. 420085, г. Казань, ул. Химиков, д. 21, кв. 15. Ахмадеев Э. И.
27. г. Тула, 18-й проезд, д. 94, СПТУ-7, тел. раб. 29-90-98. Куксенко В. Ф., зам. директора (класс КУВТ-86).
28. 614000, г. Пермь, Комсомольский пр., д. 91, тел. 44-01-31. Скрипов А. Г. (класс КУВТ-86).
29. 634000, г. Томск, СПТУ-11, ул. Герцена, д. 18; тел. 3-32-12.
30. 375040, г. Ереван, ул. Ачаряка, д. 20. СПТУ-11. Сираканян М. О.
31. 221081, г. Николаев, ул. Космонавтов, д. 66, СПТУ-11.
32. 310142, г. Харьков, ул. Тимуровцев, д. 39. Прохоров В. С.
33. 143900, Моск. обл., г. Балашиха, ш. Энтузиастов, д. 66, кв. 40. Монахов В. Т. (тел. 529-75-73, автор статьи в № 4 МПСС за 1987 г. по ПО для БК-0010).

IV. Список лиц и организаций, работающих на ПК «ЯМАХА»

1. 302016, г. Орел, ул. Комсомольская, д. 120, кв. 57. Белоножко О. П.
2. 257017, г. Черкассы, ул. Воровского, д. 11. Гаманин В. П.
3. 690105, г. Новосибирск, ул. Рельсовая, д. 2/2, кв. 20. Щекотов А. А.
4. 680030, г. Хабаровск, ул. Мухина, д. 11, кв. 87. Семешко Наталья.
5. 327030, г. Николаев, ул. Артиллерийская, д. 10, кв. 36. Ганичев Сергей.
6. 163051, г. Архангельск, ул. Энгельса, д. 101, кв. 23. Воробьев И. В.
7. 173014, г. Новгород, ул. Хутынская, д. 4, кв. 25. Мерина А. И.
8. 624000, г. Свердловск, Свердловский Инженерно-педагогический институт.

V. Компьютерные клубы и объединения

1. Любительское объединение по компьютерной технике при МТО Свердловского района г. Перми «Ц-8290».
   614039, г. Пермь, ул. Швецова, 41—26. Гладков А. В.— пред. Совета, рук., любительского объединения.
   614000, г. Пермь, ул. Пушкина, д. 13, кв. 161. Щеглов С. И. — зам. пред. Совета любительского объединения.
   614010, г. Пермь, ул. Героев Хасана, д. 4, МТО Свердловского района. Гладков А. В. (тел. 44-01-29 — после 20.00 моск. вр.; 44-11-10 — с 8.00 до 15.00 моск. вр.). Щеглов С. И. (тел. 32-77-28 — после 19.00 моск. вр.).
2. Компьютерный клуб. г. Рига, ул. Баласта Дамбис, д. 3. Ред. газ. «Советская молодежь», отдел соц.-экон. проблем молодежи (тел. 46-75-80).
3. Объединение любителей программирования.
   420015, г. Казань, ул. Галактионова, д. 2, кв. 5 (двор). Фукс А. М. Работает почти на всех типах ЭВМ и ОС: ОС ЕС, ДОС ЕС, ОС РВ, ЮНИКС, ДОС АСПО, АРМ СМ 2, СМ 4, Электроника «Д3-28». Готов вести переписку и обмен программами с клубами и любителями, работающими на любых ЭВМ.
4. Компьютерный клуб «Сапфир».
   340000, г. Донецк, ул. Словацкая, д. 61, кв. 113. Сербиненко А. В. («Компьютеры БК-0010»).
5. Клуб друзей компьютеров.
   422520, ТАССР, г. Зеленодольск, а/я 161, Фомин С. В. («Д3-28»).
6. Клуб «Плюс 86» специалистов по вычислительной технике при ЛДМ (Ленинградском дворце молодежи).
   г. Ленинград, ул. Проф. Попова, д. 47, к. 263 (ЛДМ) (234-97-12 (ЛДМ)) пятн. с. 19.00 до 22.00).
7. Клуб «Контакт» по программированию и вычислительной технике при МТО ППО «Моторостроитель». г. Пермь, СПТУ № 1.
   614000, г. Пермь, ул. Пушкина, д. 27, кв. 77, Скрипов А. Г.— пред. Совета (тел. 32-58-30).
8. Любительское объединение «Вариант» при Черемушкинском РК ВЛКСМ. г. Москва; раб. тел. 333-04-11. Задорин И. В. — пред.
9. Любительское объединение программистов «Содружество» при МОСТе. г. Москва; тел. 282-14-39.
10. Компьютерный клуб.
    309103, г. Белгород, п. Майский, ул. Вавилова, д. 30, кв. 1. Логачев В. В.

ЭВМ И ИНФОРМАТИЗАЦИЯ ОБЩЕСТВА: СТАРЫЕ МИФЫ И НОВЫЕ ПРОБЛЕМЫ

Г. Р. Громов

В фантастических романах главное это было радио. При нем ожидалось счастье человечества. Вот радио есть, а счастья нет.

Из записных книжек И. Ильфа

Почти всякий заметный технологический рывок вызывает сначала активное недоверие, всеобщий скепсис, а затем столь же массовую эйфорию радостно-нетерпеливого ожидания очередного «бога из машины». И лишь после преодоления этих двух первых этапов, изначально в значительной степени иррациональной общественной реакции, начинается этап реальной практической эксплуатации в народном хозяйстве новых средств техники, вскрывается и пласт проблем, которые они порождают, и открываемые ими возможности.

Этап полупрезрительного скепсиса в отношении производственных возможностей станков второй промышленной революции — персональных компьютеров — был в основном преодолен в нашей стране к 1983—1984 гг., однако этап ЭЙФОРИИ опасно затянулся и признаков его преодоления пока незаметно. Все еще делаются попытки измерять путь, пройденный страной в направлении массовой компьютеризации народного хозяйства, числом «отгруженных» с заводов компьютеров. Между тем по уровню надежности отечественные персональные компьютеры уступают аналогичным машинам США в 10—20 раз, а сеть их технического и программного обслуживания практически отсутствует. Стоит ли удивляться, что в этих условиях около половины всего парка выпущенных отечественной промышленностью персональных ЭВМ (ПЭВМ) до сих пор оказываются постоянно неисправными. Это означает, что заводы, производящие ПЭВМ, почти половину всей своей продукции могут заведомо записывать в прямой ущерб государству. К концу 1988 г. общая сумма такого прямого ущерба от выпуска «электронного лома» оценивается на уровне от 300 до 600 млн. руб. Вместе с ростом планов выпуска сумма гарантированных убытков народному хозяйству от производства неработоспособной у потребителя электронной техники быстро растет и (если не будут приняты необходимые экстренные меры) к 1990 г. будет измеряться миллиардами рублей!

Как предотвратить этот «плановый ущерб» народному хозяйству, наносимый под флагом массовой компьютеризации? Первое, что для этого необходимо сделать,— радикально изменить экономические критерии оценки деятельности предприятий, производящих вычислительную технику. Заводу должно быть экономически выгодно производить только надежные изделия, гарантированно обеспеченные сетью сервиса. Массовый выпуск ненадежных компьютеров, задержка с развертыванием сети обслуживания средств вычислительной техники должны приводить к разорению завода-производителя, поставляющего стране дорогостоящий «электронный лом». Пути и формы такой экономической привязки интересов предприятия (и отрасли в целом) к общегосударственным задачам выпуска надежной и квалифицированно обслуживаемой электронной техники в свое время достаточно подробно обсуждались в печати: «Макси-проблема микропроцессоров» (Известия, 1985, № 201), «Осторожно: компьютеры!» (журнал ЭКО, 1986, 7), «Надежность персональных ЭВМ и производственная загрузка фирменной сети сервиса» (журнал «Микропроцессорные средства и системы», 1985, № 4, а также тот же журнал за 1988 г., № 4) и др.

Однако руководители заинтересованных министерств все эти годы упорно делают вид, что к ним это не относится. Все остается как было: годами не замечать остро дискутируемую в печати проблему, ежегодный экономический вес которой измеряется сотнями миллионов рублей, даже сегодня — в годы перестройки и полной гласности — можно! Министерства и ведомства, занятые производством средств вычислительной техники, доказывают это с еще большим блеском и убедительностью, чем их коллеги из традиционных отраслей машиностроения. И действительно, кто же добровольно решится подвязывать свое монопольно несокрушимое сегодня благополучие к неустойчивым интересам «капризного потребителя?» Ведь для этого надо менять в корне всю структуру управления, а значит, и людей... Поэтому-то и складывается та самая «патовая» ситуация, когда никому ничего объяснять не надо: все все знают и все остается по-прежнему...

Кроме мифа об «информатизации с конвейера», согласно которому чем больше ПЭВМ сошло с конвейеров заводов, производящих ПЭВМ, тем выше следует считать уровень компьютеризации страны (на профессиональном сленге специалистов в области информационной технологии этот миф получил название «сага о ветвистых компьютерах»), за последние годы глубоко укоренились и ряд других не менее пагубных для национальной экономики мифов эпохи «электронизации всей страны». Наиболее ярко они проявляются в рамках хорошо известного не только в электронике, но и в других областях техники «верхушечного» подхода, который сотрудник НИВЦ АН СССР М. И. Лазарев как-то определил лозунгом: «Вперед к компьютеризации... минуя телефонизацию».

Пора бы, видимо, уже и осознать, что нельзя сколько-нибудь всерьез обсуждать научно-технические проблемы развития сетей ЭВМ, региональные и общенациональные банки данных и знаний и другие столь популярные сегодня в кругах «техноинтеллектуалов» проекты компьютерного «пересустройства страны», пока многие регионы, включая обширные районы столичной области (например, такие как Серпухов и его академгородки-спутники, образующие один из крупнейших промышленных и научных центров Подмосковья), не имеют сколько-нибудь регулярного выхода на автоматическую междугородную связь. В то время, как на компьютерных кафедрах многочисленных ВУЗов, в кабинетах академических и отраслевых НИИ азартно обсуждаются математически элегантные концепции иерархии глобальных, национальных, региональных и локальных сетей передачи данных, конструктор из ВНИИмотопрома в Серпухове, научный сотрудник академгородка Пущино-на-Оке или физик с Серпуховского ускорителя ведет многочасовые изнурительные переговоры с районным узлом связи по вековой давности сценарию: «Алло! Девушка, не могли бы Вы соединить меня с Москвой... Нет, я не могу после ноля часов, пожалуйста, сделайте сегодня! Девушка, я Вас очень прошу!» Все та же многим знакомая по фильмам легендарной киноэпопеи о событиях октября 1917 сцена: «Связь! Барышня, связь!...»

Сегодня уже никому, видимо, не надо доказывать, что компьютер — не «лампа Аладдина» из пещер с сокровищами третьего тысячелетия, а рабочий инструмент — станок второй промышленной революции, предназначенный для эффективной обработки информации, циркулирующей в каналах хозяйственного механизма страны. Несколько десятилетий назад было установлено, что пропускная способность информационных каналов хозяйственного механизма должна возрастать пропорционально квадрату прироста валового национального продукта (ВНП). Иными словами, нельзя увеличить, скажем, на 40 % ВНП, не увеличив одновременно вдвое пропускную способность каналов передачи и преобразования информации в хозяйственном механизме стра-

ны. Поэтому при разностороннем анализе причин экономического застоя последних двух десятилетий нельзя упускать из внимания, кроме прочего, и общеизвестный факт черепашьего роста пропускной способности информационных каналов в хозяйственном механизме страны.

Достаточно сказать, что, лидируя по стали, нефти и другим «натуральным показателям», мы нередко полностью упускаем из виду важнейшие информационные характеристики динамики современного индустриально развитого общества. Так, например, телефонная сеть в СССР в десять раз отстает от коммуникационной сети США даже по чисто количественным показателям, скажем, по общему числу абонентских точек. С учетом же качественного различия используемой аппаратуры разрыв этот оказывается существенно большим.

Наконец, с некоторых пор многотрудную борьбу с бюрократизмом стали кое-где отождествлять с много более простой борьбой с бумажным носителем документооборота. Между тем, несмотря на стремительный рост в последние три десятилетия средств электронной обработки информации, основу информационных потоков, циркулирующих в обществе, и у нас в стране и за рубежом все еще составляет бумажный носитель информации. Динамика экономического роста промышленно развитых стран до сих пор так и не позволяет им снизить темпы прироста документооборота на бумажном носителе тысячелетней традиции. Бурный рост информационных потоков на иных исторически новых «альтернативных» носителях информации (машинный носитель, видео, и т. д.) не только не уменьшает, а как будто бы даже ускоряет рост бумагооборота. Суть феномена электронного «форсажа бумаги» еще предстоит, видимо, изучать, однако полезно отдавать себе отчет в том, что общий бумажный выход американской, к примеру, экономики оценивается в 4 триллиона страниц в год и продолжает расти несмотря на то, что уже сотни миллионов канцелярских папок с документами переведены на дискеты (гибкие магнитные диски) десятками миллионов владельцев ПЭВМ. В нашей стране производство бумаги для письма и книгоиздательской деятельности в расчете на душу населения в 10 раз меньше, чем в США, а документооборот на дискетах пока практически отсутствует.

Важнейшей характеристикой интеллектуальной динамики информационного общества является уровень доступности населению богатства разнообразия книжной продукции. В настоящее время у нас в среднем на каждого грамотного человека разнообразие доступной ему книжной продукции оказывается в 10 раз меньше, чем, скажем, аналогичный показатель для России 1913 г. Как отмечают комментирующие этот трагический не только для общекультурного развития населения страны факт социологи Института книги при НПО «Книжная палата» Л. Гудков и Б. Дубинин, «социальные механизмы организации литературы и науки только в одном их узловом моменте — при переводе идей и чувств как результатов индивидуальной работы в формы письменной культуры — блокируют инновационные структуры, создающие динамику интеллектуальной жизни».

Фатальная для интеллектуальной динамики, видимо, нескольких поколений советских людей изоляция от книги (из-за принципиально порочной тиражной политики), так и не преодоленная до сих пор, ныне накладывается на новый стремительно растущий культурный разрыв на этот раз в области ВИДЕО. Поток информации, поступающей к населению с экранов видеомагнитофонов, в мире в целом уже сопоставим по «мере разнообразия» с продукцией книгоиздательской индустрии. В США ежегодно издается около 50 тысяч наименований книг и имеется в наличии 50 тысяч различных видеофильмов: учебных, справочных, художественных и т. д. По числу производимых видеомагнитофонов и, соответственно, видеофильмов на душу населения наша страна до сих пор отстает от ведущих промышленно развитых стран в 1000 раз. Это означает, кроме прочего, что в новом мире культуры, по масштабам сопоставимом с миром книги, интеллектуальный и духовный натиск наших политических оппонентов не встречает сегодня никакого сопротивления как за рубежом, так и внутри страны. На вновь открытом в конце XX века континенте культурной цивилизации все события происходят без нашего участия. Пока трудно предсказать даже самые ближайшие последствия происходящего ныне тотального одностороннего культурного разоружения для будущего страны со статусом великой державы.

Среди факторов, резко ограничивающих рост пропускной способности информационных каналов хозяйственного механизма страны, не последнее место занимает совершенно иррациональный, по-видимому, суеверный в своей основе страх, который «компетентные» организации все еще испытывают перед множительной техникой. Мало того, что объем производства средств оперативного копирования бумажных документов у нас на два-три порядка ниже, чем в любой другой промышленно развитой стране (в СССР в год производится 2 тысячи аппаратов типа «ксерокс», а, к примеру, в ФРГ свыше 200 тысяч), так даже те аппараты, что имеются, используются обычно с такими «режимными» сложностями, которые еще в несколько раз снижают их эффективность. Можно предположить, что когда-то это могло иметь смысл (если ограничения на циркуляцию общеизвестной информации вообще могут иметь хоть какой-то смысл), но сегодня, когда одних только ЭВМ, снабженных автоматическими печатающими устройствами (принтерами), в стране производится в 10—20 раз больше, чем ксероксов, продолжать держать их за железными дверями?... И, тем не менее, весьма весомый вклад в замедление циркуляции информационных потоков «режим», разработанный для ксероксов во времена оные, все еще вносит, и экономическая цена такого типа «забытых инструкций» неуклонно растет.

Что такое информатизация общества?

Информатизация — это совокупность взаимосвязанных политических, социально-экономических и технологических факторов, которые обеспечивают свободный доступ всем самодеятельным членам общества к любым (кроме небольшого класса законодательно временно закрытых) источникам информации.

Оценивать уровень информатизации масштабами выпуска ЭВМ (пусть даже и безупречно работоспособных), успехами в реализации престижных проектов искусственного интеллекта и так далее — это по существу то же самое, что оценивать успехи в развитии, скажем, сельского хозяйства числом тракторов, комбайнов, гектарами пашни, кубокилометрами мелиорации т. д.

Об этом, видимо, не стоило и упоминать, если бы попытки свести многосложную проблему информатизации общества к той или иной научно-технической задаче не встречались столь часто. Например, авторитетный ученый из Киевского центра кибернетики на полном серьезе утверждает на страницах центральной газеты, что, по его мнению, «формирование машинной информатики взамен «бумажной» это и есть информатизация» (Ю Капыгин. Фактор прогресса.— Правда, 21 июня 1988). «Это же парадокс,— считает он,— что именно в 70-е годы, когда широко развернулись работы по созданию АСУ разного класса и назначения, в сфере управления ускорились негативные процессы «разбухания аппарата», роста формализма, числа бессмысленных документов, процедур и регламентов».

На наш взгляд, «негативные процессы», о которых с недоумением упоминает в контексте массового насаждения разного рода АСУ-контор А. Капыгин, это вовсе не парадокс, а закономерный результат технократического подхода к решению задач информатизации.

После того, как очередное массированное вложение средств в голубую технократическую мечту завершается очевидным экономическим крахом, наступает фаза поиска «объективных причин» или, на крайний случай,

«парадоксов», научно объясняющих нанесенный экономический эффект («из 5 тысяч созданных организационных АСУ,— уточняет Ю. Каныгин,— большая часть представляет собой малоразвитые системы, не улучшающие технологию управления предприятиями и учреждениями»).

Справедливости ради, необходимо отметить, что крупномасштабные эксперименты эпохи АСУ оказались разорительными уроками болезненного освобождения от очередной волны технократического мышления для руководства многих корпораций практически всех промышленно развитых стран. Сегодня особенно важно поэтому сделать необходимые выводы из дорогостоящих ошибок (и своих, и чужих), чтобы, по крайней мере, не повторять их на новом витке информатизации. Один из наиболее очевидных такого типа уроков заключается в том, что реальное продвижение страны вперед по пути к информационному обществу осуществляется по известному принципу «движения эскадры»: общий темп определяет скорость самого тихоходного судна. Только поэтапное комплексное решение взаимосвязанных проблем информационной технологии во всех ее и новейших и вполне традиционных ипостасях может позволить эффективно осваивать средства, выделяемые на информатизацию. Любой искусственно упрощенный подход, как бы внешне привлекателен он ни был, заведомо гарантирует создание под флагом информатизации очередной еще более масштабной, чем печальной памяти эпоха АСУ, экономической «черной дыры», куда уже начали проваливаться сотни миллионов рублей, расходуемые ныне на производство ненадежных и никем не обслуживаемых ПЭВМ. Остановить разорительный поток «ветвистых компьютеров», создать условия для эффективного использования растущего парка ЭВМ — лишь одна из сложного ряда технологических задач информатизации, которая ни в коей мере не должна заслонять другие не менее острые: книгоиздательская индустрия и тиражная политика, видео, телефон, кабельное телевидение и другие средства коммуникации, ксероксы и т. д.

Одновременно следует начать формирование и широкое общественное обсуждение взаимосвязанных программ политического, социального, правового и экономического аспектов общегосударственной проблемы информатизации общества.

142292, Пущино М. О., НИВЦ АН СССР

Статья поступила 24 июля 1988

УВАЖАЕМАЯ РЕДАКЦИЯ!

Пишет Вам учащийся 9-го класса. Родители купили мне ПЭВМ «Электроника БК 0010» 1987 г. выпуска. А телевизор у нас «Рекорд В-312» 1979 г. выпуска. Входа «видео» в этом телевизоре нет. Я пробовал подать сигнал от компьютера на антенное гнездо ТВ, на контрольные точки КТ1 и КТ3 блока «видео» (БЛОК № 2). Во II и III случаях можно было разобрать надпись, которая должна появляться при включенном компьютере. Но экран светлый, рябит, на нем бегут полосы и сама надпись, так что точно разобрать слова невозможно. Я пробовал нажимать на клавишу ЭВМ «Инверсия экрана», но изображение не улучшилось.

Очень прошу помочь мне — объясните, пожалуйста, как подключить мой компьютер «Электроника БК 0010» к телевизору «Рекорд В-312».

С уважением Олег Калюжный

140006, Люберцы Моск. обл., ул. Космонавтов, д. 50, к. 13.

УДК 681.3.01 : 51

ИНТЕРФЕЙСНЫЕ БИС МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА К1801: МИКРОСХЕМА К1801ВП1-033

(Продолжение цикла. Начало в № 4, 1988)

Микросхема К1801ВП1-033 является многофункциональным устройством параллельного интерфейса и может работать в режимах:

интерфейса накопителя на гибких магнитных дисках;

контроллера интерфейса параллельного ввода-вывода;

контроллера байтового параллельного интерфейса (ИРПР).

Микросхема предназначена для использования совместно с микросхемой К1801ВП1-034 для организации интерфейсного устройства 16-разрядного программируемого параллельного ввода-вывода и байтового параллельного ввода-вывода, а также в качестве самостоятельного интерфейсного устройства накопителя на гибких магнитных дисках.

Режим интерфейса накопителя на гибких магнитных дисках.

Условное графическое обозначение микросхемы К1801ВП1-033 в этом режиме приведено на рис. 1, назначение выводов показано в табл. 1, электрическая структурная схема представлена на рис. 2.

Микросхема устанавливается в этот режим подачей на выводы RC0 ...RC3 напряжения высокого уровня. Об-

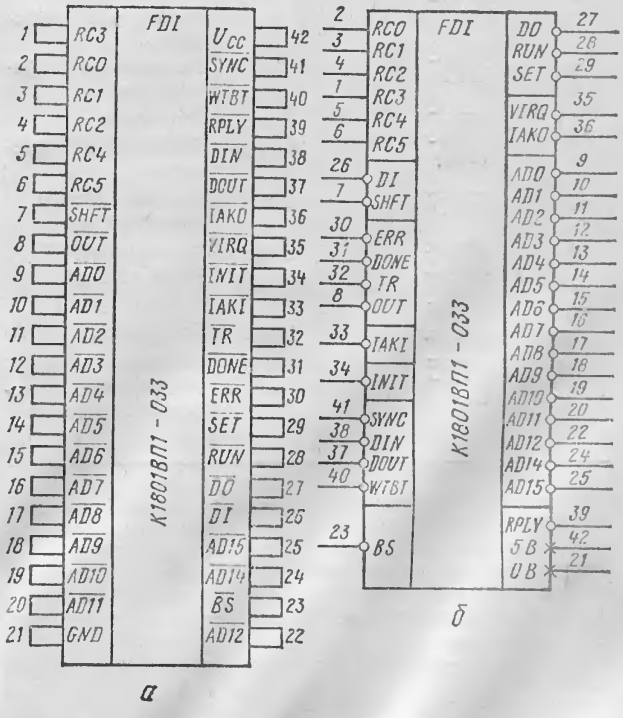


Рис. 1. Условное графическое обозначение микросхемы К1801ВП1-033 в режиме интерфейса накопителя на гибких магнитных дисках по порядку расположения (*а*) и функциональному назначению (*б*) выводов

Таблица 1

Наименование выводов интерфейса накопителя на гибких магнитных дисках

Вывод	Обозначение	Тип вывода	Наименование
1	RC3	Вход	Выбор режима 3
2	RC0		« » 0
3	RC1		« » 1
4	RC2		« » 2
5	RC4		« » 4
6	RC5		« » 5
7	$\overline{SHFT}$		Сдвиг данных
8	$\overline{OUT}$		Вывод данных
9...16	$\overline{AD0}$... $\overline{AD7}$	Вход-выход	Разряды адреса-данных
17...20	$\overline{AD8}$... $\overline{AD11}$	Вход	То же
21	GND	—	Общий
22	$\overline{AD12}$	Выход	12-й разряд адреса-данных
23	$\overline{BS}$		Внешнее устройство
24	$\overline{AD14}$		14-й разряд адреса данных
25	$\overline{AD15}$		15-й разряд адреса данных
26	$\overline{DI}$	Выход	Вход данных
27	$\overline{DO}$		Выход данных
28	$\overline{RUN}$		Пуск
29	$\overline{SET}$	Вход	Начальная установка
30	$\overline{ERR}$		Ошибка
31	$\overline{DONE}$		Завершено
32	$\overline{TR}$		Требование передачи
33	$\overline{IAKI}$		Разрешение прерывания
34	$\overline{INIT}$	Вход	Установка
35	$\overline{VIRQ}$	Выход	Требование прерывания
36	$\overline{IAKO}$	Выход	Предоставление прерывания
37	$\overline{DOUT}$	Вход	Вывод данных
38	$\overline{DIN}$	Выход	Ввод данных
39	$\overline{RPLY}$		Ответ
40	$\overline{WTBT}$	Вход	Запись-байт
41	$\overline{SYNC}$		Обмен
42	U_{cc}	—	Напряжение источника питания

мен информацией между процессором и контроллером НГМД осуществляется с помощью регистра команд состояния и регистра данных, которые считываются и загружаются программно. Формат регистра команд и состояния представлен на рис. 3. Разряды 08 ... 13 не используются. Назначение остальных разрядов следующее:

GO (пуск) — инициирует прием команды (только для записи);

F1 ... F3 — коды команд (только для записи);

US (выбор привода) — выбор одного из двух дисководов для выполнения требуемой команды (только для записи);

DONE (завершено) — выполнение команды закончено (только для чтения);

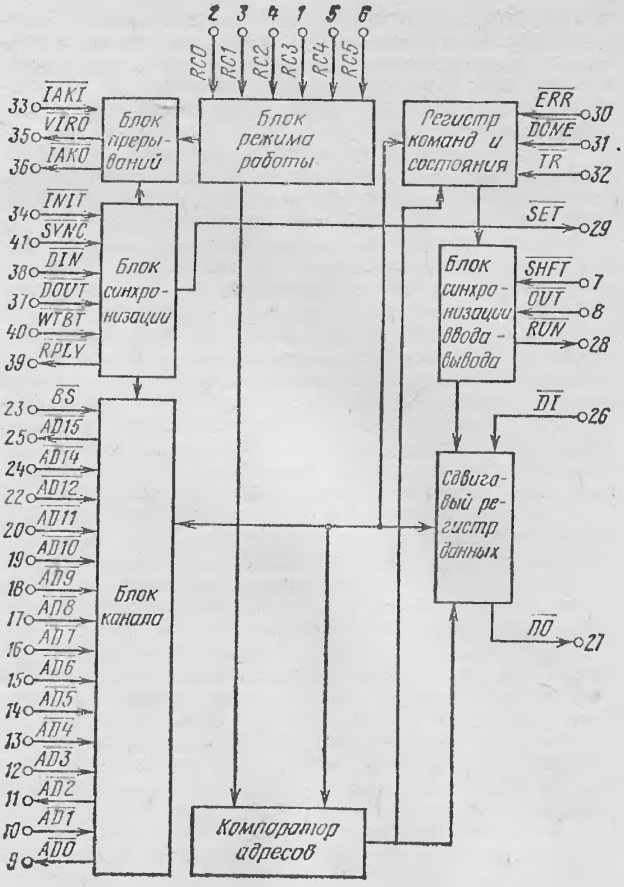


Рис. 2. Электрическая структурная схема интерфейса накопителя на гибких магнитных дисках

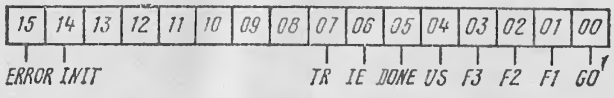


Рис. 3. Формат регистра команд и состояния

IE (предоставление прерывания) — устанавливается программно для выполнения прерывания по окончании выполнения команды; сбрасывается канальным сигналом INIT (для чтения и записи);

TR (требование передачи) — требование записи-считывания данных через регистр РД (только для чтения);

INIT (начальная установка) — устанавливается программно для приведения НГМД в исходное состояние;

ERROR (ошибка) — формируется, если в процессе выполнения команды произошла ошибка (только для чтения).

Наименования команд регистра в соответствии с кодами F1 ... F3 приведены в табл. 2.

Формат регистра данных представлен на рис. 4. Разряды 0 ... 7 содержат данные для ввода-вывода. Разряды 8 ... 15 не используются. Ниже приведен перечень интерфейсных сигналов контроллера НГМД:

$\overline{SET}$ (начальная установка) — приводит механизм и электронную часть НГМД в исходное состояние;

Таблица 2

Список команд регистра команд и состояний

Код F3	Код F2	Код F1	Команда
0	0	0	Запись в буфер
0	0	1	Чтение буфера
0	1	0	Запись сектора
0	1	1	Чтение сектора
1	0	0	Не используется
1	0	1	Чтение регистра РОС
1	1	0	Запись сектора с меткой
1	1	1	Чтение регистра РОШ

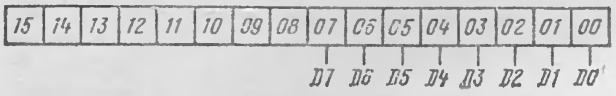


Рис. 4. Формат регистра данных

$\overline{DONE}$ (завершено) — устанавливается при выполнении команды или в случае ошибки;

$\overline{RUN}$ (пуск) — инициирует передачу команды или байта данных;

OUT (вывод) — указывает направление передачи байта информации (при низком уровне сигнала информация передается от НГМД к микросхеме);

$\overline{TR}$ (запрос передачи) — устанавливает готовность контроллера НГМД к приему-передаче байта информации;

$\overline{DI}$ (вход данных) — линия приема последовательной информации;

$\overline{DO}$ (выход данных) — линия передачи последовательной информации;

$\overline{SHFT}$ (сдвиг) — линия синхронизации приема-передачи последовательной информации; длительность отрицательного импульса не менее 200 нс, период 1 мкс;

$\overline{ERR}$ (ошибка)—формируется при обнаружении ошибки; прекращает выполнение текущей команды и устанавливает сигнал $\overline{DONE}$.

Переадресация регистров. С помощью выводов RC4 и RC5 можно переадресовать регистры микросхемы. Соответственно изменяются адреса векторов прерываний (табл. 3).

Работа микросхемы. При низком уровне сигнала $\overline{DONE}$ запись команды, содержащей Лог. 1 в нулевом разряде, в регистр команд устанавливает сигнал $\overline{RUN}$, который инициирует контроллер НГМД на прием команды. Контоллер снимает $\overline{DONE}$ и выставляет на линию $\overline{SHFT}$ серию из восьми импульсов.

После снятия сигнала $\overline{DONE}$ происходит сброс сигнала $\overline{RUN}$, а серия импульсов синхронизирует выдачу команды в последовательном коде на вывод $\overline{DO}$. В зависимости от принятого кода команды контроллер вырабатывает сигналы $\overline{OUT}$ и $\overline{TR}$ (рис. 5—8). При установленном сигнале $\overline{TR}$ в зависимости от состояния сигнала $\overline{OUT}$ обращение к регистру вызывает сигнал $\overline{RUN}$, который сбрасывается после снятия сигнала $\overline{TR}$, и серию импульсов на выводе $\overline{SHFT}$ (восемь импульсов для синхронизации адреса сектора и дорожки, семь—для синхронизации данных). По окончании выполнения команды формируется сигнал $\overline{DONE}$, который используется для работы по прерыванию (рис. 9). Требование прерывания $\overline{VIRQ}$ возникает с появлением сигнала $\overline{DONE}$ при наличии в регистре команд разрешения прерывания IE.

Таблица 3

Адреса регистров и векторов прерываний

Вывод	Уровень сигнала	Адрес РК	Адрес РД	Адрес вектора прерываний
RC4 RC5	Низкий »	177170	177172	264
RC4 RC5	Высокий Низкий	177174	177176	270
RC4 RC5	Низкий, Высокий	177200	177202	274
RC4 RC5	Высокий »	XXXXX0	XXXXX2	XXX

Примечание. X — значение разрядов адреса безразлично

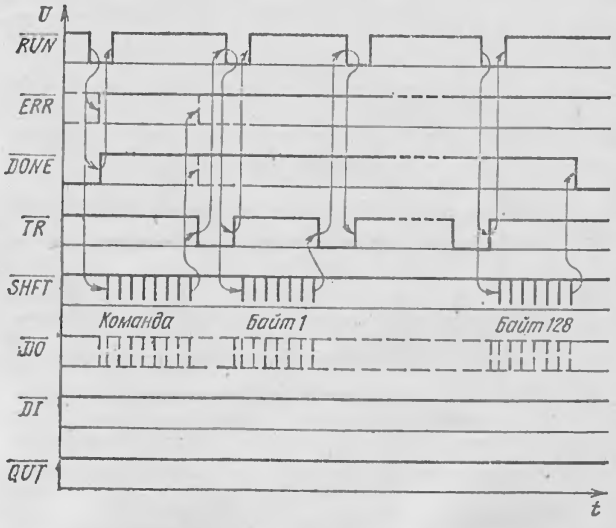


Рис. 5. Временная диаграмма цикла «запись в буфер»

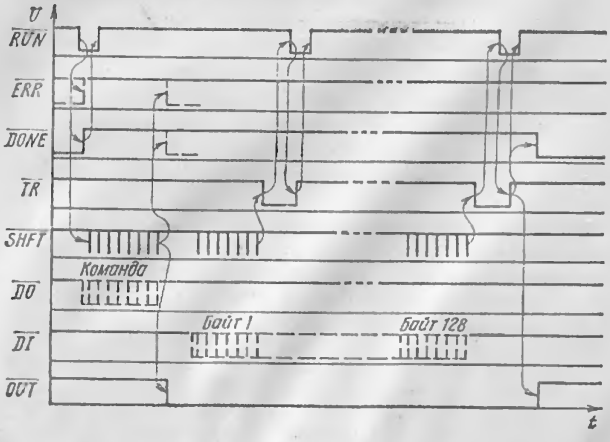


Рис. 6. Временная диаграмма цикла «чтение буфера»

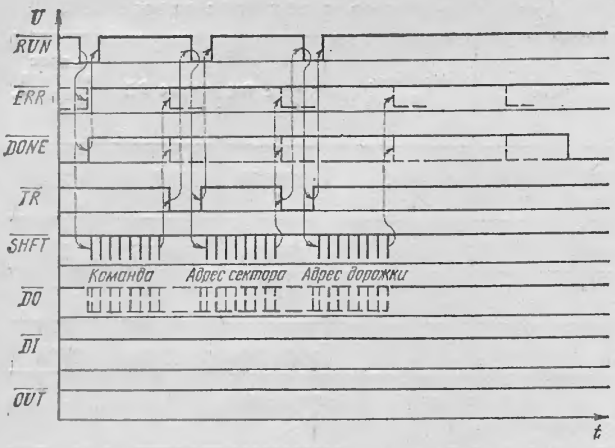


Рис. 7. Временна́я диаграмма циклов «запись сектора», «чтение сектора»

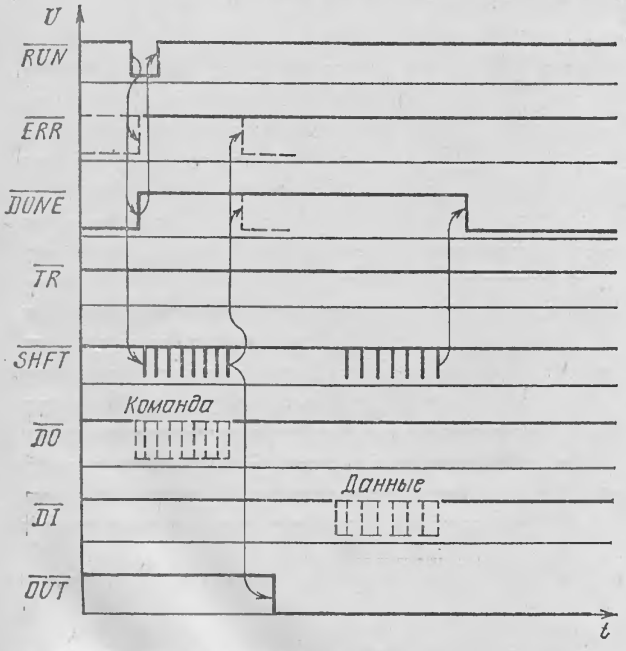


Рис. 8. Временна́я диаграмма циклов «считывание регистра состояния и ошибок», «считывание регистра ошибок»

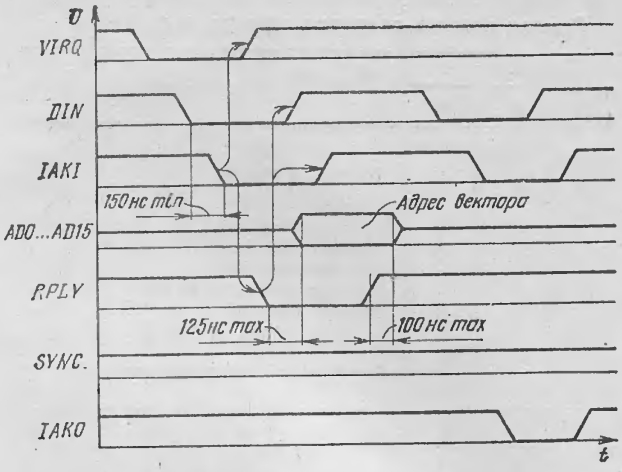


Рис. 9. Временна́я диаграмма работы по прерыванию программы

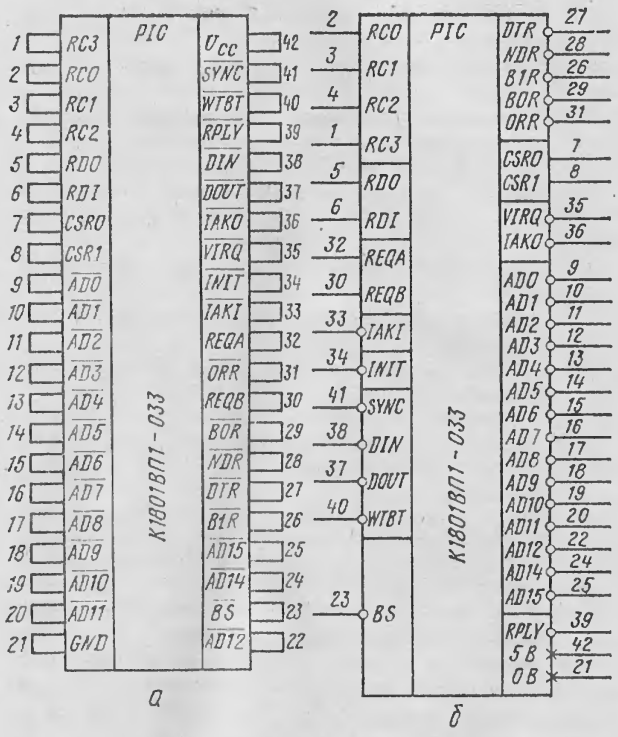


Рис. 10. Условное графическое обозначение микросхемы К1801ВП1-033 в режиме контроллера интерфейса параллельного ввода-вывода по порядку расположения (*а*) и функциональному назначению (*б*) выводов

Режим интерфейса параллельного ввода-вывода

Условное графическое обозначение микросхемы в этом режиме приведено на рис. 10, назначение выводов показано в табл. 4, электрическая структурная схема представлена на рис. 11. В этот режим микросхема переводится подачей определенных уровней напряжения на выводы RC0 ... RC3. Кроме выбора режима работы комбинация напряжений определяет переадресацию регистров и векторов прерываний (табл. 5).

Микросхема К1801ВП1-033 осуществляет прием и передачу информации с помощью регистров состояния РС, приемника РП и источника РИ. Регистры РП и РИ выполняются на двух микросхемах К1801ВП1-034 (описание БИС будет в следующем номере). БИС К1801ВП1-033 содержит РС, компаратор адресов, блок прерываний и блок управления регистрами РП и РИ. Формат регистра состояния представлен на рис. 12. Назначение разрядов РС следующее:

CSP1, CSP0, (PC01, PC00) — используются для имитации запросов прерывания в режиме автономной проверки (для чтения и записи);

REQA (требование А) — требование прерывания А (только для чтения);

Таблица 4

Наименование выводов контроллера интерфейса параллельного ввода-вывода

Вывод	Обозначение	Тип ввода	Наименование
1	RC3	Вход	Выбор режима 3
2	RC0		» » 0
3	RC1		» » 1
4	RC2		» » 2
5	RD0	Выход	Задержка ответа
6	RD1	Вход	» »
7	CSR0	Выход	Регистр состояния PC0
8	CSR1	»	Регистр состояния PC1
9...16	$\overline{AD0}$ ... $\overline{AD7}$	Вход-выход	Разряды адреса-данных
17...20	$\overline{AD8}$ ... $\overline{AD11}$	Вход	То же
21	GND	—	Общий
22	$\overline{AD12}$	Вход	12-й разряд адреса-данных
23	$\overline{BS}$		Внешнее устройство
24	$\overline{AD14}$		14-й разряд адреса-данных
25	$\overline{AD15}$		15-й разряд адреса данных
26	$\overline{BIR}$	Выход	Вывод старшего байта
27	$\overline{DTR}$		Ввод данных
28	$\overline{NDR}$		Вывод данных
29	$\overline{BOR}$		Вывод младшего байта
30	$\overline{REQB}$	Вход	Требование B
31	$\overline{ORR}$	Выход	Чтение выходного регистра
32	$\overline{REQA}$	Вход	Требование A
33	$\overline{IAKI}$		Предоставление прерывания
34	$\overline{INIT}$		Установка
35	$\overline{VIRQ}$	Выход	Требование прерывания
36	$\overline{IAKO}$	»	Предоставление прерывания
37	$\overline{DOUT}$	Вход	Вывод данных (по каналу)
38	$\overline{DIN}$	»	Ввод данных (по каналу)
39	$\overline{RPLY}$	Выход	Ответ
40	$\overline{WTBT}$	Вход	Запись-байт
41	$\overline{SYNC}$	»	Обмен
42	U_{cc}	—	Напряжение источника питания

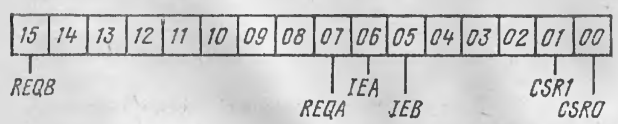


Рис. 12. Формат регистра состояния

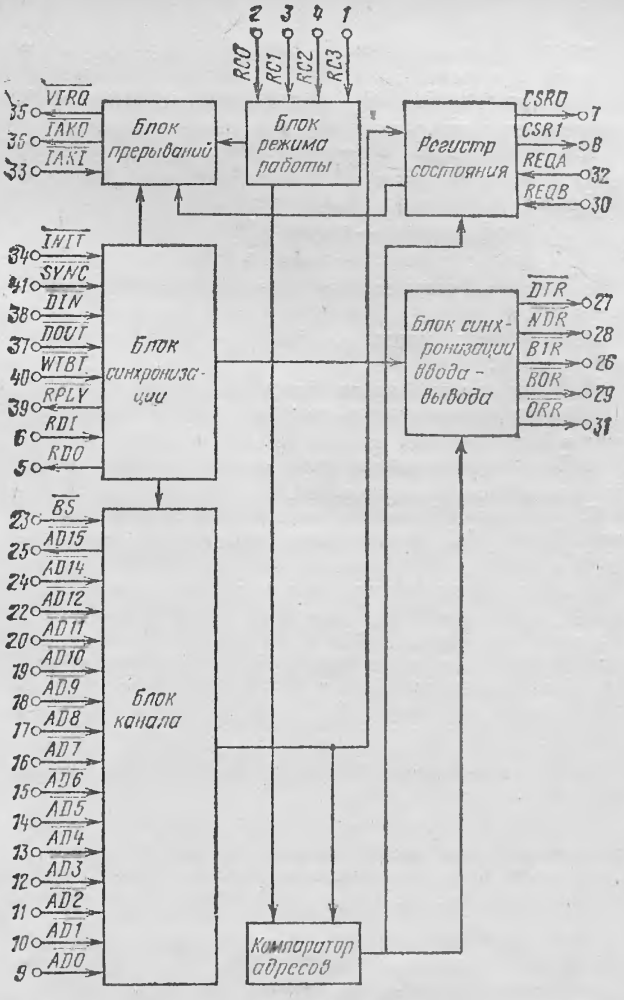


Рис. 11. Электрическая структурная схема контроллера интерфейса параллельного ввода-вывода

Таблица 5

Адреса регистров и векторов прерываний

Вывод	Уровень сигнала	Адрес РС	Адрес РИ	Адрес РП	Адрес вектора прерывания A	Адрес вектора прерывания B
RC3	Низкий	167770	167772	167774	A	300
RC0	Высокий				B	304
RC1	Низкий					
RC2	Высокий					
RC3	Высокий	167760	167762	167764	A	310
RC0	»				B	314
RC1	Низкий					
RC2	Высокий					
RC3	Низкий	167750	167752	167754	A	320
RC0	»				B	324
RC1	Высокий					
RC2	»					
RC3	Высокий	167740	167742	167744	A	330
RC0	Низкий				B	334
RC1	Высокий					
RC2	»					
RC3	Низкий	XXXXX0	XXXXX2	XXXXX4	A	XX0
RC0	Высокий				B	XX4
RC1	»					
RC2	»					

REQB (требование B) — требование прерывания B (только для чтения);

IEA (разрешение A) — разрешение прерывания A, сбрасывается сигналом $\overline{INIT}$ (для чтения и записи);

IEB (разрешение B) — разрешение прерывания B, сбрасывается сигналом $\overline{INIT}$ (для чтения и записи).

Сигналы управления регистрами РП и РИ:

$\overline{DTR}$ (ввод данных) — чтение РП;

$\overline{NDR}$ (вывод данных) — запись в РП;

$\overline{BIR}$ (вывод старшего байта) — запись старшего байта в РИ;

$\overline{BOR}$ (вывод младшего байта) — запись младшего байта в РИ;

$\overline{ORR}$ (чтение выходного регистра) — чтение РИ;

RDO (задержка ответа, выход) — выход разрыва цепочки формирования сигнала $\overline{RPLY}$;

RDI (задержка ответа, вход) — вход разрыва цепочки формирования сигнала $\overline{RPLY}$.

С помощью RC-цепочки, включенной между выводами RDI и RDO, можно увеличить ширину импульсов $\overline{DTR}$ и $\overline{NDR}$.

Работа микросхемы по прерыванию. Запрос прерывания $\overline{VIRQ}$ возникает при появлении хотя бы одного из сигналов REQA, REQB при наличии в РС разрешения прерывания IEA, IEB. При одновременном появлении сигналов REQA и REQB более высокий приоритет имеет требование B. Процедура прерывания выполняется стандартно (см. рис. 9).

Режим контроллера байтового параллельного интерфейса

Условное графическое обозначение микросхемы в этом режиме дано на рис. 13, назначение выводов приведено в табл. 6, электрическая структурная схема представлена на рис. 14. Режим задается подачей напря-

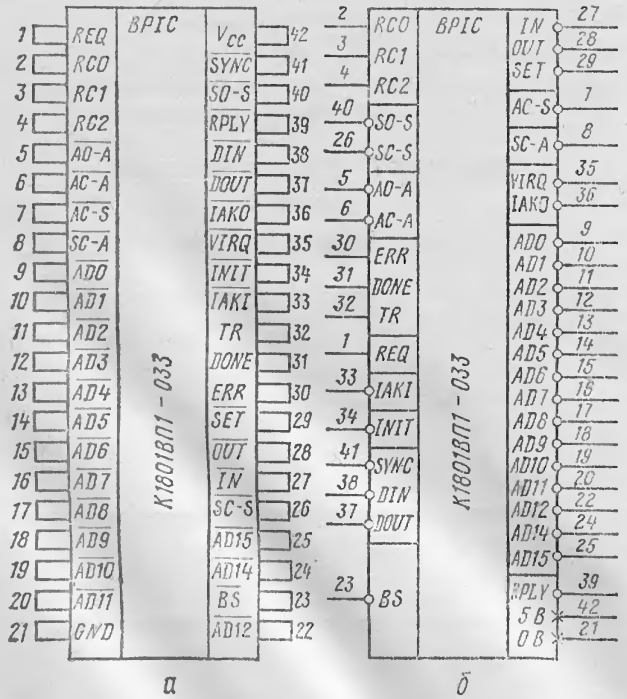


Рис. 13. Условное графическое обозначение микросхемы К1801ВП1-033 в режиме контроллера байтового параллельного интерфейса по порядку расположения (*а*) и функциональному назначению (*б*) выводов

Таблица 6

Обозначение и наименование выводов контроллера байтового параллельного интерфейса

Вывод	Обозначение	Тип вывода	Наименование
1	REQ	Вход	Требование
2	RC0		Выбор режима 0
3	RC1		» » 1
4	RC2		» » 2
5	$\overline{AO\text{-}A}$		Готовность приемника
6	$\overline{AC\text{-}A}$		Запрос приемника
7	$\overline{AC\text{-}S}$	Выход	Запрос источника
8	$\overline{SC\text{-}A}$	»	Строб приемника
9...16	$\overline{AD0}$ ... $\overline{AD7}$	Вход-выход	Разряды адреса-данных
17...20	$\overline{AD8}$ ... $\overline{AD11}$	Вход	То же
21	GND	—	Общий
22	$\overline{AD12}$	Вход	12-й разряд адреса-данных
23	$\overline{BS}$		Внешнее устройство
24	$\overline{AD14}$		14-й разряд адреса-данных
25	$\overline{AD15}$	Выход	15-й разряд адреса данных
26	$\overline{SC\text{-}S}$	Вход	Строб источника
27	$\overline{IN}$	Выход	Ввод данных
28	$\overline{OUT}$		Вывод данных
29	$\overline{SET}$		Начальная установка
30	ERR	Вход	Ошибка
31	DONE		Завершено
32	TR		Требование передачи
33	$\overline{IAKI}$		Предоставление прерывания
34	$\overline{INIT}$	Вход	Установка
35	$\overline{VIRQ}$	»	Требование прерывания
36	$\overline{IAKO}$	Выход	Предоставление прерывания
37	$\overline{DOUT}$	Вход	Вывод данных (по каналу)
38	$\overline{DIN}$		Ввод данных (по каналу)
39	$\overline{RPLY}$		Ответ
40	$\overline{SO\text{-}S}$		Готовность источника
41	$\overline{SYNS}$		Обмен
42	U_{cc}		Напряжение источника питания

жений определенного уровня на выводы RC0 ... RC2 (табл. 7). Прием и передача данных осуществляются с помощью регистров состояния источника РСИ и приемника РСП, регистров источника РИС и приемника РП. Регистры РИС и РП выполняются на одной микросхеме К1801ВП1-034, а БИС К1801ВП1-033 содержит регистры РСИ и РСП, компаратор адресов, блоки прерываний и управления регистрами РИС и РП.

Формат регистра состояния источника показан на рис. 15. Назначение отдельных разрядов следующее:

Таблица 7

Адреса регистров и векторов прерываний

Вывод	Уровень сигнала	Адрес РСИ	Адрес РП	Адрес РСП	Адрес РИ	Адрес вектора прерывания
RC0 RC1 RC2	Низкий " "	—	—	177514	177516	200
RC0 RC1 RC2	Высокий Низкий "	177560	177562	177564	176566	И 60 П 64
RC0 RC1 RC2	Низкий Высокий Низкий	177550	177552	177554	177556	И 70 П 74
RC0 RC1 RC2	Высокий " Низкий	177570	177572	177574	177576	И 170 П 174
RC0 RC1 RC2	Низкий " Высокий	XXXXX0	XXXXX2	XXXXX4	XXXXX6	И XX0 П XX4

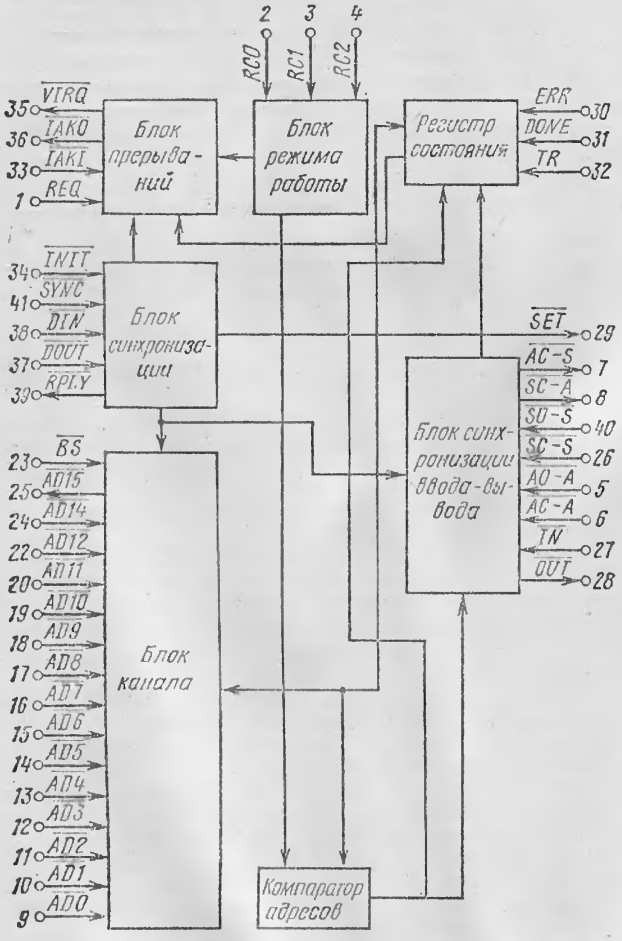


Рис. 14. Электрическая структурная схема контроллера байтового параллельного интерфейса

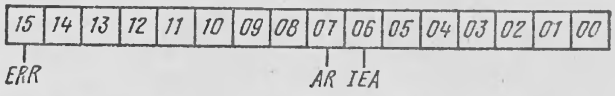


Рис. 15. Формат регистра состояния источника

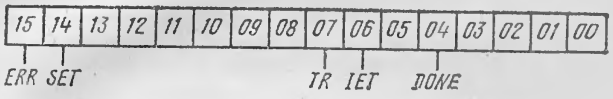


Рис. 16. Формат регистра состояния приемника

IEA — разрешение прерывания по приему; сбрасывается сигналом $\overline{\text{INIT}}$ (для чтения и записи);

AR — требование приема (только для чтения);

ERR — ошибка (только для чтения).

Формат регистра состояния приемника представлен на рис. 16. Назначение разрядов, обозначенных на рисунке, следующее:

DONE — завершено (только для чтения);

IET — разрешение прерывания по передаче; сбрасывается канальным сигналом $\overline{\text{INIT}}$ (для чтения и записи);

TR — требование передачи (только для чтения);

SET — начальная установка (только для записи);

ERR — ошибка (только для чтения).

Работа микросхемы на прием информации (рис. 17). В рабочее состояние микросхема переводится сигналом $\overline{\text{SO-S}}$. При появлении этого сигнала вырабатывается сигнал $\overline{\text{AC-S}}$ (запрос источника), ответом на который служит сигнал $\overline{\text{SC-S}}$ (строб источника). При поступлении последнего устанавливается бит AR (требование приема) в регистре состояния источника, который при наличии бита IEA (разрешение прерывания по приему) вызывает запрос прерывания. В цикле чтения регистра приемника вырабатывается сигнал $\overline{\text{IN}}$ (ввод данных) и снимается сигнал $\overline{\text{AC-S}}$. Вновь

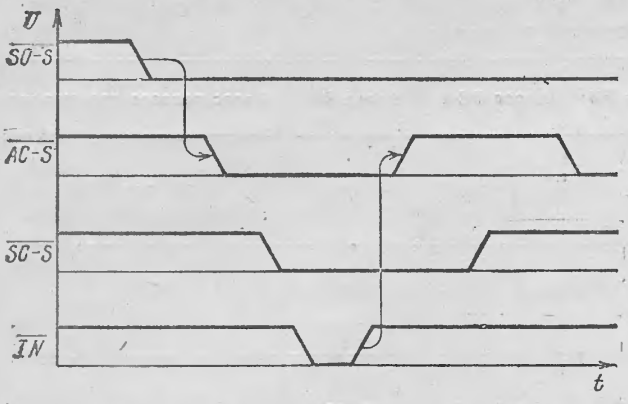


Рис. 17. Временна́я диаграмма цикла «прием данных»

установиться сигнал AC-S может только после снятия сигнала SC-S. При отсутствии сигнала SO-S (готовность источника) в регистре РСИ установлен бит ERR — микросхема к работе не готова.

Работа микросхемы на передачу информации (рис. 18). Готовность к передаче данных наступает при наличии сигналов AO-A (готовность приемника) и AC-A (запрос приемника). После записи данных в регистр источника микросхема вырабатывает сигнал SC-A (строб приемника), который сбрасывается после снятия сигнала AC-A. Во время записи в регистр

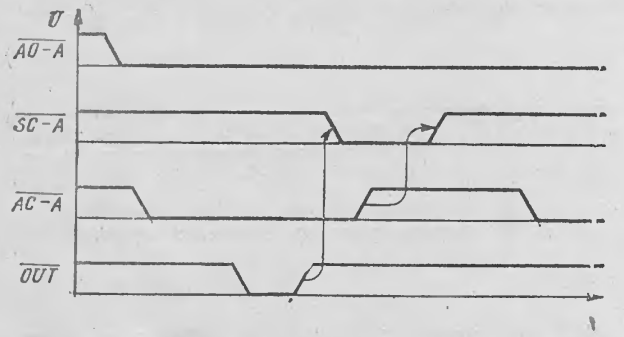


Рис. 18. Временна́я диаграмма цикла «передача данных»

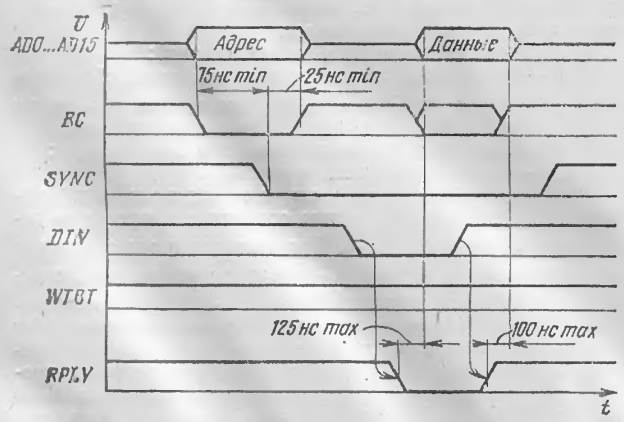


Рис. 19. Временна́я диаграмма цикла «ввод»

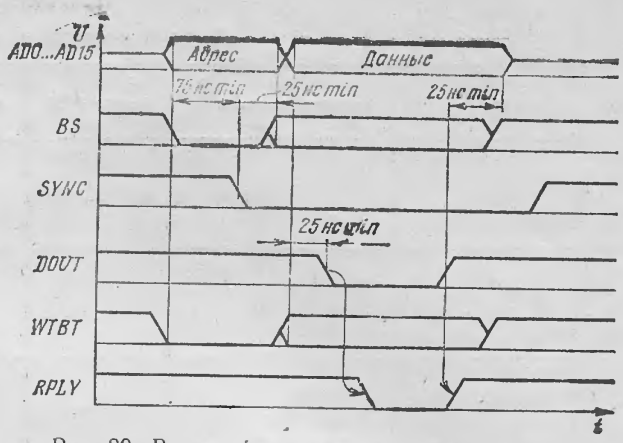


Рис. 20. Временна́я диаграмма цикла «вывод»

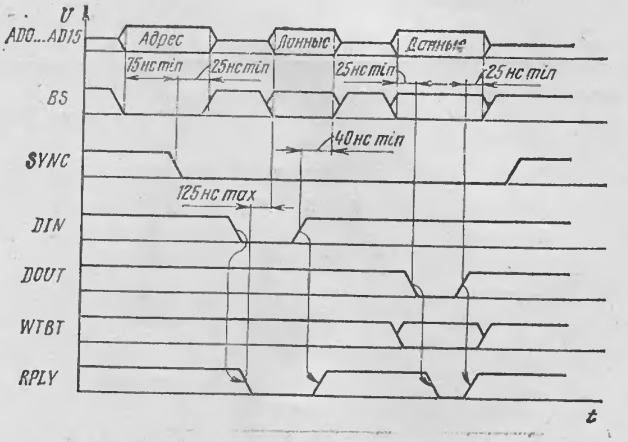


Рис. 21. Временна́я диаграмма цикла «ввод-пауза-вывод»

источника устанавливается сигнал OUT (вывод данных).

Состояния сигналов ERR, DONE, TR регистра состояния приемника соответствуют состояниям одноименных сигналов приемника. Запрос на прерывание по передаче данных возникает по сигналу REQ (требование) к установленному биту IET (разрешение прерывания по передаче) в регистре состояния.

Путем внешнего соединения вывода REQ с каким-либо выводом состояния приемника можно вызвать запрос по любому состоянию приемника. Запись Лог.1 в разряд 14 регистра состояния приемника и канальный сигнал INIT вызывают сигнал SET (начальная установка).

Основные циклы работы микросхемы К1801ВП1-033 иллюстрируются временны́ми диаграммами, приведенными на рис. 19—21. Обозначение Т соответствует передатчику, R — приемнику.

При использовании микросхемы выводы на периферийные устройства необходимо развязывать буферными ТТЛ-элементами. Допустимое значение электростатического потенциала не более 100 В. Рекомендуемые схемы подключения микросхем К1801ВП1-033 и К1801ВП1-034 при совместной работе будут приведены после описания всех интерфейсных БИС комплекта.

Г. Г. Глушкова

Телефон 208-73-23, Москва

УДК 621.328.3.049.776

Донев В. С. **Микропроцессорный комплект БИС серии СМ600** // Микропроцессорные средства и системы.— 1988.— № 5.— С. 3.

Характеризуются состав и возможности выпускаемого в Народной Республике Болгарии МПК БИС серии СМ600. Часть микросхем можно использовать в составе МП систем, построенных на МПК БИС серии КР580. Система команд МП СМ601 полностью совместима с системой команд МП МС6800 фирмы MOTOROLA.

UDC 621.328.3.049.776

Donev V. S. **CM600 microprocessor LSI family.**// Microprocessor devices and systems.— 1988.— N. 5.— P. 3.

The contents and general application fields of Bulgarian microprocessor LSI family SM600 are listed. Some LSI chips may be integrated into KP580 family-based systems. The SM601 microprocessor instruction set is fully compatible with MOTOROLA 6800 one.

УДК 681.323

Гуревич М. Х., Кузнецов А. Л., Луцкий И. Ю. Страутманис Г. Ф. **Однокристальная ЭВМ КМ1813ВЕ1 с аналоговыми устройствами ввода-вывода**//Микропроцессорные средства и системы.— 1988.— № 5— С. 5.

Подробно описана однокристальная ЭВМ КМ1813ВЕ1 со встроенными аналоговыми устройствами ввода-вывода и репрограммируемой памятью программ.

UDC 681.323

Gurevich M. H., Kuznetsov A. L., Lutsky I. Yu., Strautmanis G. F. **The single-ship microcomputer KM1813BE1 with analogue I/O ports.**//Microprocessor devices and systems.— 1988.— N. 5 — P. 5.

The structure and operation of single-ship microcomputer KM1813BE1 with integral DAC and ADC as I/O devices and built-in reprogrammable programm ROM are explained in detail.

УДК 681.327.8

Таланов В. А. **МикроЭВМ с маркой ВЭФ** // Микропроцессорные средства и системы.—1988.—№ 5.—С. 11.

Рассматриваются программные и аппаратные средства микропроцессорной вычислительной техники систем передачи и обработки информации, созданные на ПО ВЭФ им. В. И. Ленина и внедренные на производстве.

UDC 681.327.8

Talanov V. A. **Microcomputers with "VEF" trademark.**//Microprocessor devices and systems.— 1988.— N. 5—P. 11.

The microprocessor-based hardware and software products for data communication and processing developed and manufactured by VEF firm are described.

УДК 681.32.06

Домарацкий С. И., Шраго И. Л. **Организация экрана монитора ПЭВМ «Искра 1030» в программах автоматизации научных исследований** // Микропроцессорные средства и системы.— 1988.— № 5.— С. 13.

Из представленных трех типов пользовательского интерфейса выделен наиболее предпочтительный для диалога с ПЭВМ (с точки зрения пользователя). На языке ассемблера предложены листинги процедур взаимодействия с дисплеем, которые могут служить строительными блоками для программирования диалога в прикладных программах.

UDC 681.32.06

Domaratsky S. N., Shrago I. L. **The screen composition in scientific automation programs on "ISKRA 1030" personal computer**//Microprocessor devices and systems.— 1988.— N. 5— P. 13.

One of three user-frienly software interface types is selected as most preferrable for the user. The Assembly-language procedures supporting keyboard and screen interaction are proposed, which may serve as building blocks for dialogue programming in application programs.

УДК 681.3.06

Гнездилова Г. Г., Гончаров О. А., Сенин Г. В. **К типологии компьютерных игр**//Микропроцессорные средства и системы.—1988.— № 5— С. 22.

Даются назначение, классификация и описание нескольких типов компьютерных игр.

UDC 681.3.06

Gnezdilova G. G., Goncharov O. A., Senin G. V. **On the typology of computer games.**// Microprocessor devices and systems.— 1988.— N. 5— P. 22.

The general aim, classification and scenarium of several computer game types are discussed.

УДК 681.326

Александров А. В. **Автоматизированная система по выбору элементной базы для технологических контроллеров** // Микропроцессорные средства и системы.— 1988.— № 5.— С. 30.

Рассматриваются критерии разработки, а также экономические и функциональные характеристики автоматизированной экспертной системы по выбору элементной базы для технологических контроллеров.

UDC 681.326

Aleksandrov A. V. **Computerized system chooses component base for technological controllers.**//Microprocessor devices and systems.— 1988.— N. 5 — P. 30.

The project design criteria as well as technical and economical characteristics of computerized expert system, which helps to make perfect choise of component base for technological controllers, are described.

УДК 681.3.06 : 322.1

Бернов А. В., Поротов В. Н. **Аппаратно-программный комплекс программирования и отладки контроллеров на основе микропроцессоров серии К1801** // Микропроцессорные средства и системы.— 1988.— № 5.— С. 35.

Описан комплекс, обеспечивающий как автономную отладку прикладных программ, так и комплексную отладку ПО и аппаратных средств контроллеров на основе микропроцессоров серии К1801.

UDC 681.3.06 : 322.1

Bernov A. V., Porotov V. N. **Hardware-software system for K1801 microprocessor based controllers programming and debugging.**//Microprocessor devices and systems.— 1988.— N. 5 — P. 35.

The system enabling both stand-alone application programm debugging and interaction of hardware and software debugging in controllers built around K1801 microprocessor.

УДК 681.325—181.5

Пальгин П. И., Рождественский С. М., Шагурин И. И. **Микропрограммируемый комплекс для тестирования и отладки микропроцессорных систем** // Микропроцессорные средства и системы.— 1988.— № 5.— С. 39.

Рассмотрены комплекс для отладки и тестирования микропроцессорных систем на базе секционированных микропроцессоров с микропрограммным управлением.

UDC 681.325—181.5

Palgin P. I., Rozhdestvensky S. M., Shagurin I. I. **Microprogrammable unit for testing and debugging of microprocessor systems.**//Microprocessor devices and systems.— 1988.— N. 5— P. 39.

The system for testing and debugging of bit-slice microprogrammable microprocessor devices is described.

УДК 681.3.06

Бродин В. Б., Шагурин И. И. **Микропрограммируемый схемный эмулятор для отладки микропроцессорных систем** // Микропроцессорные средства и системы.— 1988.— № 5.— С. 49.

Характеризуется оригинальный универсальный схемный эмулятор на основе микропрограммируемого микропроцессорного комплекта К1804. Эмулятор настраивается на функции 8-разрядных микропроцессоров и однокристальных микроЭВМ. Он позволяет свести к минимуму объем аппаратуры, изменяемой при переходе от одного типа микропроцессора к другому, и обеспечивает доступ ко всем внутренним ресурсам эмулируемого микропроцессора, в том числе в реальном масштабе времени.

UDC 681.3.06

Brodin V. B., Shagurin I. I. **Microprogrammable circuit emulator for microprocessor systems development.**//Microprocessor i devices and systems.— 1988.— N. 5— P. 49.

An original universal circuit emulator using AMD2900-compatible microprocessor chip set is characterized. The emulator can be configured for simulation of most 8-bit microprocessors and single-chip computers, and helps to reduce hardware necessary for progect developing with various microprocessor types. It enables easy access to all internal resources of the microprocessor being emulated, even in real time scale.

УДК 681.327.8—181.4

Жукинский И. Н., Логвиненко И. В., Скрипник И. Ю., Цымбурская Л. В. **Волоконно-оптический интерфейс в адаптивной системе диагностики электроустановок** // Микропроцессорные средства и системы.— 1988, № 5— С. 82.

Рассматривается многоканальная система диагностики электроустановок, управляемая микроЭВМ по волоконно-оптическим линиям связи. В системе используются устройства «Электроника МС 8201», «Электроника МС 4101», «Электроника МС 2702». Приводится функциональная схема диагностической системы и программа избирательного опроса датчиков.

UDC 681.327.8—181.4

Zhukinsky I. N., Logvinenko I. V., Skrypnik I. Yu., Tsimburskaya L. V. **Fiber-optics interface of the adaptive diagnostic system for electrical installations.**//Microprocessor devices and systems.— 1988.— N. 5— P. 82.

The multichannel diagnostic system for electrical installations controlled by microcomputer via fiber optical cables is described. The system incorporates "Electronika MC2702" micpocontroller and "Electronika MC 8201" and "ELECTRONICA MC4101" I/O units. The block diagram of the whole system and selective sensor polling program are included.

УВАЖАЕМЫЕ ЧИТАТЕЛИ,

специалисты радио-, электронной и вычислительной техники

С 1989 года открыта широкая подписка на научно-технический сборник **«Электронная промышленность»** и отдельно на его раздел **«Новости. Сообщения. Хроника».**

Сборник публикует обзоры, статьи, информационные и рекламные сообщения по всем вопросам отраслевой экономики, финансовой деятельности, науки, техники и производства. Благодаря инженерной направленности сборник может служить пособием для разработчиков, руководителей предприятий и подразделений, работников высшей школы. Доступность изложения материала делает его интересным для широкого круга читателей. Подписная цена 32 руб. 90 коп. за 12 номеров. Индекс 3833.

Отдельное издание **«Новости. Сообщения. Хроника»**, которое входит также в основной сборник («Журнал в журнале»), представляет информацию о новых ИЭТ, оборудовании, материалах; ежемесячный дайджест информации по зарубежной электронике; хронику мероприятий научно-технической пропаганды; рекламу на товары народного потребления. Подписная цена 11 руб. за 10 номеров. Индекс 3833,1.

Подписную сумму следует направить почтовым переводом по адресу: 117415, Москва, пр. Вернадского, 39, ЦНИИ «Электроника». В почтовом переводе указать: подписка на 1989 г., название и индекс издания, фамилию, почтовый индекс отделения связи, адрес абонента.

Квитанция об оплате перевода является гарантией Вашей подписки.

Справки по телефону: 432-92-27, Москва

КОМПЛЕКС МИКРОПРОЦЕССОРНЫХ СРЕДСТВ МЕДИЦИНСКОЙ ТЕХНИКИ

(См. ст. Мамджяна Г. Г., Наумчика М. П., Прохоренко А. Н., Розмана Б. Я. и др. в № 6, 88)

ВНИИ медицинского приборостроения разработан и выпускается комплекс аппаратных и программных средств, обеспечивающих процессы проектирования и производства электронных приборов и аппаратов (прежде всего медицинского назначения) с встроенными микропроцессорами. Комплекс включает:

8- и 16-битные наборы встраиваемых унифицированных микропроцессорных модулей (ТЭЗов);

средства проектирования, отладки и тестирования;

системное, прикладное, технологическое программное обеспечение;

унифицированные конструктивы.

Достоинства:

широкий спектр предоставляемых ресурсов (набор встраиваемых модулей, средств и систем проектирования и производства, программного обеспечения, документации);

комплексное решение основных аспектов проектирования и производства интеллектуальной аппаратуры на совместимой схемотехнической и программной базе, стандартных Евроконструктивах, стандартном системном интерфейсе И-41;

возможность расширения, открытость, доступность.

Возможности сотрудничества:

передача технологии проектирования микропроцессорных приборов и аппаратов;

изготовление и поставка автоматизированных рабочих мест разработчика;

передача производственной документации на встраиваемые модули и отладочные средства;

выполнение заказов на единичный и мелкосерийный выпуск микропроцессорных модулей;

изготовление и организация рабочих мест для отладки и тестирования модулей в цеховых условиях;

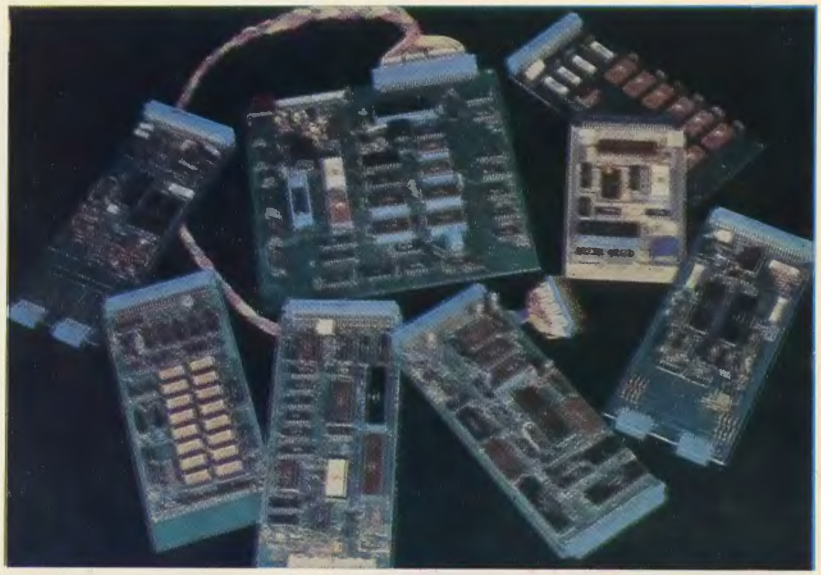
разработка конкретной микропроцессорной аппаратуры;

сопряжение приборов и аппаратов.

Области применения: медицинская техника; автоматизированные приборы различного назначения; управляющие контроллеры; АСУ ТП.



Прототипная система с программатором ППЗУ



Микропроцессорные модули, входящие в набор



Средства отладки модулей

1р. 10к.

Индекс 70588



intermicro

СССР – АВСТРИЯ

СОВМЕСТНОЕ

ИНТЕРМИКРО

ПРЕДПРИЯТИЕ



Учредителями совместного советско-австрийского предприятия «Интермикро» являются с австрийской стороны — фирма «Просистем», с советской — Всесоюзный научно-исследовательский институт проблем вычислительной техники и информатики ГКВТИ СССР.

Совместное предприятие «Интермикро» будет предоставлять заказчику широкий перечень работ и услуг в области вычислительной техники и информатики.

СПЕЦИАЛИЗИРУЕТСЯ — в решении всех вопросов заказчика по разработке, выбору, приобретению, аренде и обслуживанию вычислительных систем и программных продуктов.

ПРЕДЛАГАЕТ — необходимую Вам конфигурацию вычислительных и программных средств; получение дополнительной прибыли за счет применения современных вычислительных систем.

ОБЕСПЕЧИВАЕТ — поставку «под ключ» и сдачу в аренду вычислительных систем и программных продуктов; обучение специалистов работе с современными вычислительными средствами, пакетами, программами и технологиями; гарантийное и послегарантийное обслуживание вычислительных систем и сопровождение программых продуктов; поставку необходимой документации на русском и других языках.

В КРАТЧАЙШИЕ СРОКИ ВЫПОЛНЯЕТ — согласование конфигурации вычислительного комплекса и программного продукта для конкретного пользователя; поставку технических и программных средств «под ключ»; обучение пользователей работе с вычислительными и программными средствами; разработку программного продукта.

В дальнейшем планируется развитие совместного производства механических средств вычислительной техники в виде специализированных узлов, которые позволят повышать производительность и расширять функциональные возможности вычислительных систем для максимального удовлетворения потребностей заказчика.

Планируется также создание сети учебных центров по обучению специалистов работе с современными средствами вычислительной техники и технологией, игровых залов для предоставления вычислительных услуг населению, магазинов по продаже средств вычислительной техники, а также разработка технических средств для предоставления населению вычислительных услуг на базе региональных вычислительный сетей ГКВТИ СССР, предоставляющих доступ к информационным базам данных по регионам СССР и за рубежом.

В перспективе предполагается открытие филиалов совместного предприятия в СССР и за рубежом.

НАШИ РЕКВИЗИТЫ:

СССР, 111024, Москва, Авиамоторная, 26/5 ИНТЕРМИКРО, тел.: 252-52-66
телекс 411035 FOTON SU

ПО ЛЮБЫМ ВОПРОСАМ МЫ РАДЫ ВАС ВИДЕТЬ В ИНТЕРМИКРО.

ЖДЕМ ВАС

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ · Москва · 1988 | 5