



МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

1-2 / 1990

ISSN 0233-4844

Однокристалльный контроллер гибких дисков КР1810ВГ72А для записи, чтения и форматирования дисков с одинарной и двойной плотностью

«Искра 4816»—ряд многопроцессорных ПЭВМ, рассчитанных на рабочие станции и управляющие машины

Контроллер на основе ОЭВМ КМ1816ВЕ48 позволяет использовать бытовые магнитофоны в качестве внешнего устройства долговременного хранения данных для ЭВМ с интерфейсом ИРПР

ТВ-приемники — в качестве видеомониторов персональных компьютеров: способы сопряжения ПЭВМ с ТВ-приемниками на примере «Электроники БК-0010» и переносных телеприемников серии «Электроника»

Серийные модули отображения СМ1800 для преобразования данных в видеосигнал и построения символьных и графических изображений на черно-белых и цветных индикаторах



УНИВЕРСАЛЬНАЯ МИКРОПРОЦЕССОРНАЯ СИСТЕМА УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ЛАЗЕРАМИ

Универсальная микропроцессорная система управления (МПСАУ) технологическими лазерами обеспечивает реализацию следующих функциональных возможностей: автоматическое включение, выключение и вывод на рабочий режим; стабилизацию и оптимизацию основных параметров различных режимов работы лазера; контроль и блокировку аварийных ситуаций; диагностику возникающих неисправностей; информационное обслуживание оператора; программируемость основных технологических режимов работы; встраиваемость в лазерное оборудование; простое сопряжение с оборудованием лазерных технологических комплексов.

Система состоит из крейта БУК-Б с набором модулей (рис. 1) и выносного интеллектуального пульта управления. Напряжение питания модулей — 5 В, 5 А; 12 В, 1А; —12 В, 1 А. Все модули и источник питания выполнены в соответствии с рекомендациями Евромеханики размерами 233,4×160 мм. В состав модулей входят: 8-разрядный микроконтроллер на основе МПК БИС серии КР580; 16-разрядный микроконтроллер на основе МП К1801ВМ1 и периферийных БИС серии КР580. Оба микроконтроллера имеют одинаковую архитектуру и содержат ОЗУ емкостью 8 Кбайт, РПЗУ — 16 Кбайт, три последовательных канала типа RS-232C, 8-уровневую систему прерываний.

С объектом управления система связана через модули СО: дискретного ввода-вывода, АЦП-ЦАП, аналоговых компараторов, управления шаговыми двигателями. При необходимости разрабатываются дополнительные специализированные модули.

Пульт управления (рис. 2), выполненный на основе МПК БИС серии КР580, обеспечивает ввод с псевдосенсорной клавиатуры (до 64



Рис. 1. Крейт БУК-Б с набором модулей



Рис. 2. Пульт управления

клавиш), отображение основных параметров на четырех 4-разрядных цифровых индикаторах, индикацию основных блокировок и состояние лазера на светодиодах (до 64), предварительную интеллектуальную обработку введенной с клавиатуры информации и передачу ее в микроконтроллер.

На базе комплекта модулей (обычно 4...6) и пульта управления создается оптимальный вариант системы управления для конкретного типа лазера. Размеры МПСАУ — 280×270××220 мм, пульта управления — 450×330××70 мм.

Программное обеспечение МПСАУ ориентировано на определенный тип лазера. Оно включает драйверы ввода-вывода, программы управления процессами в режиме реального времени, библиотеки специальных программ, микробibliotheki, позволяющие автоматически генерировать основной объем прикладного ПО и т. д. Для полной отладки ПО созданы специализированные стенды-имитаторы лазеров, включающие инструментальную ЭВМ, имитатор объекта управления, интерфейсы для загрузки программ в отладочное ОЗУ микроконтроллера, программаторы ПЗУ.

Описываемая МПСАУ включена в состав различных технологических СО₂-лазеров типа ТЛ-1,5, МКТЛ-2, ТЛ-5М и может быть рекомендована к применению во всех новых разработках лазерной техники, включая экспериментальные, медицинские лазеры, лазеры для контроля, лазерное оборудование и лазерные технологические комплексы, а также может быть применена для автоматизации различного технологического оборудования и АСУТП.

140700, Шатура, Главпочтамт, НИЦТЛ АН СССР, отдел автоматизации и вычислительной техники; тел. 61-266 доб. 1-93, 2-85. Нижаде-Гавгани Э. А.

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года

ММП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД

НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 1/1990 МОСКВА

СОДЕРЖАНИЕ МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Мизин И. А.— На пути к системе персональных ЭВМ	2
Сивобород П. В., Юрченко И. В., Петренко Н. В., Ножницкий А. А., Ерошенко Н. И.— Универсальная 8-разрядная микроЭВМ КР1830ВЕ48	6
Коваленко В. А., Москалевский А. И., Солдатенко Л. М.— Контроллер гибкого магнитного диска КР1810ВГ72А	7
Воробьев Н. В., Кремлев В. Я., Меликов Г. К.— Однокристалльный оптоэлектронный преобразователь для графического манипулятора	17

ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

Бабурин А. А., Воробьев А. Д., Зильберберг И. С., Кнеллер Э. Г., Наволенко Г. Н., Павлюченко Н. П., Пац В. Б., Пчелкина Е. И.— Особенности архитектуры ПЭВМ «Истра 4816»	20
Абрамов В. Г., Брябрин В. М., Горячая И. В., Щеглова Т. Н.— ЭРУДИТ — инструмен- тальная система поддержки обучения на базе ПЭВМ	23

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Корнеев В. В., Монахов О. Г.— Параллельный Паскаль для вычислительных систем с программируемой структурой	28
Брусиловский Л. И., Скляров В. В.— Система файлового обмена KERMIT для орга- низации межмашинной связи по терминальным линиям	30
Гусев А. А., Озерец Л. С.— Драйвер для обмена данными между ОС ЕС и RT-11	33
Беркуцкий В. Я., Поливаний И. В.— Диалоговая инструментально-учебная система программирования	35
Мурсгов В. Н.— Имитационное моделирование микропроцессорных систем на базе Е-сетей для ПЭВМ	36

Операционные системы

Нафиков В. Р., Рабинович М. И.— Ядро Модуля 2 ориентированной операционной системы для встраиваемых микропроцессорных систем	37
Бульонков М. А., Дудоров Н. Н.— Интегрирующая среда непосредственного взаимо- действия в системе UNIX	40

Опыт разработки аппаратно- программных средств

Мышкин И. Л., Заречнюк Г. О.— Фортран-80 для разработки программ встроенных систем	46
Коршун И. Г., Хорошевский А. Н.— Автоматизация программирования БИС ПЗУ специализированных микропроцессорных устройств	47
Назаров С. В., Луговец А. В.— Исследование производительности ЭВМ с магистраль- ной архитектурой	49
Солонин В. Ю.— Самоконтроль универсальных программаторов	51

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Врублевский Ю. Н., Осипенко Ю. П.— ТВ-приемники — в качестве видеомониторов персональных компьютеров	54
Труханов А. И., Наумчик М. П., Чернин С. Л., Покровский Ф. Г., Ермоленко В. Г.— Малогабаритные ультразвуковые диагностические приборы на основе однокри- сталльной ЭВМ	60

Микроконтроллеры

Дианов А. П., Щелкунов Н. Н.— Одноплатные микроЭВМ для встроенных систем	64
Журавский Д. А., Барков С. Н.— Модульная микроЭВМ	68
Ходаковский Е. А.— Контроллер прерываний	72

Периферийные устройства

Асанов Р. Ш., Никитин А. С.— Имитатор интерфейса МПИ	74
Бескровный В. Г., Урсатьев А. А., Сапожникова С. Л., Шестаковский М. А.— Накопи- тель СМ5211 — внешнее ЗУ микроЭВМ «Электроника 60»	78
Давыдов В. А., Козлов В. М., Попов В. Г.— Одноплатный контроллер магнитной ленты	80
Копылов В. И., Меркулов М. Ю.— Эмулятор диска в стандарте КАМАК	84

ЦЕНТР УЧЕБНЫЙ

Белильников В. Д., Визгин Ю. И., Лебедев В. И., Лопатка К. Н.— Модуль пульта оператора измерительного прибора	86
--	----

СЛОВО К ЧИТАТЕЛЮ

Перед вами первый номер журнала «Микропроцессорные средства и системы», датированный 1990 годом. Конечно, наш постоянный читатель узнает его по внешнему виду, набору рубрик, тематике и содержанию материалов — все это, в основном, осталось без изменений. Надеемся, многие наши подписчики одобрят такое постоянство редколлегии, редакционного совета. Ведь, судя по вашим письмам, «Микропроцессорные средства и системы» стали для многих из вас привычным спутником в жизни, вошли в круг повседневного чтения.

Однако любой консерватизм формирует привычку не только к хорошему, но и к недостаткам. Не относится ли это и к нашему журналу? Чтобы избавиться от стереотипов мышления, инерции оценок и подходов к своей работе, мы решили вызвать на себя огонь нашей критики.

В последнее время мы стали получать много писем и, признаться, искренне рады укреплению связи с читателями. Но пока, видимо, рано подводить итоги. Правильнее настроить наших читателей на откровенный конструктивный критический лад и попросить побыстрее сообщить нам, что нравится, что не очень, и как сделать журнал еще лучше, еще интересней. Мы рассчитываем обратиться к вам в этом году с анкетами-вопросами и надеемся извлечь для себя из ваших ответов много ценного. Заранее благодарны всем.

И все-таки кое-какими предварительными соображениями хочется поделиться уже сейчас. Так, мы заметили, что в своих письмах вы все чаще касаетесь не только конкретных вопросов науки, техники, производства, связанных с классическим микропроцессорным направлением, но затрагиваете и некоторые общие проблемы развития мировой и отечественной информатики, вычислительной техники, кризисных и «горячих» точек. И это естественно. Сегодня специалист, каким бы узким он ни был, все теснее соприкасается со сложным и противоречивым «внешним» миром, вынужден считаться с его реалиями, да и просто черпать из окружающей жизни импульсы для творчества. И к нам, стало быть, пришла гуманизация, социализация, политизация науки и техники. До сих пор, как вы знаете, это не входило в нашу задачу. И вот теперь мы склонны, разумеется при вашем одобрении, выносить на страницы журнала и ключевые проблемы информатизации, публиковать представляющие несомненный интерес обобщающие информационные материалы.

Еще соображение. Сегодня очевидно: наше дальнейшее продвижение в области информатики, вычислительной техники, микропроцессорных средств и систем не может происходить в отрыве от развития этих направлений в других индустриально развитых странах мира, без опоры на международное разделение труда. Можем ли мы тогда равнодушно относиться к накопившемуся в этой области зарубежному опыту? По-видимому, надо смелее знакомить читателя с обзорами современной зарубежной техники, практикой ее применения, чаще помещать переводы наиболее интересных материалов из авторитетных зарубежных источников. Может быть, в интересах дела ввести с этой целью в состав редколлегии зарубежных членов?

Далее, редакционный совет, редколлегия хотели бы обсудить с вами вопрос о выпуске целевых приложений к журналу, раскрывающих проблемы и современное состояние таких актуальных направлений и сфер информатизации, как, скажем, образование, здравоохранение, охрана окружающей среды... Возможно, придется по душе и идея выпуска целевых приложений по заявкам читателей. В них можно было бы воспроизводить, например, разработанные в СССР на уровне мировых стандартов смежные решения, программные средства, в которых, мы знаем точно, нуждаются многие.

Вот и посоветуемся: какие потребуются новые рубрики? Каков, хотя бы приблизительно, перечень особо животрепещущих тем, нуждающихся в первоочередном освещении? Каков круг авторов желательно привлечь для этой работы?

Мы, конечно, осознаем трудность задач, которые собираемся взваливать на свои плечи. Сложность не только в том, что нам предстоит освоить новую проблематику. Сложность и в том, как втиснуть ее в рамки издания, выходящего шесть раз в год, без ущерба для основной — «микропроцессорной» части. Возможно, потребуется увеличить объем журнала. Может быть, лучше пойти на увеличение его периодичности — скажем, поставить вопрос о ежемесячном издании? А, может, целесообразно выпускать часть тиража в виде дискет, или распространять на дискетах только приложения к журналу и заказные программные средства — с тем, чтобы снизить расход бумаги и излишнюю типографскую нагрузку? Понятно, что с переходом на хозрасчет все эти вопросы требуют тщательной предварительной проработки.

Пожалуйста, присылайте свои соображения.

Ждем ваших писем.

Редакционный совет
Редакционная коллегия

Заместитель главного редактора
С. М. Пеленюк

Номер подготовили:

Е. И. Бабич, Г. Г. Глушкова,

В. М. Ларионова, Е. П. Граве

Корректор Е. М. Кучерявенко

Технический редактор

Г. И. Колосова

Адрес редакции журнала:

103051, Москва, Малый

Сухаревский пер., д. 9А

Телефоны: 208-73-23, 208-19-94

Сдано в набор 01.11.89

Подписано к печати 24.04.90

Формат 84×108^{1/16}

Офсетная печать Т-08918

Усл. печ. л. 10,08 Уч.-изд. л.

14,6. Тираж 87 000

Заказ 169 Цена 1 р. 10 к.

Орган Государственного

комитета СССР

по вычислительной технике

и информатике

Набрано на Ордена

Трудового Красного Знамени

Чеховском полиграфическом

комбинате

Изготовлен в

Московской типографии № 13

ПО «Периодика» ВО

«Союзполиграфпром»

Государственного комитета СССР

по печати

107005, Москва,

Денисовский пер.,

дом 30

НА ПУТИ К СИСТЕМЕ ПЕРСОНАЛЬНЫХ ЭВМ

Проблема разработки, организации массового производства и удовлетворения реально существующих потребностей в персональных ЭВМ (ПЭВМ) стоит в нашей стране очень остро, она определяет интеллектуальный уровень нации и в настоящее время из проблемы технико-экономической перерастает в проблему социально-политическую. Сегодня можно с уверенностью сказать, что ПЭВМ действительно входят во все сферы нашей жизни в качестве инструмента управления, усиления интеллектуального потенциала, средства обучения и помощи в быту и т. д.

Тридцать миллионов ПЭВМ — такова по оценкам МНТК «Персональные ЭВМ» потребность страны до 2000 г. Это, так сказать, по верхней оценке. Но и по нижней оценке получается весьма внушительная цифра — 17,5 миллиона штук. Значит, если даже исходить из минимальной потребности, то за XIII пятилетку надо произвести — 6...8 миллионов, а за XIV — 10...12 миллионов штук. В таких масштабах организовывать производство микропроцессорной вычислительной техники в нашей стране еще не приходилось.

Задача разработки и организации массового выпуска ПЭВМ впервые была сформулирована в Постановлении ЦК КПСС и Совета Министров СССР в январе 1986 г. Предусматривалось, что общий объем производства всех типов ПЭВМ в XII пятилетке составит 1,1 млн. шт. В настоящее время серийно выпускаются профессиональные ПЭВМ типа ЕС1840, ЕС1841, «Искра 1030», «Нейрон» и «Электроника-85», а также школьные ПЭВМ типа «Корвет» и «УКНЦ», о которых было рассказано в нашем журнале.

Исходя из реального положения, сложившегося в стране на конец 1985 года, можно считать обоснованным решение о производстве нескольких ПЭВМ «первой волны», достаточно разных по своим технико-экономическим характеристикам. Хотелось бы подчеркнуть вынужденность такого решения и вместе с тем целесообразность его в тот период. Оно допускало производство нескольких моделей одного класса, имеющих один прототип, и не ставило пока задачи создания унифицированной модели — в этом его вынужденность. Но в то же время оно учитывало наличие «задела» в промышленности, что позволяло выиграть в сроках — в этом его целесообразность.

Задача разработки и производства последующей унифицированной модели профессиональной ПЭВМ, получившей в дальнейшем шифр ЕС1842, была поставлена в 1987 году, в конце 1988 г. машина прошла государственные испытания, и в настоящее время организовано ее серийное производство. Профессиональные ПЭВМ ЕС1840, ЕС1841, ЕС1842, «Искра 1030» и «Нейрон», несмотря на определенные конструктивно-технологические отличия, основаны на единых архитектурных решениях, соответствующих в мире де-факто стандарту на мас-

шину ПЭВМ линии IBM — Intel-Microsoft, т. е. в них заложены соответствующие мировым тенденциям решения.

Известно, что микропроцессорная вычислительная техника может успешно развиваться только тогда, когда имеется высокий уровень развития во всех без исключения областях науки и техники, когда достигнут высокий уровень качества исходных материалов, физико-химических технологий, средств САПР, тестирования, отладки, измерения и испытаний. По всем этим компонентам, где немного, а где и существенно, наша страна отстает от развитых стран, а в совокупности это приводит к серьезному отставанию в вычислительной технике массового применения вообще. В то же время сегодня технический уровень определяется путем сравнения с лучшими зарубежными образцами. Стимулирующее воздействие такого принципа очевидно, однако, в погоне за лучшими образцами резко увеличивается стоимость разработки, а желаемый результат достигается далеко не всегда.

Исходя из наших реальных возможностей и общего уровня развития техники и технологии на период до 1995 г., при разработке ПЭВМ предпочтителен другой подход: проектирование ПЭВМ на основе технико-экономической целесообразности, предполагающей, во-первых, соответствие функциональных возможностей ПЭВМ требованиям областей массового применения в СССР, во-вторых, минимизации стоимости разработки и изготовления.

Технический уровень созданных и выпускаемых сегодня ПЭВМ нельзя признать удовлетворительным. Используемая элементная база по уровню интеграции и надежности существенно уступает аналогичной элементной базе в зарубежных моделях. Периферийные устройства имеют низкую надежность, существенно уступают зарубежным по технико-экономическим параметрам, в том числе по габаритам, массе функциональным характеристикам и стоимости. Предприятия-разработчики ПЭВМ плохо технически оснащены современными системами САПР для проектирования полужаказных БИС и СБИС, средствами тестирования, отладки, измерения и испытаний. В целом ПЭВМ «первой волны» по техническому уровню соответствует зарубежным моделям 7—8-летней давности.

МНТК «Персональные ЭВМ» совместно с Госкомитетом СССР по вычислительной технике и информатике и рядом промышленных министерств разработали Концепцию создания персональных (профессиональных и школьных) ЭВМ на 1989—1990 гг. и на период до 1995 г. В ней предложена и обоснована система ПЭВМ (СП ЭВМ) как совокупность совместимых (как минимум, снизу вверх) моделей ПЭВМ на основе единых схемотехнических, конструкторско-технологических и программных принципов. При разработке Концепции авторы исходили из того, что в мире сложился и объективно существует мировой де-факто

стандарт на 16-разрядные ПЭВМ, сформированный фирмами IBM — Intel-Microsoft. При этом речь идет в первую очередь о таких моделях ПЭВМ, как IBM PC, IBM PC/AT и совместимых с ними. Три компонента составляют основу этих ПЭВМ: базовый микропроцессор (Intel 8086, 8088, 80286), стандарт на системную шину (XT — BUS, AT — BUS — ISA) и единая операционная система (MS DOS).

Концепция рассмотрена и одобрена 28 апреля 1989 г. на совместном заседании научно-технического совета ГКВТИ СССР и Бюро Отделения информатики, вычислительной техники и автоматизации АН СССР с участием промышленных отраслей и составляет основу Программы развития отечественного парка ПЭВМ, периферийных устройств, сетевых средств и программного обеспечения.

Следование мировым стандартам не обернется запрограммированным отставанием, ибо так поступают во всем мире, следуя установившейся практике. В результате происходит объединение усилий стран мирового сообщества на наиболее перспективных направлениях развития вычислительной техники. При этом экономятся материальные и трудовые ресурсы, главным образом, за счет использования совместно разработанного программного обеспечения. Так, например, для IBM, совместимых ПЭВМ, к 1989 г. создано программного обеспечения более чем на 30 млрд. долларов.

В Концепции отмечено, что на период до 1995 г. основную долю парка 16-разрядных ПЭВМ в СССР должны составлять машины типа унифицированной модели ЕС1842 (в дальнейшем ЕС1842), а также модели АТ-подобной ПЭВМ.

Значительно более сложная ситуация в мире сложилась с 32-разрядными ПЭВМ, где мировые лидеры пока не сумели сформулировать единый стандарт, который был бы поддержан другими фирмами-производителями ПЭВМ. В Концепции значительное место уделено обоснованию архитектуры перспективных отечественных моделей 32-разрядных ПЭВМ на базе микропроцессоров типа Intel 80386, 80386SX, 80486 и др.

Бытовые и школьные ПЭВМ должны разрабатываться одновременно с профессиональными ПЭВМ и представлять собой усеченные варианты по функциональным и техническим параметрам. В результате бытовые и персональные ПЭВМ будут доступны по цене для индивидуального использования в быту и образовании, а единство элементарно-конструктивной базы создадут высокую надежность и необходимые функциональные характеристики. Это обеспечит преемственность и непрерывность образования в нашей стране.

Учитывая отставание СССР в области микро-

процессоров, предлагается создавать модели ПЭВМ сначала на основе импортных из-за рубежа микропроцессоров, а затем на базе отечественных. Опережающими темпами должна вестись разработка всей номенклатуры периферийных устройств, технических и программных средств, необходимых для объединения ПЭВМ в локальные сети.

В целях преодоления сложившегося в стране отставания в области ПЭВМ и выхода на мировой уровень в 1989—1995 гг. необходимо улучшить материально-техническую, опытно-экспериментальную и технологическую базу предприятий-разработчиков и изготовителей, оснастить их необходимыми средствами САПР, тестирования, отладки, измерения и испытаний, особое внимание уделить изготовлению полужаказных СБИС, разъемных соединителей, микросекретелей, кабельных изделий, электродвигателей, в химической, металлургической, бумажной и других отраслях — осуществить комплекс разработок по созданию чистых материалов, прецизионной электромеханики, точных технологий и др.

Предстоит создать мощности, обеспечивающие необходимый уровень технологии массового производства ПЭВМ и периферийных устройств для них, в первую очередь, мощности машиностроительных предприятий для производства специализированного технологического оборудования и контрольно-измерительной аппаратуры, а также мощности для массового производства элементной базы, в том числе БИС высокой степени интеграции.

Необходимо объединить все научные, конструкторско-технологические организации, занятые проблемой СП ЭВМ, мобильными экономическими связями путем финансирования всех разработок из единого центра, которым должен быть МНТК «Персональные ЭВМ». Финансирование должно направляться на решение определенных задач конкретным организациям. Основой организации массового производства ПЭВМ и периферийных устройств для них должен стать государственный заказ на весь планируемый выпуск с заданием технико-экономических и эксплуатационных характеристик конечной продукции без твердого закрепления всего объема производства только за отраслями-производителями, с предоставлением возможности выделения государственного заказа по серийному производству ПЭВМ министерствам-пользователям, совместным предприятиям и кооперативам. Для стимулирования массового производства ПЭВМ в государственном Заказе необходимо установить ряд нормативов, льгот и санкций, обеспечивающих его реализацию, ввести приоритетный порядок обеспечения материальными ресурсами всех предприятий, в том числе совместных и кооперативов, выпускающих ПЭВМ по государственному заказу. Важно также разработать мероприятия по подготовке массового внедрения ПЭВМ, в том числе правила использования ПЭВМ на рабочем месте.

В соответствии с ГОСТ 27201—87 каждая ПЭВМ должна быть снабжена комплектом базового программного обеспечения, включающим в себя ОС и базовым общесистемным прикладным программным обеспечением. Массовое производство и использование ПЭВМ предполагает массовое производство программных средств для них и, в первую очередь, базовых. Однако, если в отношении организации разработки и производства технических средств ПЭВМ предпринят ряд определенных шагов, то положение с программными средствами вряд ли можно признать удовлетворительным. В стране пока не создана инфраструктура промышленности программных средств ПЭВМ.

Предлагается при разработке программного обеспечения для 16-разрядных ПЭВМ ориентироваться на ОС типа MS DOS или разработанную в ИПИ АН СССР унифицированную операционную систему № «УОС». Обеспечивая полную совместимость с MS DOS, эта операционная система обладает рядом дополнительных возможностей, выгодно отличающих ее от прототипа (многозадачный режим работы, возможность использования различных систем кодировки данных, работа с расширенным объемом оперативной памяти — до 16 Мбайт, динамическая загрузка кодовых таблиц с различными символами для использования русского и др. национальных алфавитов, обеспечивается выполнение программ, созданных в операционных средах типа СРМ и др.). «УОС» и модули ее расширения, безусловно, не исключают возможность разработки ОС, подобной OS/2, и машиннезависимой ОС, подобной UNIX, которые могут стать наиболее применимыми ОС для 32-разрядных ПЭВМ как в мире, так и в СССР.

Распространяемые на отечественном рынке программные продукты, в первую очередь ОС, в основном являются результатом нелегальной адаптации западных прототипов. Следовательно, в случае принятия СССР обязательств о защите авторских прав в области программного обеспечения заводы-изготовители ПЭВМ будут поставлены в крайне тяжелое положение равно как и пользователи, уже применяющие указанные программные средства.

С целью изменения сложившейся ситуации необходимо незамедлительно начать разработку широкого спектра оригинальных программных продуктов, удовлетворяющих запросам внутреннего рынка: ОС, совместимых с наиболее популярными зарубежными; прикладных пакетов программ для пользователей-непрофессионалов; инструментальных средств для разработки прикладного программного обеспечения профессиональными программистами. Необходима четкая стратегия в области создания программного обеспечения СП ЭВМ.

Рассмотрим в самых общих чертах ее элементы. Во-первых, создание индустрии программного обеспечения должно рассматриваться как важнейшая народнохозяйственная задача. Решение этой задачи потребует значительных ресурсов и не может быть полностью осуществлено в сжатые сроки. Во-вторых, индустрия программного обеспечения должна быть сосредоточена на предприятиях трех уровней: ГКВТИ СССР и МНТК «Персональные ЭВМ»; отраслей и ведомств; вневедомственных.

В-третьих, в рамках ГКВТИ СССР и МНТК «Персональные ЭВМ» должны быть созданы специализированные предприятия по разработке программных средств. В-четвертых, работы по созданию программного обеспечения должны осуществляться в рамках Государственных целевых научно-технических программ (ГЦНТП), позволяющих проводить путем централизованного финансирования (через Госзаказ) рациональную политику в области программных средств. ГЦНТП должны строиться на конкурсной основе, финансироваться из средств ГКВТИ СССР и обеспечиваться необходимыми фондами ПЭВМ и средствами информатики для их реализации. В-пятых, необходимо разработать политику лицензионных закупок зарубежных программных продуктов.

В рамках межправительственной комиссии стран ЭВ по вычислительной технике для перспективного планирования и координации работ в области СП ЭВМ создано и успешно работает головная секция ГС-5. Секцией рассмотрена и утверждена Концепция развития 16- и 32-разрядных ПЭВМ и подготовлена Программа работ.

В стране во всех звеньях, связанных с разработкой, производством, внедрением и применением ПЭВМ, наметилась некоторая активизация, правда еще далекая от желаемой. Осуществление намеченных планов должно стать первоочередной задачей ГКВТИ СССР, Академии наук СССР и промышленных министерств. Всесоюзное общество информатики и вычислительной техники, средства массовой информации, и, в первую очередь, наш журнал внесут свой вклад в создание и применение СП ЭВМ.

На страницах журнала предполагается обсудить широкий спектр вопросов, связанных с разработкой, организацией массового производства и применения СП ЭВМ на базе микропроцессоров различных архитектур (RISC-процессоров, транспьютеров и т. д.), сопроцессоров, интеллектуального программного обеспечения, сетевых средств, средств связи, а также других средств информатики.

И. А. Мизин

УДК 681.326

П. В. Сивобород, И. В. Юрченко, Н. В. Петренко, А. А. Ножницкий, Н. И. Ерошенко

УНИВЕРСАЛЬНАЯ 8-РАЗРЯДНАЯ МИКРОЭВМ КР1830ВЕ48

БИС КР1830ВЕ48 семейства 8-разрядных однокристалльных микроЭВМ (ОЭВМ) предназначена для использования в системах обработки информации в качестве высокопроизводительного контроллера и управляющего устройства в составе различных приборов и оборудования. Эта БИС выполнена на основе КМОП-технологии с кремниевыми затворами в пластмассовом 40-выводном корпусе типа 2123.40—2.

Отличительная особенность новой БИС — универсальность областей применения (может работать как с внутренней, так и внешней программной памятью), высокая производительность и хорошая помехозащищенность, низкая потребляемая мощность. По уровням входных-выходных напряжений БИС совместима с микросхемами серий КР580, К1816, а также с ТТЛ-и КМОП-БИС серии К573.

Функциональное значение ОЭВМ определено содержимым программной памяти, которое выполнено масочным способом в процессе изготовления по картам заказа потребителей. Архитектура и структурная организация, система команд и язык ассемблера такие же, как и у ОЭВМ серии К1816 [1, 2].

Краткие технические характеристики ОЭВМ КР1830ВЕ48

Разрядность центрального процессора, бит	8
Объем внутренней программной памяти команд (ПЗУ), Кбайт	1
(возможно расширение до 4 Кбайт за счет подключения внешней памяти)	
Объем внутренней памяти данных (ОЗУ), байт	64
(возможно расширение до 320 байт за счет подключения внешней памяти)	
Число базовых команд	97
Формат команд, байт	1—2
Число таймеров-счетчиков	1
регистров общего назначения (РОН)	8
источников прерывания	2
резидентных портов ввода-вывода	3
уровней 16-разрядного стека	8
Частота импульсов внутреннего тактового генератора, МГц	1...6
Время машинного цикла (из частоты импульсов тактового генератора 6 МГц), мкс	2,5
Номинальное напряжение питания (U _{CC}), В	5
Потребляемый ток	
аппаратический (I _{CC0}), мА, не более	8
в режиме микропотребления (I _{CC}), мА, не более	100
Ток утечки на входах, мкА, не более	10

Емкость, пФ	
входа не более	10
входа выхода не более	20
нагрузки на выводах порта РО не более	200

Расположение выводов ОЭВМ приведено на рисунке.

Назначение выводов:
ОЗУ данных разбито на два банка регистров общего назначения (РОН): РОН0 с адресами 00...07; РОН1 с адресами 18...1F. Стек с адресами 08...17 и ячейки 20...3F используются только как ОЗУ данных. (Адреса ячеек памяти и коды команд ОЭВМ приведены в 16-ричной системе счисления.)

U _{CC} — напряжение питания 5 В;	
ОбВ — общий;	
BQ1, BQ2 — подключение кварцевого резонатора, LC-цепи или подача внешнего тактового импульса;	
DB0...DB7 — порт РО (шина данных);	
P1.0...P1.7 — порт P1;	
P2.0...P2.7 — порт P2;	
SR — сброс;	
SS — пошаговое выполнение программы команд;	
INT — внешнее прерывание;	
EMA — переключение для работы с внешней памятью;	
RD — разрешение чтения внешней памяти данных;	
PME — разрешение чтения внешней программной памяти команд;	
WR — разрешение записи во внешнюю память данных;	
ALE — стробирование адреса во внешней памяти;	
PK — выход строба для расширения числа каналов портов ввода-вывода;	
T0 — вход при командах условных переходов IT0, INT0; выход тактовых сигналов после выполнения команд ENT0 CLC;	
T1 — вход при командах условных переходов IT1, INT1; вход счетчика внешних событий после выполнения команды STRT CNT;	
U _{DD} — вход установки режима микропотребления.	

Для записи и выборки данных из ОЗУ применяется прямая (в коде команды три младших бита определяют адрес РОН) и косвенная (место расположения адреса указывается в команде) виды адресации. С помощью косвенной (регистровой) адресации можно обратиться к любой ячейке ОЗУ.

ПЗУ команд разбито на два банка программной памяти: МВО с адресами 0000...07FF; МВ1 с адресами 0800...0FFF. Переключаются банки программным путем.

Ввод-вывод данных и управление ОЭВМ осуществляются с помощью 8-разрядных портов РО, P1, P2. Порт РО — двунаправленный, служит в качестве шины данных при работе с внеш-

ней памятью. Порты P1 и P2 — квазидвунаправленные. Как выходные эти порты фиксируют введенные данные, как входные — не фиксируют. Три линии ввода-вывода T0, T1, INT используются как входы при реализации команд условных переходов.

Из 97 базовых команд 68 — однобайтовые. В двухбайтовых командах первый байт содержит информацию о коде команды, второй байт — непосредственные данные или младшие разряды адреса следующей команды. 54 команды выполняются за один машинный цикл, 43 команды — за два машинных цикла. Выполнение двухбайтовых команд в течение двух циклов связано с дополнительным обращением к программной памяти команд.

Режимы работы ОЭВМ

ОЭВМ может работать в следующих режимах: проверки ПЗУ, работы с внутренней памятью и внешней памятью, пошагового выполнения команд, микропотребления. Режимы работы устанавливаются комбинацией входных и выходных сигналов. В исходное состояние (сброс) ОЭВМ переводится по сигналу SR.

Режим проверки ПЗУ используется при контроле правильности информации, занесенной в ПЗУ масочным способом в процессе изготовления микросхемы. Устанавливается этот ре-

Ключ	CP	U _{CC}	40
1	T0		39
2	BQ1	T1	38
3	BQ2	P2.7	37
4	SR	P2.6	36
5	SS	P2.5	35
6	INT	P2.4	34
7	EMA	P1.7	33
8	RD	P1.6	32
9	PME	P1.5	31
10	WR	P1.4	30
11	ALE	P1.3	29
12	DB0	P1.2	28
13	DB1	P1.1	27
14	DB2	P1.0	26
15	DB3	U _{DD}	25
16	DB4	PK	24
17	DB5	P2.3	23
18	DB6	P2.2	22
19	DB7	P2.1	21
20	OB	P2.0	20

Расположение выводов ОЭВМ КР1830ВЕ48

жим при подаче на вход ЕМА напряжения $12 \pm 0,6$ В. Содержимое ПЗУ (адрес и данные) считывается через шину данных в порт P0, а адрес ячейки ПЗУ задается портами P0 (для адресов A0...A7), P2.0 и P2.1 (для адресов A8, A9).

Режим работы с внутренней памятью устанавливается подачей на вход ЕМА напряжения низкого уровня. Выполнение программы, хранящейся в ПЗУ команд, начинается после общего сброса с команды, имеющей адрес 00. Работа внешних устройств синхронизируется сигналом ALE, вырабатываемым в каждом машинном цикле.

Режим работы с внешней памятью устанавливается подачей на вход ЕМА напряжения высокого уровня (5 В) и применяется в случаях, когда для решения задачи внутренней памяти недостаточно. Использование внешней памяти программ в качестве дополнения к внутренней памяти возможно благодаря переключению банков памяти программ MB0 и MB1 командами SEL MB0, SEL MB1. Код команды принимается и младшие разряды адреса (A0...A7) выдаются во внешнее ЗУ через порт P0, старшие разряды адреса (A8...A11) — через порт P2 (P2.0...P2.3).

Память данных расширяется за пределы встроенной внутренней памяти (64 байт) с помощью шины данных. Доступ к внешней памяти данных осуществляется двухцикловыми командами MOVX A, @R и MOVX @R, A. При этом адрес данных во внешнем ОЗУ стробируется сигналом ALE, а принимаются и выдаются данные по сигналам RD и WR.

Режим пошагового выполнения команд при работе с внешней памятью для отладки и контроля программ устанавливается с помощью внешнего RS-триггера, управляемого сигналами от кнопки режима и ALE, путем подачи на вход SS. При этом завершается выполнение текущей команды; сигнал на выходе триггера в виде напряжения низкого уровня может сохранять свое состояние сколько угодно долго. Адрес следующей команды должен находиться на шине данных и в младших разрядах порта P2. Для выхода из режима и перехода к считыванию следующей команды нужно подать на вход SS напряжение высокого уровня.

Режим микропотребления задается аппаратным или программным способом. В первом случае на вход U_{DD} подается напряжение низкого уровня. В установившемся режиме работа задающего генератора блокируется, ОЭВМ удерживается в состоянии сброса, однако содержимое внутреннего ОЗУ, аккумулятора и старших разрядов регистра состояний RSW сохраняется. При подаче на вход U_{DD} напряжения высокого уровня ОЭВМ переходит в режим нормальной работы и начинает выполнение программы с нулевого адреса.

Для перевода ОЭВМ в режим

микропотребления программным способом необходимо выполнить команду IDL. В этом случае выполнение программы прекращается, внутренние фазовые и тактовые сигналы не формируются, состояние всех функциональных узлов сохраняется. Для вывода ОЭВМ из этого состояния на вход INT подается напряжение низкого уровня. Если перед этим было установлено разрешение на внешнее прерывание, то начнется отработка программы внешнего прерывания, в противном случае будет возобновлено выполнение программы с команды, следующей за командой IDL.

Для исключения возникновения тиристорного эффекта у паразитных транзисторов подложки кристалла и возможного перегрева микросхемы приняты специальные конструктивно-техно-

логические меры, обеспечившие ей нормальную работоспособность в диапазоне температур $-10 \dots +70^\circ\text{C}$.

Телефон 442-95-02, Киев

ЛИТЕРАТУРА

1. Кобылинский А. В., Липовецкий Г. П. Однокристальные микроЭВМ серии К1816 // Микропроцессорные средства и системы. — 1986. — С. 10—19.
2. Кобылинский А. В., Липовецкий Г. П., Сабадаш Н. Г. и др. СверхБИС универсальных однокристальных микроЭВМ. — Киев: Техника, 1987.

Статья поступила 9.12.88

УДК 681.322

В. А. Коваленко, А. И. Москалевский, Л. М. Солдатенко

КОНТРОЛЛЕР ГИБКОГО МАГНИТНОГО ДИСКА КР1810ВГ72А

Однокристальный контроллер гибких дисков КР1810ВГ72А позволяет работать с четырьмя дисковыми. Предназначен для записи, чтения и форматирования дисков с одинарной (режим ЧМ) в формате «IBM—3740» и двойной плотностью (режим МЧМ) в формате «IBMSYSTEM 34». Контроллер сопрягается с процессором с помощью 8-разрядного параллельного канала данных. Обмен данными между контроллером и памятью ЭВМ осуществляется непосредственно через процессор или с помощью контроллера прямого доступа к памяти.

Достоверность записи информации автоматически проверяется по контрольному коду в конце адресной и информационной частей каждого сектора. При работе необходимы внешние схемы разделения данных и форматирования информационного сигнала записи

для магнитной головки с учетом выдаваемых сигналов предкомпенсации. Контроллер формирует сигналы, управляющие запуском схемы разделителя данных и выбором его тактовой частоты. Полные форматы дорожки и сектора ГМД при работе контроллера показаны на рис. 1. Число секторов на дорожке зависит от их длины (табл. 1). Переменная длина конечного пробела определяется скоростью вращения диска, длинами секторов и пробелов, заданных программой.

Длина сектора равна 128, 256 или 512 байт при работе в режиме ЧМ и 256, 512, 1024 байт — в режиме МЧМ (табл. 2). Назначение выводов микросхемы КР1810ВГ72А приведено в табл. 3, структурная схема показана на рис. 2, электрические параметры — в табл. 4, временные диаграммы работы микросхемы — на рис. 3—9.

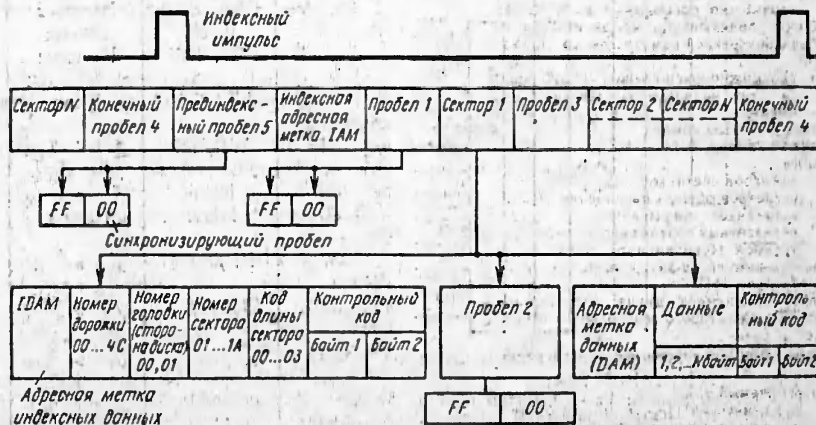


Рис. 1. Форматы сектора и дорожки ГМД

Таблица 1

Форматы ГМД

Емкость диска	Одinarная плотность				Двойная плотность			
	128	256	512	1024	128	256	512	1024
Байтов в секторе	128	256	512	1024	128	256	512	1024
Секторов на дорожке*	26	15	8	4	52	30	16	8
Килобайтов на диске	256,256	295,680	315,392	315,392	512,512	591,360	630,784	630,784

* 77 дорожек на диске.

Через последовательный интерфейс выполняются чтение или запись данных, преобразование байтов в последовательность сигналов WRD, выдача сигнала разрешения записи данных WRE и сигналов предкомпенсации PSO, PSI. В процессе считывания информации с ГМД последовательность данных в виде сигналов RDD, сопровождаемых сигналом DW, записывается через ПИ в регистры данных КГМД.

Интерфейс сопряжения обеспечивает выдачу команд управления и прием сигналов, информирующих о состоянии различных устройств НГМД через соответствующие порты входа и выхода.

Посредством логики управления режимами работы контроллера, а соответственно, и всей системой обмена

Таблица 2

Параметры сектора

Режим	Длина сектора, байт	Код длины сектора (N)	Число секторов на дорожке (SC)	Длина пробела 3 (GPL)	
				Чтение, запись	Форматирование
ЧМ	128	00	1A	07	1B
	256	01	0F	0E	2A
	512	02	08	1B	3A
МЧМ	256	01	1A	0E	3C
	512	02	0F	1B	54
	1024	03	08	35	74

Назначение выводов микросхемы КР1810ВГ72А

Вывод	Обозначение	Тип	Направление сигнала	Назначение
1	SR	Вход	От ЭВМ	Сброс
2	RD	»	»	Чтение
3	WR	»	»	Запись
4	CS	»	»	Выбор микросхемы
5	AO	»	»	Адрес
6...13	DB0...DB7	Входы-выходы	К ЭВМ	Линии двунаправленного канала данных с тремя возможными состояниями
14	DRQ	Выход	На КГДП	Запрос данных
15	DACK	Вход	От КГДП	Подтверждение прямого доступа к памяти
16	TC	»	»	Завершение обмена
17	IDX	»	От ГМД	Индексный импульс
18	INT	Выход	На ЭВМ	Запрос прерывания
19	C	Вход	—	Тактовые импульсы
20	GND	—	—	Общий
21	WRC	Вход	—	Тактовые импульсы записи
22	DW	»	От разделителя	Сопровождение данных
23	RDD	»	От ГМД	Чтение данных
24	VCO	Выход	На разделитель	Разрешение синхронизации схемы делителя данных
25	WE	»	На ГМД	Разрешение записи
26	MFM	»	На разделитель	Режим МЧМ
27	HDS	Выход	На ГМД	Выбор головки
28, 29	DS1, DS0	Выходы	На ГМД	Выбор дисководов 0...3
30	WRD	Выход	»	Запись
31, 32	RS1, RS0	Выходы	»	Сигналы предкомпенсации при записи с МЧМ
33	GLT/TRKO	Вход	»	Ошибка накопителя — нулевая дорожка
34	WR/TS	»	От ГМД	Защита записи — двусторонний диск
35	RDY	»	»	Готовность НГМД
36	HDL	Выход	На ГМД	Загрузка магнитной головки
37	FR/STR	»	»	Сброс триггера «Ошибка — шаг»
38	LCT/DIR	»	»	Снижение тока записи, направление шага
39	RW/SEEK	»	»	Чтение, запись-поиск (определяет назначение 33, 34, 37, 38)
40	U _{cc}	»	»	Напряжение источника питания 5 В ± 10%

Таблица 3

Примечание. Все параметры, кроме длины сектора, даны в шестнадцатеричном коде

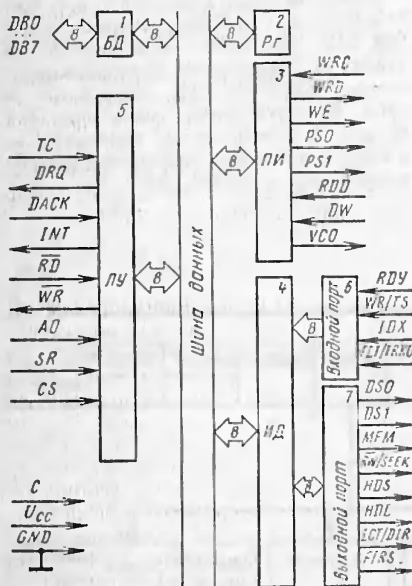


Рис. 2. Структурная схема контроллера КР1810ВГ72А:

БД — буферы данных; Рг — регистры общего назначения; ПИ — последовательный интерфейс; ИД — интерфейс сопряжения с НГМД; ЛУ — логика управления режимами чтения, записи, обмена информацией

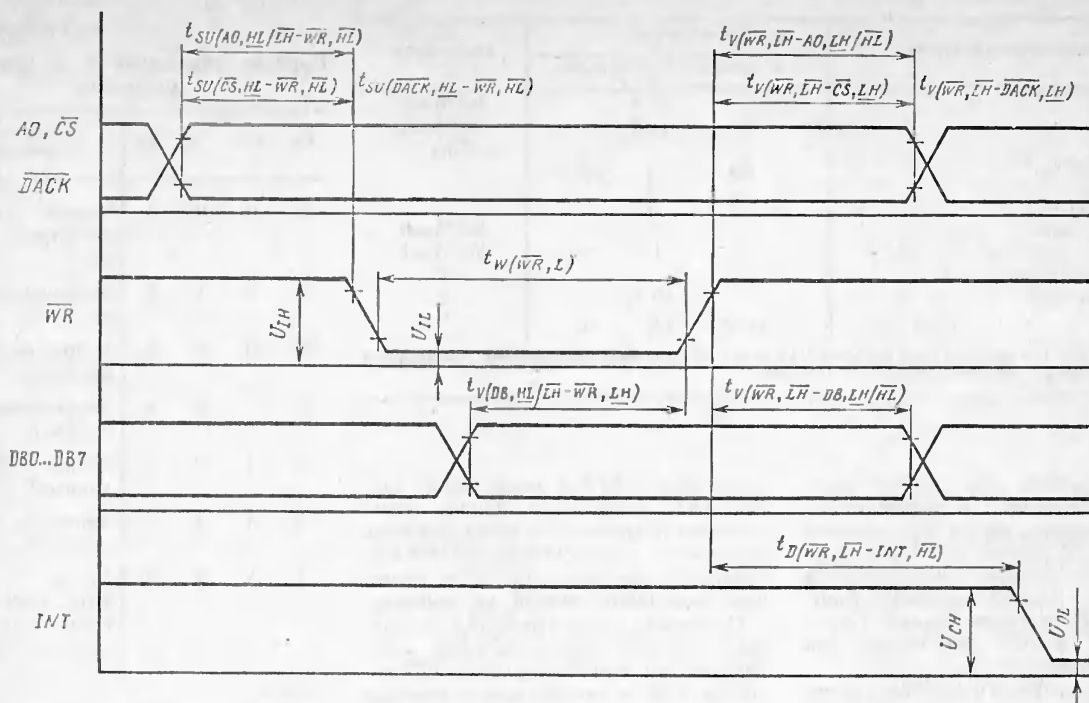


Рис. 3. Сигналы записи данных в контроллер

Таблица 4
Основные электрические параметры контроллера КР1810БГ72А, $T = -10...+70^\circ\text{C}$

Параметр, единица измерения	Статические		Примечание	
	не менее	не более		
U_{OL} , В	—	0,45	$I_{OL}=260\text{ мА}$ $I_{OH}=400\text{ мкА}$	
U_{OH} , В	2,4	$U_{CC}+0,5$		
I_{CC} , мА	—	120		
I_L , мкА	-10	10		
Динамические				
$t_D(\overline{RD}, \underline{HL}-\underline{DB}, \underline{HB}/\underline{LH})$, нс	—	200	1	
$t_D(\overline{RD}, \underline{LH}-\underline{DB}, \underline{IZ}/\underline{ZH})$, нс	20	100		
$t_D(\overline{RD}, \underline{LH}-\underline{INT}, \underline{HL})$, нс	—	500		
$t_D(\overline{WR}, \underline{LH}-\underline{INT}, \underline{HL})$, нс	—	500		
$t_D(\overline{DACK}, \underline{HL}-\underline{DRQ}, \underline{HL})$, нс	—	200		
T_{DRQ} , мкс	13	—	1	
$t_D(\overline{WRC}, \underline{HL}-\underline{WE}, \underline{LH})$, нс	20	100	1,2	
$t_D(\overline{WRC}, \underline{HL}-\underline{WE}, \underline{HL})$, нс	—	—		
$t_D(\overline{WRC}, \underline{LH}-\underline{PS}, \underline{HL}/\underline{LH})$, нс	20	100		
$t_D(\overline{WRC}, \underline{LH}-\underline{WRD}, \underline{LH})$, нс	—	—		
$t_W(\overline{WRD}, \underline{H})$, нс	$t_W(\overline{WRC}, \underline{H})-50$	—		
T_{STR} , мкс	33	—	1,2	
Режима эксплуатации				
U_{IL} , В	-0,5	0,8	$T_{11}=125\text{ нс}$ $T_C=250\text{ нс}$	
U_{IH} , В	2,0	$U_{CC}+0,5$		
T_C , нс	120	500		
$t_W(\underline{C}, \underline{H})$, нс	40	$0,52 T_C$ $0,76 T_C$		
$t_W(\overline{RD}, \underline{L})$, нс	250	—		
$t_W(\overline{WR}, \underline{L})$, нс	250	—		
Режимы работы КГМД				
<p>Многосекторный и многодорожжковый режимы. Обмен данными последовательности секторов заканчивается по сигналу T_C (завершение обмена). Если с приходом последнего байта данных сектора не выдается сигнал T_C, то КГМД записывает (считывает) данные в следующий (из следующего) сектор, увеличивая текущий номер сектора на единицу. Многодорожжковый режим определяется битом MT в коде команды. Чтение-запись дорожек и секторов ГМД выполняется последовательно до прихода сигнала T_C.</p>				
<p>ОБМЕН ДАННЫМИ В РЕЖИМЕ ПДП</p> <p>Передача (прием) данных на ГМД (от ГМД) может выполняться в</p>				

информацией с помощью ГМД принимаются команды и данные от центрального процессора, выделяются сигналы управления и прерывания, устанавливается плотность записи, опрашивается контроллер ПДП (сигнал T_C), обеспечивается выполнение требуемых команд. Фактически ЛУ является центральным процессором самого контроллера.

Регистры общего назначения включают интерфейсные регистры состояния, регистр данных, регистры номеров сектора, дорожки и др. Порядок обращения к интерфейсным регистрам приведен в табл. 5, назначение битов регистров состояний — в табл. 6.

РЕЖИМЫ РАБОТЫ КГМД

Многосекторный и многодорожжковый режимы. Обмен данными последовательности секторов заканчивается по сигналу T_C (завершение обмена). Если с приходом последнего байта данных сектора не выдается сигнал T_C , то КГМД записывает (считывает) данные в следующий (из следующего) сектор, увеличивая текущий номер сектора на единицу. Многодорожжковый режим определяется битом MT в коде команды. Чтение-запись дорожек и секторов ГМД выполняется последовательно до прихода сигнала T_C .

ОБМЕН ДАННЫМИ В РЕЖИМЕ ПДП

Передача (прием) данных на ГМД (от ГМД) может выполняться в

Таблица 5

Параметр, единица измерения	Статические		Примечание
	не менее	не более	
T_{WRC} , мкс		2; 4 1; 2	MFM=0 MFM=1 3,1
$t_w(WRC, H)$, нс	80	350	MFM=0 MFM=1 3 3
$t_w(RDD, H)$, нс			
$t_w(DW, H)$, мкс		2 1	
$t_w(IDX, H)$		10 T_C	
$t_w(\$R, H)$	14 T_C	14 —	

Примечание: 1 — значения даны для $f_C=8$ МГц (для 4 МГц значения удваиваются); 2 — параметр программируется; 3 — указаны номинальные значения.

режимах прямого (без участия центрального процессора) и непрямого доступа к памяти (через центральный процессор). В режиме ПДП для каждой посылки данных активизируется сигнал DRQ (запрос данных). Контроллер ПДП отвечает выдачей сигналов \overline{DACK} и \overline{RD} при чтении или \overline{WR} при записи.

В режиме непрямого доступа к памяти (НПДП) активизируется выходной сигнал INT и процессор контролирует (опрашивает) главный регистр состояния. Этот сигнал используется для прерывания, а бит RQM регистра состояния — для запроса данных. Системный процессор должен отвечать на запросы сигналами \overline{RD} или \overline{WR} , которые устанавливают низкий уровень INT и сбрасывают бит RQM. После заверше-

ния передачи КГМД активизирует сигнал INT, сообщая о начале фазы контроля результата. В обоих режимах выполнение команды прекращается при появлении сигналов T_C . «Не готов» или последнего сектора на дорожке.

По сигналу ошибки от КГМД прекращается запись данных на ГМД, устанавливается единица в бите E_C регистра PC0 и заканчивается команда (после установки кода прерывания 01 в битах IC регистра PC0).

ФАЗЫ ВЫПОЛНЕНИЯ КОМАНД

Каждая из 15 команд контроллера имеет фазы задания, исполнения и контроля результата (табл. 7, 8). Фаза задания описывает порядок ввода команды от ЭВМ. Первые два байта,

Порядок обращения к интерфейсным регистрам

\overline{CS}	AO	\overline{RD}	\overline{WR}	Выполняемая операция
0	0	0	1	Чтение главного регистра состояния
0	0	1	0	Запрещенная комбинация
0	0	0	0	Запрещенная комбинация
0	1	0	0	Запрещенная комбинация
0	1	0	1	Чтение регистра данных*
0	1	1	0	Запись в регистр данных*
1	X	X	X	Шина данных $\overline{DB7}.. \overline{DB0}$ в высокооммедианском состоянии

* При записи-чтении данных в режиме прямого доступа к памяти состояние входов \overline{CS} и AO может быть произвольным (X).

как правило, определяют непосредственно коды команды, остальные — необходимые параметры. Исключение составляют команды, описываемые одним байтом: «Опрос состояния прерывания» и «Определение параметров».

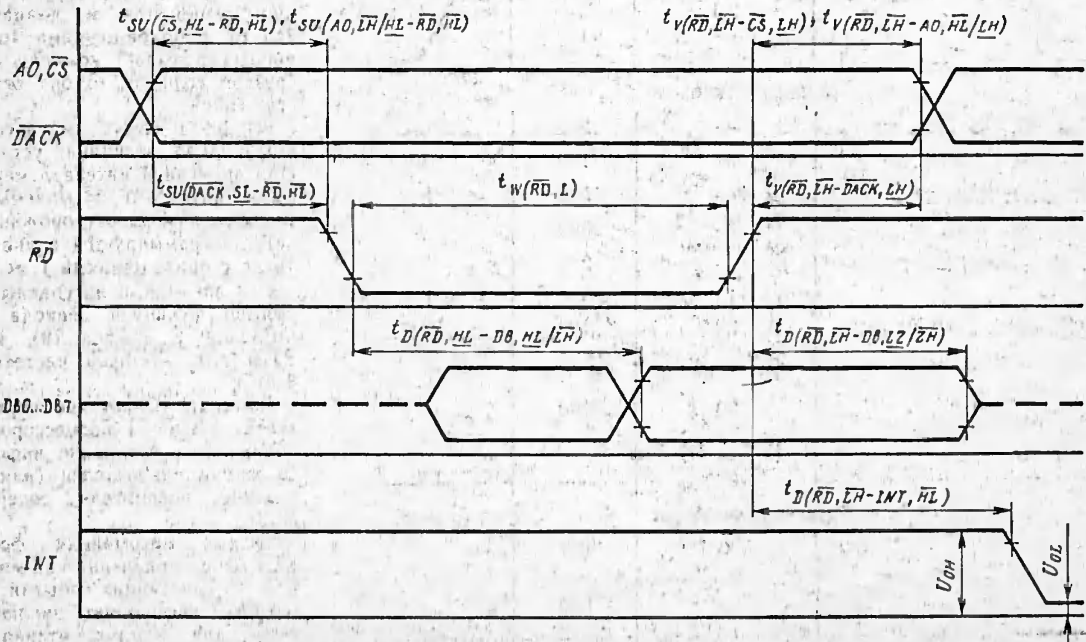


Рис. 4. Сигналы чтения данных из контроллера

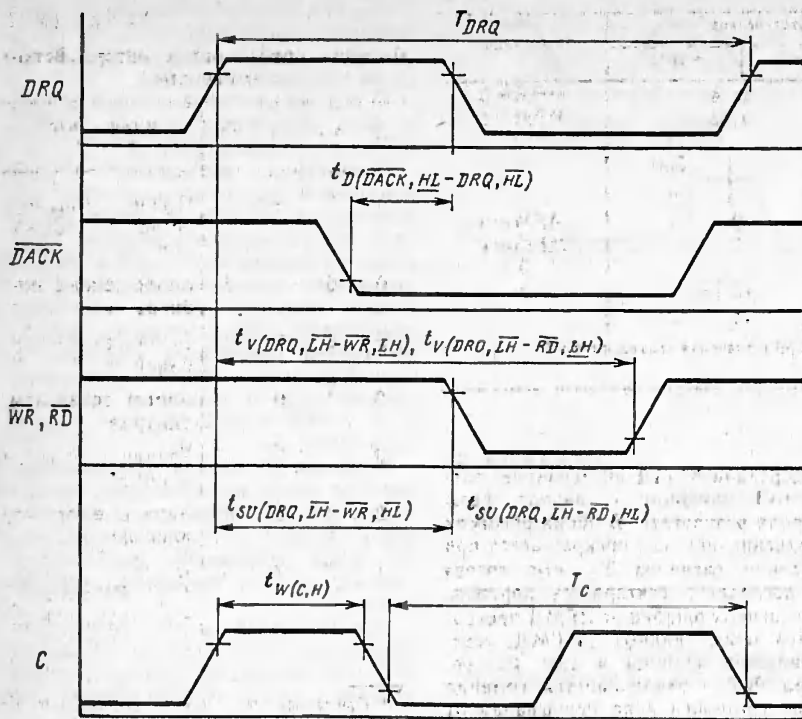


Рис. 5. Сигналы прямого доступа к памяти

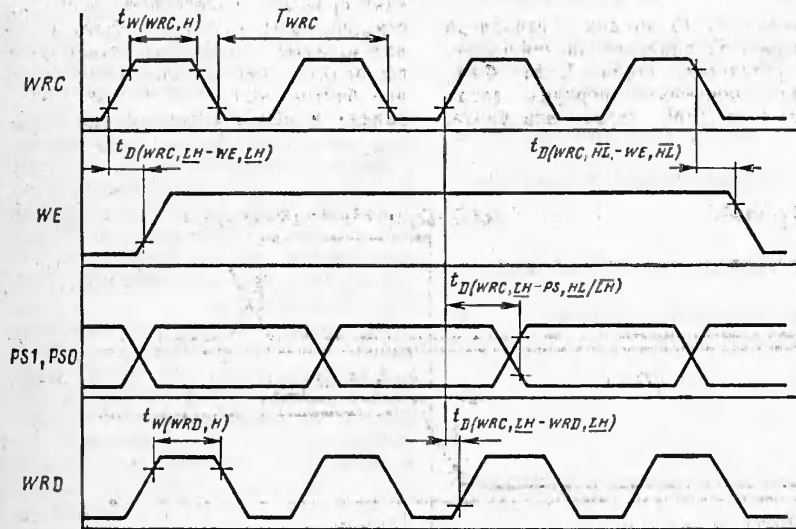


Рис. 6. Сигналы записи на ГМД

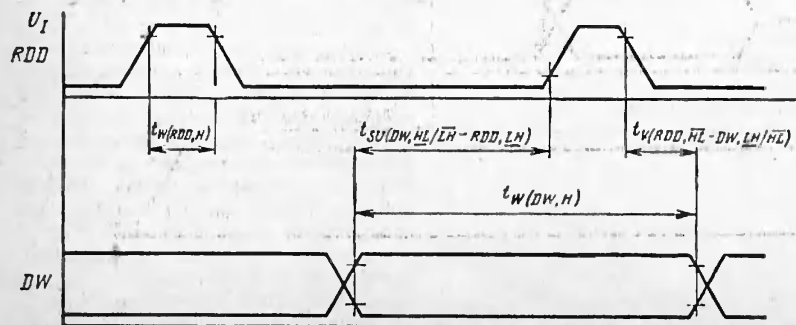


Рис. 7. Сигналы чтения с ГМД

Приведенный в табл. 7 порядок ввода исходной информации является обязательным. После ввода последнего байта задания контроллер автоматически переходит в фазу исполнения команд обмена данными или управления устройствами НГМД. Процессы обмена данными зависят от режимов ПДП и НПДП.

В фазе контроля результата процессор последовательно считывает все указанные в структуре команды байты и только после чтения последнего байта контроллер готов к выполнению следующей команды. Готовность подтверждают биты D10 и RQM главного РС, которые должны быть в состоянии 1 (PC0...PC3 опрашиваются только в фазе контроля результатов выполнения команд).

Команда «Определение параметров» используется перед выполнением любой дисковой операции (включая форматирование нового диска) для определения рабочих характеристик накопителя; задает режимы ПДП и НПДП. Состояние 0 младшего разряда третьего байта (B4) определяет режим ПДП. В процессе ее выполнения устанавливаются значения трех внутренних таймеров:

время загрузки головки HLT — 7-разрядная величина, определяющая временной интервал между загрузкой головки и началом процесса считывания или записи; программируется на интервал 2...254 мс с приращениями 2 мс;

время разгрузки головки HUR — 4-разрядная величина, задающая временной интервал между окончанием исполнительской фазы (команды чтения или записи) и разгрузкой головки; программируется в диапазоне 16...240 мс с приращениями 16 мс; если процессор выдает команду перед разгрузкой головки, то она остается загруженной;

период шаговых импульсов SRT — 4-разрядная величина, устанавливающая временной интервал между шаговыми импульсами, посылаемыми КГМД (время перехода от дорожки к дорожке); программируется в диапазоне 1...16 мс с приращениями 1 мс.

Эти временные интервалы являются прямой функцией периода тактового импульса C (вывод 19). Интервалы даны для тактовой частоты, равной 8 МГц.

Команда «Опрос состояния накопителей» выдается процессором по требованию на получение информации о состоянии накопителей (данные о состоянии накопителя содержатся в PC3).

Сигнал прерывания формируется КГМД при появлении как минимум одного из следующих событий:

КГМД выполняет заключительную фазу для команд чтения данных, дорожки, индексных и удаленных данных; записи удаленных данных; сканирования;

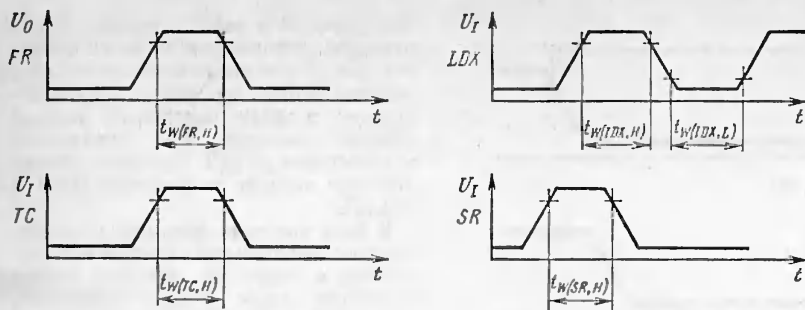


Рис. 8. Сигналы FR, LDx, TC, SR

Таблица 6

Назначение битов регистров состояний

Бит	Обозначение	Назначение
7	RQM	Главный регистр
6	DIO	Запрос данных
5	NDM	Ввод-вывод данных (направление передачи)
4	CB	Режим непрямого доступа к памяти (только при выполнении команды)
3...0	D ₃ B...D ₀ B	КГМД занят (для команд чтения или записи) НГМД под номерами 3...0 заняты. Выполняется команда «Поиск»
7,6	IC	PC0 Код прерывания 00 — нормальное завершение команды, 10 — некорректная команда, 01 — ненормальное завершение команды, 11 — состояние неготовности дисковода
5	SE	Конец поиска
4	EC	Ошибка аппаратных средств НГМД (для команд чтения или записи)
3	NR	Принят сигнал неготовности (запрос второй стороны одно-стороннего диска)
2	H	Номер головки
1,0	DS1, DS0	Выбор накопителя
7	EN	PC1 Ошибка конца дорожки (несуществующий сектор)
5	DE	Ошибка данных (по контрольному коду)
4	OR	Признак непереполнения
3,6	—	Не используется (состояние 0)
2	ND	Не найден сектор
1	NW	Признак защиты записи
0	MA	Признак недостающей адресной метки
7	—	PC2 Не используется (состояние 0)
6	CM	Контрольная метка не соответствует заданной
5	DD	Ошибка в поле данных
4	WC	Ошибка адреса дорожки
3	SH	Признак совпадения сканирования
2	SN	Сканирование не удовлетворяется
1	BC	Признак нечитаемой дорожки
0	MD	Признак отсутствия адресной метки данных
7	FT	PC3 Ошибка накопителя
6	WP	Защита записи на диске
5	RDY	Готовность
4	TO	Дорожка 0
3	TS	Двусторонний диск
2	H	Номер головки
1,0	DS1, DS0	Выбранный накопитель

сигнал «Готовность» одного из накопителей изменяет состояние; команда поиска или повторной проверки завершает операцию; КГМД запрашивает обмен данными в фазе исполнения команд записи-чтения в режиме НППД.

Прерывания, вызываемые первой и четвертой причинами, происходят при нормальных командных операциях и легко различаются процессором. Прерывания, вызываемые второй и третьей причинами, определяются только с помощью команды «Опрос состояния прерывания». При вводе данной команды сигнал прерывания переходит в низкий уровень, а причина прерывания определяется с помощью разрядов 5, 6, 7 PC0 в фазе контроля результата (табл. 9).

Команды «Поиск» и «Восстановление» не имеют итоговой фазы выполнения. Для эффективного их завершения и контроля за положением головки на диске необходимо использовать команду «Опрос состояния прерывания».

Для выявления причины прерывания процессора могут быть использованы признаки занятости КГМД (разряд D4) и режима работы НППД (разряд D5) главного PC

Разряд D5	Разряд D4	
0	0	Асинхронное событие (2), (3)
0	1	Фаза итоговой информации (1)
1	1	Требуется обмен данными (4)

Один запрос прерывания, посылаемый процессору, может быть вызван более чем одним событием. Поэтому процессору необходимо провести анализ с помощью команды «Опрос состояния прерывания» и чтения главного PC для определения причины прерывания. Таким образом обслуживаются все «угри-занные» прерывания.

По команде «Поиск» головка считывания-записи накопителя размещается над заданной дорожкой. КГМД определяет разницу между текущим и требуемым адресами дорожки и выдает соответствующее число шаговых импульсов. Если требуемый адрес дорожки больше текущего адреса, то сигнал направления (LCT/DIR, вывод 38) будет высокого уровня, если меньше — низкого уровня. Если заданная дорожка совпадает с текущим адресом, то головка не перемещается.

Период выдачи шаговых импульсов управляется таймером шагового импульса и задается (SRT) в команде «Определение параметров». После каждого шагового импульса требуемый адрес дорожки сравнивается с текущим. При равенстве адресов устанавливается признак «Конец поиска» (сигнал высокого уровня в разряде D5 PC0) и завершается выполнение команды. Если накопитель во время операции поиска находится в состоянии «Неготовность», то устанавливается признак «Неготов-

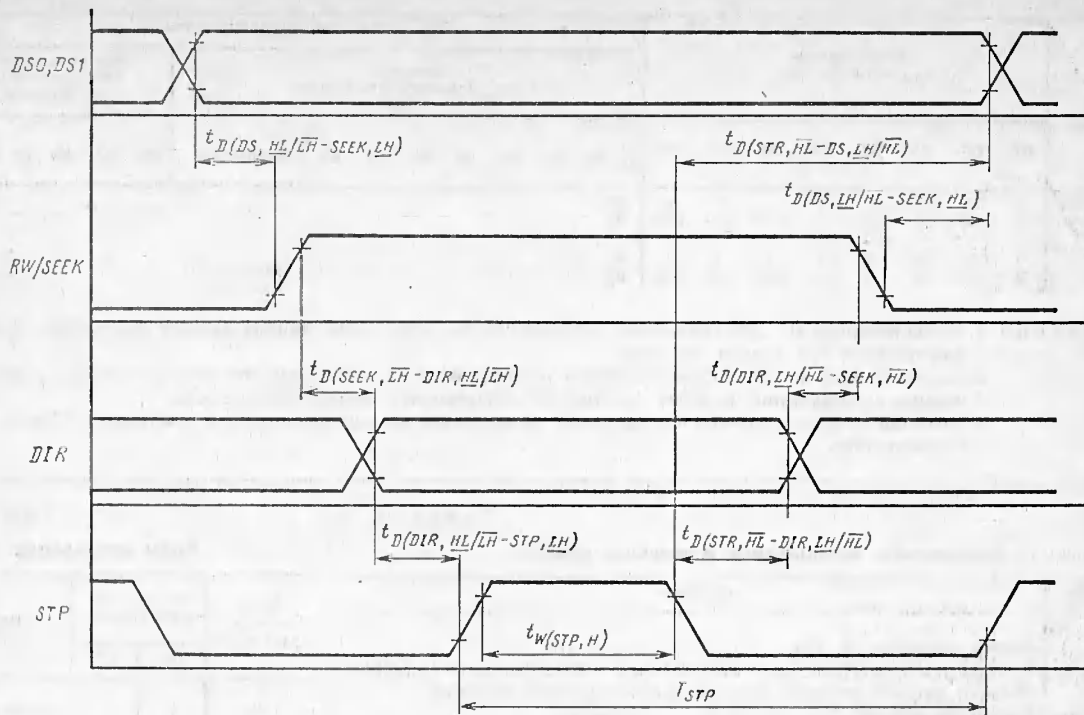


Рис. 9. Сигналы поиска

Структура системы команд контроллера KP1810BG72A

Команда	Коды команды (байты B1, B2) ¹								Фазы выполнения команды																
									Задание ² (последовательность байтов)								Контроль результата ² (последовательность байтов)								
	D7	D6	D5	D4	D3	D2	D1	D0	B1 B2	B3	B4	B5	B6	B7	B8	B9	Исполнение	B4	B2	B3	B4	B5	B6	B7	
Чтение данных	MT	MCM	SK	0	0	1	1	0	B1									Данные ГМД→ЭВМ	ST0	ST1	ST2	C	H	R	N
Чтение удаленных данных	MT	MCM	SK	0	0	1	0	0	B1	C	H	R	N	EOT	GPL	DTL									
Запись данных	MT	MCM	0	0	0	1	0	1	B1									Данные ЭВМ→ГМД							
Запись удаленных данных	MT	MCM	0	0	0	0	0	1	B1																
Чтение дорожки	0	MCM	SK	0	0	0	1	0	B1									Данные ГДМ→ЭВМ							
Чтение индексных данных	0	MCM	0	0	0	1	0	1	B1									Индексные данные ГДМ→ЭВМ							
Формирование дорожки	0	MCM	0	0	0	1	0	1	B1	N	SC	GAL	D				Индексные данные ЭВМ→ГДМ								
Сканирование на равенство	MT	MCM	SK	1	0	0	0	1	B1								Сравнение данных ГДМ и ЭВМ								
Сканирование на меньше или равно	MT	MCM	SK	1	1	0	0	1	B1	C	N	R	N												
Сканирование на больше или равно	MT	MCM	SK	1	1	1	0	1	B1																
Восстановление	0	0	0	0	0	1	1	1	B1								Установка МГ на TRO								
Опрос состояния прерывания	0	0	0	0	1	0	0	0	B1									ST0	C						
Определение параметров	0	0	0	0	0	0	1	1	B1	SRT	NLT														

код (КК) поля ID (подсчитанный для данных, записанных в первых пяти байтах) автоматически записывается, как последние два байта поля. Пробеги задаются автоматически, причем длина регулируемого пробела 3 определяется одним из параметров команды форматирования.

Метка адреса поля данных DAM формируется и автоматически записывается, как первый байт поля данных. Байт данных, указанный в командной фазе задания, заносится в поле данных (все байты) каждого сектора. Значение КК формируется из адресной метки данных и данных, записанных в секторе. За последним байтом данных размещаются два байта КК.

Форматирование дорожки начинается с физической индексной метки. Порядок распределения секторов непосредственно соответствует таблице форматирования (предоставляемой процессором). В таблице каждый сектор определяется четырьмя признаками: адресом дорожки, номером головки, адресом сектора и кодом длины сектора. Адрес дорожки в поле ID должен быть равен адресу формируемой дорожки. Адрес сектора должен быть уникальным (нет двух равных адресов). Порядок расположения секторов в таблице — это последовательность, в которой номера сектора появляются на дорожке при ее форматировании. Число входных установок (адреса дорожки, головки и сектора, код длины сектора) должно быть равно числу секторов, размещаемых на дорожке (указывается также в командной фазе).

В зависимости от расположения секторов дорожки могут форматироваться последовательно, тогда первый сектор, идущий за индексной меткой, имеет адрес сектора 1, соседний сектор — адрес сектора 2 и т. д.; или же номера секторов могут располагаться со сдвигом, например: 1, 4, 7, 10, 13 и т. д. — со сдвигом на три сектора. Такой сдвиг может быть полезен для увеличения времени, отведенного на обработку данных сектора.

Чтение данных. Для завершения спецификации фазы задания по команде «Чтение данных» требуется 9 байт. КГМД загружает головку и считывает метки адреса IDAM и поля ID. Когда адрес запрашиваемого сектора сравнивается с адресом сектора, считанным с диска, КГМД выдает побайтовые данные (с поля данных сектора) в ЭВМ. Команда «Чтение данных» автоматически выполняется в многосекторном режиме. Многодорожковый режим может задаваться с помощью признака MT (см. табл. 8). Число данных, передаваемых по одной команде на КГМД, зависит от признака многодорожечного режима, признака плотности записи и числа байтов на сектор.

В процессе выполнения команд чтения и записи для временного изменения эффективного размера дискового сектора используется специаль-

ный параметр размера сектора DTL, задающий длину 1...256 байт. Если фактический сектор больше сектора, указанного параметром DTL, в ЭВМ считывается только число байтов, указанных этим параметром, остальная часть фактического дискового сектора не считывается (данные проверяются по КК). Аналогично происходит процесс записи на диск.

Многосекторный режим чтения обеспечивается таким же образом, как и при ненулевом коде размера сектора. Параметры N и DTL всегда присутствуют при задании команд чтения-записи данных. Параметр DTL должен быть установлен на шестнадцатеричное число FF при ненулевом значении N.

Если физическая индексная метка фиксируется дважды без нахождения запрашиваемого сектора, то КГМД устанавливает признак «Необнаруженный сектор» (разряд D2 в PC1) и завершает команду считывания данных. Код прерывания устанавливается на 01 (разряды D7 и D6 PC0).

Ошибка «Необнаруженный сектор» может появиться в многосекторном режиме после чтения части секторов и в случаях, когда номер конкретного сектора упущен при форматировании дорожки или систематическая ошибка на

диске делает секторное поле ID нечитаемым.

После считывания поля ID и поля данных на каждом секторе КГМД проверяет байты КК. При обнаружении ошибки считывания (некорректный КК в поле ID или в поле данных) КГМД устанавливает признак «Ошибка данных» в PC1. Если ошибка КК появляется только в поле данных, то признак устанавливается в PC2. В любом случае КГМД завершает команду считывания данных. Код прерывания (разряды D7 и D6 в PC0) устанавливается в состоянии 01.

Если метка адреса удаленных данных считывается с диска, а признак пропуска (указываемый в командной фазе) не установлен, то КГМД выдает признак «Контрольная метка» (разряд D6 в PC2) и завершает команду чтения данных (после считывания всех данных в секторе).

По признаку пропуска КГМД пропускает сектор с адресной меткой удаленных данных и считывает следующий, т. е. признак пропуска может быть использован для игнорирования контроллером секторов с удаленными данными в процессе считывания в мультисекторном режиме.

Во время обмена данными между

Таблица 10
Значение индексных байтов в фазе контроля результата при завершении команды

MT	EOT	Сектор	Сторона	Байт информации			
				C H R N			
0	IA OF 08	1...25 1...14 1...7	0	NC	NC	R+1	NC
	IA OF 08	26 15 8	0	C+1	NC	R=01	NC
	IA OF 08	1...25 1...14 1...7	1	NC	NC	R+1	NC
	IA OF 08	26 15 8	1	C+1	NC	R=01	NC
1	IA OF 08	1...25 1...14 1...7	0	NC	NC	R+1	NC
	IA OF 08	26 15 8	0	NC	LSB	R=01	NC
	IA OF 08	1...25 1...14 1...7	1	NC	NC	R+1	NC
	IA OF 08	26 15 8	1	C+1	LSB	R=01	NC

Примечания. 1. NC означает, что значение параметра не изменяется; 2. LSB — младший значащий бит II меняет значение.

ГМД и ЭВМ контроллер должен обслуживаться системой (процессором или контроллером ПДП) каждые 27 мкс в режиме ЧМ или каждые 13 мкс в режиме МЧМ. В противном случае устанавливается признак переполнения (разряд D4 в PC1) и команда чтения данных завершается.

В фазе контроля результата команды данные зависят от состояния признака многодорожечного режима и адреса последнего обработанного сектора (табл. 10).

Запись данных. Для завершения спецификации фазы задания по команде «Запись данных» требуется девять байтов. КГМД загружает головку и считывает секторные поля ID. Когда запрашиваемый секторный адрес сравнивается с секторным адресом, считываемым с диска, КГМД принимает данные от процессора через информационную шину (канал данных) и записывает данные в поле данных требуемого сектора. Два байта КК записываются в конце этого поля.

КГМД считывает поле ID каждого сектора и проверяет байты КК. Если фиксируется ошибка считывания (некорректный КК) в одном из полей ID, он устанавливает признак «Ошибка данных» (разряд D5 в PC1) и завершает выполнение команды «Запись данных». Код прерывания (разряды D7 и D6 PC0) устанавливается в состоянии 01.

Режимы и признаки команд записи и чтения данных подобны. В исполнительной фазе команды «Запись данных» обмен данными между процессором и КГМД должен происходить каждые 31 мкс в режиме ЧМ и каждые 15 мкс в режиме МЧМ. Если временной интервал больше, КГМД устанавливает признак переполнения (разряд D4 PC1) и завершает команду.

Считывание удаленных данных выполняется подобно команде «Считывание данных». Единственная разница заключается в обработке адресной метки данных и признака пропуска. Когда КГМД фиксирует адресную метку данных в начале поля данных (признак пропуска не был установлен), считываются все данные, записанные в секторе, устанавливается признак «Контрольная метка» (разряд D6 PC2) и завершается выполнение команды. Если признак пропуска установлен, то КГМД пропускает этот сектор и продолжает считывание следующего сектора. Признак пропуска может быть использован для считывания только секторов с удаленными данными в многосекторном режиме считывания.

Запись удаленных данных выполняется в том же порядке, что и команда записи данных, за исключением того, что адресная метка удаленных данных вводится в начале поля данных вместо адресной метки обычных данных. Данной командой можно пометить «плохой» сектор, содержащий систематическую ошибку.

Команда «Чтение дорожки» аналогична команде «Чтение данных», только

вся информация (метки, адреса и данные) считывается непрерывно. Сразу же после обнаружения физической индексной метки КГМД начинает считывание индексной информации и данных с дорожки, как одного непрерывного блока данных. Даже в случае ошибки в байтах КК поля данных или поля ID КГМД продолжает считывать данные с дорожки. Он сравнивает информацию ID, считанную из каждого сектора, со значениями, указанными в командной фазе. Если информация поля ID не обнаруживается на дорожке, то в PC1 устанавливается признак «Необнаруженный сектор». Данная команда не допускает многодорожечного режима и операции пропуска данных. Она завершается после считывания последнего сектора дорожки (число секторов на дорожке указано в байте параметра дорожки в командной фазе). Если КГМД не обнаруживает адресной метки IDAM на диске после второго прохождения физической индексной метки (т. е. за два оборота диска), он устанавливает признак «Недостающая адресная метка» (разряд D0 в PC1) и завершает команду. Код прерывания (разряды D7 и D6 PC0) устанавливается на 01.

По команде «Чтение индексных данных» данные с первого корректного поля ID текущей дорожки передаются на процессор. Если на дорожке отсутствует адресная метка IDAM, устанавливается признак «Недостающая адресная метка» (разряд D0 PC1); если метка данных, — признак «Необнаруженный сектор» (разряд D2 PC1). В обоих случаях происходит завершение команды.

По командам сканирования данные, считанные с диска, сравниваются с данными, поступающими с ЭВМ (из процессора в режиме НППД или из контроллера в режиме ПДП). КГМД сравнивает данные на побайтовой основе и находит сектор диска с данными, удовлетворяющими следующим условиям: равны, меньше или равны, больше или равны системным данным.

Для сравнения используется простая бинарная (дополнение до единицы) арифметика (FF — наибольшее число, 00 — наименьшее).

Если после сравнения полного секто-

ра данных условия не удовлетворяются, номер сектора увеличивается за счет шага адреса сканирования (указанного в командной фазе) Сканирование продолжается до тех пор, пока не появится одно из следующих событий: условия для сканирования удовлетворены (равны, меньше или больше), достигнут последний сектор на дорожке или получен сигнал T_C.

При удовлетворении условий сканирования КГМД устанавливает признак «Совпадение сканирования» (разряд D3 PC2) и завершает команду сканирования. Если условия сканирования не удовлетворяются между стартовым и последним сектором на дорожке (сканируемые секторы определяются в командной фазе), КГМД устанавливает признак «Сканирование не удовлетворено» (разряд D2 PC2) и завершает команду сканирования.

Как только во время сканирования сигнал T_C от процессора или контроллера ПДП поступит на КГМД, он прекратит операцию сравнения конкретного текущего байта. При этом команда завершается. Коды состояния признаков «Совпадение сканирования» и «Сканирование не удовлетворено» для различных условий завершения сканирования приведены в табл. 11.

Если КГМД обнаруживает адресную метку удаленных данных в одном из секторов и признак пропуска имеет низкий уровень, то сектор рассматривается как последний на дорожке, устанавливается признак «Контрольная метка» (разряд D6 PC2) и завершается команда. Если признак пропуска имеет высокий уровень, КГМД пропускает сектор с адресной меткой удаленных данных и считывает следующий сектор. В этом случае также устанавливается признак «Контрольная метка» (разряд D6 PC2), показывающий, что обнаружен стертый сектор*.

* Во время выполнения команд сканирования последний сектор на дорожке должен быть считан для соответствующего завершения команды. Например, шаг адреса сканирования устанавливается равным 2, конец дорожки (последний сектор) — 26. Сканирование начинается с сектора 21, тогда будут сканированы секторы 21, 23 и 25. Следующий сектор 27 на дорожке обнаружен не будет и произойдет непредусмотренное завершение команды. Для нормального завершения необходимо, чтобы в данном примере сканирование началось в секторе 20 или конец дорожки был установлен равным 25.

Таблица 11

Коды состояния сканирования

Команда	PC2		Соотношение данных ГМД и ЭВМ
	Разряд D2	Разряд D3	
Сканирование равно	0	1	Равны
	1	0	Не равны
Сканирование меньше или равно	0	1	Равны
	0	0	Меньше
	1	0	Меньше равны
Сканирование больше или равно	0	1	Равны
	0	0	Больше
	1	0	Больше, равны

Во время сканирования данные передаются из процессора или контроллера ПДП для сравнения с данными, считанными с диска. Для исключения установки признака переполнения (разряд D4 PC0) необходимо, чтобы данные выбирались не более чем через 27 мкс (режим ЧМ) или 13 мкс (режим МЧМ). Если ошибка из-за переполнения появляется, КГМД завершает эту команду.

Недопустимые команды. Если на КГМД подается недопустимая (неопределенная) команда, он прекращает выполнение другой команды и не выдает сигнала прерывания. В основном РС устанавливаются разряды D6 и D7, указывая процессору, что контроллер требует заключительной фазы выполнения команды. В качестве завершения должно быть считано содержимое PC0, в том числе шестнадца-

теричный код 80, указывающий на прием недопустимой команды.

Необходимо отметить, что команда «Опрос состояния прерывания» должна посылаться после команды «Поиск» или «Восстановление», в противном случае КГМД будет принимать следующую команду, как недопустимую. Кроме того, после обслуживания последнего «утраченного» прерывания повторные команды «Опрос состояния прерывания» также воспринимаются, как недопустимые.

В некоторых случаях полезно использовать недопустимую команду, как команду $N_0 - O_p$ для переключения КГМД в резервное или нерабочее состояние.

Телефон 442-94-31, 442-95-35, Киев

Статья поступила 26.12.88

УДК. 621.049.77

Н. В. Воробьев, В. Я. Кремлев, Г. К. Меликов

ОДНОКРИСТАЛЛЬНЫЙ ОПТОЭЛЕКТРОННЫЙ ПРЕОБРАЗОВАТЕЛЬ ДЛЯ ГРАФИЧЕСКОГО МАНИПУЛЯТОРА

Микросхема предназначена для использования в графическом манипуляторе мышь вычислительных комплексов ДВКЗ, ДВК4 [1]. Кинематическая схема и принцип работы манипулятора полностью сохранены. Существенное упрощение и удешевление его конструкции достигается за счет того, что шесть ИМС ТТЛШ-типа, четыре фототранзистора с соответствующими нагрузками, времязадающая емкость, используемая в тактовом генераторе, заменены на две ИМС инжекционно-поле-

вого типа, совмещающие преобразователь оптической информации и систему обработки полученных сигналов (рис. 1). Две идентичные ИМС необходимы потому, что оптическая информация вводится по двум независимым ортогональным осям.

В качестве оптоэлектронного преобразователя использован нормально-закрытый полевой транзистор с управляющим р-п-переходом (ИЗПТУП), выполняющий роль переключательного элемента вентиля ИПЛ [2], имеющего

спектральную чувствительность 0,6... 1,1 мкм.

Для исключения ложного срабатывания схемы обработки и обеспечения высокой помехоустойчивости между оптоэлектронными преобразователями и схемой обработки расположены триггеры Шмитта (рис. 2). Пороги отпирания и запираания регулируются независимо благодаря изменению нагрузочного тока первого и третьего ИЗПТУП. Схема обработки синхронизируется тактовым генератором, выполненным в виде кольцевого генератора из 35 последовательно соединенных инверторов. Применение кольцевого генератора, построенного на вентилях ИПЛ, позволяет получать частоты в диапазоне 3...4 декад. Частота генерации выбирается простым подбором тока инжектора, стабилизируемого с помощью генератора, подключенного к инжектору.

Выходы манипулятора подсоединены к реверсивному счетчику, выполненному на ТТЛ-элементах, поэтому необходимо преобразование логических уровней ИПЛ, соответствующих напряжениям 0,2...0,5 В для Лог. 1 и сотым долям вольта для Лог. 0, в стандартные уровни ТТЛ.

На кристалле интегрального манипулятора мышь расположены (рис. 3) оптоэлектронные преобразователи (1, 2), триггеры Шмитта (3, 4), схема обработки (5), тактовый генератор (6), транслятор уровней (7, 8) контактные площадки (9). Штриховыми линиями показаны области инжекторов.

Размер кристалла определяется расстоянием между оптоэлектронными преобразователями 1 и 2, обеспечивающими формирование двух сдвинутых друг относительно друга на 1/4 периода импульсных сигналов, каждый из которых имеет скважность, равную двум. В данном случае расстояние соответствует размерам модулирующего диска в манипуляторе мышь, выполненного на ИМС ТТЛ-типа. Напряжение питания кристалла E ограничивается снизу значением 0,7 В. Исключение составляет схема транслятора уровней, для которой питающее напряжение должно составлять не менее 3 В.

Для оптоэлектронного преобразователя и цифровой схемы обработки, вы-

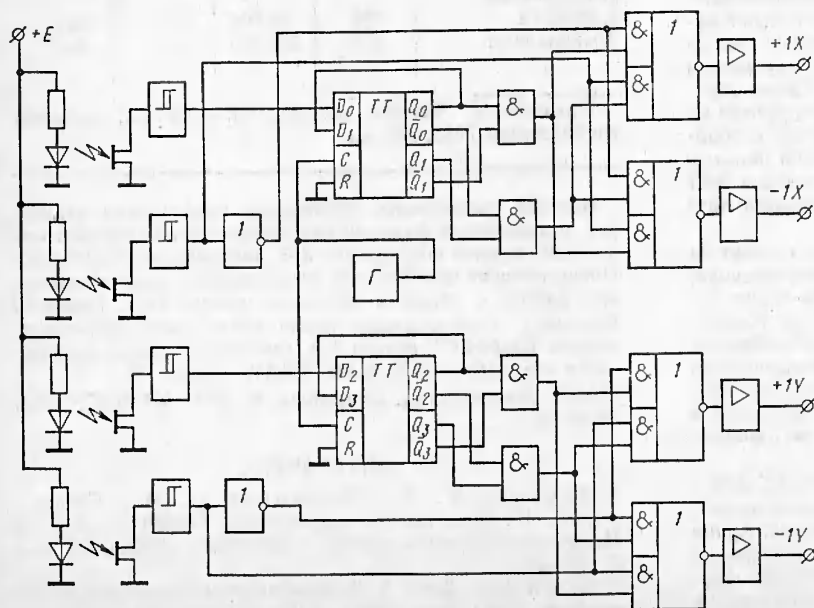


Рис. 1 Принципиальная схема электрической части манипулятора мышь

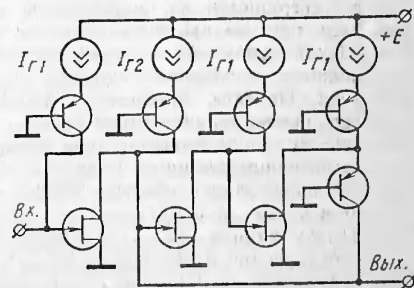


Рис. 2 Электрическая схема триггера Шмитта

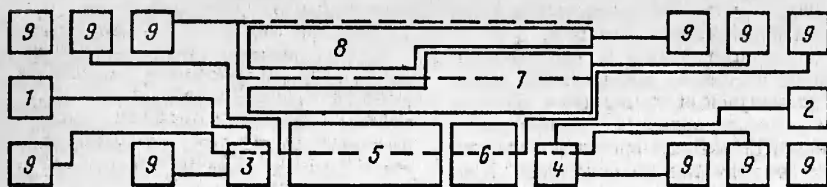


Рис. 3 Компоновочный чертеж кристалла

полненным на одном кристалле, требуется согласование уровней. Если учесть, что уровень Лог. 1 НЗПТУП является возрастающей функцией тока инжекции и составляет 0,2...0,5 В, что соответствует фотоЭДС, возникающей на переходе затвор-исток фоточувствительного НЗПТУП при его облучении, то согласование логических уровней оптоэлектронного преобразователя и схемы обработки сводится к правильному подбору тока инжектора входного каскада схемы обработки. ИПЛ-вентили сохраняют надежную работоспособность в диапазоне трех декад изменения токов инжектора [3], поэтому можно выби-

рать значения тока инжектора входного каскада в широком диапазоне уровней освещенности. Кроме того, диапазон чувствительности оптоэлектронного преобразователя может быть расширен благодаря изменению площади перехода затвор-исток фоточувствительного НЗПТУП, что позволяет заменить светодиод на источник естественного освещения, излучение которого подается на оптоэлектронные преобразователи с помощью световодов.

Простота технологического процесса изготовления, включающего всего четыре операции фотолитографии, и

возможность конструктивного выполнения всех блоков устройства без навесных элементов значительно повышают надежность, уменьшают стоимость и потребляемую мощность манипулятора.

Телефон 532-98-61, Москва

ЛИТЕРАТУРА

1. Воробьев Н. В., Безобразов В. С. Графический манипулятор мышь для персональных ЭВМ // Микропроцессорные средства и системы.— 1988.— № 3.— С. 57—59.
2. А. С. 519102 СССР. Интегральный инвертор / В. Я. Кремлев, Р. Ж. Ерманов, В. В. Лебедев.— Оpubл. 1979. Бюл. № 4.
3. Кремлев В. Я., Меликов Г. К., Бондарь И. А. Статистические характеристики элементов СБИС инжекционно-полевой логики. Технология, проектирование и надежность интегральных полупроводниковых схем. // Сб. науч. тр. / МИЭТ.— 1988.— С. 69—78.

Статья поступила 30.12.88

КРАТКИЕ СООБЩЕНИЯ

УДК 681.32

В. В. Анищенко, Е. М. Злотник, А. Р. Левин, И. К. Стежко

СЕМЕЙСТВО ЦВЕТНЫХ ГРАФИЧЕСКИХ АДАПТЕРОВ ДЛЯ ПЭВМ

Новый вариант двухплатного графического адаптера, состоящий из модулей видеопамати и видеопроцессора, использует два свободных соединителя электронного базового модуля ПЭВМ ЕС1840 [1]. Однако использование в качестве контроллера ЭЛТ БИС К1809ВГ4, вырабатывающего адреса регенерации изображения и динамической памяти, позволило исключить из схемы блок адресных счетчиков (АС) и увеличить глубину кадрового буфера до восьми растровых плоскостей размерами 1024 × 512. Таким образом, картинка размерами 768 × 512 точек, формируемая на экране цветного монитора, имеет 256 одновременно отображаемых цветов, выбираемых из программируемой палитры с общим числом цветов 4096. Кроме того, контроллер ЭЛТ обеспечивает функции панорамирования и создания двух независимых окон.

Трехплатный вариант графического адаптера состоит из модулей интерполятора, видеопамати и видеопроцессора. Два последних модуля функционально подобны одноименным модулям в двухплатном графическом адаптере. Модуль интерполятора увеличивает скорость построения изображения в видеопамати. Аппаратный линейный интерполятор реализует функцию построения вектора по алгоритму Брезенхэма [2]. Он записывает один элемент изображения в каждом «свободном» цикле видеопамати (500 нс), формируя изображение со скоростью 300 000 точ./с.

Этот вариант адаптера используется в ПЭВМ ЕС1840 и ЕС1841. Он устанавливается в электронном модуле расширения с помощью модулей передатчика и приемника. Кроме того, в трехплатном варианте графического адаптера на модуле видеопроцессора реализован блок аппаратного формирования маркера, который отображает световую метку в виде взаимно-перпендикулярных линий прямоугольной фигуры с изменяющимся положением и размерами.

Технические характеристики цветных графических адаптеров

Тип	Число одновременно отображаемых цветов	Быстродействие в режиме отображения	Панорамирование, окна	Аппаратный курсор
Двухплатный	16	36 000	Нет	Нет
Двухплатный К1809ВГ4	256	36 000	Да	Нет
Трехплатный	256	300 000	Да	Да

Примечание. Частота регенерации — 60 Гц, размеры изображения 768 × 512 мм

Базовое программное обеспечение графических адаптеров, реализующее функции международного стандарта машинной графики GKS уровня 2 В, написано на ассемблере. Предусмотрена возможность использования данного пакета при работе с языками высокого уровня (Си, Паскаль, Фортран). Пользователю также могут быть поставлены пакеты ДИАТЕСТ версии 1.0, демонстрирующий возможности стандарта GKS, и ARTDRAW версии 1.0.

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

ЛИТЕРАТУРА

1. Злотник Е. М., Киркорев С. И., Стежко И. К. Графический адаптер для ПЭВМ ЕС1840 // Микропроцессорные средства и системы.— 1988.— № 4.— С. 25—27.
2. Фоли Дж., Дэм А. Основы интерактивной машинной графики.— М.: Мир, 1985.— Т. 2.— С. 140—143.

Статья поступила 20.01.89.

«ЭЛЕКТРОННЫЙ ДИСК» ДЛЯ МИКРОЭВМ СМ1800

Эмулятор НГМД разработан в виде модуля внешней памяти (МВП) емкостью 256 Кбайт, подключаемого к интерфейсу И41 микроЭВМ СМ1800 (см. рисунок). В нем использована микросхема контроллера динамического ОЗУ КМ1810ВТ3, управляющая работой накопителя из 44 микросхем памяти К565РУ5Д. Данные записываются в модифицированном коде Хэмминга. Исправление одиночных и обнаружение многократных ошибок считывания реализовано с помощью микросхемы К555ВЖ1. Данные в МВП читаются и записываются, как в устройстве ввода-вывода (порт). 5-разрядное наборное поле номера платы позволяет адресовать до 32 плат МВП.

Внутреннее 16-разрядное слово данных преобразуется во внешнее 8-разрядное слово с помощью первого и второго регистров данных. Слова разделяются на байты младшим разрядом адреса. Синдром ошибки считывается из порта состояния. Через порт управления задается режим работы: прерывание по одиночной или многократной ошибке либо режим тестирования, в котором можно проверить узел коррекции и диагностики. Уровень прерывания для конкретной микроЭВМ выбирается пользователем и устанавливается на наборном поле.

При обмене данными по шине адреса необходимо выдать 5-разрядный код номера платы и 3-разрядный код адреса порта, а по шине данных — 3-байтовый адрес ячейки накопителя (байт данных). После окончания обработки обращения процессора к МВП узел диагностики и коррекции формирует сигнал ХАСК, подтверждающий завершение опера-

ции. Цикл чтения или записи одного сектора (128 байт) составляет 3,3 мс (у штатного НГМД среднее время доступа к данным 205 мс). Узел памяти организован двумя банками по 64 Кслов, которые выбираются непосредственно контроллером КМ1810ВТ3, управляющим также регенерацией динамического ОЗУ.

Программное обеспечение для ОС МикроДОС, ОС1800, СР/М состоит из короткого драйвера, встраиваемого пользователем в базовую систему ввода-вывода (BIOS) конкретной микроЭВМ.

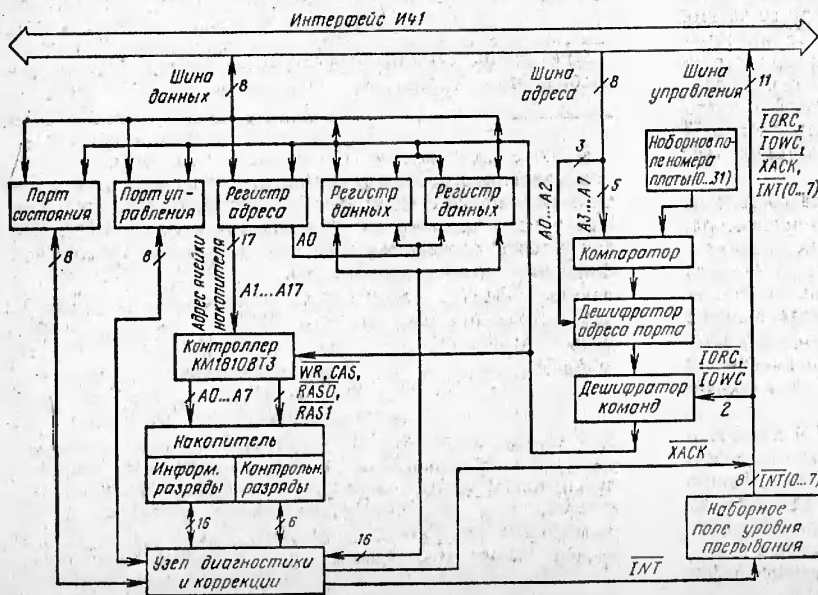
МВП выполнен на печатной плате размерами 200×320 мм. Для повышения помехоустойчивости линии адреса и тактового сигнала разнесены; линии тактового сигнала перпендикулярны линиям выхода данных. Введена дополнительная задержка сигнала CAS на 30...40 нс.

Телефон 34-43-35, Алма-Ата

ЛИТЕРАТУРА

1. Борисова А. Л., Михайлова О. Н., Рыбин И. М., Федоров А. С. Возможности создания внешнего полупроводникового запоминающего устройства СМ ЭВМ: Сб. науч. тр. — М.: ИНЭУМ, 1983. — Вып. 100. — С. 86.
2. Лукьянов Д. А. «Электроника 256К» — эмулятор диска на основе микроЭВМ «Электроника 60» и ДВК // Микропроцессорные средства и системы. — 1986. — № 2. — С. 62.

Статья поступила 24.11.87



Блок-схема меню

Институт информатизации общества и развития науки, Секция «Методологические и социальные проблемы информатизации общества» Философского общества СССР, Обнинский городской комитет КПСС

приглашают Вас принять участие в работе Второго Всесоюзного симпозиума «Проблемы информатизации общества и развития науки» с основной темой «Взаимодействие общественных, естественных и технических наук», который будет проходить в октябре 1990 г в Обнинске Калужской области.

На Симпозиуме предполагается работа следующих секций:

Глобальный процесс информатизации и медиатизации общества: концепции и тенденции,

Социально-экономические проблемы информатизации народного хозяйства,

Организация науки и развития научных исследований,

Правовые проблемы информатизации общества,

Современные информационные технологии; искусственный интеллект, экспертные системы: состояние, проблемы, перспективы,

Компьютеризация в науке и образовании,

Логические и лингвистические проблемы информационной деятельности, Информатизация общества и культура,

Молодежь и проблемы информатизации общества.

Тезисы просим присылать до 10 июня 1990 г., объемом 1—3 страницы через 1,5 интервала, в трех экземплярах, отпечатанные в рамку под ротопринт; поля: левое — 25, правое — 10, нижнее — 20 и верхнее — 35 мм, размер текста 170×240 мм, нумерация страниц карандашом, лента средней жирности, новая. Высылать в твердой упаковке без сгибов. Все цитаты должны быть выверены и подписаны на втором экземпляре, ссылки оформляются в строку в круглых скобках. В правом верхнем углу первой страницы — фамилия и инициалы, в круглых скобках — город строчными буквами, далее через два интервала посередине заглавными большими буквами, еще через два интервала — текст.

К тезисам должна быть приложена авторская справка с указанием фамилии, имени и отчества полностью, места работы, должности, ученой степени и звания, домашнего и служебного адресов и телефонов, а также номера и названия секции, в которой Вы хотели бы работать.

Тезисы присылайте по адресу: 121002, Москва, Смоленский бульвар, дом 20. Философское общество СССР, секция «Методологические и социальные проблемы информатизации общества».

Телефоны для справок: 201-50-37 и 408-57-31.

УДК 681.325.5—181.48

А. А. Бабурин, А. Д. Воробьев, И. С. Зильберберг,
Э. Г. Кнеллер, Г. Н. Навозенко, Н. П. Павлюченко,
В. Б. Пац, Е. И. Пчелкина

ОСОБЕННОСТИ АРХИТЕКТУРЫ ПЭВМ «ИСТРА 4816»

ПЭВМ «Истра 4816» относится к классу ЭВМ, рассчитанных на рабочие станции и управляющие машины (ГОСТ 201—87), решающие следующие основные задачи:

- физико-математические, экономические и САПР;
- профессиональная обработка текстов в издательствах и на производстве;
- сбор и обработка данных эксперимента;
- подготовка программ управления объектами.

16-РАЗРЯДНАЯ ПЭВМ «ИСТРА 4816»

ПЭВМ «Истра 4816» (совместная разработка ФВНИИЭМ, г. Истра и НПО Счетмаш, г. Курск) выполнена на одной 8-слойной плате размерами 300×430 мм.

Состав ПЭВМ: три процессора — один 16-разрядный КМ1810ВМ86 и два 8-разрядных КР580ВМ80; подсистема связи с внешними устройствами; ОЗУ до 4 Мбайт. Процессоры (два системных — СП и один периферийный — ПП) имеют доступ к системной магистрали. ПП, кроме того, имеет выход на магистраль связи с периферийными устройствами (магистраль ПП). Все процессоры и дисплей подключены на шину 4-портового ОЗУ.

Наличие дополнительных процессоров и большой доступной всем процессорам памяти позволяет организовать одновременно несколько процессов, часть из которых идет в фоновом режиме. Некоторые из них могут использоваться как «часовые», предохраняющие систему от случайных зависаний. Например, системный 8-разрядный процессор выполняет программы под управлением ОС CP/M, а 16-разрядный — работает под управлением ОС MS DOS, причем каждую ОС можно перезапускать, не паруя работы процессоров.

Шесть свободных мест для дополнительных печатных плат позволяют наращивать систему, а специализированная шина с большим числом сигналов — создавать переходы на специальные шины, необходимые для решения тех или иных задач (например, для создания рабочей станции исследователя с выходом на шину VME). Для этого потребуется разработать и изготовить соответствующую печатную плату межсоединений с разъемами для интерфейса магистралей. В настоящее время разработан вариант такой переходной платы с интерфейсом «Истра 4816» и разъемами «Сапфир».

Доступ к ОЗУ предоставляет арбитр системы в соответствии с фиксированным приоритетом. Производительность ОЗУ, составляющая 10 Мбайт/с, обеспечивает поток информации для процессоров и дисплея, как правило, без тактов ожидания, т. е. ОЗУ прозрачно для пользователей. При выбранных тактовых частотах и быстродействии ОЗУ на одно обращение расходуется примерно половина времени выполнения команды главным системным процессором (ГСП) КМ1810ВМ86. Оставшегося времени достаточно для дисплея и процессоров К580ВМ80, которым необходимо примерно одно обращение за 2 мкс.

Организация ОЗУ. Процессоры ПЭВМ имеют заведомо меньшую адресную сетку, чем доступный на систем-

ной плате объем ОЗУ и ПЗУ. Отдельному процессору в некоторых случаях — процессу, если он сознательно перезапускается пользователем или его программой, — выделяются определенные зоны ОЗУ. Такую организацию памяти нельзя назвать настоящей страничной — просто средствами ПП заполняются регистры, задающие положение адресного пространства каждого процессора в физическом адресном пространстве ПЭВМ.

Доступ к ПЗУ. ПЗУ предназначено, главным образом, для хранения подпрограмм эмуляции и используется ПП. Время доступа к ПЗУ не обеспечивает работу ГСП без тактов ожидания, поэтому непосредственный доступ к ним имеет только ПП. Адресное пространство ПЗУ отделено от адресного пространства ОЗУ и доступно через вспомогательные регистры. Необходимые для работы ГСП данные перекачиваются в начале работы из ПЗУ в адресное пространство ГСП в ОЗУ.

Работа с периферийными устройствами. Особенность архитектуры ПЭВМ состоит в том, что все периферийные устройства подключены к ПП и не имеют прямого доступа к СП. При выполнении ГСП команд, обращенных к какому-либо периферийному устройству, они перехватываются ПП и выполняются с помощью той периферийной аппаратуры, которая доступна и подключена к ПП. Такой подход позволяет эмулировать аппаратные средства не на уровне BIOS или других подсистем ОС, к которым некоторые прикладные программы могут вообще не обращаться, а на уровне команд самого ГСП.

Выполнение команд ввода-вывода. Используется следующая процедура:

команда распознается, приостанавливается ее выполнение аппаратными средствами, ГСП переводится в состояние ожидания;

для ПП формируется прерывание.

ПП может прочитать в регистрах адрес команды и информацию, подлежащую передаче, признаки чтения, прерывания или записи, присутствующие на магистрали СП. Имея данные об эмулируемой системе, ПП с учетом предыдущих команд ввода-вывода определяет, что именно предполагается сделать в данном случае, и ставит соответствующую задачу перед периферийным устройством. Использование специализированного ПП позволяет получить выигрыш во времени за счет переноса части задачи на ГСП. Надо также учитывать, что для большинства команд ввода-вывода эмуляция вообще не нужна.

Организация работы нескольких процессоров. Для каждого процессора с помощью регистров, задающих старшие адреса, выделяется определенная область ОЗУ. Области могут перекрываться, что допустимо только для «сотрудничающих» процессоров.

Обмен информацией между процессорами, идущими на каждом из процессоров, происходит следующим образом:

для совершенно не взаимодействующих процессоров — через файлы на НГМД или НМД (поддержки ОС);

для полностью взаимодействующих процессоров — через общие области ОЗУ (не требуется поддержки ОС, если программы пишутся пользователем самостоятельно, но нужны средства для задания перекрывающихся областей, что достигается благодаря расширению ОС командами задания областей ОЗУ);

для любых процессоров — через универсальную операцию ввода-вывода или «трубу» (требуется поддержка на уровне ОС и ПП).

Дисплей выполнен в виде одноплатного модуля, аппаратно эмулирующего все функции CGA. Информация на экране представлена картинками из нескольких процессоров, которым выделены перескакивающие области ОЗУ. Этот процесс поддерживается средствами ОС. Отображаемая информация — алфавитно-цифровая, число цветов — 16. Дисплейные команды обрабатываются средствами ПП.

Контроллер накопителя на гибких дисках эмулирует команды контроллера IBM PC с помощью ПП, выполняющего переводировку и интерпретацию команд.

Контроллер накопителя на жестких дисках также эмулирует микропрограммными и аппаратными средствами все команды IBM PC и взаимодействует с НМД, используя возможности ПП. Поддерживает работу с широким набором НМД практически любой емкости.

Конструктивные особенности. На системной плате ПЭВМ размещены все основные функциональные узлы, кроме дисплея и контроллера НГМД; процессоры ОЗУ и все устройства связи с периферийным оборудованием, т. е. порты ввода-вывода, управления динамиками, клавиатурой, последовательным и параллельным интерфейсами, система прерывания, таймеры.

Вспомогательные платы вставляются в разъемы на межсоединительной панели, связанной, в свою очередь, с системной платой. Конструкция системного блока обеспечивает доступ к любому элементу ПЭВМ в течение 2...3 мкс.

Внутренняя магистраль ПЭВМ позволяет развивать конструкцию ЭВМ и наращивать ее мощность. На нее выведены все основные сигналы ГСП, ПП и ОЗУ, что позволяет легко построить электронный переходный интерфейс на магистраль любого типа.

Технические характеристики ПЭВМ. ПЭВМ «Истра 4816» собрана целиком на отечественных комплектующих. В системном блоке размерами 147×470×380 мм расположены источник питания, системная плата, два НГМД, два Winchester накопителя и объединительная плата с восемью свободными посадочными местами. Емкость каждого НГМД — 1,2 Мбайт, Winchester — до 40 Мбайт. Масса системного блока — не более 10 кг.

Монитор — цветной или черно-белый — может замещаться и устанавливаться в соответствии с используемой платой управления. В настоящей модификации ПЭВМ поддерживаются черно-белый графический режим с 16 полутонами и разрешением 300×200 точек, цветной режим с 16 цветами и разрешением 300×200 точек, черно-белый графический режим без полутонов с разрешением 640×200 точек. Такое же разрешение и 16 цветов для символов обеспечиваются и в алфавитно-цифровом режиме. Предусмотрены четыре таблицы символов, переключаемых с клавиатуры в любой момент времени. При переключении таблиц изменяется изображение на экране, так как оно не представлено в виде графической информации в ОЗУ, а создается знакогенератором. Закачивается разработка платы, обеспечивающей 16 цветов при разрешении 640×350 точек.

На системной плате размерами 300×430 мм размещены посадочные места на 128 корпусов ОЗУ. Максимальная емкость ОЗУ — до 4 Мбайт при емкости микросхем — 256 Кбайт. Кроме того, на плате можно разместить ПЗУ и средства его адресации емкостью до 256 Кбайт. Кроме того на системной плате расположены порты и разъемы ввода-вывода Centronics, RS232, порт для клавиатуры, а также система связи и арбитража, позволяющая получить доступ ко всем ресурсам системы, ОЗУ, трем процессорам и дисплею. Все три процессора и арифметический сопроцессор (разъем для него) размещаются на плате вместе с обслуживающими их микросхемами, работают независимо и одновременно каждый в своей зоне ОЗУ. Зоны ОЗУ могут перекрываться, так что между процессорами происходит обмен информацией через эти общие области. ПЭВМ «Истра 4816» выпускается в напольном и настольном исполнениях. В варианте основного напольного исполнения можно в специальном

гнезде разметить дополнительные автономные модули, например сетевой адаптер типа «Ринг» или «Эстафета». «Ринг» разработан для ПЭВМ «Истра 4816» в Институте прикладной математики им. М. В. Келдыша АН СССР. В дальнейшем предполагается расширить функции 16-разрядной ПЭВМ «Истра 4816» за счет увеличения числа вспомогательных блоков, некоторой переконфигурации и тому подобных улучшений, не изменяющих архитектуру принципиально, а именно:

перенос контроллеров гибкого и жесткого дисков на системную плату;

разработка платы, совместимой с дисплеем EGA, т. е. платы, гарантирующей цветное изображение с 16 цветами разрешением 640×350 пиксел;

разработка плат, обеспечивающих подключение к шинам VME, MULTIBUS, IBM PC, IBM PS2 и другим, а также межсоединительных плат для одновременной работы с несколькими магистралями;

автоматическая отмена эмуляции обращений к внешним устройствам;

разработка специализированных плат преобразователей и управления периферийными устройствами;

подключение манипулятора мышь с системой управления на центральной плате. Сравнительные характеристики некоторых моделей 16-разрядных ПЭВМ приведены в таблице. Ниже дан перечень программного обеспечения, проверенного в работе (без доработки) на ПЭВМ «Истра 4816».

Перечень программ, работающих на ПЭВМ «Истра 4816»

Оболочки и утилиты

Framework II	Open Access
Side Kick	dBase II
Lotus1-2-3	SuperCalc 4
ChiWriter	SPG
Norton Utilities	Norton Comander
Evrika	Compaq (text) po
Vbench (text)	SuperCalc 3.2
MS Works	Windows
dBase III	Paradox
Quattro	MS Word 4
Personal Editor	Norton Editor
PC Tools	Code View
Powermetr (text)	Bench (text)
MS Word 3	

Средства программирования

Basic A	GW Basic
Turbo Pascal 4	Turbo Prolog
Lattice C	Fortran 77
Masm 2.0	Masm 4.0
Turbo Basic	Turbo Pascal 3
Turbo C	MS C 4.0
Lisp 85	Forth

Игры

Digger	Crazy Digger
Frogger	Striker
Patrol	Shamus
Zoom	Zaxon
Rebuding	Bomb
Karateka	Space War
Art of War	
Flights simulator	Tetris
Sopwitch	Demon
Good Runner	Lapy
Parat	Tapper
Rollo	Falcon
Larry	Decalton

Сравнительные характеристики ПЭВМ

Параметр	ГОСТ 27201-87 (ПМ4)		«Искра 1030.11»	«Искра 1030М»	EC1842	PS/2, модель 30	«Истра 4816»
	до 1991	после 1991					
1	2	3	4	5	6	7	8
Число МП общего назначения на системной плате			1	1	1	1	2
Разрядность МП	16 (32)	16 (32)	16	16	16	16:16	16:8
Число специализированных процессоров			1	1	1	1	2
Число одновременно работающих ОС	Не нормируется		1	1	1	2	2
Тип ОС			АДОС	MS DOS	MS DOS, CP/M86	MS DOS в. 3.3	MS DOS в. 3.2, CP/M86, CP/M80
Индекс производительности в базовой конфигурации			1,86	2,77	5,01	4,26	7,2
Максимальная емкость ОЗУ в базовой конфигурации ПЭВМ, Мбайт	1	1	0,512	0,512	1,0	0,64	4.0 на системной плате
Емкость ПЗУ в базовой конфигурации ПЭВМ, Мбайт	Не нормируется		0,048	0,064	0,064	0,064	0,064/0,256
Максимальная емкость ПЗУ, Мбайт			0,048	0,064	0,064	0,064	2,0*
Максимальная неформатированная емкость НГМД в базовой конфигурации, Мбайт	1	3	0,36	1	1	0,72	3,2
Емкость неформатированная НМД, Мбайт	20	40	5 (10)	10 (20)	10 (20)	20	10 (20)
Потребляемая мощность в базовой конфигурации, Вт	150	100	390	200	150	—	82
Масса ПЭВМ в базовой конфигурации, кг	15	12	—	15	15	—	10,92
Число свободных мест для установки модулей профессиональной ориентации	Не нормируется		Нет	Нет	7	3	6
Число используемых импортных ИС	—		3	5	3		0
Конструктивные особенности: число плат (размер и число слоев) базового варианта модуля процессора	Не нормируется		9 плат (Е2 2 слоя)	6 плат (Е2 6 слоев)	260×380 (10 слоев)	400×500 (10 слоев)	300×420 (8 слоев)
стоечный/настольный вариант исполнения	Не нормируется		Нет/Есть	Нет/Есть	Нет/Есть	Нет/Есть	Есть/Есть
Быстродействие, млн. кор. оп./с	1	1	1	1	2	1,5	1,5
Число адресуемых точек по протоколам: CGA EGA	640×200	720×512	640×200	640×200	640×200	640×480	640×200** 640×350
	Не нормируется		Нет	Нет	640×350		
Наработка на отказ, ч	5 000	10 000	3 000	5 000	5 000		5000***
Число используемых микросхем в базовой конфигурации			440	400	389		380
Лимитная цена в базовой конфигурации, руб.			8600	1500	6000		7600
План выпуска на 1989 г.				5600	200		500
Лимитная цена в максимальной конфигурации ПЭВМ, руб.			15 000	8900	17 000		15 000

* За счет модулей расширения ** С 1990 г. *** 10 000 в базовой конфигурации

КОНЦЕПЦИЯ 32-РАЗРЯДНОЙ ПЭВМ «ИСТРА»

1. Переход на 32-разрядную модель с сохранением аппаратной совместности с 16-разрядными ПЭВМ за счет дополнительных 16-разрядных процессоров и соответствующей организации шины.

2. Разработка внутренней шины системы производительностью 160 Мбайт.

3. Подключение к внутренней шине специализированных процессоров массивов, баз данных, арифметических и графических процессоров для аппаратной поддержки рабочих станций САПР.

АППАРАТНАЯ ПОДДЕРЖКА МНОГОПРОЦЕССОРНОЙ ОС

Создание системы управления дисплеем с аппаратной поддержкой до 16 окон в строке развертки, возможностью смены окон в каждой строке развертки (для графических окон) и в каждой строке символов (для алфавитно-цифровых окон). В каждом окне реализуется виртуальный дисплей, с разрешающей способностью до $1024 \times 768 \times 4$ и палитрой цветов, соответствующей дисплеям фирмы IBM.

Модернизация контроллеров управления внешними устройствами, особенно НМД и ИГМД, таким образом, чтобы они выполняли задания от нескольких программ одновременно с оптимизацией движения по поверхности носителя для уменьшения времени выполнения всей суммы полученных заданий.

Установка на центральной плате размерами 300×500 мм контроллера для работы с различными сетями высокоскоростным до 10 Мбит/с, различными дисциплинами и протоколами работы на линии связи.

Телефон 560-33-98, Москва

Статья поступила 16.06.89

УДК 681.3.06

В. Г. Абрамов, В. М. Брябрин, И. В. Горячая,
Т. Н. Щеглова

ЭРУДИТ — ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ПОДДЕРЖКИ ОБУЧЕНИЯ НА БАЗЕ ПЭВМ

С развитием производства ПЭВМ открываются широкие возможности для создания систем обучения нового типа. Такие системы характеризуются гибкостью, простотой в освоении, дружественным интерфейсом с пользователем, разнообразием форм воспроизведения учебной информации на экране дисплея.

Внедрение этих систем, тем не менее, замедляется из-за трудоемкости их создания. Это объясняется прежде всего тем, что в качестве инструментов для реализации обучающих систем обычно используются универсальные языки программирования, которые не предназначены специально для описания обучающих сценариев. Кроме того, обучающие системы часто разрабатываются лишь для конкретного учебного материала и определенной методики его изложения. В таком случае требуются значительные усилия для модификации заложенного материала и изменения используемой методики.

Другая особенность — в процессе реализации этих систем, как правило, участвуют не специалисты по учебным предметам, а программисты. В итоге созданный программный продукт далеко не всегда удовлетворяет поставленным дидактическим и методическим требованиям.

Таким образом, одной из насущных становится проблема разработки специальных инструментальных средств,

ориентированных на применение именно специалистами-предметниками при создании обучающих систем.

Один из возможных подходов к решению этой проблемы реализован в системе ЭРУДИТ — инструментальном программном средстве преподавателя-методиста. Она позволяет в удобной и простой форме подготовить учебный материал в соответствии с собственной методикой. Введенный материал используется в режиме обучения и контроля. Предусматриваются разные формы предъявления учебной информации, включающие игровую компоненту. Различные по сюжету и сложности они дают возможность представить учебный материал (упражнения) живо и занимательно. Игровая обстановка имеет побуждающее значение: она поддерживает азарт и интерес учащегося к повторению сеансов.

Области применения системы ЭРУДИТ определяются возможностью использования ее инструментальных средств для обеспечения эффективного процесса обучения. К числу таких приложений системы относятся: изучение родного или иностранных языков, профессиональной лексики, отдельных разделов истории, географии, элементарной математики и др. Здесь система будет полезна для тренировки памяти, сообразительности, изучения различных правил, формул, решения простейших задач, контроля качества знаний обучаемых. В процессе работы в системе пользователь приобретает еще и навыки обращения с клавиатурой ПЭВМ.

Представление учебной информации в системе

Наименьшая обучающая единица в системе — упражнение. В упражнении между элементами — порциями учебной информации устанавливаются простейшие логические связи по принципу соответствий между двумя множествами, первое из которых названо опорной композицией (ОК), второе — игровой (ИК). Под композицией здесь понимается конечное упорядоченное множество элементов одного и того же типа (текстового или графического). Все элементы в пределах композиции представляются в едином формате (т. е. помещаются в одинаковую прямоугольную область экрана). Благодаря «форматному» представлению появляется возможность удобного манипулирования элементами при реализации игровых сюжетов.

Для различных приложений ОК и ИК могут быть реализованы, например, следующими парами:

1) начальное изучение языков (родного или иностранного):

ОК — названия предметов, объектов; ИК — графические изображения предметов, объектов;

2) изучение иностранных языков:

ОК — слова, фразы на родном языке; ИК — слова, фразы на иностранном языке;

В качестве родного и иностранного языков могут фигурировать разные национальные языки (русский, английский, испанский, немецкий, французский и др.);

3) арифметика:

ОК — математические формулы, выражения; ИК — значения математических формул, выражений;

4) начальная геометрия:

ОК — изображения геометрических фигур; ИК — названия геометрических фигур;

5) история:

ОК — даты; ИК — исторические события, соответствующие датам;

6) география:

ОК — названия географических объектов; ИК — местонахождение географических объектов;

7) литература:

ОК — цитаты; ИК — авторы цитат;

8) логические задачи на поиск закономерностей:

ОК и ИК — абстрактные картинки.

Приведенные примеры не исчерпывают спектра возможных приложений.

В системе ЭРУДИТ допускаются следующие типы соответствий между элементами композиций:

простое соответствие (1:N) : ОК (множество ОК) содержит лишь один элемент; ИК (множество ИК) — N элементов. В этом случае элементу ОК соответствует единственный (первый по порядку номеров) элемент из ИК;

множественное соответствие (N:N) : в ОК и ИК содержится одинаковое число элементов. Соответствующие друг другу элементы множеств ОК и ИК идентифицируются одним и тем же порядковым номером.

В качестве упражнений первого типа (1:N) могут выступать, к примеру, следующие:

а) подобрать для слова, словосочетания или предложения на иностранном языке правильный перевод из предложенных N вариантов ответа;

б) для некоторого описываемого (или изображенного) явления, предмета или процесса выбрать правильное название (или изображение) из нескольких альтернатив. Примеры упражнений второго типа (N:N):

а) имеется некоторый набор фраз, представляющий правильный, логически связный текст на родном языке (диалог, стихотворение и т. п.) и соответствующее ему множество неупорядоченных фраз на иностранном языке; требуется расставить фразы иностранного языка в правильном порядке;

б) установить соответствия между картинками и их названиями.

Каждое упражнение снабжается заголовком, в котором формулируется инструкция для обучаемого. В структуре упражнения предусматривается задание следующих его атрибутов: степени сложности и номера игровой формы. Степень сложности — специальная количественная характеристика, влияющая на начисление баллов за упражнение. Игровая форма задает тот или иной сценарий, в соответствии с которым должно быть предъявлено данное упражнение (набор игровых форм predetermined в системе). В каждой игровой форме предусматривается определенный способ рассогласования логических связей между элементами композиций. Задача обучаемого заключается в восстановлении нарушенных связей.

Более емкая единица учебной информации — урок — задает последовательный порядок работы обучаемого с отдельными упражнениями. Причем любое упражнение одновременно может фигурировать в различных уроках. Каждый урок снабжается заголовком, в котором задается краткая формулировка целей и задач урока.

Упражнения и уроки, относящиеся к определенной тематике, объединяются в более крупную единицу учебной информации — тему, имеющую название (заголовок).

Основные режимы работы системы

Системы функционируют в двух основных режимах: «Учитель» и «Ученик». Перед началом работы пользователь настраивается на интересующий его режим с помощью графического меню и специального символического курсора. Для достижения единообразия работы в системе и регулирования обстановки пользователя предусматриваются следующие операции:

- запрос справочной информации, относящейся к текущему состоянию работы пользователя;
- переход к следующему состоянию работы;
- возврат к предыдущему состоянию работы;
- окончание сеанса работы и передача управления операционной системе;
- переключение звукового сопровождения;
- изменение палитры цветов.

Перечисленные операции выполнимы на всех этапах работы в системе.

Режим «Учитель» предназначен для ввода, структурирования и последующей корректировки информации, используемой в процессе обучения. Преподаватель начинает работу с конкретизации темы. В системе ведется специальный каталог тем, формируемый пользователем. Для этих целей предусматривается набор операций, обеспечивающих следующие возможности:

создать новую тему (занести ее заголовок в каталог);
уничтожить тему (исключить ее заголовок из каталога);
выбрать тему для дальнейшей работы с ней (сделать тему текущей).

Выбрав определенную тему, пользователь переходит (с помощью специального графического меню) в один из двух возможных подрежимов работы — с упражнениями или с уроками. Каждый из подрежимов обеспечивает различный сервис в зависимости от типа доступных объектов (упражнений или уроков). При выборе конкретного подрежима на экране дисплея высвечивается каталог с заголовками объектов соответствующего типа. Пользователь может выбрать конкретный объект (упражнение или урок) для дальнейшей работы с ним либо создать новый, присоединив его к списку уже существующих объектов того же типа. Эти действия осуществляются с помощью символического курсора путем указания на заголовок интересующего объекта.

При создании нового упражнения пользователь должен: объявить тип соответствия в новом упражнении (простое или множественное);

здать ИК и ОК, предварительно определив тип элементов в каждой из них (содержимое ОК и ИК вводится с помощью двух окон, симметрично расположенных на экране дисплея; в зависимости от указанного типа соответствий окна разбиваются на необходимое число равных частей :1 или N);

сформулировать задание (заголовок) к упражнению, т. е. установить смысл соответствия;

определить степень сложности упражнения для правильной оценки знаний обучаемого;

выбрать игровую форму предъявления данного упражнения.

При формировании композиции, содержащей текстовые элементы, преподавателю достаточно уметь работать с обычным текстовым редактором. При задании композиции с графическими элементами их нужно выбрать из встроенной в систему библиотеки «образов». В рамках режима «Учитель» имеется специальный подрежим, позволяющий поддерживать такую библиотеку. Работая с ней, пользователь может создавать, корректировать, сохранять графические изображения во внешней памяти, уничтожать их.

Если объектом работы — созданный ранее упражнение, пользователь может: отредактировать его, т. е. изменить содержимое композиций, заголовок, сложность и игровую форму; удалить из каталога; «пометить», чтобы не забыть включить его в состав какого-либо урока.

На основе созданных упражнений пользователь формирует новый урок, определяя последовательность входящих в него упражнений. При этом он просматривает каталог имеющихся в данной теме упражнений и указывает на те, которые считает нужным включить в урок. Здесь пользователю облегчат работу «помеченные» ранее упражнения.

Если объектом работы является ранее созданный урок, пользователь может отредактировать его (изменить последовательность включенных в него упражнений или заголовков) или уничтожить (исключить его из каталога).

Режим «Ученик». Конечная цель введения учебной информации — последующее ее использование в режиме «Ученик». Введенный учебный материал используется для приобретения новых знаний, контроля или проверки (повторения) уже имеющихся. Особенности данного режима заключаются в наличии разнообразных форм предъявления упражнений, включении ИК, предоставлении обучаемому свободы в выборе очередного шага обучения.

Работа обучаемого начинается с выбора конкретной темы и урока из соответствующих каталогов. Сообщив системе свое имя, обучаемый может просмотреть «десятку лучших» по данному уроку — список имен лучших учеников и количество набранных ими баллов. Обучаемый имеет возможность, повторно выполнив урок, повысить свои результаты и попасть в «десятку лучших» или переместиться на более высокий уровень в ней.

Перед выполнением урока у обучаемого запрашивается степень его подготовленности, которая учитывается при начислении баллов. Система начисления баллов организована так, что обучаемый не заинтересован в завышении или занижении значения этого показателя. Чем выше значение степени подготовленности, тем большие премии начисляются обучаемому за правильные и быстрые ответы, однако в случае неправильных ответов велики и начисляемые штрафы. Если задано низкое значение данного показателя, то штрафы будут невелики, но небольшими станут и премии.

Общее время на выполнение каждого конкретного упражнения ограничено и зависит от ряда факторов (сложность упражнения, общее число представленных в композициях элементов, специфика игровой формы, степень подготовленности обучаемого). Обучаемому предостановлена работа с последовательностью упражнений, включенных в выбранный урок. На экране высвечивается заголовок предлагаемого упражнения, в котором кратко формулируется задание обучаемому.

Информация, заключенная в упражнении, предъявляется обучаемому с использованием различных игровых форм. Каждая форма реализует определенный игровой сценарий, в соответствии с которым элементы композиций интерпретируются тем или иным образом. Общим свойством предлагаемых игровых форм является то, что в каждой из них зафиксированными остаются только элементы ОК, по отношению к которым обучаемому требуется подобрать соответствующие элементы из ИК. При этом последние могут либо явно изображаться на экране дисплея, либо быть представленными неявно (отсутствовать на экране). Обучаемый имеет возможность «играть» (манипулировать) этими элементами: предлагать свои варианты элементов, сочетать их с определенными элементами ОК, составлять из них различные комбинации. Этим, в частности, объясняется выбор названий для композиций («опорная» и «игровая»).

Игровые формы различаются в зависимости от заложенных в них схем рассогласования связей между элементами композиций. В системе ЭРУДИТ представлены три игровые формы, инвариантные относительно структуры и конкретного наполнения упражнения (набор игровых форм может быть расширен подключением новых с соблюдением требований интерфейса).

В первой игровой форме элементы ОК и ИК организованы в виде двух замкнутых цепочек, одна из которых неподвижна, а другая перемещается относительно первой. В определенные моменты времени цепочки соответствуют друг другу, образуя ряд связанных пар. Обучаемому требуется зафиксировать момент совпадения цепочек. Заметим, что для восстановления всех рассогласованных связей необязательно знание всех пар соответствий, достаточно определить хотя бы одну из них (по единственному «ключу» — соответствующей паре элементов — можно восстановить все оставшиеся связи). Можно составлять и более сложные задания, например на выявление закономерностей. В этом случае знания одного ключа недостаточно. Обучаемому требуется проанализировать состав обеих цепочек, выявить критерий, в соответствии с которым упорядочены элементы в каждой из них, и лишь тогда совместить цепочки.

Во второй игровой форме элементы ОК выводятся на экран в виде звеньев разомкнутой цепочки по порядку своих номеров. Элементы ИК представляются только в беспорядке. Обучаемый, манипулируя элементами ИК, должен сформировать из них цепочку, согласующуюся с заданной. Данная игровая форма полезна для закрепления в сознании обучаемого не очень прочных ассоциативных связей, когда отдельные компоненты некоторой конструкции достаточно хорошо усвоены, но неясно, как они согласуются друг с другом. Предлагаемая игровая форма может быть использована при составлении заданий на установление истинности высказывания, определение правильного ответа на вопрос среди предложенного мно-

жества альтернатив, упорядочение перепутанных фрагментов информации и др.

В третьей игровой форме элементы ОК поочередно выводятся на экран, элементы ИК на экране не изображаются. Обучаемый должен для каждого предъявляемого элемента ОК воспроизвести на экране (например, набирая текст с клавиатуры) соответствующий элемент ИК. Данный обучаемым ответ соотносывается с хранимым эталоном. Эта игровая форма — более сложная для обучаемого, так как не предусматривает выбора ответа из списка предложенных. Она может быть полезной в заданиях на тренировку памяти, классификацию, воспроизведение недостающего фрагмента информации и др.

В описанных игровых формах предусматриваются поощрения за правильно выполненные упражнения. При этом используются графика, мультипликация, звуковое сопровождение и другие приемы для поддержания интереса. Для успешного выполнения упражнения обучаемому необходимо проявлять не только свои знания, но и определенную сообразительность и реакцию.

Варианты эксплуатации и характеристики системы

Предлагаются два варианта эксплуатации системы ЭРУДИТ.

Первый вариант предполагает разделение (в лицах) пользователей на следующие группы: преподавателей и обучаемых. Преподаватели наполняют систему конкретным учебным материалом, а обучаемые осваивают его, выступая по отношению к первым в качестве потребителей. Задача преподавателя в этом случае заключается в подборе упражнений, обеспечивающем прогресс знаний обучаемых.

Второй вариант — совмещение в одном лице как преподавателя, так и обучаемого. В этом случае система используется для самообучения: работая в режиме «Учитель», пользователь самостоятельно подготавливает необходимый учебный материал и затем закрепляет его в режиме «Ученик». Регулярно пополняя и обновляя запас учебного материала и упражняясь на нем, пользователь поддерживает соответствующие знания в активном состоянии.

Для работы в системе ЭРУДИТ от пользователя не требуется почти никакой специальной подготовки. В случае необходимости система предоставит ему помощь. Дружелюбность и простота в использовании делают эту систему доступной как профессионалу, так и дилланту.

ЭРУДИТ реализована в рамках MS DOS на ПЭВМ типа IBM PC/XT/AT с оперативной памятью не менее 128 Кбайт и цветным графическим дисплеем в системе программирования Turbo Pascal.

Телефон 939-18-80, Москва, Татьяна Николаевна Щеглова.

ЛИТЕРАТУРА

1. Брябри В. М. Программное обеспечение персональных ЭВМ. — М.: Наука, 1988.
2. Брябри В. М., Горячая И. В. ЭРУДИТ // Информатика и образование. — 1987. — № 4. — С. 76—79.
3. Горячая И. В. Инструментальные средства для построения обучающих систем на персональной ЭВМ: Автореф. дис. ... к. ф. м. н. — Москва, 1989.
4. Горячая И. В., Щеглова Т. Н., Шербатюк Л. Л. Обучение на базе ПЭВМ / Вопросы прогр. обеспечения ЭВМ. — М.: МГУ, 1987. — С. 7—12.

Статья поступила 26.08.87

А. А. Акаев, И. А. Аккозиев, С. А. Алымкулов,
Т. М. Муратов, А. Н. Нуркамилов

ШКОЛЬНАЯ ПЭВМ МЕКТЕП

ПЭВМ построена на базе МПК БИС серии К580. В состав ПЭВМ входят резидентное ОЗУ, страничное ЗУ, контроллеры клавиатуры, кассетного магнитофона и видеотерминала, общая системная магистраль, клавиатура, блок питания. Кроме этих основных узлов, обеспечивающих непосредственное функционирование базовой модели ПЭВМ, в ее состав включен минимальный набор периферийных БИС: параллельный интерфейс ввода-вывода, контроллер прерываний, БИС таймера, необходимый для создания автоматизированных обучающих классов.

Все основные узлы ПЭВМ связаны с МП через общую системную магистраль. Конструктивно ПЭВМ выполнена в виде одной двусторонней печатной платы размерами 360×260 мм.

Основные технические характеристики ПЭВМ МЕКТЕП

Процессор	КР580ИК80А
Объем памяти, Кбайт	
ПЗУ	16
ОЗУ	64
Страничная память	128 Кбайт... 1 Мбайт
Устройство отображения	Бытовой ТВ-приемник
Внешнее запоминающее устройство	Кассетный магнитофон
Программное обеспечение	Монитор, редактор текста, интерпретатор БЕЙСИК

Резидентное ОЗУ построено на микросхемах К565РУ5 (регенерация ячеек ОЗУ осуществляется программно-аппаратным способом в очередных циклах организации прямого доступа к памяти); ПЗУ — на микросхемах К573РФ4 и К573РФ5. В ПЗУ хранятся монитор, драйверы периферийных устройств, тестовые программы самопроверки работоспособности основных узлов системы.

Организация страничной памяти позволяет использовать МЕКТЕП в системах автоматизации сложных технологических процессов контроля, делопроизводства и обучения учащихся курсам информатики и вычислительной техники в школах. Страничная память построена на элементах К573РФ4 и К565РУ5. В 16 Кбайт страничного ПЗУ хранится интерпретатор БЕЙСИК. Каждая страница имеет емкость 32 Кбайт. Максимальное число страниц — 128.

Использование в качестве дисплея ТВ-приемника определяет применение программируемых контроллеров видеотерминала КР580ВГ75 и прямого доступа к памяти КР580ВТ57, совместно выполняющих функции воспроизведения и регенерации изображения на экране ЭЛТ в кодах КОИ-7 из экранного ОЗУ. На экране отображаются 64 символа в 25 строках текста. Предусмотрены символы псевдографической информации. Знакогенератор контроллера видеотерминала, реализованный на ИМС К573РФ5, дает возможность отображать 256 символов в двух растровых матрицах: 6×8 и 8×16 точек.

С помощью одного программируемого параллельного интерфейса ввода-вывода КР580ИК55 организована информационно-управляющая связь ПЭВМ с клавиатурой и внешним накопителем на кассетном магнитофоне. Клавиатура представляет собой часть печатной платы ПЭВМ с установленными на ней 75 клавишными переключателями типа ВН. В процессе разработки учитывались особенности школ, проводящих обучение на национальном языке (программно-аппаратным способом реализованы буквы киргизского алфавита).

Контроллер кассетного магнитофона (программируемый параллельный интерфейс ввода-вывода КР580ИК55) позволяет записывать информацию на магнитную ленту последовательно бит за битом по методу двухфазного кодирования. Распаковка, упаковка и проверка на достоверность вводимой и выводимой информации выполняются подпрограммой обслуживания контроллера.

Обучающий класс на основе ПЭВМ МЕКТЕП содер-

жит ведущую ПЭВМ с дополнительным внешним устройством памяти (квазидиск) для хранения обучающих программ и 8...11 рабочих станций на основе базовой модели ПЭВМ МЕКТЕП. Обучающая программа отдельного урока хранится в банке прикладных программ. Опытные образцы ПЭВМ МЕКТЕП установлены в ряде школ Киргизии.

720057, Фрунзе, пр. Мира, 66, Фрунзенский политехнический институт; тел 44-09-76, 44-56-81

Статья поступила 4.12.88

А. В. Александров, М. Г. Весноватов, В. В. Павлов

УЧЕБНЫЙ ВИДЕОФИЛЬМ «ПЭВМ: СРЕДСТВА ОТЛАДКИ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ»

Видеофильм является логическим продолжением курса лекций, анонсированного в журнале «Микропроцессорные средства и системы» 1987, № 6. Цель курса — дать начальные знания по эксплуатации ПЭВМ, особенностям их аппаратного и программного обеспечения. В качестве учебного вычислительного средства используются ПЭВМ ряда ДВК. Видеокурс состоит из 11 лекций.

Лекция 1 — назначение ДВК, варианты исполнения и модели ПЭВМ, аппаратные средства, центральный процессор, интерфейсы, возможность подключения дополнительных устройств, команды пульта терминала.

Лекция 2 — система команд процессора ДВК, методы адресации памяти, машинные коды основных команд процессора, примеры составления простейших программ в машинных кодах.

Лекция 3 — системное программное обеспечение ДВК; понятие ОС: состав и назначение отдельных компонентов; форматы данных и файлов; типовой процесс прохождения задачи при использовании различных языков программирования.

Лекция 4 — основной компонент ОС — монитор; виды мониторов ОС ДВК, основные команды, необходимые для получения начальных навыков работы с ДВК.

Лекция 5 — понятие дистрибутива ОС; генерация рабочей версии ОС ДВК, состав системного диска и порядок его создания.

Лекция 6 — работа с текстовой информацией; различные редакторы текстов, входящие в состав ОС ДВК, их преимущества и недостатки для конкретных приложений.

Лекция 7 — языки программирования, трансляторы которых входят в ОС ДВК; краткие характеристики и основные области применения.

Лекция 8 — использование ДВК в качестве автоматизированной информационно-поисковой системы; два вида систем различной сложности.

Лекция 9 — методы и средства отладки микропроцессорных устройств: классификация отладочных средств, основные области применения, преимущества и недостатки каждого класса.

Лекция 10 — кросс-средства отладки на примере новых версий пакетов МИКРОС-816 и МИКРОС-821, особенности их использования на различных типах ЭВМ.

Лекция 11 — универсальный отладочный комплекс МИКРОСОТ, его состав, возможности, технические характеристики; особенности применения устройств, входящих в состав комплекса.

Видеофильм не предназначен для замены всего комплекта технической документации на ЭВМ, но наши лекции должны существенно облегчить Вам период начального знакомства с вычислительной техникой.

Телефон для справок об условиях поставки 468-81-75, Москва

ПОКОЛЕНИЕ С КОМПЬЮТЕРОМ*

Фрагмент беседы с Александром Квашиевским, министром по делам молодежи Польской Народной Республики

— Уверен, что высокого уровня компьютеризации мы достигнем гораздо раньше, чем утверждают скептики.

— Каким способом?

— Разными. Скорее всего, это будет компьютеризация «по-польски»: с самым разнообразным оборудованием, без единой системы, с большим процентом частного импорта. Компьютеры быстро войдут в нашу жизнь, как это было с видео и другими новинками. А, например, через посредство вашего журнала создаются условия для заключительного броска в будущее, который должна будет сделать Польша и все мы.

— Вы хотите сказать, что компьютеризация «по-польски» определяется в большой степени заграничными или частными фирмами, в которых работают, как правило, молодые люди и руководит ими молодежь. Эти фирмы сталкиваются с разными, часто весьма негативными оценками их деятельности. Как Вы смотрите на эти фирмы?

— Критицизм тех, кто считает, что каждый, зарабатывающий много денег, заслуживает осуждения, отвергаю полностью. Фирмы заполняют определенную нишу на нашем рынке и функционируют в соответствии с законом; их следует оценивать положительно. Особенно те, которые не занимаются поставками дорогой парфюмерии или чего-то подобного, а поставляют продукцию новейшей техники, необходимую для развития народного хозяйства. Как бы это ни показалось удивительным, следует с уважением относиться и к тем, кто привозит компьютеры из-за границы на собственные средства.

— Почему мы не можем производить свои компьютеры?

— Конечно, необходимо выпускать большое количество персональных компьютеров. Причин, не позволяющих это делать, — множество. Электроника всегда была в тени тяжелой промышленности, к тому же эмбарго со стороны западных стран в начале 80-х годов изолировало ее от новейших мировых достижений в этой области. Чтобы насытить рынок компьютерами, требуется как развитие массового производства, так и возможность дальнейшего свободного развития малых фирм, эластичных и эффективных, приносящих доход своим владельцам и удовлетворение работающим там людям и клиентам. Большой электроники без большой промышленности не создадим, и то, что делают малые предприятия, заслуживает внимания.

— Если вашу точку зрения разделяет большинство в правительстве, то не следует ожидать введения таможенных пошлин на микрокомпьютеры и новых налогов на компьютерные фирмы.

— Доводом в пользу стратегии поддержки электроники правительством является факт отмены пошлины на ввоз из-за границы электронных изделий. Если государственные предприятия будут производить достаточно много продукции, которая сможет конкурировать с импортом, к проблеме пошлины надо будет вернуться. Однако до этого еще далеко. То же касается и налогов на малые фирмы. Условия, в которых они функционируют, наверное, могли бы быть лучшими, но, как видно, и существующие условия дают возможность для их развития. Если, однако, в соответствии с государственным планом к 1990 г. в нашей стране начнется многосерийное производство микрокомпьютеров, малые фирмы вынуждены будут изыскивать новые формы своей деятельности.

— Чтобы выиграть битву за компьютеризацию, потребуются не только компьютеры, но и армия информатиков, электроников и т. д. Между тем, мы не можем полагаться на излишки высококвалифицированных специалистов в этой области, а, кроме того, самые лучшие из них редко идут работать в промышленность.

— Конкуренция со стороны малых фирм и тут сыграет положительную роль. Быть может, она заставит директоров предприятий обратить внимание на кадры. За многие

годы мы привыкли смотреть на кадры как на дешевый и вполне доступный ресурс. Людей для работы всегда хватало: нас много. Однако люди компетентные и предприимчивые — большая редкость не только в Польше.

А мы так и не создали систему, которая бы таким людям помогала, выдвигала их. Работающих может быть миллионы, но по настоящему творцов технического прогресса — всего сотни либо тысячи. Мы четко представляем, что следует очень активно проводить кадровую политику и лучших использовать как можно больше, но очевидные положения программы с трудом пробивают себе дорогу в жизнь.

Сегодня в воображении многих людей компьютер является машиной, которая исполняет наши желания, как только нажмешь кнопку. Но программирование компьютера, его обслуживание — это тяжелый и подчас изнурительный труд. Очевидно, разрушение стереотипа произойдет вместе с развитием эрудиции в области информатики. Необходимо понять, что проблемы, связанные с новой техникой, нельзя воспринимать упрощенно. Компьютер ставит высокие требования своему партнеру — человеку. Каждый в этом должен убедиться сам, — публикация здесь мало может помочь. Я доволен тем, что в создаваемых компьютерных клубах большая часть молодежи сама в этом может убедиться.

— Считаете ли Вы возможным допустить, что какая-то фирма, государственная или частная, повторит успех, выпавший на долю Apple в зарождении микрокомпьютера?

— Безусловно. Мало того, думаю, что фирму, которая повторит карьеру Apple, может создать «Байтек». Воспитывая новое мышление, Вы одновременно растите компьютерную генерацию. Попробуйте собрать Ваших читателей под знаменем польской «Кремневой Долины». Будем решать, какая организационная форма для этого подойдет. Давайте начнем сегодня же! Покажем всем консерваторам, разочарованным, уставшим, что можно силами компетентной молодежи создать фирму, которая будет выпускать несколько тысяч компьютеров ежегодно. Я совсем не шучу. Пусть начнется с программирования...

— Однако сегодня нелегко заявить читателям, что наши шансы на появление микрокомпьютеров связываются с призывом к ним о создании этих микрокомпьютеров.

— Компьютеров уже много. Будет серийная продукция, и она будет относительно дешевле. Будут в школах, уже есть в клубах. Среди Ваших же читателей много таких, которые размышляют об отечественной «Кремневой Долине». Это может стать знаменем «Байтека». Пусть попробуют. Я готов в рамках Государственного Фонда молодежи выделить им кредит на развитие этой деятельности. Быть может, таким способом быстрее исполнятся мечты молодежи о компьютере и современности.

Беседу вели *Грегор Онихимовски* и *Роман Познаньски*
(Перевод — *Забаринской С. П.*)

«ЖУРНАЛ НОВЫХ ПОКОЛЕНИЙ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ»

издается по поручению Координационного совета по вычислительной технике и информатике Академий наук социалистических стран в Центральном институте кибернетики и информационных процессов АН ГДР.

Журнал предусмотрен для всех видов публикаций, посвященных разработке вычислительных систем с новыми свойствами и областями применения, т. е. созданию новых поколений вычислительных систем.

Главной целью журнала является обмен информацией о последних разработках в области аппаратного и программного обеспечения, а также их применения для обработки знаний и больших массивов данных.

Выходит один том в год (4 номера). Годовая подписная цена: 120 марок ГДР.

На «ЖНПВС» можно подписаться в Главном управлении Союзпечать (просп. Маршала Жукова, 4, Москва).

* Журнал «Байтек» (ПНР). — 1987. — № 4.

УДК 681.324.06

В. В. Корнеев, О. Г. Монахов

ПАРАЛЛЕЛЬНЫЙ ПАСКАЛЬ ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ

Язык параллельного программирования Паскаль-С создан путем расширения языка Паскаль за счет введения операций порождения, уничтожения, а также взаимодействия (синхронизации и обмена данными) процессов, распределенных по машинам. При этом фрагменты программы, обрабатывающие данные, программируются на стандартном языке Паскаль, а для синхронизации процессов обработки и организации межмашинных обменов данными используются введенные операции, реализующиеся с помощью набора операций ядра ОС МИКРОС [1, 2]. Последнее содержит подпрограммы интерпретации операций над основными объектами: семафорами и процессами. Доступ пользователя языка Паскаль-С к операциям ядра ОС осуществляется обращением к подпрограмме, имеющей общее стандартизованное описание для любой операции ядра. Типы объекта, над которым производится операция, и операции задаются параметрами подпрограммы. Следует отметить, что по стилю параллельного программирования языка Паскаль-С и ОККАМ близки [3].

Описание процессов. Процесс — последовательность выполнения программы $P = \langle X, A \rangle$, где X — множество переменных, A — конечная последовательность действий над переменными из X . Множество X состоит из подмножества локальных переменных процесса U и выделенного подмножества глобальных переменных S , называемых семафорами: $X = U \cup S$. Каждое действие из A имеет вид

$$m: \text{when } \Phi(S) \text{ do } A_m(X) \text{ od } g(Y); t(Y),$$

где $\Phi(S)$ — предикат, определенный на множестве семафоров; m — метка действия; $A_m(X)$ — преобразование: $x_1 \leftarrow A_m^1(X), \dots, x_n \leftarrow A_m^n(X)$; $g(Y)$ — преобразование локальных переменных, осуществляемое вне критического интервала, задаваемого операторными скобками **do od** [4]; $t(Y)$ — либо оператор выбора следующего действия процесса, либо оператор останова. Предикат $\Phi(S)$ проверяется при достижении текущей точкой процесса метки действия m . Если данный предикат истинен, то действие выполняется, иначе процесс остается в текущей точке, ожидая выполнения предиката $\Phi(S)$. Предикат $\Phi(S)$ задается в виде

$$\Phi(S) = \left(\bigwedge_{k \in A} S_k > 0 \right) \wedge \left(\bigwedge_{k \in B} S_k = 0 \right) \wedge \left(\bigwedge_{k \in C} S_k < 0 \right),$$

где $\{S_k | k \in A \cup B \cup C\}$ — подмножество семафоров.

Информационный вектор процесса либо загружен в процессор, либо находится в очереди готовых процессов (семафор **READY**), либо помещен в очередь ждущих процессов (семафор **WAIT**). Соответственно процесс при этом находится в текущем, готовом и ждущем состояниях. Порождение процесса сводится к помещению его информационного вектора либо в процессор элементарной машины [ЭМ] [4, 5], либо в одну из очередей: **READY** или **WAIT**, а уничтожение процесса — к удалению информационного вектора из этих очередей или процессора. Программа, исполняемая процессом (тело процесса), зада-

ется в виде процедуры на языке Паскаль-Си. Адрес начала данной процедуры указывается в информационном векторе процесса на этапе его формирования.

Описание семафоров. Семафор S — тройка $\langle \text{sem}(S), M_S, I_S \rangle$, где $\text{sem}(S)$ — значение семафора S , M_S — упорядоченное множество, состоящее из $\text{sem}(S)$ элементов; I_S — множество допустимых операций над семафором S . Когда элементы множества M_S представляют собой структуры данных, семафор S называется *структурированным*. Операции над семафором S включают (исключают) элементы множества M_S и изменяют значение семафора S . Если не требуется различать и использовать элементы множества M_S , а важно лишь знать значение мощности этого множества, то семафоры S называются *неструктурированными*. Операции над этими семафорами меняют их значения.

Семафоры-устройства (**PROCESSOR** — информационный вектор текущего процесса и **MEMORY** — список свободных слов оперативной памяти ЭМ) содержат множества, элементы которого — массивы программно доступных регистров устройства. Операции над семафорами-устройствами — определение состояния, захват и освобождение устройства.

Семафоры позволяют создавать инкапсулированные абстрактные типы данных [6], определяя соответствующим образом структуру элементов семафора и набор операций над ним.

Все операции над семафорами реализуются с помощью процедуры **NUC**. Операции с элементами структурированного семафора имеют обращение **NUC** ($N, M, \text{COL}, \text{SEM}, \text{MODA}$), где **MODA** — тип операции; **SEM** — имя семафора; **COL** — адрес массива дескрипторов доступа элементов семафора; M — номер элемента семафора, после которого необходимо вставить (удалить, прочитать) N элементов, заданных в массиве дескрипторов **COL**. Набор операций, задаваемых параметром **MODA**, включает операции начальной установки семафора, внесения, удаления, чтения элементов семафора из заданного места (после указанного элемента, из начала, из конца).

Над структурированными семафорами включены операции чтения (удаления) N элементов с заданным свойством (в этом случае M — адрес программы проверки заданного свойства у элемента семафора). С помощью структурированных семафоров легко создаются такие структуры данных, как списки, очереди, стеки, буферы.

Операции над неструктурированными семафорами: задание значения семафора, равного e (где e — целое число); уменьшение (увеличение) на e значения семафора. Неструктурированные семафоры используются для синхронизации процессов, как счетчики числа ресурсов.

Операция над семафором **MEMORY**: создание списка из свободных массивов слов оперативной памяти; добавление в список свободных массивов слов, заданных дескрипторами; извлечение из списка N свободных массивов слов. С помощью этих операций динамически используется память ЭМ и порождаются-уничтожаются объекты ядра.

Операция порождения семафора **CRTSEM** заносит в дескриптор доступа значение имени порожденного семафора заданного типа. По имени семафор уничтожается операцией **DSTSEM**.

Операцией **CRTRRS** процесс порождается занесением его информационного вектора в очередь ждущих процессов. Процесс с утилизацией, используемой его памятью, уничтожается операцией **DSTPRS**. Поиск уничтожаемого процесса выполняется по имени.

Над семафором PROCESSOR определены следующие операции:

CRTPSW — создает информационный вектор в начале любого нового действия процесса и записывает в информационный вектор текущего процесса имена семафоров, необходимых для вычисления $\Phi(S)$, а затем загружает измененный вектор в процессор ЭМ посредством операции INPROC;

INPROC — проверяет значение предиката $(\bigwedge_{k \in A} S_k > 0) \wedge (\bigwedge_{k \in B} S_k = 0) \wedge (\bigwedge_{k \in C} S_k < 0)$ загружаемого процесса.

Если значение предиката равно false, то информационный вектор процесса, загружаемого в процессор, помещается в очередь ждущих процессов. Из очереди готовых процессов READY выбирается информационный вектор процесса, который загружается в процессор. Указанный предикат получает значение true, если значение всех семафоров, имена которых принадлежат множеству A, больше нуля; множеству B, равны нулю; множеству C, меньше нуля. Если значение предиката равно true, поля A, B, C получают значение 0 (нуль), а текущий процесс входит в свой критический интервал (т. е. STATE \geq 0).

Установка семафора STATE в нуль и конец критического интервала текущего процесса задаются с помощью оператора od над семафором STATE. Эта операция разрешает прерывание текущего процесса.

Взаимодействие процессов в МИКРОС происходит либо через общение переменных (семафоры), либо (если процессы расположены в разных ЭМ) посредством виртуальных линий связи операциями приема и передачи данных.

В системах с программируемой структурой [4, 5] входным и выходным полюсам ЭМ, используемым для подключения линий, приспаны метки $p, p \in \{1, 2, \dots, v\}$, где v — степень графа межмашинных связей. Линия связи, соединяющая эти полюса ЭМ и обозначенная через (i_p, j_q) , имеет в ЭМ_i выходную очередь Вых_p, а в ЭМ_j — входную очередь Вх_q. Работа линии связи активизируется, если в Вых_p есть элемент данных, а в очереди Вх_q — свободное место для принятия элемента данных. При такой организации линий интерпретация команды передачи данных из ЭМ_i в ЭМ_j сводится к вынесению в качестве элемента в очередь Вых_p данных, которые необходимо передать из ЭМ_i по линии (i_p, j_q) в ЭМ_j. Команда приема данных из ЭМ_j, выполняемая в ЭМ_i, выберет элемент данных из входной очереди Вх_q линии (i_p, j_q) . Если при выполнении команды приема обнаруживается, что входная очередь пуста, то завершение команды задерживается вплоть до момента поступления элемента данных в очередь Вх_q.

Таким образом, взаимодействие процессов посредством виртуальных линий связи обеспечивается операциями передачи SEND (M, P) и приема RECEIVE (M, P), где M — сообщение, P — номер полюса (входной/выходной очереди — семафоры INPUT_p/OUTPUT_p).

Программа на языке Паскаль-С представляет собой совокупность процессов, оформленных в виде процедур и взаимодействующих через общие переменные — семафоры (см. рисунок). Семафоры и процессы — базовые объекты, лежащие в основе системы программирования МИКРОС. Процесс-производитель, запускаемый по условию S=0, производит выборку блока длиной 10 ячеек из списка свободной памяти (семафор MEMORY) и заносит его в структурированный семафор S (в конец списка); запускаемый по условию S>0, выбирает из начала семафора S первый элемент и заносит его в список свободной памяти. Порождение семафора S и процессов и задание их предикатов происходит в головной программе.

Особенности реализации и применения языка Паскаль-Си. Введенные средства параллелизма в язык Паскаль дают возможность пользователю писать асинхронные, параллельные программы в терминах процессов и семафоров и образуют версию языка, названную системный Паскаль (Паскаль-С).

```

*****
*          ПРОГРАММА "ПРОИЗВОДИТЕЛЬ-ПОТРЕБИТЕЛЬ"          *
*          ЯЗЫК ПАСКАЛЬ-С, ОС РАФОС/ОС МИКРОС              *
*****
TYPE DESC=RECORD CASE COL:0..1 OF/ОПИСАНИЕ ТИПА
U:(ADR: INTEGER; /ДЕСКРИПТОР ДОСТУПА
BUF: ^INTEGER;
LEN: INTEGER;
1:(ADR1: INTEGER;
BUF1: ^IVEC;
LEN1: INTEGER;
END;

VAR D, DV, DV1, DV2: DESC; /ДЕСКРИПТОР ДОСТУПА
IV1, IV2: IVEC; /ИНФОРМАЦИОННЫЕ ВЕКТОРЫ
MEMORY, PROCESSOR, STATE, S: INTSEK; /СЕМАФОРЫ

PROCEDURE NUC(N, M: INTEGER, VAR COL: INTEGER,
SEM, MODA: INTEGER); /ОПИСАНИЕ ВНЕШНЕЙ ПРО-
EXTERNAL; /ЦЕДУРЫ ОБРАЩЕНИЯ К ЯДРУ

PROCEDURE P1; /ПРОЦЕСС-ПРОИЗВОДИТЕЛЬ
VAR D1: DESC;
BEGIN
NUC(0, 0, DV1.COL, PROCESSOR, CRTPSW); /НАЧАЛО КРИТИЧЕСКОГО ИНТЕРВАЛА
D1.LEN:=10; /ВЫБОР БЛОКА ИЗ 10 СЛОВ
NUC(0, 1, D1.COL, MEMORY, DLTMEM); /ИЗ СВОБОДНОЙ ПАМЯТИ
IF D1.BUF<>NIL THEN /ЕСЛИ ТАКОЙ БЛОК ЕСТЬ, ТО
NUC(1, 0, D1.COL, S, INTAL); /ЗАНЕСЕНИЕ ЕГО В СЕМАФОР S
NUC(0, 0, 0, STATE, UPDATE); /КОНЕЦ КРИТИЧЕСКОГО ИНТЕРВАЛА
END;

PROCEDURE P2; /ПРОЦЕСС-ПОТРЕБИТЕЛЬ
VAR D2: DESC;
BEGIN
NUC(0, 0, DV2.COL, PROCESSOR, CRTPSW); /НАЧАЛО КРИТИЧЕСКОГО ИНТЕРВАЛА
NUC(1, 0, D2.COL, S, DLEAL); /ВЫБОР 1-ГО ЭЛЕМЕНТА ИЗ S
NUC(0, 1, D2.COL, MEMORY, INTMEM); /ВОЗВРАТ ЕГО В ПАМЯТЬ
NUC(0, 0, 0, STATE, UPDATE); /КОНЕЦ КРИТИЧЕСКОГО ИНТЕРВАЛА
END;

BEGIN /ПРОГРАММА ИНИЦИАЦИИ
D.LEN:=5 /ПОРОЖДЕНИЕ СТРУКТУРИРО
S:=D.BUF^; /ВАННОГО СЕМАФОРА S

IV1.B:=@S; IV1.BLEN:=1; /ФОРМИРОВАНИЕ ИНФОРМАЦИОН-
IV1.PROG:=@P1; DV.BUF1:=@IV1; /НОГО ВЕКТОРА ПРОЦЕССА
DV1.BUF1:=@IV1.A; /-ПРОИЗВОДИТЕЛЯ
NUC(0, WAIT, DV.COL, MEMORY, CRTPRC); /ПОРОЖДЕНИЕ ПРОЦЕССА-ПРОИЗВО-
/ДТЕЛЯ
IV2.A:=@S; IV2.ALEN:=1; /ФОРМИРОВАНИЕ ИНФОРМАЦИОННОГО
IV2.PROG:=@P2; DV.BUF1:=@IV2; /ВЕКТОРА ПРОЦЕССА
DV2.BUF1:=@IV2.A; /-ПОТРЕБИТЕЛЯ
NUC(0, 0, DV.COL, PROCESSOR, INPROC); /ПОРОЖДЕНИЕ ПРОЦЕССА-ПОТРЕБИ-
/ТЕЛЯ

```

Программа на языке Паскаль-С состоящая из двух процессов, взаимодействующих по типу производитель-потребитель через общий семафор S.

Особенность реализации данного языка: операции над семафорами и процессами выполнены в ядре ОС; из программы пользователя, написанной на языке Паскаль-С; организованы обращения в ядро ОС к программам интерпретации этих операций. Поэтому при работе программы на языке Паскаль-С в память ЭМ должно быть загружено ядро ОС и заданы адреса некоторых семафоров ядра (MEMORY, WAIT, PROCESSOR, READY, STATE и др.), а также порождены некоторые процессы ядра (например, обслуживающие виртуальные линии связи). Для этого программа пользователя должна быть скомпонована либо с ядром, либо с таблицей меток ядра, либо, если ядро расположено в определенном месте в памяти, требуемые адреса заданы в ней самой. Кроме того, присоединяется программа инициации процессов ядра.

Особенности применения языка Паскаль-С при создании распределенной ОС МИКРОС: ограничение на объем оперативной памяти, занимаемой системными программами; невозможность использования стандартного монитора ОС РАФОС (ФОДОС, ОСДВК) в тех системах МИКРОС, в которых нет запоминающих внешних устройств (диска или ленты). По этим двум причинам разработчики вынуждены были отказаться от использования стандартного монитора (исполняющей системы) языка Паскаль/РАФОС (PASMOM) и объектной библиотеки (PASCALOBJ). Поэтому необходимо задавать область памяти для глобальных переменных в программе пользователя и при исполнении операций языка Паскаль, требующих обращений к процедурам объектной библиотеки (например, операций ум-

ножения над целыми и действительными числами, операций над множествами и файлами, операций работы с внешними устройствами), обеспечивать собственными средствами выполнение таких операций.

Заключение. Рассмотренные средства, введенные в язык Паскаль, позволяют создавать параллельные программы, ориентированные на распределенную обработку в системе МИКРОС. С помощью данных средств эффективно реализуется несколько уровней распределенной ОС для МИКРОС, обеспечивающей как режим параллельных вычислений, так и функционирование в режиме сети ЭВМ. Опыт разработчиков показывает, что программисты, знающие язык Паскаль, достаточно легко осваивают его параллельную версию.

630090, Новосибирск, 90, пр. Академика Лаврентьева, 13, ИФП СО АН СССР, лаб. вычислительных систем; тел. 35-23-71

ЛИТЕРАТУРА

1. Корнеев В. В. Элементарная машина однородной вычислительной системы с программируемой структурой // Кибернетика.— 1980.— № 1.— С. 75—81.
2. Корнеев В. В., Монахов О. Г., Тарков М. С. Ядро операционной системы ЭМ вычислительной системы с программируемой структурой // Сб. статей: Однородные вычислительные системы.— Новосибирск.— 1981.— С. 22—42. (Вычислительные системы, вып. 90).
3. Тейлор Р., Уилсон П. Язык, ориентированный на процессы и удовлетворяющий требованиям распределенной обработки // Электроника.— 1982.— Т. 55, № 24.— С. 26—34.
4. Корнеев В. В. Архитектура вычислительных систем с программируемой структурой.— Новосибирск: Наука, 1985.— 166 с.
5. Евреиннов Э. В., Хорошевский В. Г. Однородные вычислительные системы.— Новосибирск: Наука, 1978.— 319 с.
6. Данные в языках программирования / Пер. с англ.— М.: Мир, 1982.— 327 с.
7. Hansen P. V. The architecture of concurrent programs.— N. Y.: Englewood Cliffs, 1977.— 317 с.
8. Хокни Р., Джессехоуп К. Параллельные ЭВМ. Архитектура, программирование и алгоритмы.— М.: Радио и связь, 1986.— 392 с.

Статья поступила 21.07.87

УДК 681.3.06

Л. И. Брусиловский, В. В. Складов

СИСТЕМА ФАЙЛОВОГО ОБМЕНА KERMIT ДЛЯ ОРГАНИЗАЦИИ МЕЖМАШИННОЙ СВЯЗИ ПО ТЕРМИНАЛЬНЫМ ЛИНИЯМ

Традиционно для организации межмашинного обмена используются разнородные локальные сети (ЛС) [1]. Однако тиражи промышленной сетевой связанной аппаратуры значительно уступают тиражам различных типов компьютеров, поэтому приобрести ее трудно. Кроме того, сетевые контроллеры часто ориентированы на применение в составе однородных ЛС.

В этих условиях большой интерес вызывает появление ряда программных систем, обеспечивающих некоторые сетевые услуги (распределение ресурсов, файловый обмен, удаленную загрузку и удаленный запуск программ) и использующих аппаратуру асинхронных последовательных интерфейсов (терминальные линии связи). К ним относится система файлового обмена между разнородными вычислительными машинами — KERMIT [2] (см. таблицу).

Основные реализации системы KERMIT

ЭВМ	ОС	Язык реализации
IBM 370 EC ЭВМ IBM 370, EC ЭВМ CDC Cyber 170 Sperry/Univac 1100	VM/CMS, MVS/TSO MTS NOS EXEC	IBM assembler Pascal FORTRAN-77 EXEC assembler PL/1 BLISS-32, MACRO 32 MACRO 11
Honeywell DEC VAX-11, CM1700, «Электроника 82» DEC PDP 11, CMЭВМ, «Электроника 60»	Multics VMS RT, RSX, RSTS, MUMPS	
ДВК2, ДВК3, ДВК4 VAX, PDP 11, CM ЭВМ, «Электроника 60» ДВК2, ДВК3, ДВК4 DEC PRO 300 SERIES «Электроника 85»	UNIX P/OS	C BLISS-16, MACRO 11 PL/P FORTRAN RETFOR
PRIME HP1000 HP3000, Univac	PRIMOS RTE Software Tools	RATFOR FORTRAN DR ASM MS MASM86
Data General Data General I8080, I8085, Z80 I8086, I8088	AOS RDOS CP/M80 PC DOS, MS DOS CP/M86	DR ASM86 DEC-10/20 CROSS BASIC
I8086, I8088 Apple2 6502	Apple DOS	
ATARI Home Computer	DOS	

Область применения системы KERMIT

Система позволяет строить двухточечные соединения машин через терминальные порты, что, в принципе, легко осуществить, так как практически для любой ЭВМ связь с терминалом реализована стандартным образом как аппаратно (потенциальные сигналы по стандарту EIA RS-232 или токовые по стандарту 20 МА), так и программно (обмен кодами ASCII).

Система представляет собой программную надстройку над средствами ОС для работы с многотерминальной аппаратурой или линией связи. Она поддерживает функции распределения ресурсов и файлового обмена. Одна из машин, на которой запущена программа KERMIT, эмулирует терминал другой, т. е. становится «виртуальным терминалом», что обеспечивает доступ к ее ресурсам.

Функция файлового обмена выполняется запуском на обеих машинах программ KERMIT, выполняющих оригинальный протокол обмена канального уровня для передачи данных в виде последовательности кадров с подсчетом контрольной суммы кадра. Протокол обеспечивает передачу данных в виде пакетов с подтверждением и повторную передачу пакетов в случае обнаружения ошибок и помех на линии связи.

Локальный и удаленный режимы работы

Локальный режим работы позволяет распоряжаться всеми ресурсами локального компьютера пользователя. Ресурсы удаленного компьютера становятся доступными в **удаленном режиме работы**, в который управляющая программа переходит по команде CONNECT, переводящей локальный компьютер в режим виртуального терминала удаленной машины. При этом весь обмен между локальным терминалом и удаленной машиной «прозрачен»

для локальной машины, кроме специальной служебной последовательности выхода (ESC-последовательности), возвращающей локальный режим.

Для того чтобы определить в какой ОС (а следовательно, в локальном или удаленном режиме) находится пользователь, в программе KERMIT для каждой из реализаций задается по умолчанию свой промпт — строка приглашения ввода команды. Так, например, для системы PC DOS промпт Kermit-86), а для системы PT-11 промпт Kermit-11).

Можно организовать кросс-соединения, последовательно переходя от узла к узлу в удаленном режиме. Для удобства в каждой из систем помимо своего промпта нужно установить свою ESC-последовательность.

Режим SERVER

В расширенном варианте реализации протокола KERMIT удобен режим SERVER. От базового варианта, когда KERMIT запускается на обоих узлах и управляющие команды подаются пользователю с двух сторон, режим отличается тем, что все управление сеансом связи ведется со своего локального узла в локальном режиме (один запуск SERVER на удаленном компьютере). В этом режиме можно неоднократно передавать и принимать файлы без перехода в режим виртуального терминала и возвращения обратно. Кроме того, он позволяет получать каталог диска удаленной машины, стирать файлы, определять наличие свободного места на диске и т. п.

Стандартные типы связей

Протокол KERMIT обеспечивает двухточечное соединение машин. В то же время можно выделить три типа связи, характеризующихся рядом особенностей. Много-терминальную систему будем называть ХОСТ-системой [1].

ХОСТ-ПК — это традиционный тип связи, поддерживаемый протоколом. Со стороны многотерминальной ХОСТ-системы один из портов выделяется для связи с ПК, причем номер терминальной линии должен быть заранее определен для ОС ХОСТ (как правило, на этапе ее генерации). Со стороны ПК требуется только наличие связанного интерфейса.

ХОСТ-ХОСТ — в этом случае для каждой из систем нужно определить связанные терминальные линии. Организовать файловый обмен можно через промежуточный ПК, сведя этот тип связи к предыдущему, и непосредственно, но в этом случае требуется соблюдать дисциплину перехода из локального режима в удаленный и обратно для каждой из ХОСТ-систем.

ПК-ПК характеризуется тем, что принимает команды только со своей клавиатуры и не может иметь второго терминала. В этом случае программа KERMIT может использоваться для обмена файлами, однако работу приходится вести параллельно на двух машинах «вслепую».

Следует добавить, что пакет KERMIT поддерживает обмен по коммутируемым телефонным линиям связи, если имеются соответствующие телефонные модемы. Для защиты от потери информации при работе в режиме виртуального терминала (беспротокольном) можно использовать так называемый ноль-модем, т. е. линию связи с дополнительными сигналами подтверждения приема и готовности передачи (хотя для передачи данных достаточно и трехпроводной линии связи) или использовать стандартные средства ОС для работы с линиями связи (в PT-11, например, XL-хендлер).

Типы передаваемых данных

Базовая реализация протокола KERMIT предусматривает передачу только символьных файлов в коде ASCII. В расширенной реализации (например, для машин серии PDP 11) можно передавать двоичные файлы и автоматически преобразовывать двоичные данные в символьные (8 бит делится пополам и дополняется до двух символьных байтов). Обеспечивается автоматическое распозна-

вание типа файла по расширению имени файла. Для машин серии IBM 370 при передаче символьных файлов требуется предварительное преобразование кодов EBCDIC в код ASCII. Допускается программное управление контролем четности и скоростью обмена на линии связи.

Работа с файлами в системе KERMIT

Предметная область системы KERMIT — файлы данных. Полный формат имени файла в системе KERMIT имеет вид Filespec::=Node::Device:[Directory]Name.Type;Version, где поля имеют следующий смысл:

Node — имя узла в DECnet;
Device — физическое или логическое имя устройства;
[Directory] — пользовательский код UIC или номер PPN, определенный не во всех ОС;
Name.Type — собственно имя и тип файла;
;Version — номер версии, определенный в системах типа RSX.

Обычно используется сокращенный формат имени файла, в котором некоторые поля, определенные в текущей ОС, берутся по умолчанию. Кроме того, в именах файлов можно использовать групповые символы (например, *, %, ? в зависимости от системы).

Имена и дополнительные атрибуты файлов для различных ОС могут значительно отличаться, поэтому при обмене файлами происходит преобразование имен и атрибутов файлов, сопровождающееся сокращением длины имен файлов и потерей некоторых атрибутов. Определены действия в случае обнаружения в текущем каталоге принимающего устройства файла с тем же именем, которое передается.

Обмен в системе KERMIT управляется с помощью команд, список которых дан ниже. Среди них можно условно выделить пять групп команд: основные; команды в режиме SERVER; вспомогательные; команды, определяющие параметры линии и информирующие об установленных параметрах; команды ОС.

Команды системы файлового обмена KERMIT

a	Выполнить командный файл (синоним TAKE)
BYE	Выполнить выходную регистрацию удаленного SERVER
CONNECT	Связаться с удаленной системой (режим REMOTE)
COPY	Локальное копирование файла(ов)
CWD	Установить новый рабочий каталог
DELETE	Локальное удаление файла(ов)
DIRECT	Информировать о рабочем каталоге
DISCONNECT	Дать отбой на удаленной коммутируемой линии (синоним HANGUP)
DISPLAY	Внутренняя отладка
ERASE	Локальное удаление файла(ов)
EXIT	Выход из системы (синоним QUIT)
FINISH	Остановка удаленного SERVER без выходной регистрации
GET	Получить файл(ы) от удаленного SERVER
HANGUP	Дать отбой на удаленной коммутируемой линии (синоним DISCONNECT)
HOST	Выполнить системные команды локально
LOCAL	Вызвать интерпретацию команд на локальной системе
LOGFILE	Создать регистрационный файл
QUIT	Выход из системы (синоним EXIT)
PRINT	Локальная печать файла(ов)
RECEIVE	Принять файл(ы) от удаленного KERMIT
REMOTE	Префикс для команд управления файлами в SERVER
RENAME	Локальное переименование файла(ов)
SEND	Передать файл(ы) на удаленный KERMIT
SERVER	Запуск режима SERVER
SET	Изменить параметры KERMIT
SHOW	Информировать о параметрах KERMIT
TAKE	Выполнить командный файл (синоним команды)
TYPE	Локальный вывод файла на терминал
WHO	Локальная информация о регистрации пользователей (только для RSTS/E)

К основным относятся команды перехода из локального режима в удаленный (CONNECT) и обратно (ESC-последовательность), организации файлового обмена (SEND Filespec/RECEIVE Filespec). Переход в режим SERVER на ХОСТ-системе выполняется командой SERVER, а его выключение со стороны ПК — командой BYE (или FINISH). Для файлового обмена в этом режиме исполь-

зуются команды SEND Filespec/GET [Remote Filespec].

Вспомогательные команды необходимы для получения подсказки по команде HELP, запуска на выполнение косвенного командного файла (TAKE), выхода из программы (EXIT) и т. п.

Для определения линии используется команда SET с большим числом параметров. Все параметры линии, установленные данной командой или по умолчанию, можно увидеть с помощью команды SHOW. Команда HANGUP (или DISCONNECT) типична в работе с коммутируемой телефонной линией (отбой связи).

KERMIT позволяет выполнить по программе ряд команд монитора ОС (например: COPY, DELETE, DIREKTORI, PRINT, RENAME, TIPE) как на своем ПК, так и в ХОСТ-системе.

Для удобства работы с различными ОС в KERMIT имеются синонимы команд (например, TAKE — A, EXT — QUIT, DELETE — ERASE, DISCONNECT — HANGUP).

Основные характеристики протокола KERMIT

Протокол разработали и реализовали в Columbia University Center for Computer Activities (CUCCA) в 1981 г. программисты Bill Catching и Frank da Cruz [3] первоначально для машины DEC-20 и микрокомпьютера с CP/M80. Затем он был распространен практически на все массовые типы машин и ОС в университетах и колледжах США и Канады [2]. Название протокола является сокращением следующего выражения: KL-10 Eggrfree Reciprocal Microcomputer Interchange over TTY-lines (безошибочный двунаправленный для обмена между микрокомпьютерами по терминальным линиям).

В основе протокола KERMIT — сетевая модель ISO, рекомендации ANSI X3.57, X3.66 и идеи организации файлового обмена в сетях DECnet и ARPAnet. Протокол обеспечивает взаимную конфигурацию системы, согласующую параметры линии с обеих сторон; передачу пакетов данных с подтверждениями; обнаружение ошибок и помех в линии связи.

Для надежности передачи файлов по линии связанные компьютеры упаковывают данные в пакеты, содержащие контрольную информацию: маркер синхронизации, индикатор длины пакета, номер пакета для его восстановления в случае неисправности линии, 0..94 байт данных (в зависимости от типа пакета) и контрольную сумму. Все типы пакетов имеют одинаковый формат (см. рисунок) и различаются только содержанием поля данных. При потере на линии пакет передается повторно, дублированные пакеты отбрасываются. Обмен осуществляется в дуплексном и полудуплексном режимах связи. Пользователь определяет направление обмена, активная сторона которого — передатчик. Протокол поддерживает обмен данными по выделенным физическим и коммутируемым телефонным линиям связи. Все поля состоят из символов кода ASCII. Виды пакетов TYPE приведены ниже.

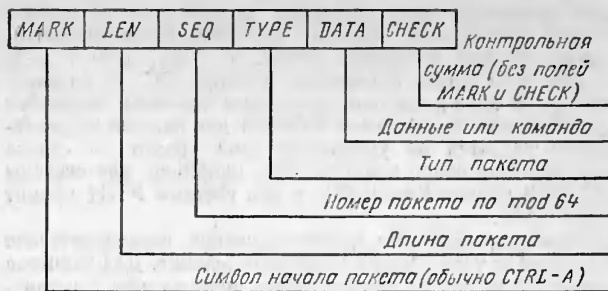
Базовый режим	Режим SERVER
D — пакет данных	A — атрибуты файла
Y — положительное подтверждение	R — инициализация приема
N — отрицательное подтверждение	C — системная команда
S — инициализация передачи	K — команда KERMIT
B — завершение транзакции	G — общие команды KERMIT с подкомандами
F — заголовок файла	I — инициализация обмена
Z — конец файла	
E — ошибка передачи	
T — зарезервирован	

Сеанс обмена пакетами, включающий передачу одного или более файлов в одном направлении, называется транзакцией. Последовательность передачи пакетов следующая:

передатчик посылает S-пакет, согласующий конфигурацию линии с двух сторон; приемник подтверждает передачу Y-пакетом;

передатчик посылает F-пакет — заголовок файла, содержащий в поле данных имя файла; приемник посылает Y-пакет с пустым полем данных или именем, под которым пересылаемый файл будет запомнен;

передатчик пересылает содержимое файла в D-пакетах;



Формат пакета KERMIT

каждый пакет положительно (Y—) или отрицательно (N—) подтверждается приемником;

когда передан весь файл, передатчик посылает Z-пакет (конец файла); если есть еще файлы для передачи, то процесс повторяется, начиная с передачи F-пакета; если все файлы переданы, то передатчик посылает B-пакет (конец транзакции).

При возникновении фатальной ошибки в процессе передачи посылается E-пакет, вызывающий аварийное завершение транзакции.

Сценарий сеанса связи

Пример типичного сценария сеанса связи приведен ниже для соединения типа ПК-ХОСТ. (в качестве ПК используется компьютер с ОС RT-11 версии 5.02, а в качестве ХОСТ-системы — с ОС TSX-Plus версии 6.01).

RT-HISJ V05.02	Приветственное сообщение ОС RT-11 на ПК, запуск программы KERMIT на ПК
.KERMIT	
Kermit-11>SET PROMPT PC)	Изменение промпта для наглядности
PC) SET LINE 2	Определение линии связи на ПК
PC) CONNECT	Переход в удаленный режим
<BK)	Символ активации ХОСТ
TSX-Plus version 6.01	Приветственное сообщение ОС TSX-Plus
Licence=***	Сообщение о лицензии
Logan please:LEONID	Входная регистрация пользователя на ХОСТ, пароль не отображается
Password:	
Welcome to the system	Приветствие системы TSX-Plus, запуск программы KERMIT с системного или текущего устройства ХОСТ
.KERMIT	
Kermit-11) SET PROMPT HOST)	Изменение промпта для наглядности
HOST)SET LINE 5	Определение линии связи на ХОСТ
HOST)SEND MYFILE.FOR	Передача файла MYFILE.FOR с ПК на ХОСТ
CTRL-/C	CTRL-/C — это ESC-последовательность, возврат в локальный режим
PC) RECEIVE	Принять на ПК файл
PC) CONNECT	Перейти в удаленный режим
HOST EXIT	Выход из программы KERMIT на ХОСТ
OFF	Выходная регистрация на ХОСТ
Connect-00:39:15 CPU-00:00:14.	Информация о продолжительности зарегистрированного сеанса
CTRL /C	Возврат в локальный режим
PC) EXIT	Выход из программы KERMIT на ПК, промпт системы RT-11 на ПК.

Перед началом сеанса связи пользователь должен знать номера связных портов. Для рассматриваемых ОС это можно сделать командой SHOW TERMINALS.

По-видимому, наиболее естественна инициатива по организации связи со стороны ПК. Поэтому после загрузки ОС на ПК запускается программа KERMIT; признаком ее готовности к приему команд является промпт Kermit-11). Затем с помощью косвенного командного файла или явного выполнения команд задаются параметры линии связи ПК. Для перехода в удаленный режим нужно выполнить команду CONNECT, по которой ПК становится виртуальным терминалом ХОСТ-машины. После этого пользователь ПК может зарегистрироваться для работы в ХОСТ-системе. Естественно, что получаемые в его распоряжение ресурсы лежат в пределах установленных привилегий. Для организации файлового обмена пользователь должен запустить на ХОСТ-системе программу KERMIT. В примере требуется передать файл с ХОСТ-машины на ПК. Поэтому пользователь выполняет на ХОСТ команду SAND с указанием имени переда-

СИСТЕМА ОБРАБОТКИ ТЕПЛОВИЗИОННЫХ ИЗОБРАЖЕНИЙ НА БАЗЕ ПЭВМ ЕС1840

Тепловизионные пирометрические системы (ТПС) контроля различных термических процессов получили широкое распространение в промышленности. ТПС — это измерительные системы, сочетающие функциональные возможности тепловизоров и пирометров излучения [1]. Для получения количественной и качественной информации о температурном распределении визируемого участка поверхности ТПС, как правило, содержат специализированные компьютерные системы обработки тепловизионных изображений.

В состав ТПС входят следующие основные блоки и устройства (рис. 1): тепловизионная камера (ТБК); анализатор тепловых изображений (АТИ);

цветной графический дисплей; управляющая ЭВМ с периферийными блоками (печатающим устройством, дисплеем, дисководом, графопостроителем и т. д.); комплект градуировочных и поверочных средств (модель черного тела (МЧТ), образцовая термопара и т. д.), которые совместно с ЭВМ образуют систему автоматизированной градуировки (САГ).

Основные технические характеристики ТПС серии ИИ-Т с датчиками для различных температурных диапазонов приведены в таблице. Система обработки тепловизионных изображений ТПС ИИ-42ТМ (рис. 2) построена на базе ПЭВМ ЕС1840 с программным обеспечением термографической обработки ПТО-1. Результаты обработки тепловизионной информации в этой системе отображаются на экране графического дисплея в виде черно-белых или цветных изображений вместе с количественной информацией о температурном распределении поверхности контролируемого объекта (рис. 3).

В программном обеспечении ПТО-1 можно выделить пять основных групп программ разного назначения:

- вычисления температуры;
- графического преобразования изображений;
- пространственно-частотной обработки (фильтрации) изображений;
- ввода-вывода для работы с ВУ;
- работы системы автоматизированной градуировки при подключении калибровочных средств.

Режим работы с изображениями выбирается из набора меню (рис. 4) с помощью клавиш управления курсором или функциональных клавиш F1...F10. Режим работы с изображениями, записанными на диске, задается с помощью меню работы с файлами.

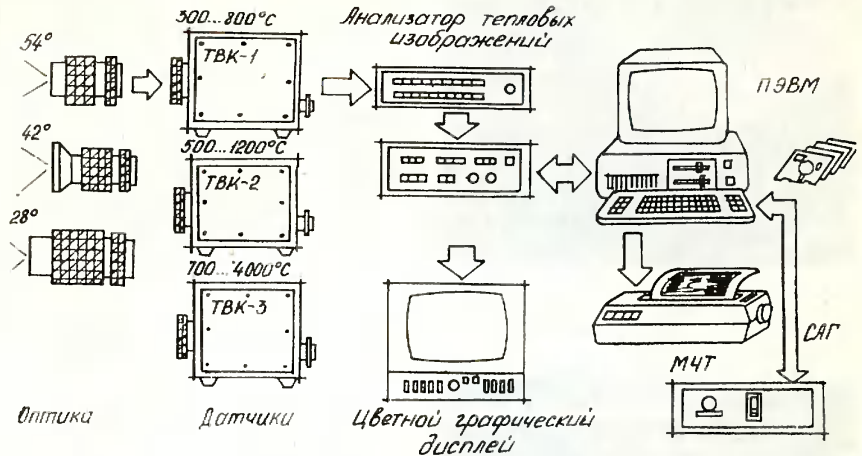


Рис. 1. Состав ТПС ИИ-42ТМ

Программы измерения температуры обеспечивают расчет значений температуры по уровням видеосигнала с учетом градуировочных характеристик (ГХ), полученных во время градуировки ТПС и загруженных предварительно в память ПЭВМ с диска. ГХ представляются в табличной форме. В основе измерений и расчетов истинной температуры лежит методика приведения системы к монохроматической и специальное математическое обеспечение [2].

Программы обработки изображений позволяют получать термопрофили, псевдорельефные, изометрические и псевдообъемные изображения тепловых полей, временные зависимости температуры в задан-

ной точке, хронограммы выбранного столбца (или строки), масштабирований, изменять положение и совмещать различные изображения.

Принцип построения псевдорельефных изображений основан на модуляции яркости изображения в соответствии с пространственным распределением градиентов. Это придает рельефность картине, позволяя наглядно представить перепады температур (рис. 5).

К особо информативным видам геометрических преобразований тепловых распределений относятся трехмерные изображения — изометрическое и псевдообъемное, основанные на представлении уровня сигнала (температуры) в виде третьей (вер-



Рис. 2. Тепловизионная пирометрическая система ИИ-42ТМ

Технические характеристики ТПС серии ИИ-Т

Характеристика	Датчик		
	ТВК-1	ТВК-2	ТВК-3
Детектор ИК-излучения	Виднокон		
Область спектральной чувствительности, мкм	0,8...2,5	0,5...1,8	0,45...1,0
Диапазон контролируемых температур, °С	300...800	500...1200	700...4000
Оптика	Пять сменных объективов		
угол зрения, град	16	28	42
фокусное расстояние, мм	250	150	90
апертурная диафрагма, ступеней	12	12	12
Параметры чересстрочной развертки	50	50	50
частота полей, Гц	625	625	625
число строк в кадре	150	210	300
Число разрешаемых элементов в строке, не менее	3	3	5
Число поддиапазонов	0,8 при Тобъекта=400	1,0 при Тобъекта=700	0,8 при Тобъекта=1000
Температурная чувствительность, °С (максимальная)	±5	±0,5	±5
Приведенная погрешность измерения, %	±2,0	±3,0	±7
Температура (аппаратурная), °С, не более	ИИ-42Т	ИИ-42ТМ	ИИ-52Т
Система обработки и отображения	«Электроника ДЗ-28»	ЕС1840 (ЕС1841, РС ХТ/АТ)	ДВКЗМ2 (ДВК4, СМ)
Тип управляющей ЭВМ	ТИО	ТИО-1	ТИО-2
Тип изображения	Черно-белое и цветное с дискретными уровнями градации яркости и цвета: 4, 8, 16 и 256		
Формат изображения в кадровом ОЗУ, элементов	128×128	128×128	512×512
Число блоков кадрового ОЗУ	3	3	4
Программное обеспечение	ТИО	ТИО-1	ТИО-2

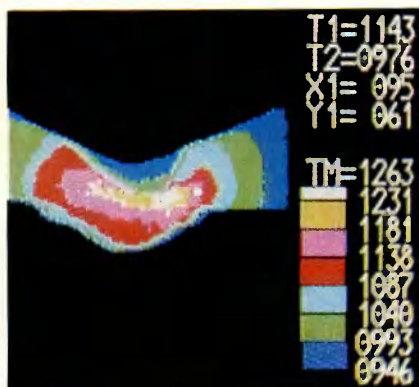


Рис. 3. Тепловое изображение нагретой при испытании до плавления металлической пластины (режим измерения температуры)

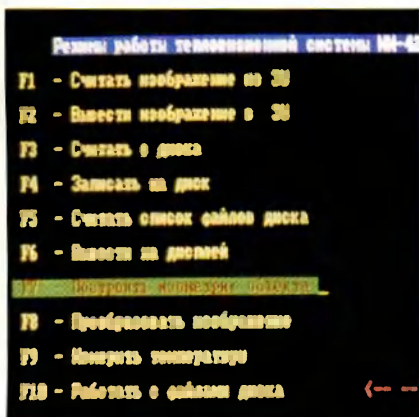


Рис. 4. Одно из четырех меню выбора режима работы



Рис. 5. Псевдорельефное изображение

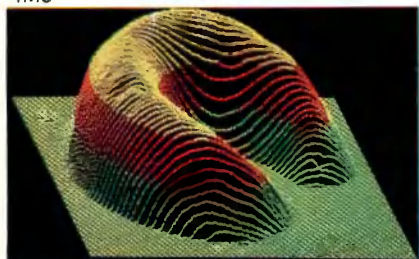


Рис. 6. Изометрическое трехмерное изображение

тикальной) пространственной координаты. Изометрическое изображение образуется из совокупности смещенных определенным образом горизонтальных термопрофилей (рис. 6).

Псевдообъемное изображение с непрерывной поверхностью получается при заполнении путем интерполяции пустых мест между термопрофилями (рис. 7). Уровень сигнала

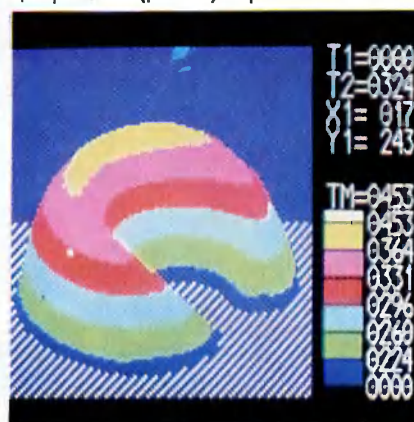


Рис. 7. Псевдообъемное изображение непрерывной поверхностью

в точках поверхности изометрического и псевдообъемного изображений устанавливается в соответствии с его распределением в основном изображении, что позволяет использовать при его анализе возможности цветового кодирования. Ввод в ЭВМ преобразуемого изображения и вывод результирующего трехмерного в одно из кадровых ЗУ анализатора осуществляется столбцами. В программе построения трехмерных изображений предусмотрена возможность получать разрезы строящейся фигуры различными плоскостями (рис. 8).



Рис. 8. Разрезанное трехмерное изображение

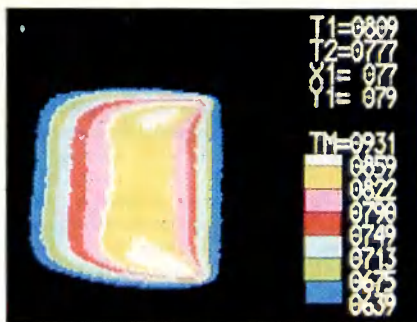


Рис. 9. Тепловое изображение полупроводниковой пластины в процессе обработки электронным пучком, подвергаемое хронографированию

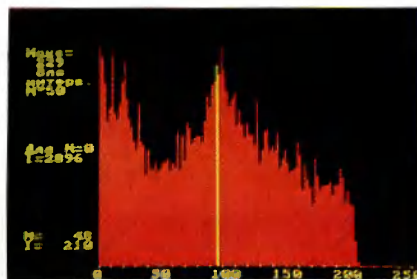
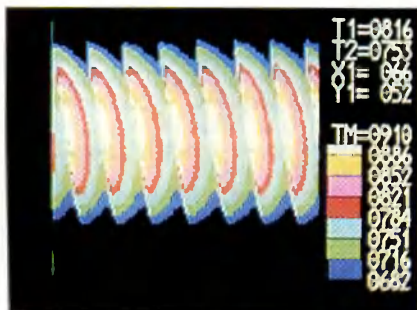


Рис. 10. Исходное изображение (а) и гистограмма распределения интенсивности его точек (б)

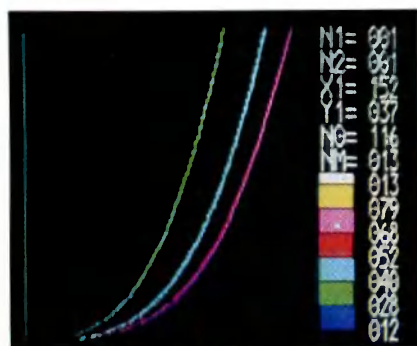


Рис. 11. Набор температурных градуировочных характеристик

На практике часто требуется исследовать характер изменения температурного распределения во времени. Для этих целей в ТПС ИИ-42ТМ имеется несколько возможностей: запись изображений в кадровые ЗУ и на диск накопления и вывода на экран дисплея временных зависимостей сигнала в маркере, получения и обработки хронограмм заданного столбца или строки. Хронограмма представляет собой фиксированное двумерное изображение развернутого по одной из пространственных координат процесса временного изменения выбранного сечения теплового поля (рис. 9).

Для статистического анализа тепловых изображений программные средства ИИ-42ТМ обеспечивают построение гистограммы распределения температур (рис. 10).

Дополнительные возможности при исследовании тепловых изображений дает программа преобразования интенсивности каждой точки изображения: функция преобразования может быть либо стандартной (линейной, степенной, логарифмической), либо произвольно заданной в режиме графического редактора. Преобразовав изображение по заданному закону, можно избавиться от фона, выделить на изображении изотермические зоны и линии, скорректировать световую характеристику приемника излучения, получить негативное изображение, повысить контрастность.

ТПС ИИ-42ТМ с программным обеспечением ПТО-1 позволяет осуществлять пространственно-частотную фильтрацию тепловых изображений, коррекцию пространственно-частотной характеристики оптико-электронного тракта с помощью обратного преобразования Фурье. Программа пространственно-частотной фильтрации повышает четкость изображения, выделяет контуры, увеличивает отношение сигнал-шум. Для очистки изображения от одиночных помех используется программа медианной фильтрации.

Градуировка любой пирометрической системы состоит из множества различных операций: съема и фиксирования показаний нескольких прибором, статистической обработки получаемых данных, составления градуировочных таблиц, слежения за температурой образцового излучателя и ее регулирование и т. д.

С помощью САГ градуировка проводится в динамическом режиме, т. е. при постепенном, медленном, но непрерывном увеличении (или уменьшении) температуры в полости МЧТ. Градуировка заканчивается при заполнении всех точек ГХ, соответствующих всем уровням цифрового видеосигнала. Предусмотрено усреднение ГХ, полученных при увеличении и уменьшении температуры. Все ГХ записываются на магнитный диск для хранения и отображаются на экране дисплея (рис. 11).

Предложенная методика проведения градуировки в динамическом режиме, воплощенная в аппаратных и программных средствах САГ, позволяет снимать одну ГХ по всем точкам с погрешностью, не превышающей значение порога температурной чувствительности системы, в течение 15—20 мин.

220600, Минск, ул. П. Бровки, 6 Минский радиотехнический институт; тел. 39-88-62

ЛИТЕРАТУРА

1. Тепловизионная пирометрическая система / Гайдукевич Ю. Ч. и др. // Электронная промышленность. — 1987. — С. 59—62.

2. Математическое описание выходного видеосигнала тепловизионной пирометрической системы / Домаренок Н. И. и др. // Современные экспериментальные методы исследования процессов тепло- и массообмена. Материалы Международной школы — семинара. — Минск. — ИГМО АН БССР. — 1987. — Ч. 2. — С. 33—40.

Статья поступила 21.02.89

В. Г. Алешин, Л. А. Белов, Д. Г. Бухаров В. М. Демидов, М. Ю. Фомичев, И. Л. Королев

СИНТЕЗАТОР СЛОЖНЫХ СИГНАЛОВ

Основу синтезатора сложных электрических сигналов составляет аналого-цифровой функциональный генератор, построенный на МПК БИС серии КР580.

Генератор (рис. 1) включает автоколебательный блок АК, блок жесткой логики БЛ, узел частотомера Ч и микропроцессорную систему МС с клавиатурой Кл и индикатором И (рис. 2). Для повышения рабочей частоты синтезатора применяются быстродействующие элементы жесткой логики, коммутируемые сигналами от МС.

Прибор работает в трех основных режимах: базовом (БР), частотной модуляции (ЧМ) и сигналов произвольной формы (СПФ). В БР блок

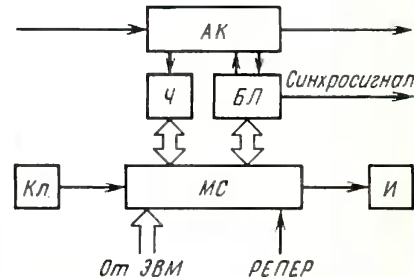


Рис. 1. Структурная схема генератора

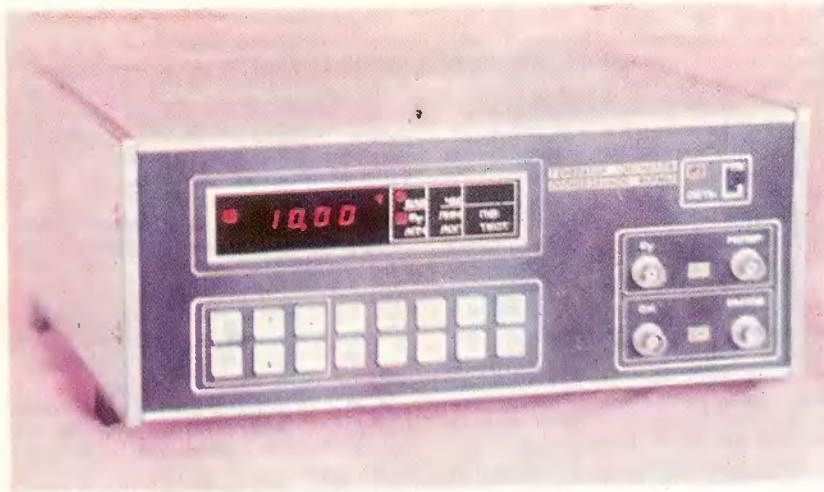


Рис. 2. Генератор сигналов произвольной формы

АК образует типовую схему функционального генератора [1], которая обеспечивает периодические выходные сигналы одной из трех типовых форм (синусоидальной, треугольной или прямоугольной) и возможность управления частотой в широких пределах за счет подачи напряжения на вход аналогового интегратора. В приборе предусмотрен ввод кодов формы, ослабления выходного сигнала (до -40 дБ относительно максимальной амплитуды 10 В на нагрузке 50 Ом), а также его частоты (в пределах двух декад от $10,00$ до $999,9$ кГц).

Встроенный автоматический частотомер Ч на основе БИС таймера К580ВИ53 с помощью МС измеряет [2] фактическое значение частоты повторения и выводит его на индикатор в виде четырех десятичных разрядов и порядка. По команде от клавиатуры сигнал частотомера может быть подан на вход управления частотой: включается система автоподстройки частоты и синтезируется любой из 10^1 дискретов в пределах двух декад. Нестабильность средней частоты определяется тактовым генератором МС и не превышает $\pm 5 \cdot 10^{-4}$.

В режиме ЧМ блоки АК и БЛ образуют схему управляемого по частоте автогенератора с кольцом дополнительной обратной связи для формирования выходного напряжения с когерентной ЧМ. Обеспечивается генерация сигнала с линейной, логарифмической или псевдосумовой когерентной ЧМ без разрыва фазы по введенным значениям нижней и верхней частот. Возможна коррекция нелинейности модуляционной характеристики автогенератора.

Появление внешнего сигнала РЕПЕР вызывает прерывание в МС, подпрограмма обработки которого прекращает перестройку частоты. Это дает дополнительную сервисную возможность автоматического поиска

частоты, вызывающей приход сигнала РЕПЕР.

В режиме СПФ по введенным отсчетам формируется сигнал в виде сплайнов первого порядка с заданной длительностью, скважностью и амплитудой. Для этого в типовой схеме функционального генератора использован интегратор с кодовой установкой крутизны и релейный элемент с кодовым управлением пологом срабатывания. Частота повторения сигнала в режиме СПФ устанавливается в диапазоне 1 Гц... 500 кГц.

В приборе предусмотрен, кроме того, режим самоконтроля ТЕСТ, в котором при помощи собственного частотомера прсверяется сохранение допустимой погрешности частоты повторения в нескольких точках всех рабочих режимов. Успешное выполнение процедур контроля сопровождается звуковым сигналом, индицирующим также попытку ввода данных, не реализуемых в данном генераторе.

Программное обеспечение занимает в ПЗУ 6 Кбайт (три БИС типа К537РФ5); оперативная память имеет объем 1 Кбайт, сохраняя все параметры режимов, введенные оператором.

Прибор потребляет от сети переменного тока частотой 50 Гц и напряжением 220 В мощностью 40 Вт, имеет массу 8 кг и габаритные размеры $130 \times 310 \times 330$ мм.

105835, Москва, Е-250, Красноказарменная, 14 МЭИ, кафедра радиопередающих устройств; тел. 362-77-95

ЛИТЕРАТУРА

1. Ноткин Л. Р. Функциональные генераторы и их применение. М., Радио и связь, 1983.
2. Алешин В. Г., Белов Л. А., Бухаров Д. Г., Демидов В. М., Фомичев М. Ю. Широкодиапазонный автоматический частотомер. Микропроцессорные средства и системы. 1988, № 6.

КОДИРОВЩИК-II

Центральное конструкторское бюро с опытным производством (ЦКБ с ОП) АН БССР разработан и изготовлен опытный образец устройства для автоматизированного считывания графической информации (Кодировщик-II).

Технический уровень изделия соответствует мировым образцам и подтверждается картой уровня.

Устройство обладает патентной чистотой по СССР, ГДР, ВНР, НРБ, ПНР, ЧССР, Кубе, СФРЮ, МНР. На Лейпцигской весенней ярмарке 1989 г. отмечено золотой медалью.

Устройство защищено авторским свидетельством по заявке № 4326752/24 от 28.11.88. Указатель координат устройства защищен свидетельством на промышленный образец № 26126.

Устройство предназначено для автоматического преобразования в цифровую форму кодов координат точек изображения, указываемых оператором, и предварительной обработки считанной информации.

Область применения

— оперативная обработка больших объемов графической информации сложных форм (графиков, схем, диаграмм, рентгенограмм и т. п.) в составе автоматизированного рабочего места на базе персональной ЭВМ ЕС1840 (ЕС1841).

Достоинства

- простота реализации;
- высокие метрологические показатели;
- эргономичность, что достигнуто благодаря удачной дизайнерской проработке, а также возможности визуальной обратной связи при помощи микроабло указателя координат;
- возможность использования амплитудно-дифференциального метода измерения координат;
- малая энергоемкость.

Основные технические данные

- формат поля считывания, мм 340×435 ;
- разрешающая способность, мм $0,025$;
- максимальная аппаратурная погрешность, мм $0,1$;
- быстродействие, точек/с 125 ;
- указатель визир с индикаторным микроабло;
- габаритные размеры, мм $615 \times 517 \times 85$;
- масса, кг 8 ;
- потребляемая мощность, Вт до 30 .

Конструкторскую документацию можете получить по договору на передачу научно-технической продукции с оказанием услуг: сопровождение изготовления, участие в наладке, испытаниях и внедрении. Имеется возможность изготовления изделий.

Адрес для справок и запросов 220600, Минск, ГСП, ул. Логойский тракт, 20, тел. 65-17-60, 65-17-09

ваемого файла, затем через ESC-последовательность возвращается в локальный режим и выполняет команду RECEIVE. Процесс передачи пакетов сопровождается индикацией передающегося файла, текущего номера пакета и числа повторов в случае ошибочной передачи пакета. По завершении передачи файла выдается соответствующее сообщение.

Ввод системы в эксплуатацию (установка)

Система KERMIT распространяется в виде дистрибутивного набора на магнитной ленте. Для установки программ KERMIT на конкретных компьютерах проще всего перенести программное обеспечение на общие внешние носители информации. Если это невозможно, то предлагается процедура установки с использованием терминальных линий связи. Для этого достаточно хотя бы на одной из машин иметь полный дистрибутивный набор. Простые программы на языках BASIC или FORTRAN-IV, приведенные в руководствах по установке для конкретных ОС, позволяют принять от полной дистрибутивной системы минимальное ядро, а затем через него и все необходимые файлы конкретного дистрибутивного набора.

Распространение дистрибутивного набора

Широкая популярность и большое число реализаций программ KERMIT (более 100), безусловно, вызваны свободным распространением, не связанным с платой за лицензию. Однако CUCCA имеет зарегистрированное право копирования (Copyright) на протокол, документацию и реализации, выражающееся в том, что при распространении KERMIT требуется сохранить название системы и подпись о праве копирования, которую можно перевести так:

«Copyright (C) 1985. Опекун — Колумбийский университет в сити Нью-Йорка. Любому пользователю и учреждению гарантируется разрешение использовать, копировать или передавать данное программное обеспечение без цели наживы при условии сохранения этой подписи о праве копирования».

Тем не менее, предусматривается получать плату за распространение KERMIT, хотя она в основном покрывает накладные расходы. В плату за распространение полного дистрибутивного набора входит стоимость магнитной ленты с программами и документацией, печатные руководства пользователя и описание протокола, внутренние транспортные расходы.

Установка системы KERMIT или создание новых программ и документации производится за дополнительную плату. Адрес Колумбийского университета, распространяющего KERMIT:

KERMIT Distribution, Columbia University Center for Computing Activities 7th Floor, Watson Laboratory, 612 West 115th Street New York, N. Y. 10025

Пример организации межмашинной связи

В ЦКБ уникального приборостроения НТО АН СССР система KERMIT применяется с 1984 г. Первоначально для связи использовались выделенные физические линии связи. Однако в последнее время вместо них применяется универсальная магистраль типа коммутируемой токовой петли (КТП-магистраль) [4]. Она состоит из многожильного кабеля (в котором поддерживаются сигналы в стандарте активной токовой петли), коммутаторов (по одному на ЭВМ), согласователей, преобразующих сигналы последовательных асинхронных интерфейсов различных машин в сигналы на шине, и источника питания.

КТП-магистраль удобна и проста в использовании. Пользовательский компьютер коммутируется с корреспондентом вручную. При этом сигнальная индикация предупреждает о возможных конфликтах (использование занятой линии) и отражает процесс обмена информацией. На КТП-магистрали обеспечивается гальваническая развязка (которой может и не быть на терминальном интер-

фейсе) для защиты сети от влияния электромагнитных помех и уравнивающих токов. КТП-магистраль по стоимости прокладки дешевле физических линий связи между компьютерами, но полностью обеспечивает все их возможности. Кроме того, она гарантирует кросс-соединения для системы KERMIT, реализация которых может потребовать много времени при непосредственном согласовании различных типов машинных терминальных интерфейсов. КТП-магистраль может быть использована и для других систем связи по терминальным линиям, например VTCOM. Дополнительную информацию по вопросам приобретения и использования КТП-магистрали можно получить по адресу: 117342, Москва, ул. Бутлерова, 15. ЦК УП НТО АН СССР.

ЛИТЕРАТУРА

1. Бойченко Е. В., Кальфа В., Овчинников В. В. Локальные вычислительные сети. — М.: Радио и связь, 1985.
2. Frank da Cruz KERMIT Users Guide, Sixth Edition, Revision 1, CUCCA, New York 10027, 13 June 1985.
3. Frank da Cruz, KERMIT Protocol Manual, Fifth Edition, CUCCA, New York 10027, 3 April 1984.
4. Брусилковский Л. И., Суворов В. А. Применение КТП-магистрали для системы файлового межмашинного обмена по терминальным линиям связи KERMIT // Тех. докл. на XXII школе по автоматизации научных исследований. — Тольятти, 1988.

Статья поступила 25.05.88

УДК 681.3

А. А. Гусев, Л. С. Озерсц

ДРАЙВЕР ДЛЯ ОБМЕНА ДАННЫМИ МЕЖДУ ОС ЕС И RT-11

Одним из эффективных методов ускорения обработки и хранения больших объемов накопленной информации — стандартизация форматов хранения данных [1], позволяющих унифицировать программное обеспечение (ПО) и облегчить каталогизацию данных. В связи с этим возникает задача совместимости единого формата с различными типами ЭВМ (в частности, ЕС, «Электроника», ДБК и т. д.) [2].

На ЭВМ ЕС выполняется предварительная обработка*. Результаты обработки записываются на магнитную ленту в стандартизованном формате хранения. Когда алгоритм последующей обработки, связанной с анализом и интерпретацией данных, можно сформировать однозначно, она также выполняется на ЕС. Однако в большинстве случаев необходим интерактивный режим, для которого более удобны персональные компьютеры.

Для обеспечения возможности работы на микроЭВМ с данными, записанными в ОС ЕС на магнитной ленте со стандартными метками (SL), создан драйвер ЕС.SYS, позволяющий читать магнитную ленту с помощью стандартных операторов языков высокого уровня, что существенно для программистов-непрофессионалов. Драйвер реализует обмен однородными массивами данных: символами или числами во внутреннем представлении ЭВМ (2-байтовые целые или 4-байтовые действительные). Форма представления данных указывается командой SET (табл. 1) Длина блока записи в ОС ЕС (параметр BLKSIZE) не должен превышать 2048 байт.

* Для экспериментов на борту спутника, проводимых в НИИФ МГУ, такая обработка состоит в преобразовании телеметрического кода в значение измеряемой физической величины (скорость счета, энерговыделение и т. д.), координатной привязке и расчете геофизических параметров в точке наблюдения.

Параметры команды SET драйвера EC.SYS

REAL	Данные считываются с ленты как действительные числа (REAL*4) во внутреннем представлении
INT	Данные считываются с ленты или записываются на ленту как целые 2-байтовые числа (INTEGER*2) во внутреннем представлении
CHAR	Данные считываются с ленты или записываются на ленту в коде ДКОИ-8
COMMA=nn	nn — восьмеричный код символа, разделяющего данные во внутреннем представлении при чтении с ленты
SIZE=n	n — размер блока при записи на ленту из RT-11. По умолчанию n=2048
LRECL=n	n — длина записи в байтах при чтении с ленты (после каждого n байт вводятся символы CR и LF)

Таблица 2

Примеры записи данных в фортране ОС ЕС и чтение в RT-11 (при символьном выводе как разделитель используется CR и LF)

Запись в OCEC	Чтение в RT-11	
	BASIC	FORTRAN
INTEGER*2 J(1024) DO 2 I=1, 1024 2 J(I)=I WRITE(11,1)J 1 FORMAT (8(128A2))	1 DIM J(1024) 2 (1) 3 FOR I=1 TO 1024 4 INPUT#1,J(I) 5 NEXT I	DIMENSION J(1024) (2) DO 2 I=1, 1024 2 READ(1,2)J(I) 1 FORMAT(17)
REAL*4 A(512) DO 2 I=1,512 2 A(I)=I WRITE(12,1)A 1 FORMAT (4(128(A4)))	1 DIM A(512) 2 (1) 2 FOR I=1 TO 512 4 INPUT#1,A(I) 5 NEXT I	DIMENSION A(512) (2) DO 2 I=1,512 2 READ(1,3) A(I)
DATA A,B,K/1.2 & 0.3E21,14/ DATA CRLF/3338/ WRITE(12,1) A, B, K & ,CRLF 1 FORMAT (F3.1,., & E7.1,.,12,A2)	1 (1) 2 INPUT#1,A,B, K	(2) READ(1,4) A, B, K 4 FORMAT (F3.1,IX, & E7.1,IX,12)

- (1) OPEN 'ECO:INP.DAT' FOR OUTPUT AS FILE #1
(2) OPEN (UNI T=1, NAME=' ECO:INP.DAT',
TYPE='OLD',
ACCESS=' SEQUENTIAL', FORM='FORMATTED',
ERR=100)

Данные, записанные во внутреннем представлении, переводятся драйвером в символьное представление (КОИ-7). Последовательности символов отделяются друг от

Пример записи в RT-11 и чтения в OCEC

1 A=1.2/B=1.2E34/K=56 2 OPEN 'EC1:D.D' FOR OUTPUT AS FILE #1 3 PRINM #1,USING #111.1.1.1/111.1.1.1>A,B,K	DATA A,B,K/1. &2.1.2E34/ (1) WRITE &(1.2)A,B,K 2 FORMAT &(F5.1,E8.1,13)	READ &(11.1)A,B,K 1 FORMAT &(F5.1,E8.1,13) (2)
1 DIM J(1024) 2 OPEN 'EC1:D.D> FOR OUTPUT AS FILE #1 3 FOR I=1 TO 1024/J(I)=I/NEXT I 4 FOR I=1 TO 1024/PRINT#1,J(I)/NEXT I	INTEGER*2 &J(1024) (1) WRITE(11,1)J 1 FORMAT &(8(128A2))	INTEGER*2 &J(1024) DO 2 I=1,1024 2 READ(11,1) J 1 FORMAT &(128((8A2)/)) (2)
(1) OPEN (UNI T=1,NAME='EC1:D.D',TYPE='NEW',FORM='FOR & MATTED',ACCESS+' SEQUENTIAL',ERR=100) (2) // GO.FTI1001 DD SIS +OLD.UNI T=T,LABEL(1,SL,IN). // DSN=D,D,VOL=SER=RT1:OCEC,DCB=(RECFM=FB, LRECL=16,BLKSIZE=204		

друга дополнительным символом, устанавливаемым командой SET. Данные, записанные в форматном виде, конвертируются из ДКОИ-8 в КОИ-7. Это относится только к алфавитно-цифровым и специальным символам. Остальные коды передаются без изменений, что позволяет записывать в ОС ЕС управляющие символы КОИ-7 для получения формата RT-11 (табл. 2).

Лента просматривается непосредственно из монитора RT-11 команд TYPE, PRINT, COPY, DUMP. Тексты из библиотек ОС ЕС переносятся в файлы RT-11 утилитами IEBGENER. Поскольку в скопированном таким образом тексте отсутствуют управляющие символы, необходимо его редактирование, состоящее в введении символов CR и LF после каждых 80 символов текста. Каталог ленты получается с помощью программы ECDIR, читающей стандартные метки и распечатывающей содержание аналогично соответствующей процедуре ОС ЕС.

Драйвер обеспечивает возможность записи данных в формате OCEC во внутреннем представлении (целые 2-байтовые числа) и форматном виде. При этом пишутся необходимые стандартные метки начала (HDR) и конца (EOF) файла. Размер формируемого блока устанавливается командой SET и не должен превышать 2048 байт. Инициализация ленты выполняется программой ECINIT.

При записи в форматном виде алфавитно-цифровые и специальные символы КОИ-7 конвертируются в символы ДКОИ-8. Управляющие символы КОИ-7 заменяются символами пробела, которые необходимо учитывать при программировании вывода (табл. 3).

Тексты, переносимые из RT-11 в ОС ЕС, нуждаются в предварительном редактировании, состоящем в замене кода символа TAB необходимым числом пробелов, дополнением каждой строки символами пробела до полной длины 80 символов, включая CR и LF. Редактирование выполняется программой ECEDIT, после которой файл можно скопировать на магнитную ленту драйвером EC.SYS и записать в библиотеку ОС ЕС той же утилитой IEBGENER. Объем драйвера — 17 блоков.

Телефон 939-18-28, Москва

ЛИТЕРАТУРА

- Mazza C. Standard Formatted Data Units Proc. Workshop on Geophys. Informatic. Moscow. Aug. 14—18, 1988.
- Abston E. Compatibility Problems: Computer-to-Computer communications and Software Transportability. Proc. Workshop on Geophys. Informatic. Moscow. Aug. 14—18, 1988.

Статья поступила 9.09.88

В. Я. Беркуцкий, И. В. Поливаный

ДИАЛоговая ИНСТРУМЕНТАЛЬНО-УЧЕБная СИСТЕМА ПРОГРАММИРОВАНИЯ

Для выработки элементарных навыков работы с ПЭВМ необходима разработка предметно-развивающих учебных компьютерных сред, наибольшую популярность среди которых в настоящее время получила система программирования «Школьница» [1], а также различные версии систем на базе языка Лого [2, 3], созданного С. Папертом в Массачусетском технологическом институте США.

Вместе с тем, не теряет своей актуальности задача создания такой учебной среды, которая «не должна требовать от ребенка сложных знаковых переозначений, большой глубины планирования, запоминания длинных последовательностей шагов и т. п.» [4], и позволяет решать достаточно сложные, не привязанные к какой-либо одной школьной дисциплине и имеющие интегрированный гуманитарный характер задачи [4—6].

В Павлодарском педагогическом институте разработана диалоговая инструментально-учебная система программирования (ДИУС), состоящая из следующих основных функциональных модулей:

МОНИТОР управляет работой системы;

ПРОГРАММИРУЕМЫЙ ЗНАКОГЕНЕРАТОР содержит представление булевых матриц всех символов входного алфавита;

ИНТЕРПРЕТАТОР (таблично-управляемый диспетчер) отвечает за обработку команд входного языка;

РЕДАКТОР вводит и корректирует текст программы;

ПОДСКАЗЧИК поясняет синтаксис входного языка, диагностирует ошибки и сбойные ситуации, возникающие в процессе эксплуатации системы;

МОДУЛЬ СВЯЗИ С ДОС отвечает на связь с принтером, НГМД;

ДИУС позволяет изучать базовые алгоритмические конструкции (команда, ветвление, циклы и т. п.) на гуманитарной основе, т. е. без привлечения математического аппарата. Эта возможность обеспечивается использованием статической и динамической графики, музыкального вывода, литературных выражений. Работа с графикой организована так, что от пользователя не требуется предварительных знаний о декартовой системе координат. Для получения сложных изображений и графической динамики не нужны вычислительные операции. Простая и более естественная, чем в БЕЙСИКе, система кодировки музыкальных знаков также не требует долгого предварительного освоения. Средства работы с литературными величинами вполне традиционны: склейки, вырезки и т. п. Заметим, что музыкальные и графические характеристики входного языка системы ДИУС (простота кодировки, быстрота исполнения, набор предоставляемых возможностей) заметно превосходят аналогичные характеристики языка BASIC APPLESOFT. Преподаватель имеет возможность либо выбрать наиболее приемлемую для него основу подачи материала (музыка, графика, текст), либо использовать сразу весь спектр элементарных операций.

Входной язык системы ДИУС можно расширять. Преподаватель получает возможность создавать новые подпрограммы и объявлять их частью языка. При этом вызов подпрограмм синтаксически не отличается от обращения к встроенным командам. Таким путем преподаватель приспособливает язык к своим нуждам, подчиняя его собственной методической схеме. Используя расширение, можно создавать новые графические примитивы*

вы, новые языки графического вывода, реализовывать графомызыкальные, графотекстовые, текстомызыкальные команды и т. п. Так, в рамках ДИУС достаточно просто моделируется язык программирования Лого, графические средства и элементарные операции БЕЙСИКА, различные исполнители и т. п.

ДИУС позволяет организовывать диалог с пользователем на основе лексики родного языка. Опыт показывает, что язык программирования, построенный на такой основе, усваивается учащимися гораздо быстрее. В настоящее время реализована версия ДИУС на базе русской лексики, однако с помощью генератора системы достаточно просто могут создаваться версии на основе лексики любого другого языка, если литеры алфавита этого языка представимы точечными матрицами 8×8.

Система ДИУС поддерживает дружественный интерфейс, обеспечивая подсказку неопытному пользователю на различных этапах его работы, но в то же время позволяет более опытному пользователю обходиться без подсказки. ДИУС, являясь учебной системой, большое внимание уделяет диагностике ошибок, выдавая ее в развернутом виде. Например, вместо сообщения типа СИНТАКСИЧЕСКАЯ ОШИБКА система выдает более конкретное сообщение: В КОМАНДЕ ОТСУТСТВУЕТ ЗАПЯТАЯ. Помимо диагностики ошибок и сбойных ситуаций система в любой момент времени по запросу пользователя напоминает ему правила записи некоторых команд языка.

Система ДИУС работает в трех режимах: интерактивном, программном, инструментальном. Последовательное их использование в процессе обучения методически обусловлено. На первом этапе в интерактивном режиме (т. е. в режиме немедленного исполнения одной вводной команды) обучаемые осваивают элементарные операции: работу с графикой, музыкой, множественным присваиванием, выводом алфавитно-цифровой информации. Уже на этом этапе школьники получают возможность решать далеко не элементарные задачи. В процессе работы в интерактивном режиме постепенно назревает потребность в долговременном хранении вводимых команд и поэтому переход к программному режиму не вызывает затруднений. В режиме программирования осваиваются более сложные алгоритмические структуры и управляющие конструкции: ветвление, цикл ПОВТОР, цикл ПОКА, работа с массивами и т. п. По мере усложнения программ назревает потребность в использовании подпрограмм, что и обеспечивает инструментальный режим.

Все директивные команды ДОС и команды редактирования ДИУС закреплены за соответствующими функциональными клавишами и выполняются немедленно по их нажатию, что заметно упрощает изучение указанных команд, а их дальнейшее использование позволяет избежать путаницы, которая возникает, например, при первом знакомстве с системой программирования БЕЙСИК. Таким образом, решая свои задачи в рамках ДИУС, учащийся концентрирует внимание главным образом на тексте своей программы.

Входной язык ДИУС содержит основные конструкции программирования большинства современных языков: массивы, циклы, ветвление, подпрограммы и т. п. Таким образом, знание входного языка ДИУС обеспечивает успешное освоение многих современных языков программирования. Блочная структура входного языка, графосинтаксические конструкции, средства комментирования позволяют писать программы с хорошо обозримой управляющей структурой. Язык обладает сравнительно небольшим набором команд (четыре элементарные операции и пять управляющих конструкций), однако набор стандартных функций в нем такой же, как и в языке БЕЙСИК. Язык позволяет в полном объеме работать с символической информацией, получать сложные графические изображения, эффективно работать с динамической графикой, вводить и выводить алфавитно-цифровую информацию, осуществлять музыкальный вывод.

Настоящая версия системы ДИУС разработана для ПЭВМ АГАТ и написана на языке ассемблера 8-разрядного микропроцессора 6502, который обладает достаточно

традиционной для 8-разрядных процессоров архитектуры, поэтому можно говорить об адаптации системы ДИУС к другим ПЭВМ, используемым в сфере образования.

637002, Павлодар, ул. Мира 60, Павлодарский пединститут; тел. 75-29-95.

ЛИТЕРАТУРА

1. Звенигородский Г. А., Глаголева Н. Г., Земцов П. А., Налимов Е. В., Цикоза В. А. Программная система «Школьница» и ее реализация на персональных ЭВМ // Микропроцессорные средства и системы. — 1984. — № 1.
2. MSX-LOGO Reference manual. Logo Computer System Inc., 1985. — 142 p.
3. Williams G. Logo for the Apple II, the TI-99/4A and

the TRS-80 Color Computer // Byte. — 1982. — Vol. 7, — № 8. — P. 163—193.

4. Научная программа психолого-педагогических исследований по проблеме «Новые технологии обучения и развития детей» // Информатика и образование. — 1989. — № 4. — С. 3—6.
5. Ершов А. П., Звенигородский Г. А., Первин Ю. А. Школьная информатика (концепции, состояние, перспективы). — Новосибирск, 1979. — 52 с. — (Препринт / АН СССР. Сиб. отделение. Вычисл. центр; № 152).
6. Иванов А. Г., Карпова А. В. Перспективы применения языка Лого в обучении. — М.: Наука, 1987. — С. 80—88.

Статья поступила 31.08.89

УДК. 681.53.072

В. Н. Мурогов

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ НА БАЗЕ Е-СЕТЕЙ ДЛЯ ПЭВМ

В разработанной системе имитационного моделирования за основу взят аппарат Е-сетей [1, 2], выгодно отличающийся от сетей Петри.

Отличия Е-сети: входить и выходить из позиции может не более одной дуги; существует несколько типов позиций, которые могут содержать не более одной фишки; очереди имеют произвольное число фишек; фишки обладают набором атрибутов (в данной системе это вектор целых чисел типа long);

с каждым переходом Е-сети (отрезок прямой линии) можно связать временную задержку, процедуру преобразования атрибутов проходящих через него фишек и разрешающую процедуру, если переход имеет входную разрешающую позицию.

По выразительным средствам Е-сеть, как и сеть Петри, имеет структуру ориентированного графа, у которого вершины одного типа образуются позициями, а другого — переходами. Дуги могут связывать только переход с позицией или позицией с переходом.

Следует отметить, что для усиления выразительных средств Е-сетей снято ограничение на общий вид процедур, разрешающих преобразование атрибутов и вычисления временных задержек для перехода. В созданной системе эти процедуры могут иметь абсолютно произвольный вид и реализовывать любые вспомогательные вычисления и действия, необходимые пользователю.

В качестве базового набора типов элементарных сетей был выбран набор Г. Натта и добавлены две элементарные сети — генератор и поглотитель фишек.

Процесс работы Е-сети. В результате срабатывания переходов фишки перемещаются из входных позиций переходов в выходные. Для этого на каждом

шаге модельного времени определяются переходы, готовые сработать, и выполняются следующие действия: проверка возможности активизации перехода и условия завершения фазы его активности; разрешающая процедура; определение длительности активной фазы перехода с помощью процедуры вычисления задержки; процедура преобразования атрибутов.

Система имитационного моделирования EVA (E-net Valuation Adviser) включает в себя:

- тестовый редактор;
- графический редактор EEE для задания структуры;

специальный язык EDEF, разработанный на базе языка Си для описания процесса функционирования Е-сети; компилятор PREPARE языка EDEF, генерирующий модельную программу на языке Си с помощью определений, связанных редакторами EEE и EDIT; графический отладчик GRDEB; компилятор TCC и редактор связей TLINK языка Turbo C; вспомогательные компоненты (пакетные файлы, с помощью которых вызываются программы PREPARE, TCC и TLINK указанной модели и создается ее файл).

Объем всех программных компонент системы EVA занимает 500 Кбайт дисковой памяти. Процесс создания модели заключается в задании двух исходных файлов описания модели и их последовательного преобразования до получения исполняемого файла модели системы (см. рисунок).

Функции графического редактора EEE: создание и удаление позиций переходов и дуг; маркирование позиций; просмотр создаваемой сети (перемещение и масштабирование); загрузка ранее созданных сетей из файлов библиотеки моделей, их стыковка; хранение полученной сети в файле; синтаксический контроль правильности соединения позиций и переходов; вывод на матричное печатающее устройство или графопроектор изображение создаваемой сети.

Программа на языке EDEF — набор разделов, который может встречаться не более одного раза и является фрагментом программы на языке Си. Переменные можно разделить на пользовательские (для целей исследователя) и системные (определяющие состояние и параметры имитационной модели). Основные системные переменные следующие:

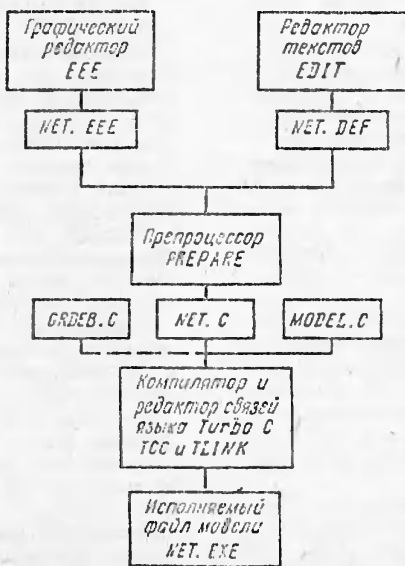


Схема процесса создания модели и взаимосвязей между файлами

finish — модельное время, при котором прекращается имитация;
 clock — текущее модельное время только для чтения;
 g — значение разрешающей позиции текущего перехода;
 d — значение временной задержки текущего перехода;
 S[i].atr [j] — j-й атрибут i-й простой позиции.

Разделы программы предназначены для описания переменных и функций пользователя (A:); задания начального состояния модели (I:); определения значений разрешающих позиций (P:); вычисления временных задержек срабатывания переходов (D:); преобразования атрибутов фишек при срабатывании переходов (T:); указания элементов E-сети, для которых проводится сбор статистики (S:); описания процедуры, которая выполняется по окончании процесса моделирования (E:).

Описание E-сети на языке EDEF создается с помощью любого редактора текстов. Если модель имеет имя NET, то файл должен иметь имя NET.DEF. В отличие от существующих систем имитационного моделирования на базе E-сетей язык EDEF позволяет во всех разделах использовать функции на языке Си любой сложности. Это дает возможность «обрамления» процесса имитации программами пользователя: создания собственного интерфейса между моделью и исследователем и точек прерывания моделирования по достижении определенных состояний, динамического изменения параметров модели.

После создания файлов структуры E-сети и ее описания пользователь с помощью пакетного файла UNION.BAT, вызывающего компилятор PREPARE, объединяет их в файл NET.C программы на языке Си. Компилятор PREPARE синтаксически контролирует структуру E-сети.

После компиляции файлов NET.C и MODEL.C создается исполняемый файл NET.EXE. Файл NET.EXE отладочного варианта позволяет получить модель с помощью графического отладчика GRDEB. При отладке модели в графическом режиме можно просматривать модель или отдельные участки в зависимости от выбираемого масштаба графического изображения E-сети и ее расположения на экране.

Функции отладчика GRDEB следующие: трассировка перемещения фишек по E-сети в течение одного очередного такта работы модели (единицы модельного времени); прогон модели с текущего момента модельного времени до указанного времени; прогон модели до ближайшего момента модельного времени срабатывания хотя бы одного перехода системы; выдача промежуточных статистических данных на экран; выбор нового расположения E-сети на экране и ее масштаба.

Москва, ул. Кропоткинская, 13/7,
 ИФТП АН СССР; тел. 248-82-07

ЛИТЕРАТУРА

1. Nutt G. J. Evaluation Nets for Computer System Performance Analysis.— AFIPS FJCC.— 1972.— Vol. 41, pt. 1.— P. 279—286.
2. Костин А. Е., Шаньгин В. Ф. Организация и обработка структур данных в вычислительных системах.— М.: Высшая школа, 1987.

Статья поступила 29.11.88

ОПЕРАЦИОННЫЕ СИСТЕМЫ

УДК 681.3.06

В. Р. Нафиков, М. И. Рабинович

ЯДРО МОДУЛЯ 2 ОРИЕНТИРОВАННОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ ДЛЯ ВСТРАИВАЕМЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Операционные системы (ОС) для встраиваемых микропроцессорных систем (МПС) ориентированы, в той или иной степени, на микропроцессоры конкретной архитектуры, а порой и конкретного типа. Значительно большую гибкость обеспечивает создание ОС, ориентированных не на аппаратную среду, а на среду языка программирования. Разработанное ядро распределенной ОС под названием «Вторник» входит в состав программного обеспечения МПС управления и диагностики карьерных экскаваторов объединения «Ижорский завод». Ее отличает высокая степень переносимости: для переноса ядра требуется изменить не более 20 тщательно прокомментированных строк. Ядро является самостоятельным программным изделием и может использоваться во встраиваемых системах различного назначения с одним центральным процессором (ЦП).

Структура ядра

С точки зрения проблемного программиста, ядро «Вторник» представляет собой библиотеку пассивных процедур с упрощенными до предела средствами контроля. В отличие от большинства подобных систем примитивы, реализуемые процедурами ядра, оперируют с абстракциями, отражающими прагматический (а не теоретический) подход к структурированию ПО МПС.

Три абстракции являются основными в ядре «Вторник»: процессы, ресурсы и события. Каждой из них соот-

ветствуют свои типы данных (записи и указатели) и примитивы (процедуры), использующие эти типы данных. Отдельному процессу, ресурсу или событию ставятся в соответствие конструкции-дескрипторы. Процесс представляет собой программу пользователя, снабженную средствами для выполнения в ядре. Ресурс означает нечто, требующее доступа в режиме «взаимного исключения»: данные, устройства, а также любые другие компоненты МПС. Событие — это такое происшествие в МПС, о возникновении которого следует известить пользовательские процессы. События используются для синхронизации процессов.

Управление процессами

В ядре «Вторник» процесс представлен тремя конструкциями: дескриптором, телом и рабочей областью.

Дескриптор процесса — это структура, описывающая характеристики процесса и содержащая указатель на его контекст при сопроцессорных передачах управления. Конкретный процесс идентифицируется адресом своего дескриптора. Дескриптор размещается в пользовательской программе или создается средствами «Вторник».

Тело процесса — это процедура, реализующая алгоритм процесса. Процесс завершается выполнением примитива «Уничтожить процесс». Одно и то же тело может использоваться несколькими процессами — копиями, что обеспечивается повторной входимостью процедур в языке Модула 2.

Рабочая область — это стек для размещения переменных, контекста процесса при сопрограммных передачах управления, временного хранения данных. Каждый процесс имеет свою рабочую область, размер которой зависит, в частности, от реализации компилятора.

Процесс может находиться в одном из трех состояний: активности, готовности и блокировки. В каждый момент вре-

мени только один процесс является активным, т. е. выполняется на ЦП. В состоянии готовности находятся процессы, стоящие в очереди к ЦП. Процесс, находящийся в состоянии блокировки, не выполняется и не претендует на доступ к ЦП. Причины блокировки процесса следующие: вызов соответствующей процедуры или ожидание освобождения ресурса, аппаратного прерывания, объявления события, завершения отсчета промежутка времени. В очереди к ЦП готовые процессы упорядочены по приоритету, называемому здесь статусом процесса и определяющему права на ресурсы. Активный процесс может перейти в состояние блокировки, уступив ЦП. При этом активным становится процесс, имеющий наивысший статус. Если этим статусом обладают несколько процессов, то в ЦП передается тот, который стоит первым в очереди. Для равномерного распределения процессорного времени между процессами с одинаковым статусом используется круговая диспетчеризация, реализуемая через квантование. Несомненно, что этот механизм присущ скорее системам с разделением времени, чем системам реального времени, к которым относятся «Вторник». Задачам реального времени в большей степени соответствуют детерминированные методы планирования выполнения процессов, такие как составление расписаний, что требует существенных временных затрат.

Изначально в системе существуют стартовый процесс и процессы драйверов внешних устройств, к которым нет доступа со стороны пользовательских программ. При запуске системы стартовый процесс имеет наибольший статус. Затем статус стартового процесса понижается до наименьшего, и если перед этим создан хотя бы один пользовательский процесс, то ему и будет передано управление.

Для управления процессами в ядре «Вторник» предусмотрен набор примитивов в виде процедур языка Модуля 2, достаточный для успешного функционирования развитой встраиваемой МПС. **Основные примитивы:**

СОЗДАТЬ ПРОЦЕСС. Заполняется дескриптор процесса и связывается с его телом и рабочей областью. Процесс переводится в состояние готовности.

УНИЧТОЖИТЬ ПРОЦЕСС. Процесс блокируется. Освобождается память, занятая под его дескриптор и рабочую область.

ПРИОСТАНОВИТЬ ПРОЦЕСС. Происходит блокировка процесса. Процесс может блокировать сам себя.

ВОЗОБНОВИТЬ ПРОЦЕСС. Процесс переходит в состояние готовности. С помощью программ обработки события блокированный процесс может перевести себя в состояние готовности.

ИЗМЕНИТЬ СТАТУС ПРОЦЕССА. Вместе со статусом изменяется и положение процесса в очереди к ЦП.

Число процессов, которые можно создать с помощью ядра, ограничивается лишь размером динамической памяти (кучи). Итак, средствами ядра «Вторник» МПС может быть синтезирована в виде множества процессов.

Управление ресурсами

Основная функция любой ОС — управление ресурсами. Имеются в виду ресурсы вычислительной установки и самой ОС. Управление различными ресурсами пользовательских программ обычно возлагается на пользователя. При мультипрограммном управлении ресурсами разных типов возникает одна и та же проблема — организация «взаимного исключения» при доступе к ресурсам. Во многих ОС она решается путем организации доступа к ресурсам в особом системном процессе, не прерываемом пользовательскими процессами.

В ядре «Вторник» для доступа к системным и пользовательским ресурсам применен механизм семафоров со счетчиком. Ядро позволяет выполнить следующие операции над ресурсами:

СОЗДАТЬ РЕСУРС. Ресурс идентифицируется своим дескриптором. Перед использованием ресурса необходимо создать и инициализировать его дескриптор.

ЗАХВАТИТЬ РЕСУРС. Механизм захвата ресурса предназначен для программиста, т. е. вне зависимости от состоя-

ний ресурса управление возвращается в тело процесса только после того, как ресурс захвачен. На самом деле процесс может захватить либо свободный ресурс, либо ресурс, уже захваченный этим процессом. Если один процесс пытается занять ресурс другого, то он блокируется и ставится в очередь к этому ресурсу. Очередь упорядочивается по статусу процессов. Каждый захват ресурса увеличивает на единицу содержимое счетчика доступа.

ОСВОБОДИТЬ РЕСУРС С УМЕНЬШЕНИЕМ СЧЕТЧИКА ДОСТУПА. Освободить ресурс может только его владелец. При этом счетчик уменьшается на единицу. Если счетчик обнуляется, то ресурс считается свободным, а владельцем становится процесс, стоящий первым в очереди.

ОСВОБОДИТЬ РЕСУРС СО СБРОСОМ СЧЕТЧИКА ДОСТУПА. Операция аналогична предыдущей, но содержимое счетчика не уменьшается, а сбрасывается.

ПРОВЕРИТЬ СОСТОЯНИЕ РЕСУРСА И ЗАХВАТИТЬ ЕГО, ЕСЛИ ОН СВОБОДЕН. Если время ожидания ресурса критично, можно предварительно проверить его состояние. Операции проверки и захвата объединяются в одну непрерывную.

Два системных ресурса ядра — ЦП и куча — недоступны пользователю непосредственно (только через механизм доступа к ресурсам). В операции инициализации дескриптора ресурса программист может указать адрес и интервал процедуры передиспетчеризации. Ядро будет вызывать эту процедуру с указанной периодичностью в контексте процесса, являющегося драйвером таймера. С помощью этого механизма пользователь может синтезировать свой алгоритм перераспределения ресурса.

Управление событиями

Событие идентифицируется дескриптором, который должен быть создан и инициализирован. Процессы могут объявлять события и обрабатывать их. Если процесс предполагает обрабатывать событие, он должен сообщить об этом ядру «Вторник». Такая операция называется заказом на обработку события. При этом необходимо указать так называемую процедуру-обработчик события (ПОС). Заказ на обработку запоминается и хранится независимо от того, сколько раз событие будет объявлено, до тех пор, пока процесс не откажется от него. ПОС вызывается при каждом объявлении события и выполняется с наивысшим статусом в контексте процесса, заказавшего обработку. Операция объявления события имеет две формы: для определенного процесса или для всех процессов, заказавших обработку.

Ядро «Вторник» позволяет обрабатывать так называемые разовые события, когда допускается только один заказ на обработку, и ядро уничтожает дескриптор после первого объявления этого события. Ядро позволяет процессам выполнять следующие действия над событиями:

СОЗДАТЬ СОБЫТИЕ. Создается дескриптор события.

УНИЧТОЖИТЬ СОБЫТИЕ. Дескриптор события уничтожается.

ЗАКАЗАТЬ ОБРАБОТКУ СОБЫТИЯ. Указывается ПОС и код запроса.

ОТКАЗАТЬСЯ ОТ ОБРАБОТКИ СОБЫТИЯ. Уничтожается первый из запросов от данного процесса.

ЗАПРЕТИТЬ РАСПОЗНАВАНИЕ СОБЫТИЙ. При объявлении события, обработка которого заказана данным процессом, выполнение ПОС задерживается, но не отменяется.

РАЗРЕШИТЬ РАСПОЗНАВАНИЕ СОБЫТИЙ. Все задержанные попытки выполнить ПОС успешно выполняются.

ЖДАТЬ ОБЪЯВЛЕНИЯ СОБЫТИЯ. Разрешается распознавание событий. Если задержанных ПОС нет, то процесс блокируется.

ОБЪЯВИТЬ СОБЫТИЕ. Можно передать данные процессам, обрабатывающим это событие. Таким образом происходит обмен сообщениями между процессами.

Последние три операции предназначены для организации диспетчеров — процессов, обрабатывающих мно-

гие события, возникающие асинхронно. Запрет распознавания событий необходим для предотвращения потери информации о возникновении следующего события. Схема построения диспетчера приведена на рис. 1. Ядро «Вторник» создает пять дескрипторов для предопределения событий. Четыре из них объявляются системными процессами, представляющими собой драйверы внутрипроцессорных прерываний в следующих ситуациях: сбой источника питания, срабатывание аппаратуры защиты памяти, переполнение в операциях с плавающей запятой, исчезновение порядка в операциях с плавающей запятой. Пятое событие объявляется при обнаружении недопустимого параметра в вызове процедур ядра. Пользовательские процессы могут заказать обработку этих событий.

Дополнительные средства

Процедуры отсчета времени — традиционные процедуры установки и получения системного времени, задержки выполнения процесса. Кроме того, имеется процедура, позволяющая отсчитывать временной промежуток асинхронно. Выполнение процесса при этом не прекращается, а по завершении отсчета времени драйвер таймера вызывает пользовательскую процедуру, которая, например, может объявить событие.

Процедуры обработки прерываний предназначены для написания драйверов. Одна из них позволяет определить, какой процесс выполнялся в момент прерывания.

Процедуры организации стартового процесса. В функции стартового процесса (рис. 2) входит инициализация глобальных переменных модулей и создание хотя бы одного пользовательского процесса.

```
(* ДИСПЕТЧЕР: обработка n событий, возникающих      *)
(* асинхронно.                                         *)
PROCEDURE DISPATCHER; (* Тело процесса-диспетчера. *)
BEGIN
  (* Предполагается, что дескриптор собы- *)
  ... (* тия уже создан.                       *)
  (* Далее следуют заказы на обработку *)
  (* событий:                               *)
  (* EVENTn - указатели на дескрипторы *)
  (* событий.                               *)
  (* CODEn - коды, присваиваемые заказам *)
  (* EVENTPROC - системная процедура *)
  (* обработки событий.                   *)
  LINKTOEVENT(EVENT1,CODE1,EVENTPROC);
  LINKTOEVENT(EVENT2,CODE2,EVENTPROC);
  ...
  LINKTOEVENT(EVENTn,CODEn,EVENTPROC);
  ...
  LOOP (* Начало цикла по обработке событий. *)
  (* Процедура WAITFOREVENT ожидает объявления *)
  (* какого-либо события, помещает переданное *)
  (* сообщение в буфер и возвращает код запроса. *)
  CASE WAITFOREVENT(BUFFER,SIZE(BUFFER)) OF
    CODE1: (* Произошло 1-ое событие. *)
      ... (* Обработка 1-го события. *)
    ; CODE2: (* Произошло 2-ое событие. *)
      ... (* Обработка 2-го события. *)
    ;
    ...
    ; CODEn: (* Произошло n-ое событие. *)
      ... (* Обработка n-го события. *)
  END (* CASE *)
END (* LOOP *)
END (* DISPATCHER *);
```

Рис. 1. Схема построения диспетчера событий

```
MODULE TEST;
(* Пример оформления стартового процесса *)
(* *)
FROM SYSTEM IMPORT ADDR;
(* Импортируется статус процесса. Это пере- *)
(* численный тип. В примере используется зна- *)
(* чение STBACKGROUND. *)
FROM DEFINESYSTEM IMPORT PROCESSTATUS;
(* Импортируются процедуры ядра: *)
FROM NUCLEAR IMPORT
  CREATEPROCESS, PROCESSPOINTER, STARTSYSTEM;
(* Импортируется пустая процедура: *)
FROM TIMER IMPORT IDLE;

(* Тело пользовательского процесса: *)
PROCEDURE UserBody;
BEGIN
  ...
END UserBody;

(* Указатели на дескрипторы процессов. *)
VAR User1,User2: PROCESSPOINTER;

(* Тело данного модуля является концом тела *)
(* стартового процесса. *)
BEGIN
  (* Создается две копии пользовательского *)
  (* процесса: *)
  User1:=CREATEPROCESS(UserBody,STBACKGROUND,1000);
  User2:=CREATEPROCESS(UserBody,STBACKGROUND,1000);
  (* Запуск пользовательских процессов: *)
  STARTSYSTEM();
  (* Цикл стартового процесса: *)
  LOOP
    IDLE();
  END;
END TEST.
```

Рис. 2. Оформление тела стартового процесса

Процедуры завершения системы, определяемые программистом, могут быть использованы, например, для сохранения содержимого промежуточных буферов в драйверах или вывода сообщения оператору.

Обработка ошибок

Контроль допустимости параметров при обращении к ядру необходим только при разработке ПО МПС, а на стадии эксплуатации он не нужен, поэтому в ядре «Вторник» используется асинхронная обработка ошибок. При обращении к ядру с недопустимым параметром объявляется событие, дескриптор которого экспортируется из ядра. При отладке пользовательские программы могут предусматривать обработку этого события.

Реализация ядра «Вторник»

Первая версия ядра реализована для МП серии K1801 в ОС RSX-11M средствами системы программирования Модуль 2/PB. Это позволило даже этап комплексной отладки выполнить в среде RSX-11M, а собственно МПС использовать только на этапе испытаний. Ниже приведены характеристики модулей ядра «Вторник» для МП серии K1801. Данная версия ядра содержит кроме драйвера таймера драйверы внутрипроцессорных прерываний по недопустимому адресу, аварии источника питания, а также драйверы сопроцессора плавающей запятой.

Характеристики модулей ядра «Вторник» для МП серии K1801

Число модулей	3
Размер модулей, строк определений	518
реализации	1180
Размер реализационно-зависимой части, строк	10

Вторая версия ядра «Вторник» работает в ОС MS DOS (IBM/PC) без драйверов внутрипроцессорных прерываний.

Телефон 481-84-77, Ленинград

1. Вирт Н. Программирование на языке Модуля 2.— М.: Мир, 1987.— 224 с.
2. Дейтел Г. Введение в операционные системы.— М.: Мир, 1987.— 359 с.
3. Bondy J. Uninitialized Modula 2 Abstract Object Instances, Yet Again // SIGPLAN Notices.— 1987, Vol 22, No 5.— P. 58—63.

Статья поступила 28.02.89

УДК 681.3.06

М. А. Бульзиков, Н. Н. Дудоров

ИНТЕГРИРУЮЩАЯ СРЕДА НЕПОСРЕДСТВЕННОГО ВЗАИМОДЕЙСТВИЯ В СИСТЕМЕ UNIX

В настоящее время большое внимание уделяется организации удобного и эффективного пользовательского интерфейса с ПЭВМ [1]. Однако активность ПЭВМ в диалоге с пользователем приводит к многократному увеличению информации, выводимой на терминал. И если для ПЭВМ, имеющих прямой доступ к экрану, это ограничение не критично, то для многопользовательских ЭВМ, работающих в режиме разделения времени и имеющих удаленные терминалы, оно становится камнем преткновения. Видимо, поэтому экранные пользовательские интерфейсы остаются привилегией ПЭВМ.

В данной работе мы хотели бы поделиться опытом создания интерфейса работы с системой UNIX [2].

Коммуниканты и их взаимодействие

Один из основных принципов, лежащих в основе ОС UNIX — любая программа делает только то, для чего она непосредственно предназначена, но делает это хорошо [2]. Например, вряд ли естественно относить вывод текста на печатающее устройство к основным функциям редактора текстов. Однако этот принцип вступает в противоречие с необходимостью возможностью пользователя выбирать себе ту операционную среду, которая наилучшим образом соответствует его текущим требованиям. Если действие пользователя выходит за пределы возможностей данной системы, то он должен обратиться к некоторой другой, сменить текущую обстановку. Штатные средства позволяют либо вызвать новую систему (как процедуру), либо полностью на нее переключиться. При этом, однако, во вновь вызванной системе приходится начинать с нуля, и текущая обстановка может быть потеряна. Ситуация усугубляется тем, что инициализация системы (включая ее загрузку в память ЭВМ) — довольно трудоемкое действие, что чаще приводит к вынужденному нежеланию смены обстановки или нарушению вышеуказанного принципа.

Для преодоления этой трудности было введено и реализовано понятие коммуникантов. Центральное место в системе коммуникантов занимает базовый исполнитель, вокруг которого формируется совокупность систем-коммуникантов. Базовый исполнитель формирования и передачи команд на исполнение ОС поддерживает создание и взаимодействие коммуникантов. При первом вызове некоторого коммуниканта базовый исполнитель запоминает его имя, которое связывает с номером процесса, сопоставленного этому коммуниканту ОС. Отличительная черта коммуникантов — они не завершаются до тех пор, пока существует базовый исполнитель или сами не будут уничтожены насильственно (например, командой kill). Таким образом, выход из любого коммуниканта — это не полное окончание его работы, а временное переключение на другой коммуникант; повторный вызов — возобновление его работы в том месте, где он был покинут. При этом, естественно, сохра-

няется вся обстановка, наработанная в коммуниканте с момента его первого вызова. Для единообразия базовый исполнитель рассматривается как один из коммуникантов. Переключение коммуникантов производится командой *compt имя* — *коммуниканта сообщение*, которую можно передать базовому исполнителю любым из коммуникантов. Частный случай *имени* — *коммуниканта* — точка (.), означающая коммуникант, от которого было передано управление к текущему коммуниканту. Это позволяет имитировать обычный вызов (типа вызова процедуры). Сообщение в команде *compt* может быть использовано вызываемым коммуникантом для определения своего дальнейшего поведения. С точки зрения каждого из коммуникантов базовый исполнитель — сопрограма, выполняющая команды ОС, включая теперь и переключение к другим коммуникантам. Хотя реально все сообщения проходят через базовый исполнитель, с точки зрения пользователя, происходит непосредственное переключение (возможно, с передачей сообщения) от одного коммуниканта к другому.

Помимо базового исполнителя *Scr* реализованы текстовый редактор *Bed*, графический редактор *Vec* и система интеграции объектов *Doc*. Коммуниканты предоставляют свои экранные интерфейсы работы с объектами ОС. Кроме этого, разработаны некомунирующие программы, предоставляющие экранный интерфейс при работе с одним объектом. Выбор способа реализации программы в виде коммуниканта должен основываться на сложности и необходимости создания локальной обстановки программы.

Состав коммуникантов можно расширить. Планируется, например, создание системы управления процессом разработки программ. При этом не требуется вносить какие-либо изменения в уже имеющиеся системы. Это позволяет говорить о системе коммуникаций как об *интегрирующей среде*. Подобного рода средства есть на ПЭВМ. Наша цель — создать в рамках штатной многопользовательской ОС средства комплексации систем в дополнение к имеющимся средствам комплексации программ.

Экранный интерфейс

Экранные интерфейсы предназначены для отображения текущего состояния некоторой программы на экране дисплея таким образом, чтобы любое действие пользователя вызывало изменение изображения на экране. А так как большинство программ, реализующих экранный интерфейс, работают в режиме непосредственного ввода, т. е. ЭВМ сразу воспринимает нажатие любой клавиши, то должна быть обеспечена реакция на любое нажатие. Однако для удаленного терминала именно передача сообщения является наиболее дорогостоящей операцией. Поэтому при разработке экранного интерфейса требуется тщательно анализировать способы реакции, минимизируя объем сообщений. Здесь возникает противоречие с аппаратной независимостью программного продукта. К сожалению, возможности различных типов терминалов слишком разные и переносимость экранного интерфейса на них вынуждает использовать лишь те, которые имеются в любом терминале (например, описанные в файле *termcap*). Такая универсальность приводит к тому, что теряются возможности конкретного терминала, выходящие за рамки минимальных и позволяющие увеличить не только наглядность выводимой информации, но и ее компактность. Поэтому представляется правильным сосредоточить усилия на переносимости той части экранного интерфейса, которая определяет, что надо выводить. Ту часть, которая определяет, как надо выводить, приходится разрабатывать для каждого терминала заново.

В основном, при изменении способа отображения выводимой информации нельзя обойтись «косметической» адаптацией программы. Тем не менее, некоторые приемы сохраняются для большинства терминалов. Например, каждый интерфейс должен иметь заголовок, постоянно присутствующий на экране и дающий пользователю, по крайней мере, информацию о системе, в которой он работает. Чаще всего редактирующие команды терминала изменяют

часть экрана правее и ниже текущей позиции курсора. Поэтому размещение заголовка в верхней части экрана позволяет во многих случаях избежать его перевывода, сократив тем самым объем передаваемой информации.

Следует позаботиться и о том, чтобы способ появления изображения не раздражал пользователя. Большинство терминалов способны стереть весь экран или его часть. Однако мгновенное опустошение сильно заполненного экрана действует на пользователя шокирующе. Поэтому замена информации на экране выводом ее поверх старой кажется нам предпочтительнее. Аналогичным «раздражителем» является мелькание информации (особенно служебной), например, индикация номера строки и (или) столбца в текстовых редакторах. Использование подобных «раздражителей» оправдано лишь в особых случаях, когда требуется повышенное внимание пользователя. Перечень рекомендаций можно без труда продолжить, но даже применение наиболее простых из них позволяет существенно сократить время реакции диалоговых программ и, следовательно, повысить эффективность работы.

Базовый исполнитель Scg

Базовый исполнитель реализован как модификация штатного интерпретатора командного языка Cshell [3] в двух направлениях. Во-первых, выполнены коммуникации, о которых говорилось выше и, во-вторых, — экранный интерфейс, заменивший обычный телетайпный режим работы. Были успешно опробованы и другие альтернативы реализации базового исполнителя: например, за основу взят интерпретатор командного языка Shell, а экранный интерфейс выполнен в виде отдельного коммуниканта.

Scg позволяет работать в двух режимах: каталога и команд. В первом случае пользователь видит перед собой текущий каталог и может выбирать, перемещая курсор, тот или иной файл этого каталога. Во втором случае пользователь может манипулировать совокупностью введенных им ранее команд. Наличие первого режима говорит об определенной *объектно-ориентированности*, так как позволяет применять операции к файлам, которые являются основными обрабатываемыми объектами, не набирая их имен.

Реализации первоначальных версий Scg были основаны на чистом принципе объективной ориентации. Однако оказалось, что работа в таком режиме весьма неестественна. Это относится в основном к глобальным командам, не имеющим аргумента (например, команда `date` печати текущего времени). Неудобства возникают также и в случае, когда обрабатываемый объект находится вне текущего каталога. Поэтому была введена возможность совмещения объектно-ориентированного и *командного* диалогов. Scg позволяет ввести любую команду Cshell, отредактировать ее и передать на исполнение. При этом пользователю не приходится набирать вручную имена файлов.

Основная часть вывода Scg приходится на изображение каталога. Уменьшение объема этой части возможно, во-первых, за счет уменьшения числа файлов в одном каталоге (например, разбиением на подкаталоги) и, во-вторых, за счет сокрытия служебных или вторичных файлов, к которым относятся объектные модули, копии и т. п. Для пользователя стимулом к реализации первой возможности является задержка при выводе большого каталога и затруднительность ориентации в нем. Вторая возможность реализуется автоматически, если поддерживаются либо атрибуты, либо соглашения об именах файлов, позволяющих определить их вторичность (например, имена с суффиксом «.b»). Естественно, что пользователь может перейти в режим «полного» каталога, если ему желательно видеть все файлы. Мгновенный снимок экранного интерфейса Scg:

```
Screen (Базс ВПТК «Старт») /u/soft/Doc
saved Doc ERROR Makefile doc.c
doc.d doc.h get—key.c catalog.c screen.c
show.c system.c topics.c ity.c
cd/usr/new/lib
```

В вырожденном случае экранный интерфейс отвечает лишь за визуализацию текущего каталога, а пользователь работает в телетайпном режиме, набирая команды в отдельной командной строке, непосредственно следующей за изображением каталога. Важно то, что от пользователя не требуется никаких действий для переключения из одного режима в другой. Это достигается *разделением вводимых символов*, при которых управляющие символы обрабатываются объектно-ориентированной компонентой, а обычные — командной.

Таким образом, в любой момент работы Scg состояние диалога определяется выбранным файлом и состоянием командной строки. Возможность переопределения эффектов клавиш в ходе работы позволяет пользователю «настраивать клавиатуру» согласно либо своим привычкам, либо характеру выполняемых работ. Определение эффекта клавиши может быть параметризовано. Например, строка

```
cr %n %i
```

определяет шаблон команды, перед исполнением которой вместо %n представится имя выбранного файла, а вместо %i — командная строка. Кроме того, с каждой такой командой связываются три атрибута, указывающие на необходимость перерисовки экрана после выполнения команды, проверки применимости команды и ожидания нажатия какой-либо клавиши перед перерисовкой экрана.

Первый атрибут позволяет минимизировать перерисовку экрана для команд, имеющих малый объем вывода или вообще ничего не выводющих («`date`», «`ls —l %n`», «`mv %n.b %n`» и т. п.). Последний обеспечивает сохранение информации, выведенной некоторой командой.

Проверка применимости определяет, удовлетворяет ли выбранный файл некоторым условиям. В качестве такого условия выбрано следующее: «файл обычный текстовый». Оно дает возможность не редактировать и не выводить на экран каталоги, готовые программы и т. п.

Кроме того, для каждого файла известно, является ли он каталогом, что отражается на экране повышенной яркостью. Наличие определений эффектов клавиш

```
cd %n
cd ..
```

позволяет перемещаться по всей файловой системе, используя лишь эти две клавиши и клавиши управления курсором.

Экранный интерфейс Scg реализует еще одно средство — сохранение команд, набранных пользователем вручную. Он может выбрать (как обычно, с помощью управления курсором) любую из этих команд, отредактировать ее и выполнить повторно. Scg ограничивает число запоминаемых команд числом строк экрана, но это ограничение команд не критично, поскольку имеется возможность «забыть» ненужные команды.

Dos — система интеграции объектов

Если Scg предоставляет лишь средства, непосредственно опирающиеся на базовую ОС, то Dos дает возможность пользователю самому описывать типы объектов и организовывать из них произвольные структуры. Структура совокупности объектов лишь косвенно связана с древовидной структурой файловой системы UNIX и представляет собой ориентированный граф. Каждая вершина графа называется оглавлением и представлена отдельным файлом UNIX'a. В этом файле хранится имя оглавления и описание совокупности объектов, которыми, в частности, могут быть снова оглавления. Описание каждого объекта состоит из имени типа объекта, имени самого объекта и строки, связывающей этот объект с ОС (в простейшем случае — именем файла), например, текст «Коммуниканты и их взаимодействие» «com».

Имя типа объекта совпадает с именем некоторого меню, которое задается на специальном языке Menu и определяет действия и ответные реакции, допустимые для данного типа объектов. Внешне меню изображается своим

именем, за которым следует набор альтернатив. Пользователю предоставляется возможность указать курсором желаемую альтернативу и выбрать ее. При этом выполняются действия, указанные в данной альтернативе применительно к данному объекту. Способ выбора альтернативы также специфицируется в описании меню.

Мгновенный снимок текущего состояния экранного интерфейса **Doc**, приведенный ниже, состоит из заголовка текущего оглавления, перечня объектов из этого оглавления и текущего меню. Текущее меню определяется типом текущего объекта, который указывается стрелкой. Выбор текущей альтернативы меню приведет к редактированию файла, связанного со вторым объектом оглавления.

Doc (Барс ВНТК «СТАРТ») Интегрирующая система

Введение

→ Коммуниканты и их взаимодействие

Экранный интерфейс

Базовый исполнитель — Scg

Система интеграции объектов — Doc

Рисунки

Текст: Вывод Редактор Форматирование

Описание меню состоит из совокупности альтернатив. С каждой альтернативой связано одно или несколько действий. Оно может быть либо внутренним, т. е. изменяющим внутреннее состояние меню, либо внешним. К внутренним действиям относятся, например, *next* и *pred*, переназначающие текущую альтернативу на следующую и предыдущую соответственно. (Конечно, это сразу отражается на экране дисплея.) Текущая альтернатива выбирается действием *select*.

Внешние действия либо выдают сообщения монитору **Doc**, например, *return «in»*, *return «out»*, что приводит к смене текущего объекта и оглавления, либо передают некоторую команду базовому исполнителю. Команды формируются аналогично тому, как это делалось в **Scg**, например *exec «bed %f.txt»*. Однако в **Doc** параметры команды (здесь — *%f*) трактуются как переменные, значения которых устанавливаются либо монитором **Doc**, либо с помощью специальных внутренних действий **Menu**. Переменные *'t'* и *'d'* устанавливаются монитором равными соответственно строке, связанной с текущим объектом, и имени файла, хранящего текущее оглавление. Переменную *'l'* монитор **Doc** устанавливает равной имени «временного файла». Так, если переменная *'l'* установлена равной *«com»*, то выполнение последнего действия приведет к передаче базовому исполнителю команды *bed com.txt*. Программист может сам устанавливать значения переменных, например посредством присваивания *let 'w' = «62»*. После этого команда в действии *exec «pr — %w %f.txt»* раскроется как *pr—62 com.txt*.

Продемонстрируем аппарат меню на примере описания меню для работы с текстами.

```
def text =
  menu «Текст»
  open
  (0,12) «Вывод» 'w':
    print (-1,12) «Минуточку ...»
    exec «lyper —%w %f.txt > %t»
    let 's' = «%t»
    call sprint—text;
  *(0,20) «Редактор» 'r':
    let 's' = «%f.txt»
    call edit;
  (0,30) «Форматирование» 'f':
    print (-1,30) «Минуточку ...»
    exec «lyper —%w —f %f.txt > %t && mv
    %t %f.txt»
    print (-1,30)»
  use Main
  end
```

Известно, в какой точке экрана было вызвано то или

иное меню. Относительно этой точки идет отсчет координат (y, x) для всех сообщений, которые печатаются в ходе интерпретации меню.

Части, повторяющиеся в различных альтернативах или различных меню, можно вынести и идентифицировать именем. К таким частям относятся совокупности действий и альтернатив (*call* и *use*). Например,

```
def edit = ## ## редактирование файла
  proc
    exec «bed %s»
    geshow; ## ## обновить экран
```

Одно меню может вызывать другое, например, **do формат Печати (1, 0)**.

Важное свойство меню — его модифицируемость. Это значит, что программист может переопределить уже имеющееся или определить новое меню или действие, изменив таким образом поведение всей системы, даже в ходе ее работы. Эта возможность реализуется внутренним действием, «считывающим» файл, содержащий совокупность определений: *menu «draw. M»*. Указанное средство позволяет программисту создавать обстановку, соответствующую характеру тех объектов, которыми предстоит оперировать пользователю. Например, считывание файла, содержащего определение

```
def edit =
  proc
    exec «ed %s»
    geshow;
```

приведет к тому, что далее вместо редактора **Bed** будет вызываться штатный редактор **ed**.

Редактор **Bed**

Возможности редактора **Bed** включают большинство из общепринятых, в том числе сохраняемый набор макрокоманд, двунаправленный поиск и т. п. Однако специфическая обстановка повлияла на их форму и (или) способ реализации. Так, оказалось, что реализация обычной оконной системы при одновременном редактировании нескольких файлов требует слишком большого обмена с терминалом. Поэтому предпочтительнее предоставить пользователю возможность «мгновенного» переключения с одного файла на другой, нежели одновременной визуализации нескольких файлов.

Процесс редактирования естественно рассматривать как взаимодействие с экранным интерфейсом, и, следовательно, он должен удовлетворять требованиям, предъявляемым к любому экранному интерфейсу.

Состояние редактирования заключается не только в содержании редактируемого файла, но и текущем режиме (вставки/замены), регистре (русский/латинский), образце поиска и многом другом. Поддерживать на экране всю эту информацию нецелесообразно. По-видимому, должны быть отражены лишь те элементы состояния редактирования, которые, с одной стороны непосредственно влияют на выполнение текущих команд и, с другой, не слишком часто меняют свое значение. Большие возможности предоставляет способность терминала менять форму курсора, так как обычно в его окрестности сосредоточено внимание пользователя.

Как и для любого коммуниканта, выход из редактора **Bed** рассматривается не как завершение работы, а как временная приостановка, связанная с обращением к другой системе, в частности со сменой редактируемого файла. Именем нового редактируемого файла является сообщение *file* в команде *bed file*. При этом **Bed** «не знает», к кому он обращается, передавая команду *com*. Вызывается тот коммуникант, из которого **Bed** был вызван в последний раз. Это позволяет работать с одним редактором, вызывая его поочередно, скажем, из **Doc** и **Scg**. Можно организовать и одновременную работу двух (и более), присвоив каждому из них уникальное имя. В этом случае в каждом

экземпляре будет поддерживаться своя обстановка, например соответствующая редактированию программ или редактированию текстов.

С точки зрения вызывающей системы *Bed* неотличим от любого другого редактора: каждый сеанс имеет начало и конец, но, с точки зрения пользователя, вызов осуществляется «мгновенно», так как *Bed* постоянно загружен в память.

Кроме рассмотренной выше реализована еще одна форма коммуникации, состоящая в передаче команд на выполнение базовому исполнителю. (В этом случае нет затрат на загрузку базового исполнителя, который также постоянно находится в памяти.) Сделать это можно, либо сформировав команду из текущей редактируемой строки, либо описав соответствующую макрокоманду редактора. В первом случае, естественно, используются все редактирующие команды редактора. Особенно полезны при этом команды вставки текущего образца поиска и имени файла, что позволяет, например, не выходя из редактора, найти и выдать все вхождения образца поиска в текущем файле или во всех файлах текущего каталога.

«Упрятать» передачу сообщения базовому исполнителю в макрокоманду можно рассматривать как подсистему редактора, а сам редактор — как ведущую систему. Так, коммуникации дают возможность без изменения самого редактора ввести в него команды форматирования или печати текста, а для программы, находящейся в редактируемом файле, — трансляции или запуска, что приближает редактор, по крайней мере по поведению, к *Turbo*-системам.

Заключение. Описанную систему коммуникантов можно рассматривать как своего рода конструктор систем, каждая из которых реализует некоторый пользовательский интерфейс. Важное свойство системы — ее открытость, т. е. возможность подключения новых коммуникантов для нового типа объектов или способа взаимодействия. В основном даже самые простые средства взаимодействия значительно повышают эффективность использования ЭВМ.

Резидентное расположение коммуникантов в памяти дает эффект «мгновенного» переключения из одной обстановки в другую, что в сочетании с общими соглашениями о визуализации и взаимодействии делает систему коммуникантов единой, с точки зрения пользователя.

Телефон 35-43-44, Новосибирск

ЛИТЕРАТУРА

1. Денниг В., Эсиг Г., Маас С. Диалоговые системы «Человек — ЭВМ». Адаптация к требованиям пользователя. Сер. Математическое обеспечение ЭВМ. М. Мир, 1984 — С 112
2. Кристиан К. Введение в операционную систему UNIX. — М. Финансы и статистика, 1985. — 318 с.
3. Диалоговая единая мобильная операционная система ДЕМОС. Руководство программиста. Интерпретатор команд CSHELL. — Калинин: НПО «Центрпрограмм-систем», 1985. — 158 с

Статья поступила 16.07.88

УДК 681.3.06

М. Я. Инденбаум, В. А. Курылев

ИСПОЛЬЗОВАНИЕ И РАЗВИТИЕ СИСТЕМЫ МОДУЛЯ 2

Создателю языка программирования Модуля 2 [1, 2] Н. Вирту удалось разработать мощное и компактное средство программирования, сочетающее в себе многие свойства языков Ада, Си, Паскаль, Модуля и допускающее эффективную реализацию на мини- и микроЭВМ. Для Модуля 2 характерны такие черты, как четкий синтаксис, концепция модуля, поддержка определения абстрактных

типов данных, локализация системно (и машинно)-зависимых свойств, параллельное и низкоуровневое программирование, процедурный тип и работа с массивами переменной длины.

Модуля 2 сочетает язык и технологию программирования в рамках одной системы [3], значительно увеличивает надежность программного обеспечения (ПО) и является перспективным средством программирования, особенно для встроенных систем управления. Цель настоящего сообщения — описание опыта развития Модуля 2 для использования в обучении и производстве.

Исходная система Модуля 2, созданная коллективом авторов во главе с Н. Виртом, ориентирована на работу в ОС RT-11SJ (РАФОС монитор SJ). Для использования Модуля 2 на ЭВМ типа «Электроника 100/25», «Электроника 79», СМ4 разработаны системно-зависимые средства поддержки функционирования Модуля 2 в среде ОС RSX-11M (ОС РВ). Модифицированные определения некоторых объектов (главным образом это касалось интерфейса с файловой системой) приняты за основу и разработаны новые модули их реализации для ОС РАФОС монитор SJ. Полученная в результате версия системы Модуля 2 инвариантна относительно ОС РВ и РАФОС на уровне библиотечных модулей определения.

В настоящее время Модуля 2 используется в ряде предприятий для разработки общего системного ПО, программирования встроенных систем управления, а также в учебном институте в курсах программирования.

Особенности существующей версии системы Модуля 2: в полном объеме система функционирует на ЭВМ «Электроника 100/25», «Электроника 79», ДВК3;

возможен режим трансляции и компоновки программ в ОС РВ с последующим выполнением загрузочного модуля в среде РАФОС;

система русифицирована. Компилятор модифицирован для расширения синтаксиса, обеспечивающего написание идентификаторов на русском языке: литеры «Ш», «Щ», «Э», «Ч» отнесены к буквам, введен символ подчеркивания «_»;

для ЭВМ с процессором плавающей арифметики существует версия компилятора, обеспечивающая работу с вещественными числами двойной точности;

в компоновщик введен режим разделения кодов и переменных загрузочного модуля, что позволяет создавать программы для выполнения в ПЗУ;

введено дополнительное функциональное управление работой отладчика;

для ОС РВ разработаны библиотечные модули обработки асинхронных прерываний;

введены единые средства работы с командными файлами, для ДВК3 в библиотеку модулей включен графический пакет;

для ОС РВ заканчивается разработка интерактивного отладчика, построенного на базе исходного.

Язык Модуля 2 позволяет обойтись без программирования на ассемблере, однако может возникнуть необходимость использования отлаженных и эффективных подпрограмм, написанных на ассемблере ранее. Система Модуля 2 не предоставляет стандартных средств подключения модулей, созданных в другой программной среде. Ассемблерную программу можно включить в резидентное ядро при генерации системы, но такой путь не приемлем для программ прикладного характера.

Для решения данной задачи в ОС ДВК разработана методика слияния загрузочного ассемблерного файла формата SAV с загрузочным файлом программы на Модуля 2 формата LOD. Последний выполняется стандартно. Ассемблерный модуль (пакет модулей) включается следующим образом:

пишется модуль определения на Модуля 2 с соответствующими именами процедур и формальными параметрами;

разрабатывается модуль реализации на Модуля 2, в котором каждая процедура содержит один оператор — вызов интерфейсной процедуры, с помощью которой будет выполнен вход в ассемблерный пакет подпрограмм. Данная про-

цедура должна быть описана последней в модуле реализации и не содержит операторов (пустая);

модули определения и реализации компилируются, выполняется компоновка и выписывается адрес начала кодов интерфейсной процедуры;

для ассемблерных подпрограмм пакета пишется на ассемблере небольшой управляющий (главный) модуль, обеспечивающий соответствие в передаче параметров и выборе подпрограммы пакета, например по присвоенному ей номеру;

выполняется компоновка управляющего модуля и пакета стандартной программой LINK ОС ДВК, причем в качестве адреса загрузки указывается запомненный ранее адрес интерфейсной процедуры;

запускается специально созданная для этой цели программа слияния, которая модифицирует загрузочный файл модуля реализации, вставляя на место интерфейсной процедуры коды из ассемблерного загрузочного файла. Новый файл имеет формат системы Модуля 2 и используется обычным образом.

Описанная методика была применена для включения в библиотеку программ Модуля 2 написанного ранее на ассемблере пакета графических программ ЭВМ ДВКЗ.

Телефон 9-11-30, Смоленск

ЛИТЕРАТУРА

1. Вирт Н. Программирование на языке Модуля 2.— М.: Мир, 1987.
2. Жезмир Л. М., Кибиткин В. В., Осетинский Л. Г. Язык программирования Модуля 2 // Зарубежная радиоэлектроника.— 1986.— № 11.— с. 38—52.
3. Кибиткин В. В. Адаподобная технология программирования и ее реализация для СМ ЭВМ // УСиМ.— 1986.— № 12.— С. 71—77.

Сообщение поступило 26.07.88

УДК 681.3.06

С. И. Борисов, Л. В. Попов

ИНТЕГРИРОВАННАЯ ОПЕРАЦИОННАЯ ОБОЛОЧКА, РЕАЛИЗОВАННАЯ В ОС РАФОС

Время и качество решения прикладных задач в значительной степени зависит от того, насколько интерфейс диалогового общения с ЭВМ удобен пользователю. В последнее время все большее признание получает концепция создания интегрированных операционных оболочек (ИОО), разработанных, например, для ОС UNIX, MS DOS и др. [1—3].

Предлагаемая ИОО представляет собой надстройку над ОС РАФОС, позволяющую использовать все ПО вычислительной системы в режиме диалога, организованную в форме меню и обеспечивающую следующие возможности: максимальную детализацию сообщений, предназначенных начинающим пользователям; ввод директив до появления пояснительного текста на терминале; минимальное время отклика. Информация о структуре вычислительной системы хранится в базе данных, которая формируется и модифицируется с помощью специализированного редактора.

Основная структурная единица сеанса работы пользователя с ИОО — пункт диалога. Каждому пункту соответствуют два файла из базы данных с указанием набора директив, доступных в пункте; пояснительных текстов и инструкций, выводимых на экран терминала; имен прикладных программ, командных файлов, следующих пунктов диалога.

Экран терминала разбивается на окна, отображающие имя текущего пункта диалога; данные, соответствующие пункту диалога; имена установленных дискет и (или) сменных магнитных дисков; предписание установить дискету в случае отсутствия файлов, необходимых для выполнения выбранной директивы; указание на ввод информации; последнюю директиву (или последовательность директив, разделяемых «;»), введенную пользователем; сообщение об ошибках; текст дополнительной инструкции; режим работы (исполнение прикладной программы после ввода соответствующей директивы, создание командного файла из выбранных программ и его исполнение, получение справочной информации о структуре ПО вычислительной системы); назначение функциональных клавиш, предназначенных для вывода на экран терминала следующей страницы пояснительного текста или дополнительной инструкции, изменения степени детализации сообщений, ограничения объема выводимой информации на терминал, выхода в предыдущий пункт диалога, окончания сеанса работы.

При регистрации в базе данных нового пользователя создается модель, в которой указаны: идентификатор; фамилия, имя, отчество; уровень квалификации в прикладной области (1—3); уровень квалификации в области вычислительной техники и программирования (1—3); имя пункта диалога, с которого начинается сеанс работы; режим работы и объем информации, выводимой на терминал, установленные в предыдущем сеансе. Модель может корректироваться автоматически по анализу времени реакции и числу совершаемых ошибок или администратором вычислительной системы по просьбе пользователя.

Предлагаемая ИОО реализована на языках Паскаль, MACRO 11 и требует не менее 32 Кбайт оперативной памяти.

Телефон 552-79-64, Ленинград

ЛИТЕРАТУРА

1. Брябрин В. М. Программное обеспечение персональных ЭВМ.— М.: Наука, 1988.— 272 с.
2. Брябрин В. М., Блинов Д. М. Интегрированная система для решения прикладных задач // Микропроцессорные средства и системы.— 1987.— № 1.— С. 28—30.
3. Гнездилова Г. Г. Интегрированная операционная среда персональной ЭВМ // Микропроцессорные средства и системы.— 1987.— № 1.— С. 31—34.

Сообщение поступило 21.02.89

УДК 681.3.06

У. Ф. Фейзханов

ПАКЕТ ПРОГРАММ ДЛЯ БЫСТРОДЕЙСТВУЮЩЕГО ПЕРИФЕРИЙНОГО ПРОЦЕССОРА МТ-70

Использование БПП МТ-70 при обработке больших массивов аналоговых данных существенно ускоряет процесс анализа данных. Например, быстрое преобразование Фурье для 1024 комплексных точек выполняется на ДВКЗ за 1,3 с, а на МТ-70 — за 30,6 мс, т. е. в 40 раз быстрее. К сожалению, базовое программное обеспечение, которое поставляется с БПП МТ-70, громоздкое и не удобное для использования в прикладных программах. БПП МТ-70 имеет собственную память емкостью 64 Кбайт, и пользователь сам должен заниматься ее распределением под таблицы синусов и косинусов, массивы промежуточных и основных данных. После выполнения операций поль-

зователь должен смотреть, нет ли переполнения. Например, для выполнения БПФ действительного массива требуется 15 обращений к базовому пакету, программа становится трудно читаемой, ее сложно модифицировать.

Предлагаемый пакет программ позволяет максимально использовать аппаратные возможности МТ-70. Память МТ-70 подразделяется на две части: в первой находятся блоки данных пользователя, во второй — таблицы синусов и косинусов и служебная информация. Данные в МТ-70 организованы в формате поблочной-плавающей запятой, т. е. блок данных состоит из 16-разрядных мантисс: порядок, общий для всего блока. Для получения истинного значения необходимо мантиссу умножить на два в степени порядок. Память данных разбивается на блоки одинакового размера, определяемого подпрограммой SIZE. Блоки данных могут быть действительными и комплексными. Каждый блок содержит информацию о порядке блока и числе нулей перед старшей значащей цифрой. Перед выполнением операций сложения и вычитания блоки приводятся к одинаковому порядку. Во избежание переполнения или потери точности блоки нормализуются так, чтобы максимальная мантисса находилась в пределах 8192...16384.

Пакет состоит из 40 подпрограмм, которые по функциональному назначению можно разбить на три группы: организующие обмен данными между памятью МТ-70 и оперативной памятью ЭВМ блоками и отдельными элементами блока;

вычисляющие прямое и обратное БПФ, сложение, вычитание и умножение блоков, определяющие максимальное и минимальное значения блока;

служебные подпрограммы чтения-записи порядка блока и числа нулей перед старшей значащей цифрой, приведения блоков к одному порядку.

Чтобы результаты вычислений одной программы были доступны другим программам, выполняющимся последовательно, необходимо воспользоваться следующими двумя подпрограммами:

FRZ — запоминаются параметры пакета: размер блоков, порядок и адрес блока в памяти МТ-70;

AFRZ — восстанавливаются все параметры пакета из служебной зоны памяти МТ-70.

Первая программа должна заканчиваться обращением CALL FPZ, а вторая — начинаться CALL AFRZ.

Небольшие переделки в МТ-70 позволяют организовать обмен данными в оперативной памяти объемом 248 Кбайт. Под управлением однозадачного монитора RT11SJ расширенную память можно использовать в качестве электронного диска. Следующие четыре программы предназначены для такой работы:

DOF (NZ, NADR, NSL) — открыть файл с номером NZ, размер записи;

NSL, NADR — адрес, с которого производится запись-чтение в (из) блок;

DRD (NBL, NZ) — последовательное чтение из файла в блок;

DRD (NBL, NZ, I) — чтение с произвольным доступом (I — номер записи);

DWD (NBL, NZ) — последовательная запись из блока в файл;

DWD (NBL, NZ, I) — запись с произвольным доступом (I — номер записи);

DCF (NZ) — закрыть файл.

Объем памяти, занимаемой пакетом программ вместе с базовым программным обеспечением, куда входят 40 программ нижнего уровня, составляет 6,5 Кбайт.

Телефон 535-08-06, Москва

Сообщение поступило 28.02.89

УДК 631.3.06

А. В. Лопатин, Б. А. Позин

ФОРМАТИЗАТОР ИСХОДНОГО ТЕКСТА ПАСКАЛЬ-ПРОГРАММ

Одним из основных языков в составе программного обеспечения профессиональной ПЭВМ семейства ЕС1840 является Паскаль-M86, в котором в полной мере реализованы средства структурного программирования, а также возможности, обеспечивающие создание программ большого размера — до нескольких десятков тысяч операторов. Однако разработка таких программ затрудняется необходимостью раздельной компиляции. Все части программы должны быть собраны воедино на уровне исходного текста. Использование аппарата вставки исходного текста при компиляции позволяет размещать исходный текст программы в нескольких файлах.

В данной работе предлагается программа-форматизатор ЛИСТИНГ, обеспечивающая форматизацию исходного тек-

Дата : 13.11.87 ВРЕМЯ : 00:16:13
ИМЯ ФАЙЛА : C.PAS ЛИСТ 1

```

1 program Testing;
2 {
3
4   Testing - драйвер для пакета CheckSet
5 }
6
7 { $I CheckSet.PAS }
8 {
9   {
10    пакет CheckSet
11    проверка и модификация таблицы
12    FALSE - строка в таблице представлена
13    TRUE  - строка в таблице не представлена
14  }
15 }
16
17 type
18 ( тип формального параметра )
19 CheckString = string[80];
20 var
21   CheckVar : record
22   LengthSet : Integer;
23   SetString : array[1..1000] of string[10];
24 end;
25 function CheckSet(InStr : CheckString) : Boolean;
26 var
27   Target : string[10];
28   I : Integer;
29   Old : Boolean;
30 begin
31   with CheckVar do
32     begin
33       Target:=copy(InStr,1,10);
34       Old:=false;
35       I:=1;
36       while (i <= LengthSet) and (not Old) do
37         begin
38           if Target=SetString[i] then Old:=true
39           else i:=i+1;
40         end;
41       if not Old then
42         begin
43           LengthSet:=LengthSet+1;
44           SetString[LengthSet]:=Target;
45           CheckSet:=false;
46         end
47       else CheckSet:=true;
48     end;
49   end;
50 end;
51
52 var
53 ( действительный параметр )
54 TestString : string[40];
55 Result : Boolean;
56 label Label_Read;
57 begin
58   { Установка начальной длины таблицы }
59   CheckVar.LengthSet:=0;
60   Label_Read :
61   readln(TestString);
62   if TestString <> '&' then
63     begin
64       Result:=CheckSet(TestString);
65       if not Result then write(ln,TestString,'M');
66       writeln(ln,TestString);
67       goto Label_Read;
68     end;
69 end.

```

ста программ, написанных на языке Паскаль, и полученные листинга с явной прорисовкой логической структуры. Использование листинга, выпущенного форматизатором ЛИСТИНГ, позволяет значительно упростить понимание логики функционирования программ и поиска в тексте программы «лишних или недостающих» операторов END.

При анализе исходного текста программа ЛИСТИНГ выполняет следующие действия:

разделение операторов, т. е. каждый оператор печатается на отдельной строке;

выделение процедур и функций с помощью логических скобок, *меток*, печатая их увеличенным шрифтом, и *зарезервированных слов*, печатая их с двойной плотностью;

прорисовку структурных конструкций вида BEGIN — END, CASE — END, RECORD — END, REPEAT — UNTIL с помощью логических скобок.

Параметры печати задаются форматизатору одноуровневым функциональным меню, при этом функции выполняются одним нажатием функциональной клавиши. Дополнительные возможности программы: *задание активного дисковода*, *типа шрифта* («убористый» или стандартный), *плотности* печатаемого текста (67 или 93 строки на странице); *формирование страниц* (в соответствии с вертикальным размером листа бумаги АЦПУ ЕС — 305 мм), т. е. печать на каждом листе заголовка, содержащего имя печатаемой программы и номер листа (заголовок первого листа дополнительно содержит дату и время); *обработка файлов*, включаемых в исходный текст программы директивой {\$I имя файла}; *нумерация строк* в соответствии с их размещением в файлах. Номера строк, находящихся в включаемых файлах, печатаются с двойной плотностью.

На рисунке дан пример листинга, полученного с помощью программы-форматизатора ЛИСТИНГ. Используемый здесь текст программ Testing распечатан в режиме обработки включаемых файлов с выделением логической структуры и зарезервированных слов и с нумерацией строк. Первые шесть строк листинга содержат начало программы Testing, в шестой строке стоит директива включения файла, поэтому за ней (начиная с седьмой строки листинга) следует текст пакета, размещенного в файле CheckSet. Pas. Нумерация строк текста пакета, данная с двойной плотностью, соответствует расположению строк в файле. В тексте пакета встречаются две строки с номером 27. Это значит, что в исходном файле слово BEGIN находится в той же 27-й строке, что и слово WHILE, но для прорисовки логической структуры слово BEGIN было напечатано на отдельной строке. В 16-й строке исходного текста программы Testing содержится пример выделенной метки.

При разработке большого комплекса программ (более 30 тыс. операторов Паскаля) применение форматизатора помогает создавать тексты программ, которые удобно сопровождать, независимо от того, специалист какой квалификации разрабатывал ту или иную часть комплекса.

Существующие версии программы ЛИСТИНГ функционируют на ПЭВМ семейства ЕС1840 под управлением ОС типа М86 или Альфа-ДОС.

Телефон 203-25-00, Москва

Сообщение поступило 20.09.88

ОПЫТ РАЗРАБОТКИ АППАРАТНО-ПРОГРАММНЫХ СРЕДСТВ

УДК 621.317:681.3.62

И. Л. Мышкин, Г. О. Заречнюк

ФОРТРАН-80 ДЛЯ РАЗРАБОТКИ ПРОГРАММ ВСТРОЕННЫХ СИСТЕМ

Для выбора оптимальных программных средств проектирования ПО встроенной системы вычисления полярных координат на базе МП КР580ИК80А с отведенной областью ПЗУ 16 Кбайт проведен анализ нескольких компиляторов, доступных в среде ОС1800: БЕЙСИК-80, Паскаль-МТ+, Фортран-80. Основным критерий выбираемого языка — минимальный объем генерируемого соответствующим компилятором исполняемого модуля (с учетом библиотечных средств). Тестирование проводилось с помощью однотипного алгоритма вычисления математического выражения и использования форматированного ввода-вывода:

- 1: ввести массивы X1(100), X2(100)
- 2: цикл по i от 1 до 100
- 3: вычислить $Y = \sin(X1(i)) / \cos(X2(i)) \times \ln(i)$
- 4: вывести Y в заданном формате
- 5: конец цикла
- 6: конец программы,

где X1, X2 — массивы чисел в формате с плавающей или фиксированной точкой; Y — число с плавающей точкой.

Программа по такому алгоритму была написана на языках Фортран-80, БЕЙСИК-80 и Паскаль-МТ+ и скомпилирована соответствующими компиляторами в среде ОС1800. Результаты, приведенные в табл. 1, позволяют сделать следующие выводы:

наибольшей компактностью при решении задач математического характера обладает модуль, сгенерированный компилятором Фортран-80;

примемлемо также использование компилятора БЕЙСИК-80 в случае реализации коротких алгоритмов, где длина исходного текста не превышает 100 операторов; компилятор Паскаль-МТ+ для работы со встроенными средствами математической обработки практически не пригоден. Общий объем библиотек математических операций с плавающей точкой (FPREALS.ERL) и трансцендентных функций (TRANCEND.ERL) составляет 12 Кбайт и полностью переносится в исполняемый модуль. Это приводит к увеличению объема модуля сверх допустимых пределов

Таблица 1

Тип компилятора	Объем Кбайт		Длина	
	компилятора, на диске	прикладных библиотек	скомпилированной программы, байт	рабочего модуля после компоновки, Кбайт
Фортран-80	26	FORLIB.REL 27	201	9
БЕЙСИК-80	31	BASLIB.REL 49	566	12
Паскаль-МТ+	166	PASLIB.ERL 24 FPREALS.ERL 8 TRANCEND.ERL 4	263	14

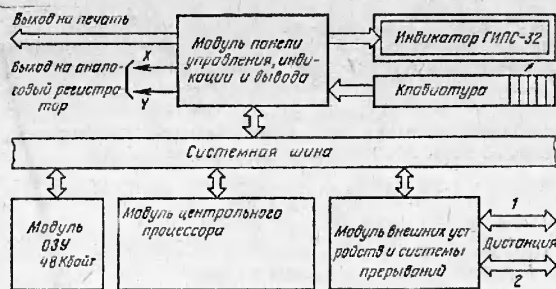


Рис. 1. Структурная схема системы вычисления полярных координат

(>16 Кбайт). Поэтому при проектировании встроенной системы (рис. 1) для вычисления полярных координат в качестве основного языка был выбран Фортран-80. В задачи системы входили сбор данных измерения дистанции, селекция сбросов и ложных данных, сглаживание результатов измерений по методу наименьших квадратов, вычисление полярных координат и вывод результатов на цифровые и графические регистраторы. Для удобства отладки система спроектирована по модульному принципу. Модуль центрального процессора (ЦП) содержит МП КР5801К80А с тактовым генератором (ТГ) и системным контроллером (СК), усилитель шины адреса, дешифратор адресов памяти и ПЗУ объемом 16 Кбайт, занимающее адресное пространство 0Н...3FFFH. Модуль ОЗУ объемом 48 Кбайт построен по принципу «прозрачной» динамической памяти и занимает остальное адресное пространство 4000Н...FFFFH. Модуль внешних устройств и системы прерываний принимает данные о дистанции от измерительных систем 1 и 2 и выработывает запросы прерываний МП на обслуживание соответствующих устройств в реальное время. Модуль управления, индикации и вывода обслуживает матричную индикаторную газоразрядную панель, клавиатуру, печать и аналоговый самописец.

Программное обеспечение состоит из четырех модулей: SF — набор драйверов внешних устройств и имитация ввода-вывода ОС1800 (ассемблер); DEFECT — подпрограмма сбора и селекции входных данных (Фортран); MINSQR — подпрограмма обработки массивов по методу наименьших квадратов (Фортран); ALG — подпрограмма расчета координат и вывода результатов (Фортран). Данные компиляции приведены в табл. 2. После компоновки скомпилированных подпрограмм с библиотекой FORLIB. REL длина исполняемого модуля составила 15,5 Кбайт, в том числе библиотечных подпрограмм — 8,5 Кбайт. Общая длина исполняемого модуля не превысила отведенного пространства (16 Кбайт)

Разработка программ и комплексная отладка совместно с аппаратной частью велась с помощью отладочной микроЭВМ (рис. 2). Она включает модуль ЦП, ОЗУ 64 Кбайт, интерфейс НГМД «Электроника НГМД-7012».

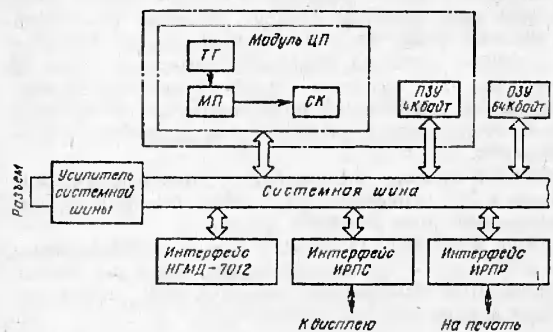


Рис. 2. Структурная схема микроЭВМ разработки и отладки программ

Название программы/ длина (строк)	Длина кодов в системе		Использование		
	16-рич- ной	10-рич- ной	ввода- вывода	трансцен- дентных функций	чисел с плавающей точкой
SF	0663H	1635	—	—	—
DEFECT/35	0309H	777	+	—	+
MINSQR/62	06ACH	1708	—	—	+
ALG/126	0BDDH	3037	+	+	+
Всего		7157			

интерфейс ИРПС для связи с терминалом, интерфейс ИРПР для вывода данных на печать и усилитель системной шины для подключения к шине отлаживаемого модуля. В ПЗУ модуля ЦП находятся монитор-отладчик и базовая система ввода-вывода ОС1800.

В процессе отладки из отлаживаемого блока удаляются модули ЦП и ОЗУ и системная шина инструментальной микроЭВМ подключается к внутренней шине блока. При этом их адреса не должны перекрываться с адресами внешних устройств, уже имеющимися в микроЭВМ.

Предварительно производится контрольная проверка модулей отлаживаемого блока простыми тест-программами, позволяющих оценить их работоспособность. Набор подпрограмм первоначально компонуется в виде загрузочного модуля ОС1800 с адреса пользовательских программ 0100H без разделения области кодов и данных. Отлаживаемая программа загружается либо непосредственно пользователем, либо с помощью отладчика DDT или SID. Ассемблерный модуль отлаживается полностью указанными отладчиками.

Применение языка Фортран-80 позволило разработать эффективное с точки зрения объема исполняемой программы ПО для встраиваемой системы и значительно снизить время и затраты на ее разработку.

Телефон 74-21-13, Львов

Статья поступила 18.03.88

УДК 681.326.3:681.3.06

И. Г. Коршун, А. Н. Хорошевский

АВТОМАТИЗАЦИЯ ПРОГРАММИРОВАНИЯ БИС ПЗУ СПЕЦИАЛИЗИРОВАННЫХ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ

Один из наиболее трудоемких этапов проектирования специализированных микропроцессорных устройств (СМПУ) — разработка и отладка рабочих программ, а также их запись в БИС ПЗУ.

Процесс разработки рабочих программ для СМПУ состоит из оперативной корректировки их текста и перезаписи информации в ЗУ. При этом возникает проблема переноса данных от эталонного носителя в ЗУ без потери информации. Кроме того, с течением времени потеря информации может произойти в самой БИС ЗУ, что связано с восстановлением переключек, воздействием импульсных помех. В связи с этим эталонный текст программы должен храниться на одном из внешних носителей.

В литературе [1, 2] рассматриваются варианты средств автоматизации программирования ПЗУ, однако они либо выполнены на устаревшей материальной базе и не обеспечивают многих практически важных функций, либо в них не приведены не только тексты и описания программного обеспечения, но и его алгоритмы.

Предлагаемая методика программирования БИС ЗУ — вариант универсального микроассемблера для СМПУ произвольной конфигурации, позволяет располагать адреса в любом порядке, группировать данные в произвольное число полей микрокоманды и представлять их в выбранной системе счисления, а также записывать подробные комментарии в тексте микропрограммы.

Программа ИГЛА, реализующая эту методику, написана на языке МАКРО 11. Ее полный текст и подробное описание приведены в [3]. Программа имеет объем загрузочного модуля около 16 Кбайт, работает под управлением ОС РАФОС и обеспечивает автоматизацию всего процесса программирования ЗУ, включая множество операций, предшествующих программированию (формирование исходного текста микропрограммы, его корректировка, верификация и т. д.). Программа не накладывает ограничений на тип ПЗУ и число полей микрокоманды и может работать с микропрограммами, объем текста которых превышает емкость одной дискеты (возможен перенос информации с одного типа ЗУ на другой).

ИГЛА обеспечивает следующие основные режимы работы: запись информации из файла в ЗУ, чтение информации из ЗУ в файл, сравнение информации, содержащейся на различных носителях, синтаксический контроль записи адресов и данных, корректировка данных текстовых файлов.

Содержимое ЗУ хранится в виде текстовых или двоичных файлов на устройствах прямого доступа. В текстовом файле содержится информация для программирования нескольких кристаллов ЗУ, поэтому в одной таблице можно описать полный текст микропрограммы с произвольным числом полей микрокоманды. Первый столбец включает в себя 16-разрядные адреса ЗУ, следующие — 8-разрядные данные соответствующих кристаллов ЗУ для этих адресов. Содержимое каждого кристалла ЗУ хранится в отдельном столбце. Адреса и данные могут записываться, по усмотрению пользователя, в двоичной, восьмеричной, десятичной или шестнадцатеричной системах счисления. Значение системы кодируется программой с помощью специальных символов, которые фиксируются непосредственно перед численным значением адреса и данных. Эти символы образуют разделители между столбцами данных, а также выделяют адресный столбец; данные с одним адресом, но располагающиеся в различных столбцах — поле микрокоманды, которая может состоять из нескольких строк текста. Число пробелов не лимитировано. Конец строки данных отображается символом «.». Система счисления кодируется специальными символами, в адресном столбце: «=» — двоичная, «&» — восьмеричная, «@» — десятичная, «H» — шестнадцатеричная; в столбцах данных: «:» — двойная, «+» — восьмеричная, «%» — десятичная, «#» — шестнадцатеричная.

В качестве комментариев можно использовать любые символы, кроме зарезервированных для кодирования системы счисления адресного столбца. Пример таблицы, содержащей микропрограмму с длиной поля микрокоманды, равной 32, приведен ниже.

МИКРОПРОГРАММА СОПРОЦЕССОРА ДЛЯ РЕШЕНИЯ СИСТЕМЫ
ЛИНЕЙНЫХ УРАВНЕНИЙ С ДВУМЯ НЕИЗВЕСТНЫМИ
И ИЗВЛЕЧЕНИЯ КОРНЯ КВАДРАТНОГО

Тип ПЗУ — K656PT5

Нумерация битов	76543210			
Кристалл	1	2	3	4

АДРЕС K4K5C3-C1U3-U1 E3-E0M8-M1 K6A6-A4A3-A0 СИФ6-Ф4Ф3-Ф0

МОНИТОР

ОЖИДАНИЕ, СБРОС, ОЧИСТКА РО ЧЕТВЕРТОГО ПРОЦЕССОРА.

ГД 0025:11 011 000:0000 0000:0 010 1011:1 100 0000.

ВКЛЮЧЕНИЕ ШИННОГО ФОРМИРОВАТЕЛЯ НА ПРИЕМ.
M1 В АС ПЕРВОГО ПРОЦЕССОРА, ПЕРЕХОД С ЗАПИСЬЮ В РЕ-
ГИСТР КОМАНД.

2Д 0008:11 000 000:0000 0000:0 111 1000:1 000 1011.

УСТАНОВИТЬ «ПУСК». ОЧИСТИТЬ РО ЧЕТВЕРТОГО
ПРОЦЕССОРА
ЗАХВАТ МАГИСТРАЛИ ЕСТЬ?

ЗУ 0015:11 011 000:0000 0010:0 011 0101:1 100 0000.
ПЕРЕХОД ПО СОДЕРЖИМОМУ РЕГИСТРА КОМАНД НА РЕШЕНИЕ
СИСТЕМЫ УРАВНЕНИЙ ИЛИ ИЗВЛЕЧЕНИЕ КОРНЯ КВАДРАТНОГО.
4Д 0024:11 011 000:0000 0000:0 110 0110:1 001 0000.

МОДУЛЬ ВВОДА

В РО ПЕРВОГО ПРОЦЕССОРА — АДРЕС НАЧАЛА АРГУМЕНТОВ
В АС ПЕРВОГО ПРОЦЕССОРА — ВВЕДЕННОЕ ЗНАЧЕНИЕ
АРГУМЕНТА

РО В РА, РО-1 В РО ПЕРВОГО ПРОЦЕССОРА, ВКЛЮЧЕНИЕ ЕГО
КАНАЛА А

1Д 0006:11 000 000:0100 0000:0 000 1100:0 01 0000.

РО+1 В РО, ШИННЫЙ ФОРМИРОВАТЕЛЬ ВКЛЮЧЕН НА ПЕРЕДАЧУ,
ВКЛЮЧЕНИЕ КАНАЛА А ПЕРВОГО ПРОЦЕССОРА.

2Д 0056:11 000 000:0100 0000:0 011 1100:0 011 0000.

В крайнем левом столбце записаны адреса в шестнадцатеричной системе; в остальных четырех столбцах — текст микропрограммы в двоичной системе. Пробелы в пределах столбца используются для выделения полей микрокоманды. Комментарии, расположенные произвольным образом, облегчают чтение и понимание микрокоманды (микропрограммы).

Режимы чтения и корректировки приводят к модификации таблицы данных, которая записывается в новом выходном файле, а предыдущая сохраняется в старом файле на случай возможных сбоев аппаратной части или ошибочных действий пользователя.

Если входной файл с именем, указанным пользователем, отсутствует, то он создается программой в режимах чтения и корректировки. Этот файл содержит таблицу с заполненным в соответствии с заданной системой счисления адресным столбцом и пустым столбцом данных и используется программой в качестве входного. Если пользователь указывает номер столбца больший, чем число столбцов во входной таблице, то при работе в режимах, модифицирующих таблицу данных, в выходную таблицу дописываются пустые недостающие столбцы.

Для записи информации можно использовать как текстовый файл, содержащий таблицу с описанным выше способом хранения информации о программировании ЗУ, так и бинарный (двоичный) файл. В текстовом файле пользователь фиксирует начальный и конечный адреса записи и номер кристалла (столбца). Для двоичного файла задаются начальные номера блока и слова в этом блоке, а также указывается, какой байт каждого слова следует записывать в ЗУ: младший или старший. С помощью двоичных файлов в ЗУ можно программировать загрузочный модуль рабочей программы пользователя. Перед записью информации программа может произвести синтаксический контроль адресов и данных таблицы входного текстового файла. Программа обнаруживает следующие ошибки:

символ в адресном столбце или столбце данных, не являющийся цифрой в указанной пользователем системе счисления;

символьная запись числа в столбце данных (адреса больше максимального 8-разрядного (16-разрядного) двоичного числа).

В режиме чтения информация из ЗУ помещается в заданный пользователем столбец таблицы текстового файла, который совпадает с номером кристалла. Начальное и конечное значения заданного диапазона адресов могут совпадать. В процессе чтения программа создает выходной файл, содержащий модифицированную таблицу данных в соответствии с указанной пользователем системой счисления.

В режиме сравнения анализируется информация, записанная в ЗУ, и информация, содержащаяся в файле, либо содержимое двух столбцов данных.

Сравнение файл-файл возможно только для текстовых файлов, а ЗУ-файл — как для текстовых, так и для двоичных файлов. При обнаружении различий информация записывается в выходной файл отличий:

A3 — XXXX D3 — ББББББББ ДЕ — ББББББББ АЕ —
XXXX % — ДДД.

где АЗ, ДЗ (АЕ, ДЕ) — адрес и данные сравниваемого (эталонного) носителя; «%» — процент несовпадений при 100-кратном чтении по одному и тому же адресу ЗУ; % — 0 при сравнении файлов, ХХХХ — слово (2 байта) адреса в шестнадцатеричной системе счисления; ББББББББ — байт данных в двоичной системе счисления; ДДД — байт значения процента несовпадений в десятичной системе счисления.

Режим корректировки предназначен для внесения изменений в таблицу данных: очистка и удаление столбца данных, вставка пустого столбца и замена его содержимого (заполнителем, который вводится пользователем по запросу с терминала (один байт) и помещается по всем указанным адресам и данным с терминала по всем адресам из указанного диапазона или из корректирующего файла). Во всех режимах, кроме режима удаления столбца, система счисления данных в корректируемых столбцах определяется пользователем. В качестве корректирующих столбцов можно использовать столбцы корректируемого файла. Это позволяет изменять систему счисления данных указанного столбца.

Ядро программы — аппаратно независимо, т. е. может использоваться при работе с любым программатором, подключенным к ДВК или СМ ЭВМ и реализующим режимы записи и чтения ЗУ. Подпрограмма ZAPDAN инициирует запись байта данных на программирующем устройстве, RIDDAN — чтение, CLERP реализует вспомогательные функции на устройстве, например инициализацию операции очистки ПЗУ. Ядра программы и подпрограмм связаны тремя глобальными переменными: ADRES модифицируется ядром и содержит значение (слово) адреса ЗУ, по которому производится чтение-запись данных;

DANNYE — ядром при записи и аппаратно зависимой частью при чтении данных;

JSWIGL — только ядром и передает в нулевом бите младшего байта информацию о типе ЗУ (0, если ПЗУ или ОЗУ, и 1, если РПЗУ).

Подпрограммы ZAPDAN, RIDDAN и CLERP образуют программный интерфейс между ядром программы и программирующим устройством. Они обеспечивают прием адреса и данных из ядра в режиме записи, установку в программирующем устройстве требуемого режима работы, прием адреса из ядра и передачу данных в ядро в режиме чтения.

При использовании программы ИГЛА для работы с программирующим устройством конкретного типа необходимо выполнить следующее:

подключить программирующее устройство к каналу ЭВМ через соответствующий контроллер;

написать процедуры ZAPDAN, RIDDAN и CLERP на языке МАКРО 11 в соответствии с рекомендациями [4];

оттранслировать эти процедуры и прикомпоновать их к ядру.

310141, Харьков-141, пр. Ленина, 14, ХИРЭ, каф. ТОЭ; тел. 63-08-73

ЛИТЕРАТУРА

1. Лукьянов Д. А. Схемотехника универсальных программаторов ПЗУ // Микропроцессорные средства и системы. — 1985. — № 3. — С. 84.
2. Жихарев В. И. Программатор на базе микроЭВМ «Электроника ДЗ-28» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 40—43.
3. Коршун И. Г., Хорошевский А. Н., Борисов Е. В. Автоматизация программирования БИС ПЗУ устройств управления на микропроцессорах. — Харьков, 1986. — Деп. в УкрНИИТИ 1986, № 2105-Ук 86.
4. Дианов А. П., Щелкунов Н. Н. Методика программирования микросхем ПЗУ // Микропроцессорные средства и системы. — 1985. — № 3. — С. 75.

Статья поступила 23.09.87

УДК 681.324

С. В. Назаров, А. В. Луговец

ИССЛЕДОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ЭВМ С МАГИСТРАЛЬНОЙ АРХИТЕКТУРОЙ

Оценка зависимости производительности ЭВМ от интенсивности обмена данными с периферийными устройствами (ПУ) имеет важное значение для пользователя при обосновании выбора ЭВМ, режима обмена и определения накладных расходов.

Постановка задачи. Пусть на ЭВМ реализуется некоторый класс задач, характеризуемый смесью команд (S), частотой выполнения операций ввода-вывода (f), объемом блока передаваемых данных (e_0), режимом обмена (B) и взаимной удаленностью (l) блоков данных при обмене с накопителем на магнитных дисках. Если помнящая производительность ЭВМ для этого класса задач P_p , то реальная определяется величиной $P_p = K_p \cdot P_n$, где K_p — коэффициент снижения, учитывающий информационный обмен.

$$K_p = F_1(e_0, f, B, l),$$

где $e_0, f \geq 0$, $e_0 \cdot f \leq E_{max}$; $B=0$ (без буферизации); $B=1$ (с буферизацией); $0 \leq l \leq L_{max}$; E_{max} — макс скорость обмена с ПУ (Кбайт/с); L_{max} — число блоков магнитного диска (МД).

Значение E_{max} зависит от технических характеристик ПУ. Например, для НМД — это скорость вращения диска и объем информации, который может быть считан за один оборот МД.

В эксперименте работа прикладных программ моделируется смесью команд в процессе выполнения которых через равные промежутки времени $\Delta t = 1/f$ инициируются операции чтения блоков данных объемом e_0 с МД. Процентный состав команд этой смеси приведен в табл. 1. Операции умножения и деления с фиксированной точкой и арифметические операции с плавающей точкой в смеси команд ГИБСОН-1 заменены операциями сложения и вычитания с фиксированной точкой и операциями сдвига. Необходимо отметить, что в архитектуре рассматриваемых ЭВМ существует восемь различных способов адресации операндов, существенно влияющих на время выполнения команды. При проведении эксперимента предполагалось, что в командах смеси применяются только регистровая, автоинкрементная и индексная адресации операндов с вероятностью появления 0,4, 0,2 и 0,4 соответственно (при использовании другой смеси способы адресации могут быть изменены).

Таблица 1

Смесь команд

Обозначение	Название	Содержание смеси, %
MOV	Загрузка (запоминание)	31,2
ADD, SUB, INC, DEC	Индексная арифметика	18,0
ADD, SUB, INC, DEC, ADC, SBC	Арифметика с фиксированной точкой	13,9
ASZ, ASR, ROZ, ROR, BIS, XOR	Сдвиг (логические операции)	11,2
JSR, RTS, CCC, CZR и др.	Прочие	5,4
CMP	Сравнение	3,8
BVC, BNE, BZT, SOB, JMP	Ветвление	16,5

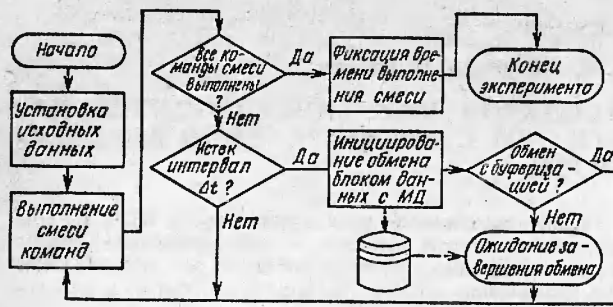


Схема проведения эксперимента

Если K — число команд в смеси, T_s — время их выполнения при заданной интенсивности обмена с ПУ, то $P_p = K/T_s$.

Программные средства, обеспечивающие проведение такого эксперимента, включают управляющий модуль ОСРРР, измерительную программу МГР, программы обработки прерываний от таймера TIME10 и TIME, а также модуль SM, моделирующий смесь команд. Выделение смеси команд в отдельный программный модуль позволяет применять измерительные средства для оценки производительности ЭВМ на любой смеси команд.

Разработанные программные средства позволяют оценивать производительность ЭВМ, учитывая взаимодействие с МД (см. рисунок), устройством печати и дисплеем как в отдельности, так и в совмещенном режиме.

Управляющий модуль ОСРРР иницируется ОС после загрузки в оперативную память ЭВМ. В диалоге с оператором устанавливаются исходные данные, вызывается программа МГР, обрабатываются ее выходные данные и определяется номинальная производительность процессора. Для измерения T_s используется программа МГР и производится оценка производительности ЭВМ для всех заданных оператором интенсивностей обмена с ПУ.

За один прогон программы ОСРРР может быть проведена серия экспериментов с исходными данными V , e_s , Δt , I .

Для каждого из этих параметров задаются начальное и конечное значения, а также дискретность их изменения.

С помощью программы МГР определяется время выполнения смеси команд, моделируемой модулем SM, в двух режимах: в условиях обмена данными с ПУ и без организации такого обмена. В первом случае программа МГР, получив от модуля ОСРРР управление, настраивается на принятые исходные данные; в векторе прерываний от таймера адрес системной программы обработки прерываний заменяется адресом программы TIME10 и управление передается модулю SM. Эти действия через каждые 20 мс обеспечивают запуск программы TIME10, которая выполняет операции ввода-вывода на ПУ с заданными параметрами и подсчитывает время проведения эксперимента.

В случае, если программа МГР выполняется в режиме определения времени выполнения смеси команд без обмена с ПУ с целью определения номинальной производительности процессора, для обработки прерываний от таймера вместо программы TIME10 используется программа TIME. TIME не обращается к ПУ, а только изменяет счетчик времени.

После определения реальной производительности ЭВМ программа ОСРРР вычисляет коэффициент снижения этой производительности. Результаты экспериментов, выдаваемые на печать, включают в себя значение номинальной производительности и таблицы, содержащие коэффициенты K_s для всех значений параметров.

Комплекс измерительных программ выполняется под управлением ОС, совместимых с RT-11 (Рафос, Фодос или ОСДВК). Программа ОСРРР разработана на языке Паскаль, остальные — на ассемблере. Объем загрузочного

модуля, объединяющего все программные средства, составляет 4 Кбайт без учета рабочих областей.

Результаты эксперимента. Рассчитана производительность микроЭВМ «Электроника 60М», микроЭВМ ДВК2М и мини-ЭВМ СМ4 в зависимости от интенсивности обмена с НГМД типа «Электроника НГМД-70», «Электроника ГМД-7012» и «Электроника НМД СМ5402» соответственно. Для микроЭВМ «Электроника 60М» производительность оценивалась в условиях одновременного обмена информацией с двумя устройствами: НГМД и дисплеем типа I5ИЭ-00-013. Эксперименты проводились под управлением ОСДВК и SJ-монитора ОС РАФОС.

Номинальная производительность микроЭВМ «Электроника 60М», микроЭВМ ДВК2М и мини-ЭВМ СМ4 на используемой смеси команд составляют соответственно 96, 113 и 240 тыс. команд/с. Коэффициенты снижения производительности для исследуемых ЭВМ линейно падают в зависимости от частоты обращения к внешним запоминающим устройствам (ВЗУ) и существенно зависят от режима обмена.

Если данная смесь команд подходит для задачи пользователя, то, подставив свои значения исходных данных в таблицу значений коэффициента K_s (табл. 2, 3, 4), можно рассчитать производительность ЭВМ.

В режиме обмена с буферизацией значение K_s выше, чем в режиме без буферизации, так как в последнем слу-

Таблица 2

Значения коэффициента K_s для микроЭВМ «Электроника 60М»

e_s , Кбайт	$f=0,5$ Гц		$f=1,0$ Гц		$f=2,0$ Гц		$f=3,33$ Гц
	$I=0$	$I=400$	$I=0$	$I=400$	$I=0$	$I=400$	$I=0$
0,5	$\frac{0,99}{0,93}$	$\frac{0,99}{0,77}$	$\frac{0,98}{0,83}$	0,54	$\frac{0,95}{0,70}$	$\frac{0,95}{—}$	$\frac{0,91}{0,48}$
1,0	$\frac{0,98}{0,91}$	$\frac{0,98}{0,71}$	$\frac{0,95}{0,81}$	0,45	$\frac{0,90}{0,60}$	$\frac{0,90}{—}$	$\frac{0,83}{0,30}$
1,5	$\frac{0,97}{0,89}$	$\frac{0,97}{0,70}$	$\frac{0,93}{0,75}$	$\frac{0,93}{0,39}$	0,85	$\frac{0,85}{—}$	—
2,0	$\frac{0,96}{0,83}$	$\frac{0,96}{0,66}$	$\frac{0,91}{0,69}$	$\frac{0,91}{0,33}$	$\frac{0,81}{0,37}$	$\frac{0,81}{—}$	—

Примечание. В числителе дано значение K_s при $V=1$, в знаменателе — при $V=0$.

Таблица 3

Значения коэффициента K_s для микроЭВМ ДВК2М

e_s , Кбайт	$f=0,5$ Гц		$f=1,0$ Гц		$f=2,0$ Гц		$f=3,33$ Гц
	$I=0$	$I=400$	$I=0$	$I=400$	$I=0$	$I=400$	$I=0$
0,5	$\frac{0,99}{0,92}$	$\frac{0,99}{0,77}$	$\frac{0,98}{0,82}$	$\frac{0,98}{0,48}$	$\frac{0,96}{0,66}$	$\frac{0,96}{—}$	$\frac{0,93}{0,52}$
1,0	$\frac{0,98}{0,90}$	$\frac{0,98}{0,75}$	$\frac{0,96}{0,78}$	$\frac{0,96}{0,43}$	$\frac{0,92}{0,58}$	$\frac{0,92}{—}$	$\frac{0,86}{0,41}$
1,5	$\frac{0,97}{0,87}$	$\frac{0,97}{0,70}$	$\frac{0,94}{0,73}$	$\frac{0,94}{0,38}$	$\frac{0,88}{0,51}$	$\frac{0,88}{—}$	$\frac{0,79}{—}$
2,0	$\frac{0,96}{0,84}$	$\frac{0,96}{0,59}$	$\frac{0,92}{0,67}$	—	$\frac{0,84}{0,39}$	$\frac{0,84}{—}$	—

Примечание. В числителе дано значение K_s при $V=1$, в знаменателе — при $V=0$.

Таблица 4

Значения коэффициента K_p для мини-ЭВМ СМ4

С _с , Кбайт	f=2,0, Гц		f=3,3, Гц		f=5,56, Гц		f=10,0, Гц	
	I=0	I=2000	I=0	I=2000	I=0	I=200	I=0	I=2000
1	0,997	0,997	0,995	0,995	0,990	0,990	0,982	0,982
	0,99	0,88	0,93	0,81	0,90	0,65	0,81	0,34
6	0,990	0,990	0,983	0,983	0,970	0,970	0,943	0,943
	0,93	0,85	0,85	0,73	0,80	0,53	0,68	0,13
11	0,982	0,982	0,970	0,970	0,948	0,948	0,904	0,904
	0,89	0,77	0,81	0,64	0,69	0,36	0,34	—
16	0,974	0,974	0,957	0,957	0,928	0,928	—	—
	0,76	0,70	0,65	0,51	0,38	0,16	—	—

Примечание. В числителе дано значение K_p при $V=1$, в знаменателе при $V=0$

час процессор значительное время «простаивает», ожидая завершения операций ввода-вывода, а производительность ЭВМ зависит от размещения блоков данных на магнитном диске (в процессе обмена с буферизацией этот параметр на производительность ЭВМ не влияет, так как переключение головок чтения-записи производится без участия процессора)

По полученным в результате экспериментов значениям в процессе считывания блоков информации различного объема строятся зависимости потерь производительности процессоров микро- и мини-ЭВМ в расчете на один байт

передаваемых данных от объема блока. Эти затраты существенно зависят от применяемого способа передачи данных по внутренней магистрали ЭВМ между ПУ и ОП и с ростом объема блока нелинейно сокращаются в пределах от 5,5 до 4,5 команд на байт для программно-управляемого обмена с прерываниями и от 0,75 до 0,2 команд на байт для обмена по каналу прямого доступа. Для микроЭВМ «Электроника 60М» в условиях одновременного ввода данных с НГМД и вывода на дисплей установлено, что одновременный обмен с устройствами данного типа увеличивает потери производительности процессора на передачу одного байта информации более чем в два раза по сравнению с последовательным обменом.

Выводы. Производительность ЭВМ существенно зависит от интенсивности и режима обмена с ПУ. Наибольшая производительность ЭВМ достигается в режиме обмена с буферизацией. При этом следует учитывать, что одновременный обмен с несколькими ПУ для ЭВМ с магистральной архитектурой вносит потери производительности, не равные сумме потерь производительности по обмену с каждым ПУ. Дополнительные потери производительности зависят от типа ЭВМ, способа передачи данных и ПУ

Среди сравниваемых микро- и мини-ЭВМ наилучшими характеристиками производительности обладает мини-ЭВМ СМ4.

ЛИТЕРАТУРА

1. Драммонд М. Е. Методы оценки и измерения дискретных вычислительных систем. М.: Мир, 1977
2. Машины вычислительные электронные цифровые общего назначения. Общие технические требования ГОСТ 16325—76. Приложение 2. Методика определения производительности ЭВМ общего назначения. М.: Изд-во стандартов, 1976. С. 14—28.
3. Феррари Д. Оценка производительности вычислительных систем. М.: Мир, 1981

Статья поступила 30.09.87

УДК 681.326

В. Ю. Солонин

САМОКОНТРОЛЬ УНИВЕРСАЛЬНЫХ ПРОГРАММАТОРОВ

Производительность настройки универсальных программаторов* значительно повышается при введении специального блока контроля (БК). Трудности построения схемы БК заключаются в том, что уровни вершины и основания импульсов, формируемых программатором, поступающих на его многочисленные выходы, отличны от TTL-уровней. Параметры таких импульсов невозможно проконтролировать, соединив выходы программатора с его входами, так как входы реагируют только на отсутствие или наличие напряжения без конкретизации его значения, хотя некоторые неисправности программатора проявляются и в искажении уровней импульсов.

В БК число элементов измерения значений напряжений удалось свести к одному, применив последовательный алгоритм контроля. Для этого используются два пороговых элемента на транзисторах V30, V31 и стабилитронах V28, V29 (см. рисунок). Их входы через диоды подключены к выходам всех 24 ключей Q0...Q7, A0...A15 (такое чис-

ло ключей необходимо для программа-тора при включении в номенклатуру программируемых микросхем, например КР556РТ1, КР573РФ3 и др.) и к выходам программируемых источников на напряжения E2, E3. Блок контроля содержит соединения выходов ключей программатора Q0...Q7, A0...A15 с его входами Q0_{пр}...Q7_{пр}, A0_{пр}...A15_{пр}, которые в программаторе через диодно-резисторные цепи считывания соединены с выводами микросхемы-порта КР580ВВ55А*. Указанное соединение позволяет проверить цепи считывания во время контроля ключей программатора. Для считывания Лог 0 входами программатора при закрытом состоянии ключей выходы последних соединены с общим проводом через резисторы R1 R24. Делители на резисторах R28, R30 и R29, R31 обеспечивают разные напряжения U₁, U₂ срабатывания поро-

* Лукьянов Д. А. Схемотехника универсальных программаторов ПЗУ // Микропроцессорные средства и системы. — 1985. — № 3. — С. 84—88 (См. рис. 3 на с. 86, фрагмент схемы в правом нижнем углу.)

вых элементов, отличающиеся друг от друга на минимальный допуск установки выходных напряжений программатора, определенный в паспортах программируемых микросхем, например, U₁=9 В, U₂=10 В

Для контроля срабатывания ключа, подающего на выходную шину E0 на напряжение 5 В, используется резистор R27, обеспечивающих Лог 0 на шине E0, когда указанный ключ закрыт. Наличие напряжения питания 12 В контролируется с помощью стабилитронно-резисторной цепи R25, R32, V27, а наличие напряжения питания — 5 В — с помощью ключа на элементах R26, R30, R34, V32. Входы В0...В7 программатора, являющиеся выводами порта микроЭВМ, на которых при программировании каждый персональный блок формирует двоичный код для распознавания, использованы для передачи информации о работе блока контроля в микроЭВМ. Выходы программатора, сигналы которых имеют TTL-уровни (например, выходы «Разрешение», «Проверка», подключенные к выводам какого-либо регистра или порта микроЭВМ), соединены непосредственно со входами В6, В7 программатора. С помощью программ контроля сведения о неисправностях отображаются на дисплее программатора.

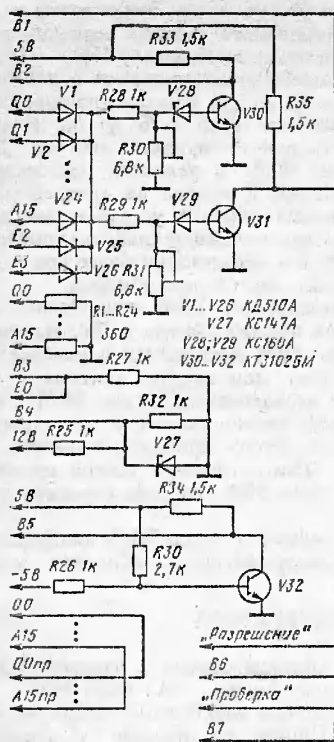


Схема блока самоконтроля

Программатор контролируют по алгоритму, реализуемому программой, занесенной в ПЗУ микроЭВМ. Вначале проверяют напряжения питания 12 В и — 5 В. Лог. 1 на входах В4, В5 свидетельствует о наличии этих напряжений, Лог. 0 — о их отсутствии. Создают различные логические состояния на ТТЛ-выходах «Разрешение» и «Контроль». Проверяют прохождение этих сигналов на входы В6, В7. При этом обнаруживают замыкания ТТЛ-выходов и их обрывы. При большем числе ТТЛ-выходов с открытым коллектором их можно соединять между собой и подключать к одному из входов В

программатора. В таком случае необходимо поочередно подавать сигналы на ТТЛ-выходы, каждый раз контролируя поступление сигнала на этот вход.

Затем проверяют работоспособность ключей. Для этого закрывают все ключи программатора, формируют напряжение U_1 на выходе программируемого источника напряжения Е1, подключенного в программаторе ко входам ключей. Проверяют наличие нулевого напряжения, свидетельствующего о закрытии ключей, на всех входах программатора Q0...A15_{пр}, к которым через БК подключены выходы ключей программатора. Поочередно по одному, чтобы не перегрузить источник Е1, открывают и закрывают указанные ключи. По состоянию входов Q0...A15_{пр} определяют ключ, который не открывается, т. е. напряжение на этом выходе не повышается; устраняют обнаруженные неисправности ключей, индицируемые на дисплее программатора, например: «Ключ 16 не закрывается».

Далее обнуляют программируемые источники напряжения Е2, Е3. Поочередно открывают и закрывают их выходные ключи. Появление при этом Лог. 0 на входе В1 свидетельствует о неисправности программируемого источника напряжения Е2 или Е3 (отсутствие возможности сброса его выходного напряжения). Формируют напряжение U_1 на выходе источника Е2. При открытом выходном ключе источника К2 вход В1 должен быть в состоянии Лог. 0, а вход В2 — в состоянии Лог. 1. Если на входах В1 и В2 Лог. 0 или Лог. 1, то источник Е2 не настроен. Закрывают выходной ключ источника Е2. Переход при этом шины В1 в Лог. 1 означает срабатывание этого ключа. То же выполняют с программируемым источником напряжения Е3. Устанавливают напряжение источника Е1, равное U_1 . Поочередно по одному открывают и закрывают ключи, соединенные с выходами Q0...A15. При каждом открытии ключа считывают информацию с В1 и В2. Вход В1 должен быть в состоянии Лог. 0, а вход В1 — Лог. 1, в противном

случае — неисправен соответствующий ключ (например, обрыв его шины питания). Если это имеет место при открытии любого ключа, значит, источник Е1 не настроен. Открывают и закрывают ключ Е0. Проверяют правильность логических состояний на входе В3.

Целесообразно программу контроля составлять таким образом, чтобы при неисправности цепи контроль повторялся бы до устранения этой неисправности оператором. То есть, организуется цикл контроля каждой цепи прожига, выход из которого происходит при формировании этой цепи правильного сигнала. Для настройки источника Е1 можно организовать дополнительный цикл проверки всех ключей. Циклы облегчают устранение неисправностей и настройку программируемых источников напряжения. Например, при наличии на дисплее программатора надписи «Е2 не настроен» вращают рукоятку потенциометра настройки источника Е2 программатора до изменения надписи, свидетельствующей о настройке источника Е2 и обнаружении очередной неисправности.

Выходы БК могут быть подключены и к другим входам программатора, например к некоторым входам Q0...A15_{пр}. Для контроля микроЭВМ программатора могут использоваться известные алгоритмы, применяемые для контроля вычислительных машин.

Таким образом, при использовании БК программатор сам указывает, какой его ключ или программируемый источник напряжения, или его другая цепь прожига неисправны или не настроены, что облегчает и ускоряет изготовление и ремонт универсальных микропроцессорных программаторов.

Описанная схема и алгоритм ее работы использованы в универсальном программаторе, производство которого освоено экспериментальным заводом Конотопского СПКТБ (245780, Сумская обл., Конотоп, ул. Семашко, 4, СПКТБ).

Телефон 3-27-77, Конотоп Сумской обл.

Статья поступила 18.03.88

М. С. Суханов Г. Е. Шевелев

ИНТЕРПРЕТАТОР БЛОК-СХЕМ

Язык блок-схем — одно из наглядных и простых средств отображения алгоритмов. Имеются языки высокого уровня, приближенные по своим изобразительным возможностям к блок-схемным представлениям. Авторы попытались исключить этап кодирования программы на языке высокого уровня. Разработанная система состоит из редактора и интерпретатора.

Редактор позволяет создавать и корректировать блок-схемы в оперативной памяти ЭВМ с отображением их на дисплее. При этом используется методология «человек в мире объектов»*. В данном случае «мир объектов» состоит из блоков алгоритма и соединяющих эти блоки связей. Пользователь может произвольно перемещаться в этом «мире», создавая или уничтожая блоки и связи,

а также редактировать текст внутри блоков. Каждая операция осуществляется нажатием одной, соответствующей ей, клавиши. Допустимые типы блоков: Начало, Конец, Ввод, Вывод, Вычислительный, Условный.

Интерпретатор выполняет созданную с помощью редактора блок-схему в отладочном (выполнение очередного блока) либо автоматическом (выполнение всей блок-схемы) режиме. При необходимости можно прервать выполнение блок-схемы и возвратиться к редактированию. В состав системы также входят школа пользователей и подсистема проведения произвольных лабораторных работ, связанных с использованием блок-схем (например, по курсам программирования, вычислительные методы и др.).

Интерпретатор блок-схем функционирует на микроЭВМ ДВК2М и мини-ЭВМ СМ4 в операционных системах типа РАФОС. Общий объем системы на диске 100 Кбайт Требуемый объем оперативной памяти 42 Кбайт

634004, Томск, ул. Советская, 84, Кибернетический центр при Томском политехническом институте, каф. прикладной математики: тел. 4-81-55

* Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации // Микропроцессорные средства и системы. — 1986. — № 1 — С. 44—46, 58.

УДК 681.3

В. С. Дубровин, В. Г. Коломыц, А. Д. Чернявский

ИНФОРМАЦИОННЫЙ ЦЕНТР В СОСТАВЕ ДВК И ВИДЕОМАГНИТОФОНА

Аппаратные средства

Видеомагнитофон (ВМ) «Электроника ВМ12» подключается к микроЭВМ «Электроника МС1201» посредством байтового параллельного интерфейса ИРПР [1]. Модуль сопряжения, обеспечивающий согласование сигналов ИРПР с сигналами управления ВМ (см. рисунок), связан с платой микроЭВМ через разъем ХТ2.

На элементах DD1.1, DD1.2, DD1.3, DD2.1, DD2.2, DD3.1 собран узел формирования сигналов управления интерфейса ИРПР ЗП-ИН, ЗП-ПН, СТР-ИН; на VT1 и DD4.1 — формирователь импульсов движения. Сигнал на

VT1 подается с датчика движения ВМ (блок датчика вращения А6). Для микроЭВМ нулевой разряд данных от источника информации интерфейса ИРПР ДО-ИН является датчиком состояния ленты ВМ. Расход ленты учитывается подсчетом импульсов движения.

Разряды данных приемника информации интерфейса ИРПР ДО-П...Д3-П соответствуют режимам работы ВМ: ДО-П — Стоп; Д1-П — Воспроизведение; Д2-П, Д3-П — Перемотка назад и вперед.

Сигнал Стоп формируется одновибратором DD4.2 при установке в единицу нулевого разряда данных ДО-П. Выходы 1, 5 оптопар V6, V7, V8, V9 подключаются параллельно контактам, отвечающим за соответствующий режим работы (блок коммутации А9 ВМ). Такое подключение позволяет управлять ВМ вручную и от ДВК.

Программные средства

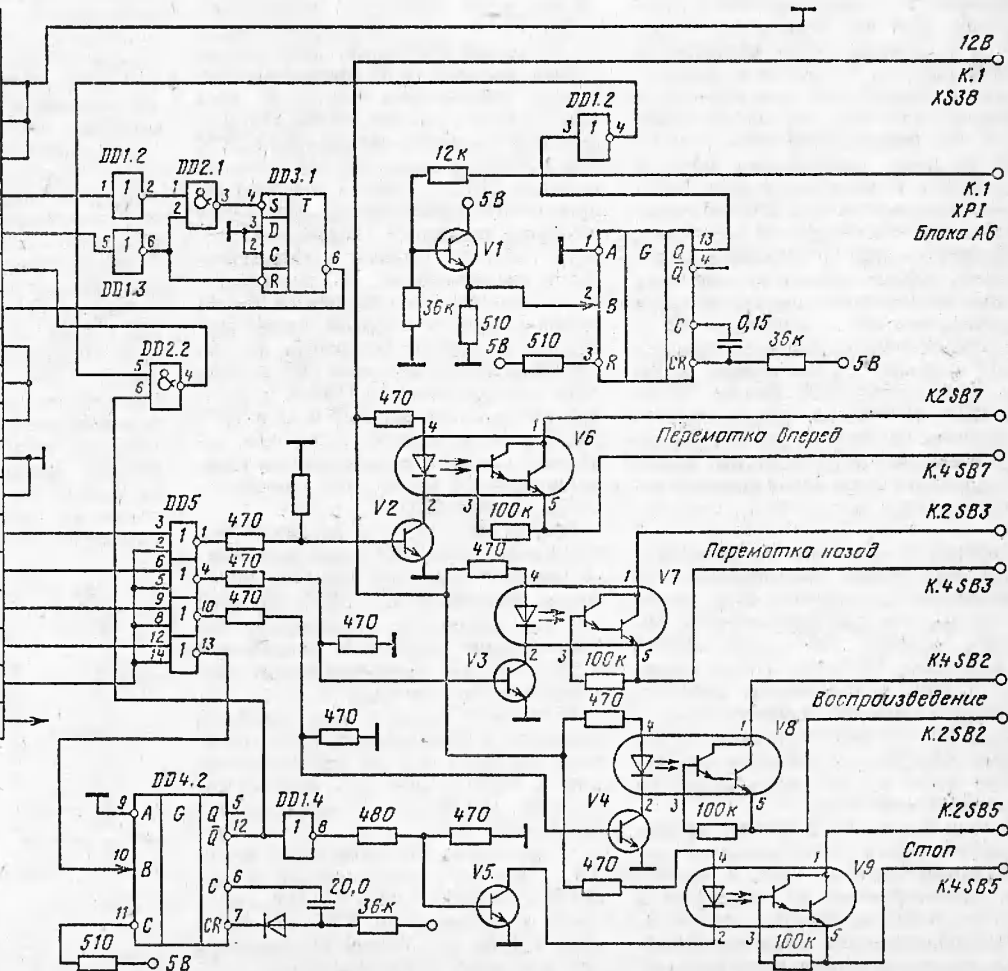
Для управления ВМ в пространстве адресов страницы ввода-вывода задействованы четыре регистра с адресами

„Электроника МС1201.01“

ХТ2

Название сигнала	Контакт
ГИ-ИН	19
ГП-ПН	37
Ошиб. 1В	2
ЗП-ИН	17
ДО-ИН	12
ЗП-ПН	36
СТР-ИН	35
Флаг В	16
Требов. В	44
Треб. перед.	13
Заверш. 1В	7
Г/Д	29
Д3-П	5
Д2-П	8
Д1-П	60
ДО-П	59
СТР-ПН	15
5В	57

DD1 — К155ЛН1
DD2 — К155ЛАЗ
DD3 — К155ТМ2
DD4 — К155АГ3
DD5 — К155ЛЕ1
V1...V5 — КТ315В
V6...V9 — А0Т110А



„Электроника ВМ12“

Принципальная схема модуля сопряжения микроЭВМ «Электроника МС1201.01» с видеомагнитофоном «Электроника ВМ12»

177270 — регистр состояния приемника (RS1), 177272 — регистр данных приемника (RD1), 177274 — регистр состояния передатчика (RS0), 177276 — регистр данных передатчика (PD0). В простейшем варианте обмен данными осуществляется в режиме готовности устройства. Перед приемом-пересылкой данных опрашиваются регистры состояния до появления единицы в седьмом бите регистра. RD1 необходим для подсчета импульсов, поступающих из ВМ, число которых пропорционально числу оборотов вала приемной кассеты.

Режимы работы ВМ: остановка двигателя, запуск на воспроизведение, перемотка назад и вперед — управляются через RDO, являющийся одновременно регистром команд. Код команды заносится в первые четыре бита регистра.

Программы управления ВМ включают процедуры на языке Паскаль:

УДК 681.325.5:181.4

Ю. Н. Врублевский, Ю. П. Осипенко

ТВ-ПРИЕМНИКИ — В КАЧЕСТВЕ ВИДЕОМОНИТОРОВ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

Непрофессиональным пользователям ПЭВМ предлагаются три варианта сопряжения своего компьютера с телевизионным приемником (ТВП)

Вариант 1 самый простой и доступный. Для его реализации потребуется соединить гнездо видеовыхода ПЭВМ кабелем, входящим в комплект поставки. Однако этот путь недоступен подавляющему большинству пользователей, ибо большинство отечественных ТВП не имеет необходимых внешних разъемов, а те немногие модели, которые располагают входом ВМ для соединения с видеомагнитофоном (например, ТВП «Электроника Ц-430/432» и ему подобные), требуют подачи на этот вход помимо видеoinформации еще и внешнего управляющего напряжения 12 В.

Использование зарубежных моделей ТВП, например DE100 фирмы SABA или PAL COLOR1120 фирмы TELEFUNKEN (ФРГ), не может служить выходом из положения, так как наряду с их очевидной недоступностью возникает проблема сопряжения цветовой видеoinформации в различных стандартах.

Вариант 2 — достаточно простой и наиболее доступный. Пользователь либо приобретает стандартный блок сопряжения видеовыхода компьютера с антенным гнездом ТВП (для ПЭВМ «Электроника БК-0010» — это входящий с 1988 г. в комплект поставки ВЧ-модуль, который переносит видеoinформацию на частотах четвертого или пятого ТВ-каналов), либо сам изготавливает такой же или аналогичный по характеристикам блок.

В этом случае он сохраняет дружеские отношения с министерствами, производящими бытовые ТВП, и Минбытом, ремонтирующим их, но терпит в качестве воспроизводимой на телеэкране видеoinформации из-за «затягивания» фронтов видеопульсов в радио тракте, а также «наживает врагов» в лице своих соседей и Государственной

EXCOM (YAR NCOUNT, KODKOM:INTEGER),

где NCOUNT — число сосчитанных импульсов во время выполнения команды с кодом KODKOM.

На международной выставке в Лейпциге демонстрировалась информационная система выбора видеосюжетов с рекламой экспонатов, представленных на выставках.

103482, Москва, НИИИЦ; тел. 531-14 03, 532-86-90

ЛИТЕРАТУРА

1. Одноплатные микроЭВМ ряда «Электроника МС 1201» / Дшхунян В. Л., Борщенко Ю. И., Отрохов Ю. Л., Шишарин С. А. // Микропроцессорные средства и системы. — 1984. — № 2. — С. 8—13.
2. Мини- и микроЭВМ семейства «Электроника» / Толстых Б. Л., Талов И. Л., Цвинский В. Г. и др. — М.: Радио и связь, 1987.

Статья поступила 8.04.88

инспекции электросвязи Минсвязи СССР (при использовании самодельных ВЧ-модулей), призванной следить за уровнем издаваемых устройствами (в том числе бытовыми) радиопомех.

Вариант 3 наиболее эффективный, но не самый доступный. Для пользователя, желающего по каким-либо причинам использовать антенный вход ТВП, остается вариант доработки имеющегося в его распоряжении ТВП в УОИ или в УОИ, но с сохранением основной функции ТВП — приема сигналов вещательного телевидения. В этом случае возможна подача на вход видеосуилителя ТВП как полного видеосигнала, так и раздельных R-, G-, B-сигналов.

Из рассмотренных вариантов сопряжения следует, что первый, самый очевидный из них, не реализован до сих пор лишь из-за отсутствия ТВП со входами для подключения ПЭВМ, а стандарт, предусматривающий ввод в ТВП информации от видеомагнитофона, морально устарел и требования его удовлетворяются в крайне незначительном числе моделей ТВП.

Отсутствие стандарта на характеристики видеовыхода ТВП также сдерживает разработку единых норм на параметры видеовыходов ПЭВМ. По этой причине авторы не анализируют все многообразие вариантов сопряжений ПЭВМ — ТВП, а рассматривают наиболее доступные модели.

В качестве примера даны варианты сопряжения видеовыхода ПК «Электроника БК-0010» [3] (и его модификаций) с переносными ТВП типа «Электроника Ц-430/432» и «Электроника Ц-431/433».

«Электроника БК-0010» имеет выход ТВ, к которому подключается кабель ВЧ-модуля либо кабель ТВ для соединения с видеовыходом ТВП по варианту 1 (рис. 1). Можно использовать для этих целей и антенный вход.

В варианте с переносными ТВП типа «Электроника 404Д» доработка сво-

дится к соединению сигнальной жилы гнезда ДМВ, используемого в качестве видеовыхода, со входом видеосуилителя ТВП через разделительный конденсатор порядка 1 мкФ.

Иначе решается задача согласования выхода БК-0010 с ТВП цветного изображения: по видеотракту путем подачи на вход ВМ полного видеосигнала положительной полярности или по ВЧ, подавая тот же видеосигнал через ВЧ-модуль на антенный вход ТВП. Однако при этом возникает паразитные цветные окантовки контуров изображения, особенно сильно проявляющиеся по краям экрана. Это связано со спецификой серийно выпускаемых трехлучевых кинескопов и системой статического и динамического сведения лучей, не обеспечивающих совмещение трех электронных лучей в одной триаде на любой точке плоскости люминофорного экрана.

Так, например, норма на расхождение лучков на экране ТВП «Электроника Ц-430/432» составляет в центре экрана (внутри круга диаметром 100 мм) не более 0,6 мм, а на периферии экрана 1,0 мм. Минимальная длительность информационных импульсов БК-0010 равна 0,3 мкс, что занимает на экране расстояние 1,1 мм. Выведенное на экран изображение от информационных импульсов такой дли-

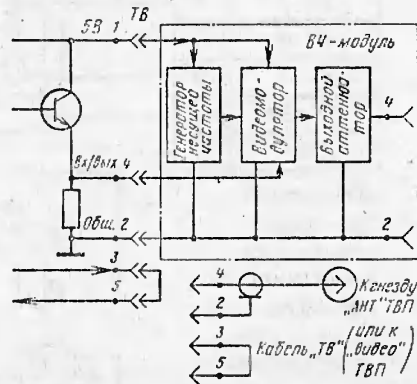


Рис. 1. Схема сопряжения видеовыхода ПЭВМ «Электроника БК-0010» с ТВП

тельности трудно различимо, особенно на краях экрана, из-за расхождения пучков даже в допустимых для цветного ТВП пределах. Норма на расхождение электронных лучей в цветных ТВП с большим экраном еще выше.

В новых моделях «Электроника БК-0010Ш», «Электроника БК-0011» и других выведен дополнительный разъем ЦТВ (рис. 2) для информационных сигналов трех основных цветов: красного R, зеленого G, синего B — и синхросмеси (строчных и полевых синхронимпульсов).

В большинстве эксплуатируемых ТВП цветного изображения, в том числе и ТВП «Электроника Ц-430/432», блок видеоусилителей (ВВУ) обеспечивает привязку сигналов яркости (цветности); матрицирование, т. е. получение первичных сигналов цветности R, G, B; усиление сигналов R, G, B видеоусилителями с последующей подачей сигналов на соответствующие катоды кинескопа; гашение по строкам и кадру; выполнение настроечных операций (статический и динамический баланс блоча) и оперативных регулировок яркости, контрастности, а часто и насыщенности.

В последних моделях ТВП («Электроника Ц-431/413») блоки цветности и видеоусилителей объединены в так называемый блок цвета (БЦ). При этом канал яркости, выполняющий роль ВВУ, собран на двух многофункциональных микросхемах К174УК1, К174АФ5 или на одной из ИМС К174ХА17, КР0021ХА4 (зарубежные аналоги ТДА3501, ТДА3562А). Микросхемы К174ХА17, КР1021Х44 имеют специальные выводы для непосредственной подачи на них сигналов цветности R, G, B положительной полярности размером 1 В при нагрузке 75 Ом, а также вывод для коммутации видеосигнала, т. е. отключения части тракта ВВУ, мешающей передаче номинальных сигналов R, G, B от ПЭВМ.

В выпускаемых отечественной промышленностью ТВП цветного изображения нет специального разъема, с помощью которого можно коммутировать поступающие от ПЭВМ информационные R-, G-, B-видеосигналы и синхросмесь, как нет и внешнего переключателя, позволяющего отключать радиотракт на время прохождения информационных сигналов. Поэтому при сопряжении ЦТВ-выхода с ТВП цветного изображения необходимо дорабатывать ТВП, а иногда и ПЭВМ. Наиболее приемлемым представляется использование разъема ВМ в качестве коммутирующего разъема для стыковки. От разъема ЦТВ (конт. 1) БК-0011 синхросмесь по кабелю ЦТВ поступает на вход разъема ВМ (конт. 2, «видео»). Корпус БК-0011 (конт. 2 разъема ЦТВ) с помощью того же кабеля соединяется с общей шиной ТВП (конт. 3 разъема ВМ). Подобная коммутация необходима для синхронизации строчной и кадровой разверток любого выбранного типа ТВП синхросмесью ПЭВМ. Размах

синхросмеси на ВМ-входе «видео» должен быть в пределах 0,7...1,4 В при входном сопротивлении 75 Ом. При отсутствии разъема ВМ синхросмесь от ПЭВМ должна поступать в селектор синхронимпульсов ТВП, который необходимо согласовать с открытым выходом ПЭВМ.

Исходя из высказанных предпосылок предлагается доработка ТВП цветного изображения.

ТВП «Электроника Ц-430/432»

Схема доработки в блоках ТВП показана на рис. 3. Если после доработки подключить выход ЦТВ БК-0011 с помощью кабеля ЦТВ к ТВП (вход ВМ) и замкнуть перемычкой центральные жилы разъемов X1 и X2, то ТВП можно эксплуатировать в качестве видеомонитора.

Выходные сопротивления ПЭВМ не должны шунтировать по постоянному и переменному току входы блока видеоусилителей (плата ВВУ). Размах информационных сигналов R, G, B, поступающих от ПЭВМ через разъем ВМ на ВВУ ТВП, должен быть по-

рядка 0,5...1,0 В. Оптимальные значения сопротивлений согласующих резисторов $R_{доп1}$, $R_{доп2}$, $R_{доп3}$ (см. рис. 2) в этом случае составляет 3,3...4,3 кОм.

Отключать радиотракт ТВП, работающего в режиме монитора, можно путем замыкания штырей разъема X18, расположенного на кроссплате (КП). Для того чтобы переключить ТВП в режим работы УОИ без вскрытия корпуса, следует контакты X18 соединить жгутом с центральными жилами разъемов X1, X2 антенного блока (АБ), предварительно отсоединив и изолировав концы, подходящие к X1, X2. Замыкание центральных жил разъемов X1 и X2 в этом случае можно выполнить с помощью внешней перемычки. Таким образом, подключение внешней перемычки, замыкающей центральные жилы разъемов X1 и X2, переводит ТВП в режим УОИ, а отключение перемычки — в режим работы.

К недостаткам подобных переделок ТВП «Электроника Ц-430/432» следует отнести то обстоятельство, что при ра-

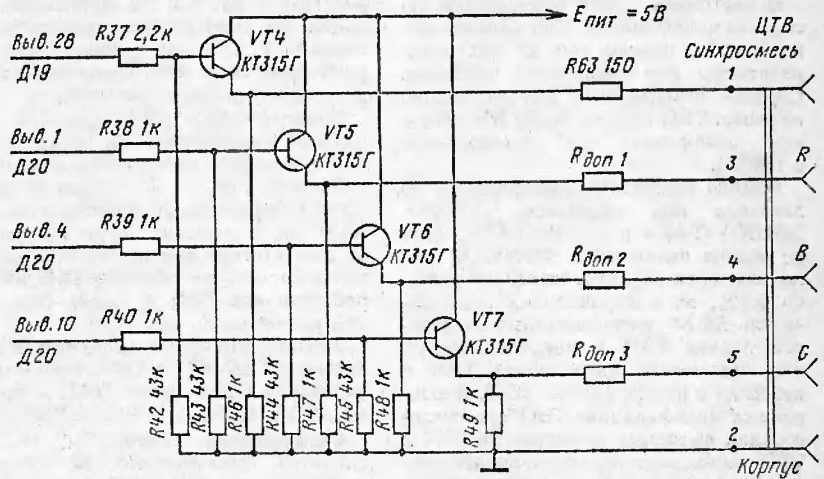


Рис. 2. Схема выходных каскадов ПЭВМ (разъем ЦТВ)

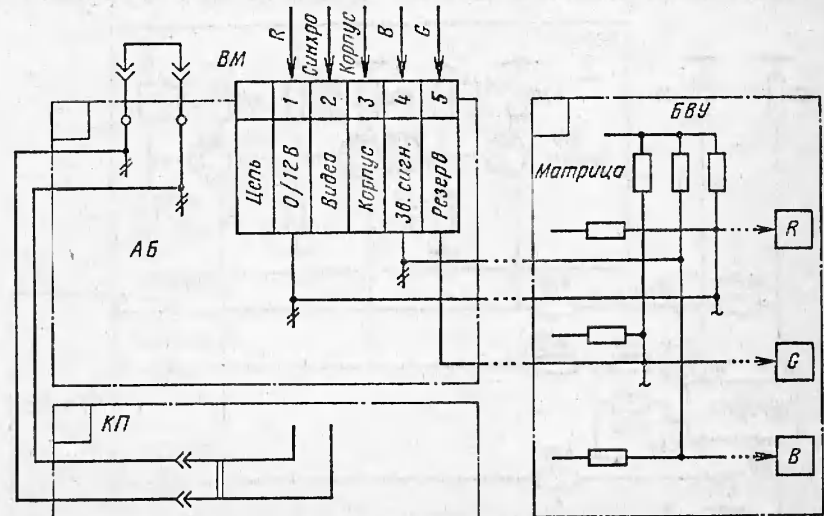


Рис. 3. Схема доработки ТВП «Электроника Ц-430/432» до монитора

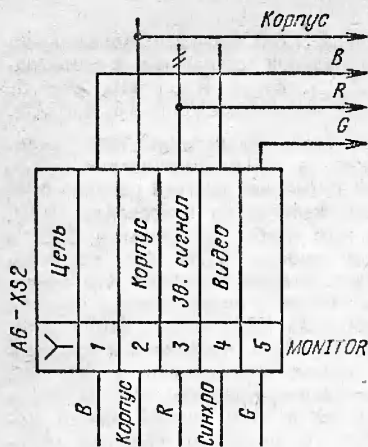


Рис. 4. Переделка монтажа розетки ТВП «VIDEOSOUND C-431»

боте аппарата в качестве телевизора нельзя принимать ТВ-сигналы на телекопическую антенну или работать от видеоманитфона без вскрытия корпуса и переделки ТВП заново.

ТВП «Электроника Ц-431/433»

В настоящее время выпускается несколько модификаций ТВП «Электроника Ц-431», причем три из них предназначены для экспортных поставок. Серийно выпускаемый для внутреннего рынка ТВП пока не имеет ВМ-входа, что осложняет его согласование с ПЭВМ.

Однако экспортные модификации телевизора под названием «VIDEOSOUND C-431» и «EUROMATIC 431» не только имеют вход MONITOR для подключения видеоманитфона (разъем XS2), но и специальный переключатель А6-S1, расположенный на верхней планке ТВП и предназначенный для отключения радиотракта ТВП и перевода в режим работы УОИ. В выбранных модификациях ТВП при согласовании выходных каскадов БК-0011 с ТВП необходимо перемонтировать концы, подводимые к розетке для подключения монитора (XS2), например в со-

ответствии с рис. 4. Номинальный размах R-, G-, B-сигналов положительной полярности, снимаемых с выходных каскадов БК-0011, должен быть равен 1 В на внешней нагрузке 75 Ом.

Электрическая схема согласования выходов первичных R-, G-, B-сигналов БК-0011 с каналом яркости (БВУ), разработанным на ИТС К174УК1, К174АФ5, показана на рис. 5. Канал яркости выполняется на основе типового включения ИМС, поэтому предлагаемая электрическая схема согласования справедлива для любых БВУ, собранных на указанных ИМС. Отличие от типового включения состоит лишь в том, что в цепи цветоразностных сигналов последовательно включаются резисторы 470 Ом, которые являются коллекторной нагрузкой соответствующих инвертирующих каскадов, собранных на транзисторах VT1...VT3. Все три инвертора полностью идентичны. Резисторы R2, R4, R6 служат для подбора требуемых размахов первичных сигналов цветности на выходах 2, 4, 6 К174АФ5. Номинальные значения сопротивлений резисторов R1, R3, R5 выбранных типовых ВЧ транзисторов средней мощности VT1...VT3, как правило, гарантируют требуемую АЧХ сигналов, а следовательно, и цветовую четкость.

Конструктивно резисторы 470 Ом удобнее расположить на плате блока цвета. Инвертор монтируется на отдельной плате, а R-, G-, B-сигналы и общий провод от розетки (например, типа А6-ХS2) через плоский жгут поступают на соответствующие контакты. Доработанный подобным образом ТВП может работать как ТВП и, после отключения радиотракта, как УОИ.

Значительно проще выполняется согласование БК-0011 с ТВП, если видеосигналы блока цвета (БЦ) собраны на ИМС К174ХА17 или КР1021ХА4.

Оптимальный размах R-, G-, B-сигналов, поступающих на соответствующие контакты ИМС, должен быть порядка 0,3...0,6 В на нагрузке 75 Ом.

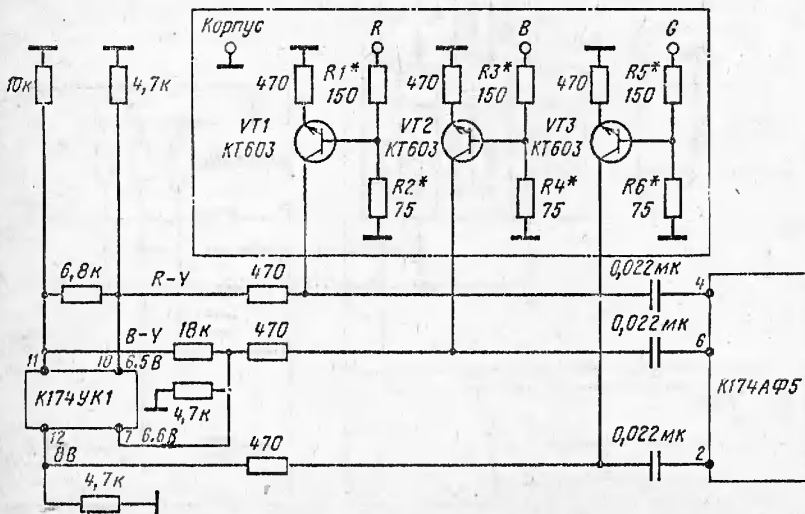


Рис. 5. Схема согласования ПЭВМ с каналом яркости

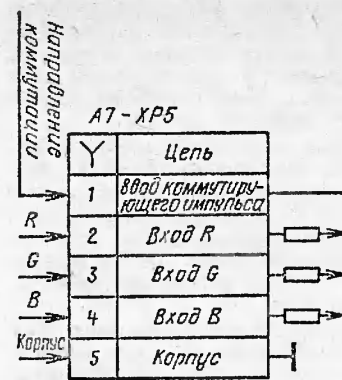


Рис. 6. Схема ввода сигналов R, G, В в блок цвета, имеющий специальные входы.

Поэтому обычно на плате БЦ предусматривают дополнительные резисторы порядка 100 Ом, последовательно включенные в цепь каждого сигнала (рис. 6).

Информационные R-, G-, B-сигналы и синхросмесь от БК-0011 по кабелю ЦТВ поступают на разъем А6-ХS2 ТВП «VIDEOSOUND C-431» и через розетку XS2, контакты которой перемонтированы в соответствии с рис. 4, по плоскому жгуту — на разъем, расположенный на плате ВЦ (или БВУ). При этом ТВП в режим УОИ переключается с помощью специального переключателя режимов работы.

Радиотракт при использовании К174ХА17 или КР1021ХА4 можно освободить, если подать напряжение 1...12 В на конт. 1 разъема А7-ХР5 (рис. 6). Напряжение коммутации можно подать через жгут на конт. 1 разъема ХР5 любым путем в зависимости от типа ТВП. В частности, если в ТВП цветного изображения входная розетка шестиконтактная, то один контакт можно использовать для подачи напряжения записания радиотракта.

Регулировка ТВП в режиме работы от ПЭВМ практически не требуется, так как размах R-, G-, B-информационных сигналов, поступающих от ПЭВМ, равны размахом R-, G-, B-видеосигналов (режим приема ТВ-сигналов) в точках подсоединения к контактам в БВУ или номинальному значению для К174ХА17, КР1021ХА4. Это условие выполняется подборкой резисторов (см. рис. 2,5). В принципе, небольшую подрегулировку можно проводить в каждом из каналов БВУ, изменяя уровни и размахы выходных сигналов и проводя визуальный контроль по изображению. При этом необходимо добиваться (при сохранении статического и динамического баланса белого) наиболее четкого изображения на экране как в режиме приема ТВ-сигналов, так и при работе от ПЭВМ.

Телефон 2-90-17 доб. 3-19, 3-96, Павловский Посад Моск. Обл., Врублевский Ю. И.; 597-85-32, Ленинград, Осипенко Ю. П.

А. А. Трушин

ПРОСТЫЕ СРЕДСТВА ВВОДА-ВЫВОДА ДЛЯ УПРАВЛЯЮЩИХ МИКРОПРОЦЕССОРОВ

К числу самых простых средств ввода-вывода, предназначенных для устройств управления на основе МП, можно отнести цифровую клавиатуру на 12...16 клавиш и цифровой индикатор на 6...10 разрядов. Схемы сопряжения получаются несложными — примерно 5...10 микросхем средней степени интеграции или одна БИС КР580ВВ79.

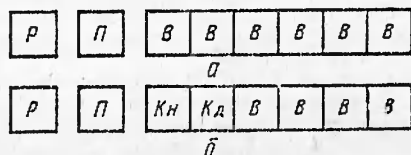
В качестве примера рассмотрим клавиатуру из 12 клавиш (цифры 0...9, С — сброс, В — ввод) и 8-разрядный семи-сегментный индикатор. С их помощью выполняются запуск и останов технологических режимов, набор (ввод) данных, просмотр (индикация) ранее введенных данных или текущих технологических параметров, тестирование основных узлов блока.

На индикатор в закодированном виде выводятся следующие данные: режим (Р), параметр (П), величина (В) — основные; и дополнительные: канал (Кн), кадр (Кд). Каждый из параметров кодируется одной или несколькими цифрами и на индикаторе занимает определенные разряды (см. рисунок).

Разряд Р обозначает основную функцию, выполняемую в данный момент блоком: Пуск, Останов, Сброс, Набор, Индикация, Регулирование, Измерение, Тест. Таких режимов, как правило, не более 10. В одном или двух разрядах П закодированы названия параметров для ввода или индикации: температура, скорость ее изменения, время, давление, толщина, коэффициенты регуляторов и т. п. В — численное значение какого-либо параметра или вспомогательный код. Если разрядности величин разных параметров неодинаковы, то неиспользуемые разряды можно не высвечивать. Кодом может быть, например, ответ процессора о результате тестирования какого-либо узла или индикация состояния исполнительного органа (закрыто-открыто или др.). Кн — номер канала, если блок управляет одновременно несколькими одинаковыми каналами. Д — совокупность параметров для управления объектом в течение определенного времени. Например, в случае необходимости регуляции температуры по сложному во времени профилю Кд может включать в себя параметры температуры, скорости ее изменения, коэффициенты закона регулирования для отдельной линейной части профиля.

Для любого блока при распределении разрядов индикатора по назначению и составлении процедур общения с клавиатурой и индикатором следует обеспечивать в любой момент времени однозначность индикации и режима работы блока, чтобы при любых манипуляциях с клавиатурой оператор знал в каком состоянии (режиме) находится блок. Ввод объектов Р, П, В... осуществляется нажатием любой последовательности цифровых клавиш 0...9, в конце которой обязательно нажимается клавиша В. Клавиша С используется для вывода на индикатор дежурной информации.

Порядок ввода объектов определяется алгоритмом работы. Часть объектов может вводиться в любой последовательности, например выбор режима (Р). Для других (например для параметров одного кадра) заранее установлена определенная последовательность чередования. Алгоритм



Варианты назначения разрядов индикатора

допускает многократное нажатие клавиши В без ввода цифр: соответствующие объекты будут выведены на индикатор в заданной последовательности. Это упрощает процедуру набора и просмотра при большом объеме данных. В программе обработки вводимых объектов желательно предусмотреть возможность ввода любого числа цифр до нажатия клавиши В, с тем чтобы сдвигом влево можно было убрать неправильно набранные цифры. После нажатия В набранные цифры будут обрабатываться программой МП.

При необходимости одновременной индикации нескольких параметров можно объединить несколько индикаторов, но целесообразно использовать более трех, четырех. Для удобства работы с клавиатурой и индикатором отдельные разряды индикаторов можно разнести в пространстве, поместив под ними соответствующие надписи, а на передней панели блока — краткую табличку с расшифровкой закодированных объектов.

На основе вышеизложенных принципов разработаны и изготовлены образцы комплектов аппаратуры для установок молекулярно-лучевой эпитаксии (пять типов блоков управления).

426054, Ижевск, Фруктовая, 41, кв. 17, Трушину А. А.

Сообщение поступило 12.10.88

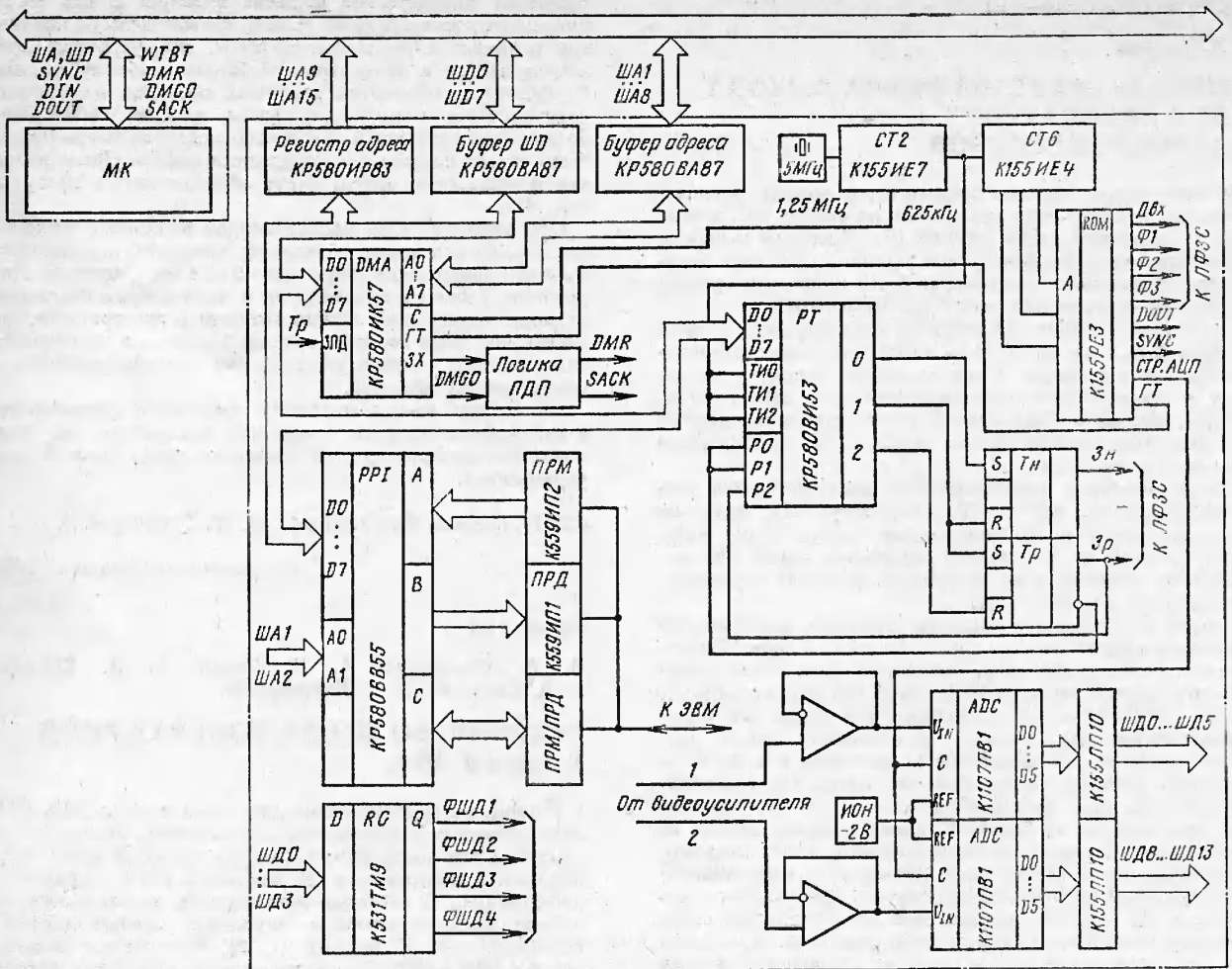
УДК 681.3.049

В. А. Филипович, А. К. Есман, А. А. Ермилов,
А. А. Савченко, В. Н. Дубровский

УСТРОЙСТВО ВВОДА ИЗОБРАЖЕНИЙ В МИНИ-ЭВМ

Устройство предназначено для ввода в мини-ЭВМ СМ4 полутонных и факсимильных изображений. Исходным документом для ввода может служить страница книги, журнала и т. д., содержащая как текстовую, так и графическую информацию. В качестве фотодатчиков используются линейные фотоприемники с зарядовой связью (ЛФЗС) К1200ЦЛ1 или К1200ЦЛ2 [1, 2], содержащие соответственно 1000 и 2000 фоточувствительных элементов. Устройство представляет собой настольный прибор, состоящий из подставки, на которой размещается исходный документ, и вертикальной штанги с блоком сканирования изображения. Блок сканирования перемещается по всей длине вертикальной штанги, что позволяет считывать изображения различного формата и проводить макросъемку объектов. Максимальный формат исходного документа составляет 210×300 мм. Блок сканирования включает в себя объектив «Гелиос-44», подвижную каретку, на которой размещены ЛФЗС и шаговый двигатель с микровинтом. При вращении оси шагового двигателя ЛФЗС перемещается вдоль плоскости построения объективом изображения, осуществляя построчное сканирование.

Устройство управляется 16-разрядным одноплатным микроконтроллером (МК) [3], выполненным на базе МП К1801ВМ1. Плата размерами 155×200 мм² размещается на подставке. МК выполняет связь с мини-ЭВМ, предварительную обработку информации об изображении, управляет всеми функциональными блоками устройства. Объем ОЗУ МК составляет 8 Кбайт, ПЗУ — 4 Кбайт. На отдельной плате, также размещенной на подставке, собраны блок управления ЛФЗС, узел ПДП, схема АЦП видеосигнала, интерфейс ИРПР, схема управления шаговым двигателем. Работа всех элементов устройства синхронизируется блоком управления ЛФЗС на базе таймера КР580ВИ53 (см. рисунок). Кроме таймера данная схема включает кварцевый генератор 5 МГц, счетчик-делитель частоты К155ИЕ7, счетчик К155ИЕ4, ПЗУ К155РЕ3, триггеры затворов накопления и разрешения. Счетчик К155ИЕ7 вырабатывает импульсные последовательности с частотами 1,25 МГц и 625 кГц, которые поступают соответственно на



Функциональная схема устройства

БИС ПДП и микросхему К155ИЕ4, адресуемую ПЗУ. ПЗУ формирует трехфазные импульсы опроса ЛФЗС, импульсы стробирования АЦП, сигнал готовности для БИС ПДП и сигналы SYNC и DOUT для записи считанной с ЛФЗС информации в ОЗУ МП в цикле ПДП.

Нулевой канал таймера управляет длительностью импульса на затворе накопления ЛФЗС, что позволяет программно изменять чувствительность устройства в зависимости от степени освещенности исходного документа. Первый и второй каналы таймера формируют требуемый импульсный сигнал на затворе разрешения ЛФЗС, а также определяют необходимую длительность трехфазных последовательностей импульсов опроса [1, 2]. Импульсы опроса с выходов ПЗУ и триггеров $T_{И1}$ и $T_{И2}$ поступают далее в блок сканирования, где преобразуются по уровню и подаются на ЛФЗС. В непосредственной близости от ЛФЗС расположены усилители видеосигнала КР544УД2, информация с выходов которых подается на входы интегральных АЦП К1107ПВ1. При использовании ЛФЗС К1200ЦЛ1 работает только один АЦП, а при ЛФЗС К1200ЦЛ2 — оба. АЦП квантуют информацию с выходов ЛФЗС на 64 уровня. Цифровая информация далее через буферы К155ЛП10 подается на шину данных МК.

Информация с выходов АЦП записывается в ОЗУ МК в режиме ПДП под управлением БИС КР580ИМ57. Для синхронизации записи данных с выходов АЦП в ОЗУ вместо сигналов ЧТЗУ и ЗПВВ БИС ПДП используются сигналы SYNC и DOUT, формируемые ПЗУ К155РЕ3, что

связано с особенностями работы ЛФЗС. Сигнал ЧТЗУ используется для управления буферами данных К155ЛП10. Информация со всего ЛФЗС записывается в ОЗУ в течение одного цикла ПДП. После завершения цикла ПДП МК переводит каретку на следующую строку изображения и передает считанные данные в мини-ЭВМ. Мини-ЭВМ связана с МК по двум независимым каналам: интерфейсу в стандарте ИРПР, выполненному на базе БИС КР580ВВ55, и каналу ПДП. Предпочтительнее использовать более быстродействующий канал ПДП, реализованный с помощью модуля в стандарте КАМАК.

Шаговый двигатель (ШД) управляется программно: четырехфазные импульсы управления ШД снимаются с выходов регистра К531ТМ9 и поступают в блок сканирования, где усиливаются по мощности и амплитуде и подаются на обмотки ШД.

МК содержит аппаратные и программные средства, позволяющие использовать для поиска неисправностей метод сигнатурного анализа. Программа сигнатурного анализа хранится в ПЗУ и запускается изменением адреса запуска МК при включении питания с помощью переключателя, установленного на плате МК.

При использовании ЛФЗС К1200ЦЛ2 устройство позволяет вводить исходное изображение с разрешением по строке порядка 10 точек/мм для формата 210×300 мм, а при ЛФЗС К1200ЦЛ1 — 5 точек/мм. Прибор К1200ЦЛ2 обладает лучшей чувствительностью в зеленой и синей областях спектра, что немаловажно при вводе цветных

изображений. Разрешающая способность определяется конструкцией механического узла сканирования и составляет 10 точек/мм. Яркость каждой точки исходного изображения квантуется с помощью АЦП на 64 уровня шкалы серого цвета. Время считывания одной строки изображения составляет порядка 10 мс, время перехода с одной строки на другую определяется характеристиками ЦД и составляет 30 мс. С увеличением частоты опроса ЛФЗС время считывания одной строки изображения может быть уменьшено до 2...4 мс, что, однако, потребует применения более мощных источников для освещения документа вследствие снижения чувствительности ЛФЗС.

В настоящее время ведутся работы по созданию на базе описанного устройства сканера для ЭВМ ЕС1841.

Телефон 65-61-12, Минск

УДК 681.327.12

А. Е. Перепечкин

КОНТРОЛЛЕР БЫТОВОГО МАГНИТОФОНА НА ОСНОВЕ ОЭВМ КМ1816ВЕ48

Контроллер позволяет использовать бытовые средства магнитной записи в качестве внешнего устройства долговременного хранения данных для любых ЭВМ, имеющих

1. Камлаков И. Д., Кленов В. П., Костюков Е. В. // Электронная промышленность.— 1982.— № 7.— С. 7—9.
2. Василевская Л. М., Костюков Е. В., Павлова Э. В. // Электронная промышленность.— 1982.— № 7.— С. 10—13.
3. Пилипович В. А., Есман А. К., Ермилов А. А., Савченко А. А. // Микропроцессорные средства и системы.— 1987.— № 1.— С. 46.
4. Пилипович В. А., Есман А. К., Ермилов А. А., Савченко А. А. // Приборы и системы управления.— 1987.— № 3.— С. 30.

Статья поступила 21.01.88

в своем составе модуль, реализующий протокол обмена по интерфейсу ИРПП [1]. На плате контроллера размерами 100×120 мм кроме ОЭВМ КМ1816ВЕ48 размещены шесть микросхем средней степени интеграции (рис. 1).

Контроллер работает в режимах записи и чтения. В режиме записи байт информации, поступающий из интерфейсно-

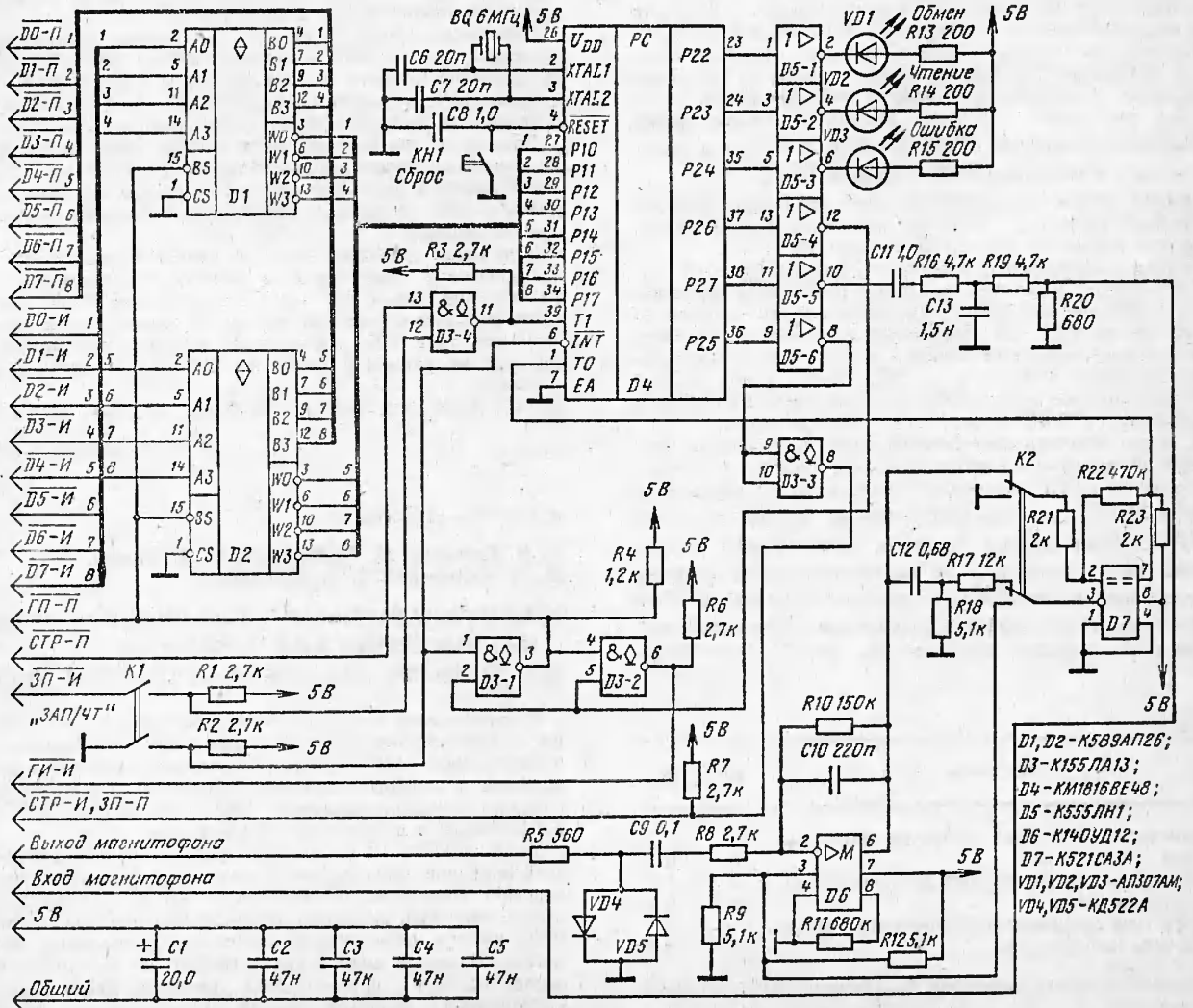


Рис. 1. Принципиальная электрическая схема контроллера бытового магнитофона

го модуля ЭВМ, преобразуется в сигнал последовательного кода с добавлением одного стартового (Лог. 0), двух стоповых (Лог. 1) битов и разряда четности. При этом Лог. 1 соответствует один период частоты 2082 Гц, а Лог. 0 — один период частоты 1041 Гц. Последовательность байтов, принятая контроллером из ЭВМ без временной задержки, оказывается разделенной на магнитной ленте лишь двумя периодами частоты 2084 Гц, соответствующими двум стоповым битам. Если очередной байт информации из ЭВМ по какой-либо причине не поступает, то контроллер после двух стоповых битов предыдущего байта продолжает формировать сигнал той же частоты. Это дает возможность создавать с помощью программных задержек межблочные и межфайловые промежутки на ленте.

В режиме чтения контроллер анализирует сигнал, поступающий с линейного выхода магнитофона, и выполняет обратное преобразование информации в параллельный код. В начале режима происходит поиск межблочного или межфайлового промежутка, равного 300 периодам частоты 2084 Гц. При этом по каждому из 256 периодов данной частоты определяется отклонение скорости движения ленты от скорости записи. Если оно превышает $\pm 33\% \dots +19\%$ от номинального значения 2084 Гц, контроллер переходит из режима чтения в состояние «ошибка» (см. таблицу); в противном случае, выполняются операции идентификации байта, обнаружения ошибок и передачи данных в интерфейсный блок ЭВМ.

Данное состояние контроллера сопровождается миганием светодиода D3 с частотой в несколько герц. Характер ошибки определяется комбинацией сигналов, поступающих одновременно по шинам данных в интерфейсный блок ЭВМ и на светодиоды D1, D2. Выход контроллера из состояния «ошибка» происходит при нажатии кнопки «сброс».

Для управления передачей данных в режиме записи используются сигналы ГП-П, ЗП-П и СТР-П, а в режиме чтения — сигналы ГП-И, СТР-И и ЗП-И.

Канал записи на магнитную ленту представляет собой буферный усилитель с фильтром верхних и нижних частот. Уровень выходного сигнала составляет 100 мВ.

Канал воспроизведения включает предварительный усилитель-фильтр с защитой по входу и компаратор напряжения. Изменение полярности входного сигнала с помощью переключателя К2 дает возможность применять магнитофоны с четным и нечетным числом инвертирующих усилительных каскадов. Амплитуда входного сигнала для данной схемы канала воспроизведения может колебаться в широком диапазоне: от 20 мВ до нескольких десятков вольт.

Следует отметить одну важную особенность работы контроллера в режиме чтения. Согласно протоколу обмена интерфейса ИРПР контроллер, получив сигнал запроса на прием ЗП-И, устанавливает данные и выдает строб СТР-И. Если данные приняты, интерфейсный модуль ЭВМ снимает сигнал ЗП-И, на что контроллер отвечает переводом сигнала СТР-И в пассивное состояние. При этом время такого ответа, обусловленное инертностью программной работы контроллера, может составлять

ИДМП C000, Coff

C000	35	86	4D	BE	FD	BD	FA	FE	62	55	9A	83	14	3E	46	08
C010	09	AB	8A	24	FD	14	3E	27	A9	BC	08	F8	12	24	FD	14
C020	3E	19	04	27	FE	14	3E	F8	77	A8	EC	1B	F9	12	34	FD
C030	14	3E	04	37	FE	14	3E	FE	14	3E	56	3A	04	08	14	47
C040	9A	7F	14	47	8A	80	03	16	4B	04	47	62	83	9A	E3	27
C050	62	14	BC	55	BF	18	14	D6	F2	5C	04	54	EF	56	9A	AB
C060	BF	FF	8A	08	14	CD	23	FB	6A	F2	74	23	F7	6A	F2	78
C070	BB	01	04	DC	BB	00	04	DC	EF	64	14	D6	F2	7A	8A	04
C080	27	A8	A9	BE	08	14	D6	F2	8F	F8	53	7F	19	04	92	F8
C090	43	80	EE	97	A8	04	9B	77	A8	04	85	14	D6	F2	A4	F9
C0A0	12	A7	04	AB	F9	12	AB	BB	02	04	DC	0A	B2	B0	04	B2
C0B0	56	B6	BB	03	04	DC	F8	39	9A	DB	04	7A	36	C4	56	C2
C0C0	8A	20	04	BC	26	CC	56	CA	9A	20	04	C4	83	14	BC	65
C0D0	42	AA	27	62	55	83	14	CD	23	F7	6A	83	F8	39	E7	E7
C0E0	43	30	3A	14	ED	8A	10	14	ED	9A	EF	04	E3	ED	10	16
C0F0	F3	04	EF	ED	EF	83	00	00	00	00	00	00	00	00	00	00

Рис. 2. Дамп программной памяти ОЭВМ КМ1816ВЕ48

в худшем случае до 30 мкс. Это обстоятельство иногда приводит к повторному считыванию ЭВМ одного и того же байта информации. Надежно застраховать себя от такой случайности можно путем введения в драйвер ввода-вывода при чтении небольшой программной задержки из нескольких операторов NOP. В остальном программное обслуживание интерфейсного модуля ничем не отличается от стандартного, принятого для данной ЭВМ.

Программа контроллера (рис. 2) размещается в программной памяти ЭВМ и занимает 246 из 256 байт ПЗУ. Напряжение питания от интерфейсного модуля ЭВМ составляет 5 В, потребляемый ток 650 мА. Средняя скорость передачи информации через контроллер — 120 байт/с.

Данные на магнитную ленту следует записывать при максимальном уровне, что обеспечивает максимальную крутизну фронта нарастания сигнала на выходе магнитофона и тем самым уменьшает вероятность появления ошибок при считывании.

Испытания показали, что при записи и воспроизведении цифровой информации на бытовом кассетном магнитофоне типа «Весна 205» с магнитной лентой МК60 число сбоев не больше, чем при работе со стандартными накопителями типа РК-1, в которых используется принцип фазовой модуляции с такой же (32 бит/м) плотностью записи.

252057, Киев, ул. Победы, 56, комн. 238; тел. 441-24-37

Статья поступила 12.02.88

УДК 681.325—181.58:615.471

А. И. Труханов, М. П. Наумчик, С. Л. Чернин,
Ф. Г. Покровский, В. Г. Ермоленко

МАЛОГАБАРИТНЫЕ УЛЬТРАЗВУКОВЫЕ ДИАГНОСТИЧЕСКИЕ ПРИБОРЫ НА ОСНОВЕ ОДНОКРИСТАЛЛЬНОЙ ЭВМ

Разработанные ультразвуковые диагностические приборы — эхоэнцефалоскоп, эхоофтальмометр, эхотомоскоп — представляют собой простые, дешевые, малогабаритные приборы с цифровой памятью сравнительно небольшого объема, микроконтроллером (МК) на основе ОЭВМ КМ1816ВЕ35 и встроенным ТВ-монитором.

Унифицированный модуль МК, предназначенный для всех трех приборов, обслуживает функциональную клавиатуру, передает данные на знакогенератор для отображения на экране монитора, управляет перемещением визиров и режимами работы, обеспечивает расчет информативных параметров. В состав модуля кроме ОЭВМ [1] вошли ЗУ на основе К573РФ5, параллельный интерфейс ввода-вывода КР580ВВ55А, программируемый трехканальный таймер КР580ВИ53, регистр управляющего слова КР580ИР82

Вид ошибки	D2	D1
------------	----	----

Недопустимо большая скорость движения ленты	0	0
Недопустимо малая скорость движения ленты	0	*
Ошибка четности	*	0
Отсутствие готовности ЭВМ принять очередной байт информации	*	*

0 соответствует отключенному, * — выключенному состоянию светодиода.

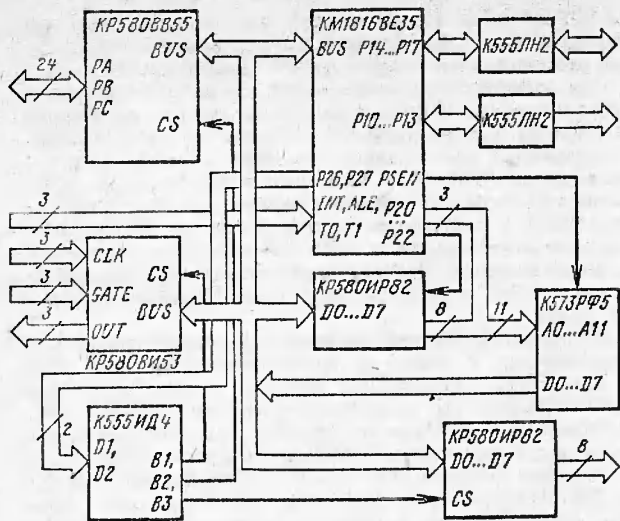


Рис. 1. Функциональная схема унифицированного микроконтроллера ультразвуковых приборов

(рис. 1). МК размещен на плате размерами 160×100мм (формат E1).

Все периферийные БИС МК — параллельный интерфейс, программируемый таймер и регистр подключены к шине процессора. На выходной разъем модуля выведены следующие основные сигналы, доступные пользователю: 24 линии ввода-вывода параллельного интерфейса, сформированные в три 8-разрядных порта; восемь линий порта P1 процессора (P10...P17); тестовые и контрольные сигналы процессора (T0, T1, ALE и т. д.); управляющие и выходные сигналы таймера; 8-разрядный код регистра управляющего слова (PUC). Три линии порта P2 (P20...P22) служат для взаимодействия ОЭВМ с внешней памятью типа К573РФ5, две (P26, P27) используются для управления через дешифратор К555ИД4 выбором периферийных БИС.

Цифровой эхоэнцефалоскоп предназначен для применения в неврологической и нейрохирургической практике. Прибор содержит два ультразвуковых датчика (рис. 2). В режиме работы с одним датчиком на экране монитора отображается эхограмма в гистограммном представлении и шкала для измерения расстояния в миллиметрах; в режиме с двумя датчиками — две эхограммы, расположенные одна под другой с размерностью 256×128 элементов, шкала расстояний, трансмиссионная метка, соответствующая

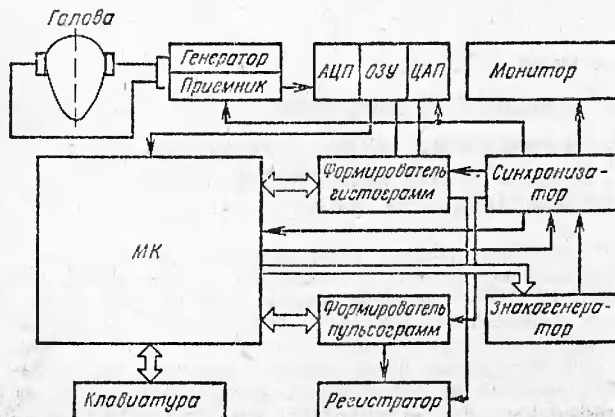


Рис. 2. Структурная схема эхоэнцефалоскопа

щая соосному расположению излучающего и приемного преобразователей с противоположных сторон головы. В правой части экрана предусмотрен вывод пульсограммы, которая формируется на основе анализа пульсаций выбранного с помощью вертикального визира эхо-сигнала. Кроме того, на экране отображаются исходные данные о пациенте и дате обследования, результаты измерений. Микропроцессорный измеритель позволяет оценивать по программно-управляемым визирам расстояние между выбранными структурами мозга и рассчитывать ряд информативных параметров — индексы желудочка, срединной камеры, мозгового плаща, медиальной стенки, а также вентрикулярный. [2].

Как показали проведенные медицинские исследования [3], имеется определенная зависимость между пульсациями отраженных сигналов и внутричерепным давлением. В приборе реализован алгоритм автоматического расчета индекса пульсаций, значение которого отображается на экране. В режиме «стоп-кадр» эхо- и пульсограмма фиксируются в памяти прибора и могут быть выведены на регистратор. Программа управления прибором занимает объем порядка 1600 байт.

Цифровой эхоофтальмометр предназначен для автоматизированной биометрии глаза, диагностики его травм и заболеваний (рис. 3). В приборе использован ряд унифицированных цифровых узлов эхоэнцефалоскопа: знакогенератор, запоминающее устройство, функциональная клавиатура, МК. Для одновременного автоматизированного измерения трех линейных размеров глаза — передней камеры, хрусталика, передне-задней оси — в прибор введены устройства стробирования и обнаружения, контроллер выборки, 3-канальный блок измерителей.

Прибор позволяет адекватно определить продольные линейные размеры оптической оси глаза, сократить время обследования за счет полной автоматизации процесса поиска, выделить и измерить информативные параметры, повысить точность и надежность результатов дополнительной обработкой диагностической информации с учетом индивидуальных значений скоростей распространения ультразвука в конкретных тканях. Независимость оценки амплитуд эхосигналов от разброса значений затухания ультразвука в глазу и повышение точности измерения временных интервалов обеспечиваются адаптивной системой временной автоматической регулировки усиления (ВАРУ) приемного тракта. Высокое быстродействие достигается благодаря последовательно-параллельной обработке диагностической информации аппаратными средствами на этапах выделения, сбора и накопления данных аппаратными и программными средствами на этапе расчета размеров глаза. Предварительная обработка состоит в формировании закона управления ВАРУ, оценке амплитудных и временных параметров эхосигналов и принятии решения о соответствии выбранного направления зондирования оптической оси глаза. Результат измерения временных интервалов

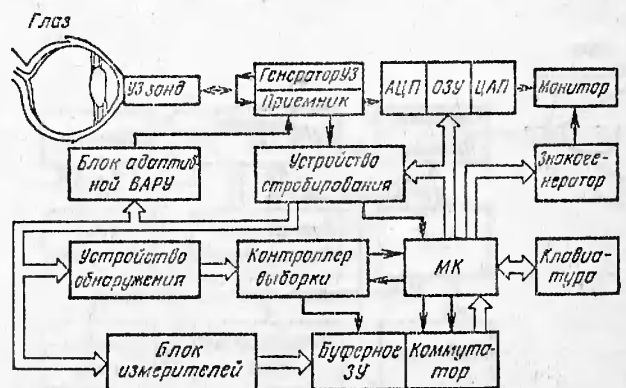


Рис. 3. Структурная схема эхоофтальмометра «Микропроцессорные средства и системы» № 1, 1990

между эхосигналами фиксируется только при выполнении критерия максимальных амплитуд трех эхосигналов.

Введение принципа избыточности, при котором из большого числа измеренных значений отбирается массив данных, отвечающих сформулированному амплитудному критерию, позволило повысить надежность информации, получаемой по заданному направлению зондирования. В то же время усреднение независимых результатов в выделенном массиве уменьшает случайную погрешность измерения.

Для компенсации погрешностей измерения от различия скоростей распространения ультразвуковых колебаний в структурах проводились экспериментальные исследования, позволившие выявить диапазоны изменения скоростей для нормы и различных патологий. Ввод индивидуальных значений скоростей с учетом патологических отклонений, имеющих у обследуемых пациентов, позволяет скорректировать указанную погрешность и повысить точность измерения. Программа управления прибором имеет объем порядка 1300 байт.

Цифровой эхотомоскоп с двумерным механическим сканированием предназначен для определения характера патологий мягких тканей внутренних органов и диагностики в акушерстве и гинекологии (рис. 4). Особенность режима двумерного сканирования объекта, а также необходимость перемещения маркеров по всему полю экрана и расширения поля функциональных клавиш потребовали разработки новых модулей знакогенератора, формирователя адресов, запоминающего устройства. Специфика прибора состоит также в том, что при выводе информации на ТВ-монитор используется специализированная развертка по сектору с углом раstra 90°.

Выбор и задание одного из восьми режимов работы прибора («стоп-кадр» изображения, три режима обработки изображения с преобразованием по фиксированному закону амплитуд эхосигналов, режим инвертирования изображения, увеличение масштаба, выбор номера кадра, вывод тестовой шкалы) осуществляются путем формирования управляющего слова, фиксируемого в регистре КР580ИР82.

Взаимодействие со знакогенератором предусматривает формирование на экране символической (буквы и цифра, 32 знака) и графической (макеры) информации. Подпрограмма формирования маркеров, управления их движением, т. е. формирование и отслеживание координат, осуществляется подпрограммой МАРКЕР. Маркер может перемещаться по окружности влево и вправо, по радиусу вверх и вниз.

При изменении одной из координат любого маркера автоматически по подпрограмме ФОРМУЛА вычисляется расстояние между ними. Эта подпрограмма в свою очередь, обращается к библиотечным подпрограммам КОРЕНЬ, УМНОЖЕНИЕ, СЛОЖЕНИЕ, ДЕЛЕНИЕ, ПРЕОБРАЗОВАНИЕ для перевода в десятичную систему счисления полученного значения. Результат заносится в ОЗУ процессора и выводится на экран подпрограммой ЗНАКОГЕНЕ-

РАТОР-ВЫВОД. Предусмотрена фиксация маркеров при выходе за допустимые пределы и сброс. Координаты маркеров отслеживаются подпрограммой ПРЕРЫВАНИЕ.

При расчете срока беременности по результатам измерения с помощью маркеров одного из трех размеров плода выбирается соответствующая таблица из ПЗУ ОЭВМ, сравниваются рассчитанные значения размера плода с табличными. Найденное табличное значение срока беременности в неделях переводится подпрограммой ПРЕОБРАЗОВАНИЕ в десятичную систему счисления. Полученное значение записывается в ОЗУ процессора и передается на знакогенератор. Программа работы МК размещается во внешнем ПЗУ типа К573РФ5 и занимает объем порядка 1,6 Кбайт.

Программное обеспечение приборов разрабатывалось и отлаживалось с помощью кросс-системы МИКРОС-048, и микроЭВМ «Электроника 60М» в составе комплекса 15ВУМС28-025 [4] и отладочной панели ОУ-48С [5].

Применение ОЭВМ серии КМ1816 совместно с БИС параллельного интерфейса КР580ВВ55, таймером КР580ВИ53 и регистром КР580ИР82 в составе унифицированного модуля МК позволило существенно расширить функции приборов, снизить их вес и повысить надежность. Апробированные схемные и программные решения могут быть использованы при создании ультразвуковых медицинских приборов более высокого класса для выполнения специализированных функций в составе мультимикропроцессорных систем.

125422, Москва, Тимирязевская, 1, ВНИИМП;
тел. 211-44-22

ЛИТЕРАТУРА

1. Кобылинский А. В., Липовицкий Г. П. Однокристалльные ЭВМ серии К1816 // Микропроцессорные средства и системы.— 1986.— № 1.— С. 10—19.
2. Методы исследования в клинической неврологии: Учебно-методическое пособие / Московский медицинский институт им. Н. И. Пирогова. М.,— 1983,— 57—59 с.
3. Энина Г. И., Робуле В. Х. Эхоульсография мозговых сосудов.— Рига: Зинатне, 1982.
4. Чернин С. Л., Береговская Е. М. Ультразвуковой эхоофтальмометр на базе однокристалльной микроЭВМ КР1816ВЕ48 // Микропроцессорные средства и системы.— 1987.— № 3.— С. 59—62.
5. Иванов В. И., Лобанов В. И., Митрофанов А. В. Отладочные средства для малозарядных однокристалльных микроЭВМ // Микропроцессорные средства и системы.— 1984.— № 2.— С. 42—45.

Статья поступила 18.03.88

Тело
пациента

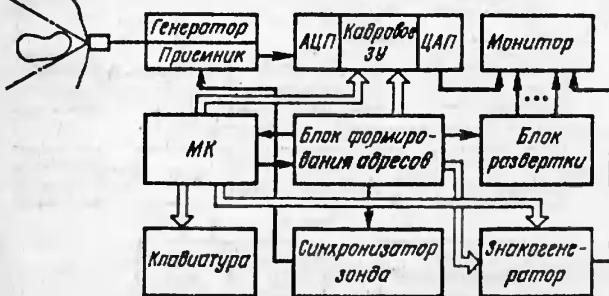


Рис. 4. Структурная схема эхотомоскопа

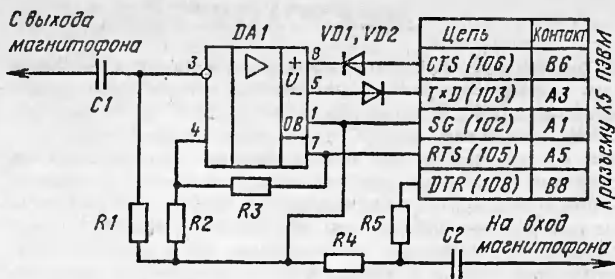
УДК 681.327

Д. А. Хохлов, Г. В. Завидовский

СОПРЯЖЕНИЕ ПЭВМ РОБОТРОН-1715 С КАССЕТНЫМ НАКОПИТЕЛЕМ НА МАГНИТНОЙ ЛЕНТЕ

Применение бытового кассетного магнитофона в качестве внешнего кассетного накопителя на магнитной ленте (КНМЛ) ПЭВМ дает возможность, кроме прочего, обеспечить совместимость различных ПЭВМ, имеющих разные форматы записи-считывания.

Для организации ввода-вывода информации из ПЭВМ Роботрон-1715 на кассетный магнитофон целесообразно использовать сигналы интерфейса V24 (разъем Х5), аппаратная и программная поддержка которого реализуется на БИС последовательного двухканального ввода-вывода U856 [1]. Для вывода информации из ПЭВМ необходимо



Принципиальная схема устройства сопряжения:

DA1 — К1400УД7; R1, R2 — 12 кОм, R3 — 1 МОм, R4 — 1 кОм, R5 — 12 кОм; C1, C2 — 0,1 мкФ; VD1, VD2 — КД521А

обеспечить программное управление значением выходного сигнала, а для ввода — возможность программного анализа значения входного сигнала. Данная БИС позволяет, кроме поддержки интерфейса V24, программно устанавливать в каждом канале сигналы PTS (105) и DTR (108) и опрашивать стандартные для данного интерфейса сигналы [2] CTS (106) и DCD (107) (в скобках приведены обозначения сигналов в соответствии с интерфейсом Стык С2).

Сигналы RTS и DTR устанавливаются следующим образом:

SMD	EQU	OFH;	адрес регистра управляющего слова
	MVI	A, 05H;	переключение БИС U856 в режим
	OUT	CMD;	управления RTS и DTR
	MVI	A, DATA;	установка требуемых значений
	OUT	CMD	сигналов PTS и DTR

Значения байта DATA определяется требуемым значением сигналов; сигнала RTS — вторым, а DTR — восьмым битами.

Значения сигналов CTS и DCD анализируются следующим образом:

MVI	A, 10H ;	переключение БИС в режим ввода
OUT	CMD ;	значений сигналов
IN	CMD ;	ввод значений сигналов
ANI	MSK	

Значение байта маскирования MSK определяется требуемыми сигналами. При этом сигналу DCD соответствует четвертый, а CTS — шестой биты.

Устройство выполнено на базе операционного усилителя с положительной обратной связью (см. рисунок). Питанию напряжения подаются на операционный усилитель через разъем X5 с выходов микросхем К170АП2, поскольку в ПЭВМ Роботрон-1715 напряжения питания не выведены на внешние разъемы. С вывода А5 разъема PTS (105) снимается напряжение 10 В (состояние Лог.0), с вывода А3 — напряжение — 10 В (состояние Лог.1). Сигнал А5 при работе устройства должен быть программно установлен в состояние Лог.0.

Авторами разработана и успешно эксплуатируется программа MRIP.COM, работающая в среде ОС CP/M, обеспечивающая запись-считывание информации с бытового кассетного магнитофона на любой НГМД ПЭВМ, изменение текстовых файлов, вывод каталога дисков и др. Программа MRIP.COM может использоваться также на микроЭВМ SM1800, Правец 8М.

Дифференциально-фазовый метод записи-считывания информации с контролем ошибок с помощью циклического избыточного кода (CRC), увеличивает надежность режима. Эффективность использования ПЭВМ возрастет, если применить специальное устройство сопряжения ПЭВМ с бытовым магнитофоном [3, 4] с формированием необходимых временных соотношений аппаратным способом.

ЛИТЕРАТУРА

1. Algorithmen der Mikrorechentechnik (Maschinenprogrammierung und Interpretiertechniken des U880), B. Lampe, G. Jorke, N. Wengel, VEB, VERLAG Technik BERLIN, 1983.
2. Мячев А. А., Иванов В. В. Интерфейсы вычислительных систем на базе мини- и микроЭВМ. — М.: Радио и связь, 1986. — 248 с.
3. Хохлов Д. А. и др. Устройство сопряжения магнитофона с автоматизированной системой контроля // Обмен производственно-техническим опытом. — 1986. — Вып. 12. — С. 54, 55.
4. А. С. 1337915 (СССР). Устройство для магнитной записи и воспроизведения цифровой информации / Д. А. Хохлов, М. Я. Масленков. — Оpubл. 1988, БИ № 34.

Статья поступила 6.12.88

УДК 621.034.01

А. В. Марчуков, И. В. Полуэктов, А. В. Исаев,
А. И. Шичев, Ю. А. Бакулин

ЛОКАЛЬНАЯ СЕТЬ НЕЙРОН-Т

Локальная кольцевая вычислительная сеть (ЛВС) НЕЙРОН-Т предназначена для объединения ПЭВМ «НЕЙРОН-И9.66», «Искра 1030.11», ЕС1840.

В комплект ЛВС входят сетевой контроллер, персональная карта для подключения к системной шине ПЭВМ, блок доступа в среду волоконных оптических линий связи, кабель, сетевые программные средства.

Сетевые программы обеспечивают множественный доступ к периферийным устройствам, прикладным программам и файлам данных.

В сети «НЕЙРОН-Т» реализованы следующие функции: коллективное использование отдельными ПЭВМ периферийных устройств;

поддержка индивидуальных паролей записи-считывания; возможность включения и отключения отдельных ПЭВМ без помех для работы сети в целом;

совместный доступ к файлам и прикладным программам;

конфигурация сети, рассчитанная на спулинг; гибкое и полное разделение любых ресурсов любого компьютера.

Аппаратные средства обеспечивают скорость передачи данных 1 Мбит/с. Сетевой контроллер выполнен в виде отдельного блока размерами (220×220×20) мм. Среда передачи данных — коаксиальный кабель, крученая пара, оптоволоконно.

Максимальное расстояние между отдельными ПЭВМ — 600 м; кодирование-декодирование — манчестерское; метод доступа — передача маркера.

Сетевое программное обеспечение предоставляет следующие дополнительные возможности: электронную почту, сетевую диагностику, удаление базы данных,

Для работы в сети ПЭВМ должна иметь накопитель на гибком магнитном диске, минимальный объем оперативной памяти 128 Кбайт и ОС MS DOS3.1...3.3.

ЛВС «НЕЙРОН-Т» позволяет подключать до 254 ПЭВМ. Контроллер ЛВС выполнен в виде печатной платы в отдельном корпусе и насчитывает около 30 микросхем. Для использования оптоволоконных линий связи создан блок доступа в среду, который может служить повторителем (длина сегмента возрастает до 1200 м).

Для определения спроса на сетевые компоненты просим высылать заявки с указанием числа и типов ПЭВМ, предназначенных для объединения в сеть по адресу: 634004, Томск-4, Советская, 84, УНПК Кибернетика

Объединенные под данной рубрикой статьи познакомят читателя с наиболее широко используемым в практике классом одноплатных микросистем, называемых микроконтроллерами (МК). В состав МК входит полный набор средств, образующих микроЭВМ: процессор, память, подсистема ввода-вывода.

Однако в отличие от микроЭВМ МК предназначены для встраивания в аппаратуру пользователя в качестве центрального элемента управления. Специфика их применения накладывает определенные требования на состав и функционирование программно-аппаратного обеспечения, приводящие к структурным и архитектурным отличиям, выделяющим МК в особый класс микросистем с характерными только для них задачами системного проектирования.

Микроконтроллеры, как правило, периодически выполняют одну и ту же прикладную программу, для хранения которой используется встроенное ПЗУ. По этой причине большинство МК не имеет встроенных контроллеров гибких или жестких магнитных дисков. Прикладные программы МК работают в реальном масштабе времени, предъявляя требования к организации системы прерываний и построению программ. Особое внимание обращается на габаритные размеры МК и потребляемую мощность.

УДК 681.325

А. П. Дианов, Н. Н. Щелкунов

ОДНОПЛАТНЫЕ МИКРОЭВМ ДЛЯ ВСТРОЕННЫХ СИСТЕМ

Семейство мМС12ХХ — набор одноплатных микроконтроллеров (МК) и микроЭВМ с различными функциональными свойствами, объединенных единым для всего семейства рядом архитектурных, структурных и конструктивных признаков. Функциональные отличия модулей и их программно-аппаратная совместимость обеспечивают быструю компоновку и развитие микросистем (МС) в соответствии с конкретными прикладными требованиями.

В основе семейства (табл. 1) — одноплатный 8-разрядный МК общего назначения мМС1204 [1], построенный на базе микропроцессора (МП) КР580ВМ80. Этот МК обладает минимальным набором внутрисплатных ресурсов, достаточным для построения простейших прикладных систем управляющего типа с быстродействием до 500 тыс. оп./с. В системах с повышенной скоростью счета (до 2 млн. оп./с) должен использоваться базовый модуль мМС1212 [2] — 16-разрядная одноплатная МС на основе МП К1810ВМ86. Ресурсы мМС1204 и мМС1212 могут быть увеличены с помощью модулей расширения специального и общесистемного назначения, среди которых —

Таблица 2
Модули памяти семейства мМС

Тип модуля	Объем, Кбайт		Тип системной магистрали
	ОЗУ	ПЗУ	
мМС3101	64	—	МИ1200
мМС3701	64	16	МИ1200
мМС3702	128	64	МИ1210

мМС1204+3701 достаточно для построения небольшого автономного программирующего комплекса мМС0602 [1].

МК мМС1205 ориентирован на многоплатные системы с большой памятью (до 1 Мбайт). Его системная магистраль совместима с 16-разрядной межмодульной шиной МИ1210, формируемой мМС1212, что обеспечивает прямое использование плат расшире-

Таблица 3

Двухплатные системы семейства мМС

Тип модуля памяти	Объем ОЗУ/ПЗУ, Кбайт					
	мМС1204	мМС1205	мМС1206	мМС1212	мМС1213.1	мМС1213.2
мМС3101	64/4	64/32	—	—	—	—
мМС3701	65/20	65/48	—	—	—	—
мМС3702	—	129/96	—	130/128	255/192	640/192

набор плат памяти (табл. 2). В двухплатных системах (табл. 3) расширяются не только ОЗУ (до 64 Кбайт и более), но и ПЗУ МС для хранения прикладных программ и исходных данных (например, ресурсов памяти

ния, первоначально разработанных для мМС1212, например модуля комбинированной памяти мМС3702. Другой 8-разрядный МК мМС1206 полностью заменяет систему мМС1204+3701 при построении недорогих и компактных

Таблица 1

Одноплатные микроЭВМ семейства мМС

Тип		Число разрядов	Объем		Канал		Таймер	Запросы на прерывания	Тип системной магистрали	I _{св.} А	Число ВИС
модуля	микропроцессора		ОЗУ, байт	ПЗУ, Кбайт	ИРПС	ИРПП					
мМС1204	КР580ВМ80	8	1	4	2	1	2×16	8	МИ1200	1,1	25
мМС1205	КР580ВМ80	8	1	32	2	1	2×16	8	МИ1210	1,3	27
мМС1206	КР580ВМ80	8	64	64	2	1	2×16	8	МИ1200	1,3	36
мМС1212	КМ1810ВМ86	16	2	64	2	—	2×16	9	МИ1210	1,6	28
мМС1213.1	КМ1810ВМ86	16	128	128	2	—	2×16	9	МИ1210	1,7	39
мМС1213.2	КМ1810ВМ86	16	512	128	2	—	2×16	9	МИ1210	1,7	39

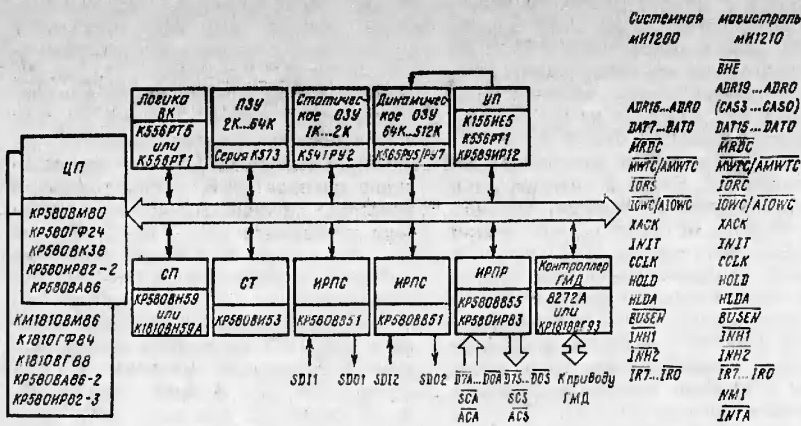


Рис. 1. Организация микроЭВМ семейства ММС

МС, микроЭВМ и микровычислительных комплексов, например системы проектирования микропроцессорных средств [3]. МК ММС1213 имеет две модификации, отличающиеся объемами ОЗУ, и функционально перекрывает двухплатную 16-разрядную систему ММС1212+3702. Область их использования — реализация ядер небольших 16-разрядных микроЭВМ. В отличие от предыдущих модулей ММС1206 и ММС1213 допускают постановку операционных сред типа ДОС1800 и БОС1810 соответственно. Для этого они должны быть расширены контроллерами гибких магнитных дисков. Простота логической организации, строгость отбора состава аппаратных средств и их функциональная завершенность — основополагающие принципы проектирования всего семейства.

Все МК и микроЭВМ имеют единую магистрально-модульную архитектуру (рис. 1), открытую для расширения с помощью системной магистрали. Модули ММС1204 и ММС1206 поддерживают 8-разрядную магистраль МИ1200 с 16-разрядной адресной шиной, остальные используют внутрисистемную магистраль МИ1210 с 16-разрядной шиной данных и 20-разрядным адресом (прямое расширение МИ1200). Внутрисистемная магистраль МК ММС1205 совместима с МИ1210, однако доступны только байтовые операции доступа. Расширение шины данных выполнено по схеме

$$\text{VHE} = \text{ADRO}$$

$$\text{DAT}_{i+8} = \text{DAT}_i, i = 0-7$$

В МК ММС1205 системная шина данных DAT7...DAT0 дополнительно буферизирована, что обеспечивает физическую изоляцию его внутрисистемных ресурсов от внешнего расширения. При доступе к внутрисистемным ресурсам этот буфер всегда передает локальные данные на системную магистраль.

Центральный процессор (ЦП) 8-разрядных МК выполнен по типовой 5-кристальной схеме [1, рис. 2], за

исключением ММС1206, в котором отсутствует буферизация адреса. Модуль имеет слаботочную адресную шину ($I_{OL} \leq 1,8 \text{ мА}$, $C_L \leq 100 \text{ пФ}$), рассчитанную на построение двух- или трехплатных систем. В ЦП ММС1205 и ММС1206 введена логическая схема (рис. 2), выделяющая сигнал начала командного цикла M1. DD1.1 формирует вспомогательный сигнал WAIT, выводимый на светодиод. ЦП ММС1205 занимает площадь $95 \times 75 \text{ мм}^2$.

Центральный процессор 16-разрядных МК ММС1212 и ММС1213 $100 \times 80 \text{ мм}^2$ реализован на базе МП К1810ВМ86 [2, рис. 2]. В ЦП ММС1213 вход немаскируемого прерывания NMI используется в качестве пульта запроса. Дребезг контактов клавиши пульта прерывания, установленной непосредственно на плате МК, устраняется на входе NMI с помощью асинхронного RS-триггера. К1810ВМ86 расширяется числовым сопроцессором К1810ВМ87 модулем вставляемым в панельку основного МП.

Подсистема памяти МК ММС1205 аналогична подсистеме памяти

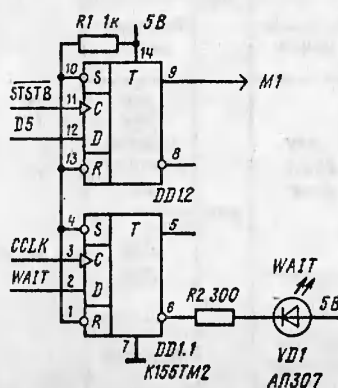


Рис. 2. Аппаратное расширение ЦП

ММС1204 [1, рис. 2], но для установки ПЗУ использована панелька [1, рис. 4], сократившая площадь, занимаемую подсистемой памяти с $120 \times 75 \text{ мм}^2$ до $105 \times 75 \text{ мм}^2$, и увеличившая допустимые объемы постоянной памяти. Логика выбора кристаллов (ВК) памяти предусматривает четыре варианта размещения ПЗУ в адресном пространстве МС, программируемых аппаратно (рис. 3). Выбор варианта размещения зависит от типа используемого ПЗУ и его физического объема. Область ОЗУ смещена с адреса 1000H (ММС1204) на 0800H и не зависит от варианта размещения ПЗУ. Перемещение области ОЗУ потребовало соответствующего переопределения системного стека и области данных, используемых монитором-отладчиком ММОН1200 [4], однако их внутренняя структура сохранена.

Сигналы управления (WE и CE ОЗУ К541РУ2) формируются схемами малой логики (в ММС1204 использовалось ПЗУ К556РТ5 логики ВК).

Оперативная память ММС1206 выполнена на приборах динамического типа серии К565 и имеет объем до 64 Кбайт (восемь микросхем К565РУ5). Память ОЗУ реализована по схеме [5, рис. 4] с «прозрачным» для ЦП алгоритмом регенерации и не требует дополнительных тактов ожидания с его стороны (данные обрабатываются с максимальной для МП КР580ВМ80 скоростью).

Для установки схем ПЗУ модуль имеет одну универсальную панельку (рис. 4), в которую могут быть установлены микросхемы типа 27512 имеющего объем в 64 Кбайт (табл. 4).

В модуле ММС1206 предусмотрены четыре варианта распределения адресного пространства памяти программ и данных между ОЗУ и ПЗУ (рис. 5). Карта памяти выбирается программно с помощью двух свободных выводов РС4 и РС5 программируемого периферийного адаптера (ППА) КР580ВВ55 [1, рис. 1]. Карты дейст-

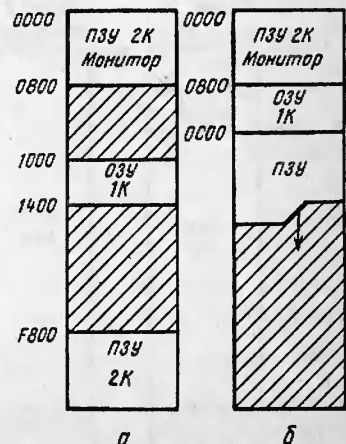


Рис. 3. Карта памяти ММС1204 (а), ММС1205 (б)

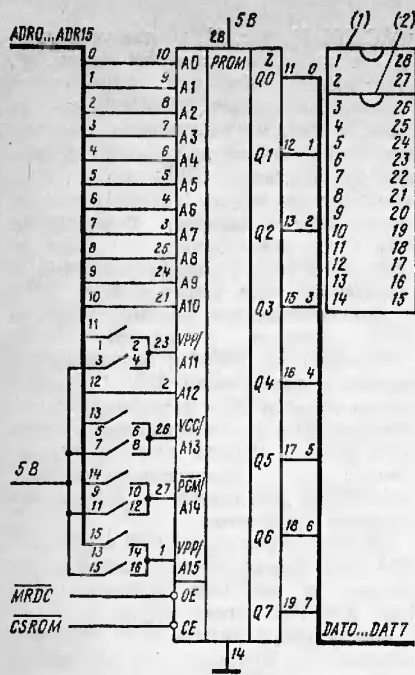


Рис. 4. Схема подключения УСРПЗУ

вительны только для операции чтения; запись данных возможна только в ОЗУ.

При сбросе система возвращается к базовой карте памяти 1 с ПЗУ

(2 Кбайт) в начале адресного пространства. В этой части ПЗУ размещается либо монитор мМОН1200 [4], либо программа инициализации системы, перекачивающая монитор ОС ДОС1800 или ДОС1810 из ПЗУ в ОЗУ с адресами 0F800H... 0FFFFH.

Карта 2 — основная при работе загруженной ОС. Карта 3 резервируется для организации ПЗУ-диска. Область ОЗУ 0F600H...0F7FFH служит окном взаимодействия между ПЗУ-диском и системой. В некоторых случаях может потребоваться набор аппаратно-зависимых подпрограмм ввода-вывода, которые не удается разместить в области монитора. Типовой пример такой программы — драйвер контроллера гибких магнитных дисков (ГМД). Для их хранения может быть использовано ПЗУ карты 4. Подсистема памяти мМС1206 с логикой ВК занимает площадь 155×80 мм², что не на много превышает площадь, отводимую для этих же целей в мМС1204 и мМС1205 со статическим ОЗУ.

Основное усовершенствование мМС1213: непосредственно на плате реализован 16-разрядный блок динамического ОЗУ объемом 128 Кбайт (16 БИС К565РУ5) или 512 Кбайт (16 БИС К565РУ7). Он использует «прозрачный» для МП алгоритм регенерации памяти без тактов ожидания со стороны ЦП. Схема управления памятью рассчитана на применение БИС К565РУ7, размещаемых в нижней половине мегабайтного простран-

ства адресов. Память реализует как байтовый, так и пословный доступ в соответствии с временными диаграммами работы МП К1810ВМ86 [2, рис. 3].

Принцип построения блока динамического ОЗУ мМС1213 такой же, как в одноплатном модуле мМС3702 [5, рис. 6]. Отличие мМС1213 — использование сигнала ALE в цикле доступа (рис. 6), входящего в состав локального интерфейса МП К1810ВМ86, но отсутствующего в составе системной магистрали (блок памяти работает без дополнительных тактов ожидания).

Для реализации произвольной логики в мМС1213 применены программируемые логические матрицы (ПЛМ) серии К556 [6]. В более ранних моделях семейства для этих целей применялись программируемые ПЗУ К556РТ5 с плавкими перемычками.

Широкое использование ПЛМ в блоке ОЗУ мМС1213 обеспечило создание действительно универсальной схемы управления памятью (УП) динамического типа, реализованной в виде автомата, алгоритм работы которого программируется с помощью плавких перемычек. На основе ПЛМ можно проектировать не только схемы комбинационной логики, но и триггерные узлы (рис. 7), обратные связи которых выполняются внешними соединениями. DD4, DD5 (рис. 6) реализуют 9-разрядный адресный мультиплексор «3 в 1» и два сигнала выборки (ROMS — внутрислотовый ПЗУ и LS — внутрислотовый средств):

ROMS = ADR16·ADR17·ADR18×
×ADR19

IOS1 = в соответствии с [2, рис. 1, 6]
IOS = IOS1·ADR11·ADR12×

×ADR13·ADR14·ADR15

RAMS = ADR19

LS = ROMS+IOS+RAMS

IOS и RAMS — вспомогательные сигналы выбора внутрислотовых средств ввода-вывода (ВВ) и ОЗУ. Временные диаграммы работы схемы управления памятью приведены на рис. 8.

Логика формирования сигнала ROMS рассчитана на ПЗУ (64 Кбайт), размещенное в конце мегабайтного адресного пространства. Однако с помощью двух новых панелей (рис. 4) возможно увеличение объема до 128 Кбайт. При этом логика формирования сигнала ROMS должна быть изменена.

При совпадении областей ПЗУ мМС1213 с встроенным ОЗУ (аналогично мМС1206) попытка записи приводит к вводу данных в ОЗУ (рис. 7). Логика формирования сигналов CASL и CASH рассчитана на будущие модификации микроЭВМ, так как карта размещения ПЗУ в мМС1213 программно не управляется (доступ к скрытому ОЗУ закрыт). Площадь, занимаемая подсистемой памяти нового 16-разрядного модуля, составляет 115×50 мм² (накопитель ОЗУ) плюс 85×85 мм² (схема УП, логика выборки кристаллов и ПЗУ) против 110×90 мм² в модуле мМС1212.

Таблица 4

Выбор перемычек и способа установки ПЗУ

Объем, Кбайт	Тип ИФРПЗУ	Перемычка	Способ установки
2	К573РФ2/РФ5	3—4,7—8	(2)
4	2732	1—2,7—8	(2)
8	К573РФ4/РФ6	1—2,7—8,11—12,15—16	(1)
16	27128	1—1,5—6,11—12,15—16	(1)
32	27256	1—2,5—6,9—10,15—16	(1)
64	27512	1—2,5—6,9—10,13—14	(1)

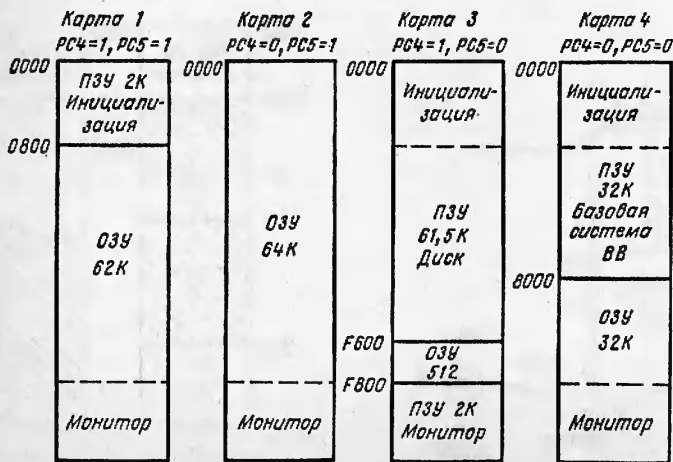


Рис. 5. Карты памяти мМС1206

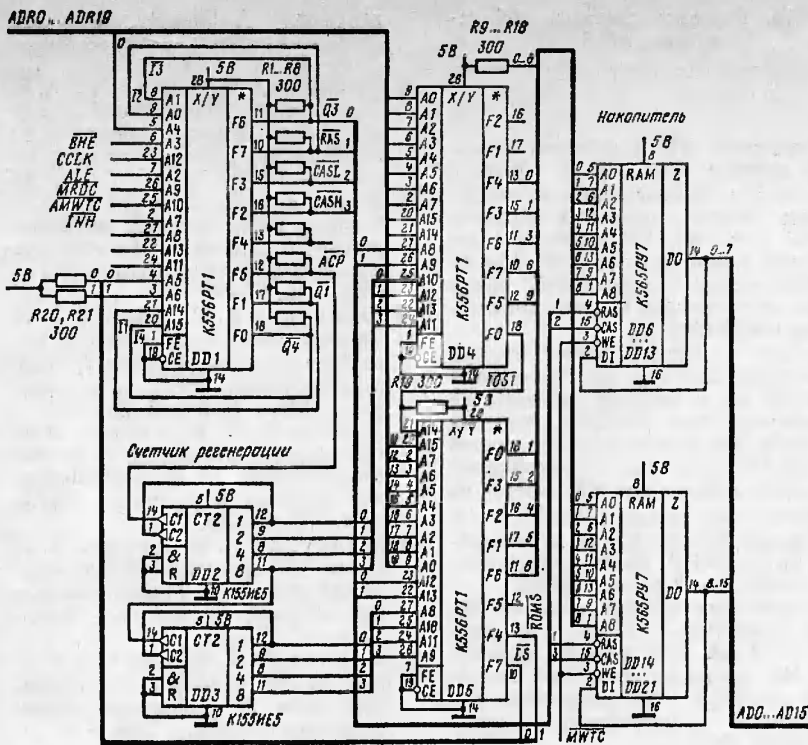


Рис. 6. Модуль ОЗУ 512 Кбайт
одноплатной микроЭВМ мМС1213

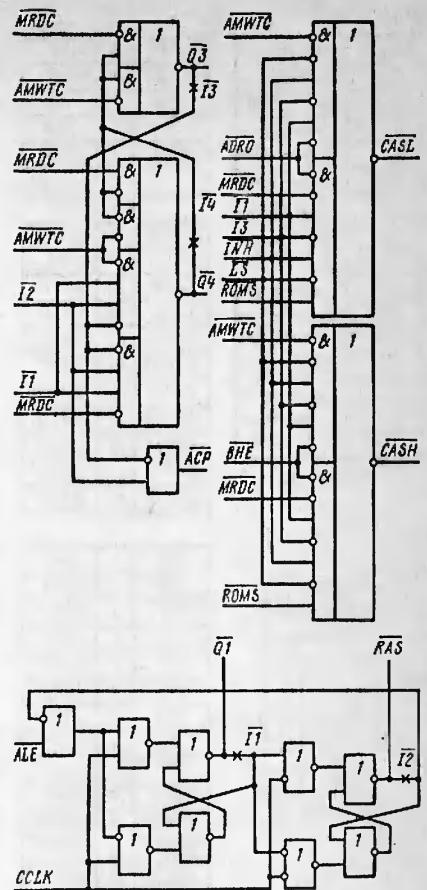


Рис. 7. Логика ПЛМ DD1

Во всех 8-разрядных МК использована одна и та же базовая организация подсистемы ВВ и средств под-

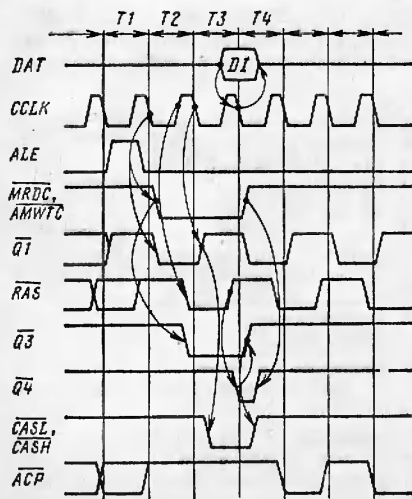


Рис. 8. Временные диаграммы работы
схемы управления памятью

держки режима реального времени [1]; занимаемая ею площадь — $210 \times 70 \text{ мм}^2$. Особенность подсистемы ВВ мМС1205 — наличие нового 8-разрядного порта вывода данных с адресом 011Н, построенного на БИС К589ИР12. Этот системный порт формирует старшую часть адреса ADR19...ADR16, сигнал INH2 программного запрета локального ОЗУ и сигнал TEST, выводимый на светодиод (рис. 9). Таким образом, расширенное пространство памяти (1 Мбайт) мМС1205 организовано в виде 16 блоков памяти по 64 Кбайт каждый. Простейшая техника переключения блоков памяти мМС1205 накладывает ряд ограничений на построение программ с межблочной передачей управления. Наличие единой для всех блоков внутрисплатной памяти, которая выполняет роль межблочной об-

	D7	D6	D5	D4	D3	D2	D1	D0
011H	TEST	INH2			ADR 19	ADR 18	ADR 17	ADR 16

Рис. 9. Формат данных системного
порта 011Н

ласти взаимодействия, ослабляет эти ограничения.

В модуле мМС1206 были усовершенствованы выходные буферные каскады ИРПС. Каскад на двух транзисторах КТ203 и КТ3102 заменен на схему с одним операционным усилителем К140УД6 (рис. 10), выполняющую те же функции, но занимающую меньшую площадь. Кроме того, свободные выходы PC4 и PC5 задействованы для выбора карты памяти (рис. 5). В новых моделях семейства подсистему ВВ предполагается расширить за счет ввода в нее контроллера ГМД.

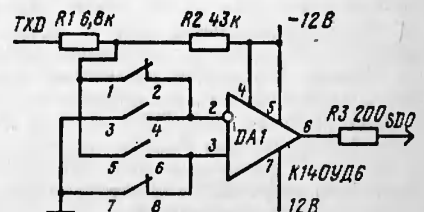


Рис. 10. Выходной буферный каскад
ИРПС

Б	А
SD02 16	SDI2
SD01 15	
GND 14	SDI1
D7A 13	GND
D6A 12	D7S
D5A 11	D6S
D4A 10	D5S
D3A 9	D4S
D2A 8	D3S
D1A 7	D2S
DOA 6	D1S
SCA 5	DO3
ACA 4	SC3
GND 3	AC3
GND 2	
GND 1	GND

ГРПМ1-31

GND 8
SD02 7
SDI2 6
5
4
GND 3
SD01 2
SDI1 1

МРН8

БHE	1	ADRO
IR0	2	ADR1
IR1	3	ADR2
IR2	4	ADR3
IR3	5	ADR4
IR4	6	ADR5
IR5	7	ADR6
IR6	8	ADR7
IR7	9	ADR8
ADR16	10	ADR9
ADR17	11	ADR10
ADR18	12	ADR11
ADR19	13	ADR12
INH1	14	ADR13
INH2	15	ADR14
16		ADR15
NMI	17	CCLK
GND	18	GND

РППМ16-72

Рис. 11. Разводка сигналов МК семейства ММС

Подсистема ВВ и средства реального времени ММС1213 и ММС1212 [2, рис. 5] — тождественны, за исключением замены выходных буферов ИРПС (рис. 10). По аналогии с семейством 8-разрядных МК оба 16-разрядных модуля обслуживаются одним и тем же системным монитором-отладчиком ММОН1210 [8].

Модули семейства имеют единый конструктив печатных плат размерами 135×240 мм и краевым разъемом на 72 контакта типа РППМ16, предназначенным для вывода внутрисистемной магистрали (рис. 11). Периферийный интерфейс 8-разрядных МК выведен на 31-контактный разъем ГРПМ1. В 16-разрядных МК отсутствует параллельный канал ИРПР, поэтому предусмотрен 8-контактный разъем МРН8.

Потребляемый ток для младшей модели семейства — 1,1 мА, для старшей — 1,7 мА. Для питания прикладных МС, построенных на базе микроЭВМ семейства ММС, разработан ряд компактных источников серии ММС90ХХ [9].

Телефон 408-62-22, Москва
Техническая документация на изготовление микроЭВМ серии ММС поставляется центром НТТМ «Физтех»: 141700, г. Долгопрудный Московской обл., аб/я 46

ЛИТЕРАТУРА

1. Шелкунов Н. Н., Дианов А. П. Универсальный одноплатный микро-

контроллер // Микропроцессорные средства и системы. — 1986. — № 5 — С. 65—69.

2. Шелкунов Н. Н., Дианов А. П. Одноплатный 16-разрядный микроконтроллер общего назначения // Микропроцессорные средства и системы. — 1987. — № 1. — С. 77—83.
3. Дианов А. П., Шелкунов Н. Н. Система проектирования микропроцессорных средств // Микропроцессорные средства и системы. — 1987. — № 5. — С. 83—86.
4. Шелкунов Н. Н., Дианов А. П. Техника программирования 8-разрядных микроконтроллеров // Микропроцессорные средства и системы. — 1986. — № 6. — С. 23—28.
5. Дианов А. П., Шелкунов Н. Н. Организация динамической памяти микросистем // Микропроцессорные средства и системы. — 1987. — № 4. — С. 75—80.
6. Шелкунов Н. Н., Дианов А. П. Процедуры программирования логических матриц // Микропроцессорные средства и системы. — 1986. — № 2. — С. 71—76.
7. Дианов А. П., Шелкунов Н. Н. Модули программирования логических схем // Микропроцессорные средства и системы. — 1988. — № 1. — С. 40—44.
8. Шелкунов Н. Н., Дианов А. П. Техника программирования 16-разрядных микроконтроллеров // Микропроцессорные средства и системы. — 1987. — № 2. — С. 11—14.
9. Дианов А. П., Шелкунов Н. Н. Малогабаритные источники питания для микросистем // Микропроцессорные средства и системы. — 1987. — № 3. — С. 73—76.

Статья поступила 11.01.88

УДК 681.326

Д. А. Журавский, С. Н. Барков

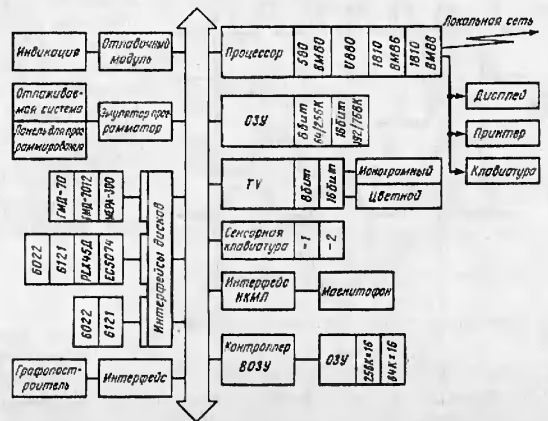
МОДУЛЬНАЯ МИКРОЭВМ

МикроЭВМ состоит из отдельных независимых друг от друга и от процессора модулей, объединенных общей системной магистралью типа И-41 (см. рисунок). Программное обеспечение (ПО) удовлетворяет требованиям полной совместимости с ОС СР/М и прямого использования стандартных системных и прикладных программ. Минимальный комплект (несложный контроллер) — один модуль центрального процессора с расположенными на нем интерфейсами, ОЗУ и ПЗУ. Такую ЭВМ можно подключить к дисплею, локальной сети и набору контролируемых устройств. Максимальный комплект (микроЭВМ) — восемь из 14 плат. Все модули совместимы друг с другом и поддерживаются сервисными функциями ПО.

Конструктивно микроЭВМ состоит из системного блока и блока питания. Модули системного блока находятся в корзине с разъемами типа РППМ16-288 (основа конструкции микроЭВМ «Электроника 60»), которые объединены печатной кроссплатой.

Из модулей процессора, контроллера телевизионного адаптера, сенсорной и кнопочной клавиатуры, связи с магнитофоном, подключения блоков ОЗУ 256К или 64К, свя-

зи с графопостроителем, отладочного эмулятора и программатора ПЗУ, интерфейса НГМД «Электроника 7012», универсального дискового контроллера и контроллера пя-



Блок-схема микроЭВМ

тидной емкости накопителя с повышенной плотностью легко собрать систему требуемой конфигурации, выполняющую конкретные функции.

При использовании микроЭВМ в качестве инструментальной под управлением ОС CP/M необходимы следующие модули: процессора, контроллера внешней памяти, оперативной памяти (более 64 Кбайт, если внешнее устройство — диск, и более 128 Кбайт, если — лента) и устройства отображения — любого дисплея, подключаемого к интерфейсам на процессорном модуле, или телевизора, подключаемого к модулю телевизионного контроллера. Операционная система, адаптированная для МП КР580ВМ80 и U880, записана в ПЗУ процессорного модуля, поэтому внешняя память нужна только для хранения программ-утилит ОС. Для МП К1810ВМ86 резидентная часть ОС хранится и загружается с устройства внешней памяти по команде монитора.

Процессорный модуль разработан в четырех различных исполнениях, включающих соответственно МП: КР580ВМ80, К1810ВМ86, К1810ВМ88, U880 с формирователями-буферами шин адреса, данных и управления; набор наиболее часто используемых интерфейсов; контроллер прерываний; системный таймер; диспетчер памяти, а также панель под РПЗУ с УФ-стиранием емкостью от 2К×8 (K573PФ2 или 2716) до 64К×8 (27512). На модуле МП КР580ВМ80 или можно установить локальное ОЗУ (две микросхемы 1К×4 типа К132PУ8 или К541PУ2) и применять его в качестве дешевого одноплатного микроконтроллера.

Модули оперативной памяти с автономной регенерацией разработаны в двух вариантах: 8-разрядный — на основе микросхем К565PУ3, К565PУ5, К565PУ6 общей емкостью 64 и 256 Кбайт; 16-разрядный — на основе К565PУ5, К565PУ7 емкостью 384 Кбайт и 1,5 Мбайт. Модули доступны во всем адресном пространстве, работают в пошаговом режиме без потери информации, используются в качестве системной памяти процессора и электронного диска.

Модули телевизионного контроллера также рассчитаны на работу с 16- и 8-разрядными шинами данных. Размер изображения в черно-белом режиме 512×256 точек, в цветном — 256×256 точек для обоих типов контроллеров. 8-разрядный контроллер имеет две градации яркости черно-белого изображения и четыре цвета; 16-разрядный — четыре градации яркости черно-белого изображения и 16 цветов и оттенков.

ОЗУ и ПЗУ телевизионного модуля — теньевые, т. е. могут включаться и отключаться процессором от системной магистрали. В ПЗУ записана программа обслуживания контроллера (эмуляция алфавитно-цифрового дисплея, вывод графических примитивов). Для 16-разрядного процессора эта программа находится в ПЗУ на процессорном модуле.

Данная программа полностью эмулирует систему команд стандартного дисплея VT52 (24 строки по 85 символов), дополненную командами видеоинверсии, подчеркивания символов, вставки и удаления строки.

Следует отметить, что при вводе-выводе изображения на экране полностью отсутствует эффект «снега», кроме того, в контроллере аппаратно реализован «ролик», т. е. сдвиг экрана по вертикали вверх и вниз, что существенно ускоряет обработку функции «перевод строки».

Универсальная кнопочная клавиатура с большим набором сервисных возможностей, с несколькими режимами кодировки клавиш, с параллельным и последовательным интерфейсами построена на ОЭВМ КР1816ВЕ35. Она поддерживает матрицу на 104 (до 114) клавиши из которых 40 (до 56) — функциональные. Они задают четыре стандартных набора кодов, используемых текстовыми редакторами и другими прикладными программами ОС CP/M, а также коды дисплеев VT52 и VT100. Клавиатура может работать в кодах КОИ-7 и КОИ-8.

Модуль связи с магнитофоном и графопостроителем содержит традиционную схему подключения к магнитофону для обеспечения совместимости со стандартом МХ и различными форматами записи ПЭВМ Микроша, РК-86, БК-0010. Кроме того, в этом контроллере аппаратно реали-

зовано кодирование байта при записи и декодирование — при чтении, а также поиск заголовка файла.

Информация записывается на ленту блоками по 1 Кбайт с дублированием для повышения надежности. Программа чтения с магнитофона расположена в ПЗУ (добавлена к набору встроенных функций процессора консольных команд ССР ОС CP/M, таких как просмотр каталога, распечатка файла и т. д.). Таким образом, можно работать без накопителя на магнитных дисках, загрузив электронный диск с магнитофона и прочитав с ленты все необходимые программы. Перед выключением микроЭВМ результаты работы сохраняются на ленте программной записи.

Скорость обмена с магнитофоном составляет 6300 бит/с, что обеспечивает запись 1 Мбайт информации на кассету МК-60.

Отладочный модуль отображает состояние всех управляющих сигналов системной шины на светодиодах выносного пульта, а также 20-разрядной адресной шины и 16-разрядной шины данных на 7-сегментных светодиодах матрицах; обеспечивает пошаговый и замедленный режимы работы процессора.

Программатор и эмулятор объединены в одном модуле. Эмулятор позволяет имитировать ПЗУ (2 Кбайт) с организацией на 8 и 16 бит данных (выбирается программно переключением режима работы эмулятора). ОЗУ эмулятора включено как теньевое со стороны центральной ЭВМ. В эту память программа обслуживания эмулятора заносит отлаживаемую версию. Далее ОЗУ переключается в режим эмуляции ПЗУ отлаживаемой системы.

С помощью программатора считывается содержимое ПЗУ или заново программируются РПЗУ типов К573PФ2, К573PФ5, 2716, 2732, К573PФ4, К573PФ6, К558PРЗ, 2764, 27128, 27256, 27512 и аналогичных по алгоритму программирования емкостью до 64 Кбайт.

На описываемой микроЭВМ поставлены ОС: CP/M версии 2.2 для процессоров КР580ВМ80 и U880; CP/M версии 1.0 для К1810ВМ86, сочетаемая с широко известными в нашей стране ОС1800, ОС1810, МикроДОС-8 и МикроДОС-16. В настоящее время имеются трансляторы практически со всех распространенных языков высокого уровня: Паскаль, ПЛ/М, ПЛ/1, Си, Ада, Фортран; текстовые редакторы (Вордмастер, Вордстар, Супертекст), интерпретатор и полностью совместимый с ним компилятор языка БЕЙСИК; кросс-системы для процессоров i8048, i8051, 6805, 6809, 6502, i8086; язык для ведения баз данных Дбэйс-2; программа файлового обмена с ОС RT-11, несколько версий ассемблеров и компоновщиков; символичный отладчик; игровые и прикладные программы.

Телефон 963-21-00, Москва

УДК 681.325

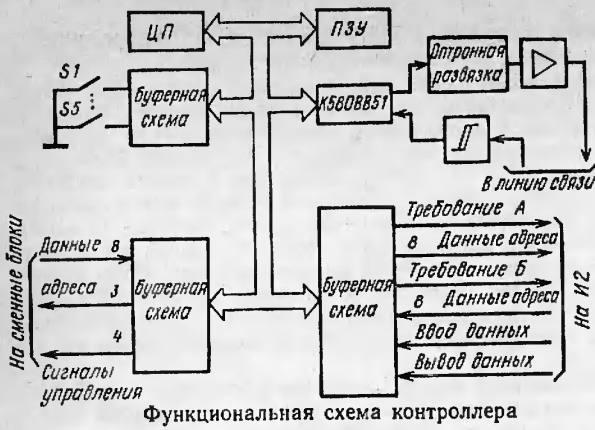
В. А. Бабин, Н. П. Журавлев, А. С. Овечкин, А. И. Судаков, В. П. Юричев

АВТОМАТИЗИРОВАННАЯ СИСТЕМА КОНТРОЛЯ ЗАГРЯЗНЕНИЯ ВОЗДУХА

Автоматизированная система контроля загрязнения воздуха (АСКЗВ) выбросами микробиологических производств — локальная распределенная система с частичной децентрализацией управления на уровне контрольной станции. Ее особенность — прямое измерение отдельных параметров с помощью преобразователей и расчет зоны распространения загрязняющих веществ в жилом секторе.

АСКЗВ завода по производству белково-витаминных концентратов (БВК) включает в себя пять (по числу источников выброса) анализаторов белковых веществ, подключенных к пяти контрольным станциям (КС). Метеодатчики (температуры воздуха, скорости ветра) подсоединены к входам шестой КС. С помощью четырехпроводной линии связи, выполненной витыми парами, информация от КС передается на каналный процессор (КП). Принцип организации линии связи — моноканальный. Система

ОДНОПЛАТНЫЙ МИКРОКОНТРОЛЛЕР С МАЛЫМ ПОТРЕБЛЕНИЕМ ЭНЕРГИИ



Функциональная схема контроллера

обладает возможностью расширения: к КП может быть подключено до 30 КС. На первом этапе использовалась микроЭВМ «Электроника 60М», в настоящее время — ДВКЗМ. Обмен с микроЭВМ осуществляется побайтно с помощью стандартных сигналов И2 (требование А, Б и ввод-вывод данных).

КС и КП — каркасы с выдвигаемыми блоками в стандарте УТК 20. Схемотехника контроллеров КП (ККП) и КС (ККС) полностью совпадает (см. рисунок), отличие — в программе управления, записанной в РПЗУ типа К573РФ1. Конструктивно они выполнены на МП К580ИК80 на двусторонней плате размерами 200×240 мм.

С помощью программы управления ККС идентифицируется адрес КС, выдаются команды управления и принимается информация от сменных блоков. Обмен информацией с И2 и формирование слова-состояния технических средств — функции ККП. Пять сменных блоков (СБ) могут быть одновременно подключены с помощью кроссплаты к ККС (общее число разработанных СБ — девять). СБ преобразуют стандартные сигналы датчиков (0...20 мА, 0...5 мА, 0...10 В), частотные или амплитудно-кодированные в цифровой код, который считывается ККС.

Интерфейс ККС — упрощенный, включает в себя 8-разрядную шину данных, 3-разрядную шину адреса и 4-разрядную шину сигналов управления (запись, чтение, выбор сменного блока, считывание первого и второго байтов).

Особенности контроллеров — отсутствие ОЗУ и возможность работы как в ККП, так и в ККС. Информация обрабатывается с помощью внутренних регистров процессора. Каждый контроллер имеет свой адрес, устанавливаемый движковыми переключателями S1...S5. Информация выдается в линию связи микросхемой последовательного интерфейса К580ВВ51 побайтно. Через оптронную развязку сигналы поступают на усилитель. В устройстве реализована защита от короткого замыкания, а также от одновременной работы двух и более аналогичных устройств.

Усилитель позволяет передавать информацию по кабелю на расстояние до 5000 м при числе приемников до 8 и до 1000 м — при их увеличении до 30. В качестве буферных схем используются ИМС серий К561 и К155.

ККП опрашивает КС, входящие в состав системы, а ККС передает информацию об измеренных параметрах на КП. Полный кадр, передаваемый ККС, — 18 байт, 12 из них — информационные. Для передачи информации о каждом параметре используются два байта.

При совпадении адресов (принятого от КП и собственного) контроллер обрабатывает команду и формирует сигналы управления сменными блоками и датчиками, затем производит съем информации с выходных регистров сменных блоков, выполненных на ИС с тремя состояниями. После кодирования информация транслируется в линию связи. Программное обеспечение контроллеров разработано на ассемблере с помощью микроЭВМ ВЭФ.

123371, Москва, Волоколамское шоссе, д. 91, ВНИИ БП; тел. 491-42-45. Овечкин Андрей Сергеевич

Статья поступила 18.10.88

Портативная аппаратура должна быть рассчитана на минимальное энергопотребление. Среди многообразия современных МП БИС наиболее удобно использовать МП-комплект серии К588 [1]. Его достоинство — программная совместимость на уровне машинных кодов и аппаратная — по интерфейсу МПИ с микро- и мини-ЭВМ ряда «Электроника 60» и СМ4. Это позволяет использовать их в качестве инструментальных ЭВМ при разработке ПО контроллеров, полностью реализуя преимущества работы в операционной системе с трансляторами языков высокого уровня.

Основные технические характеристики контроллера

Напряжение питания, В	5±10 %
Потребляемый ток, А	0,25
Быстродействие, тыс. оп./с	300

В состав микроконтроллера входят: ЦП с 16-разрядным арифметическим логическим устройством (АЛУ) и управляющей памятью (УП) микропрограмм, реализующей систему команд микроЭВМ «Электроника 60»; ПЗУ (16 Кбайт); ОЗУ (4 Кбайт). Обмен данными с внешними устройствами осуществляется через магистральный интерфейс [2].

Принципиальная схема ЦП приведена на рис. 1. В его состав включаются: интегральные микросхемы (ИМС) 16-разрядного АЛУ, пять ИМС УП, системный контроллер и три 8-разрядных приемопередатчика. Особенность ИМС серии К588 — наличие схем синхронизации, определяющих фазы работы БИС.

Функционирование АЛУ начинается по спаду сигнала сопровождения микрокоманды (рис. 2), который выдается УП [3]. Выполнение микрокоманды состоит из комбинации следующих фаз:

приема (прием и дешифрация микрокоманды, иногда прием операнда по магистрали данных);
чтения (чтение информации из регистра общего назначения (РОН), аккумулятора, регистра состояния (РС), канала АЛУ и ее обработка);

записи (окончание обработки информации и ее запись в приемник: РОН, аккумулятор, РС, канал);

выдачи результатов операции в магистраль данных.

УП формирует последовательность микрокоманд по заданной команде 4 и устанавливается в исходное состояние при подаче уровня Лог. 0 на входы R0 и R1 (рис. 3).

Для согласования внутреннего интерфейса процессора с внешним стандартным интерфейсом МПИ используется системный контроллер (СК), формирующий сигналы квитирования KB1, KB2, KB3, КП для управления УП и АЛУ. 4-разрядная магистраль ПР0...ПР3 сообщает УП коды прерываний. С помощью сигналов ОБМ, ЧТ, ЗП, ПЗП, ОТВ по 16-разрядной шине передаются команды адресов, данных и векторов прерываний АД0...АД15 [5]. СК вырабатывает сигналы ЗМ, РЗМ, ПЗ для организации прямого доступа; С1, С2 — для управления направлением передачи магистральных приемопередатчиков. ЦП — асинхронное устройство, поэтому для его работы необходимо достаточное время ожидания сигнала СПИ, поступающего от памяти или внешних устройств, которое ограничивается устойчивостью работы ЦП и снижением производительности микроконтроллера. Для этого к выводу КОШ подключается RC-цепь с постоянной времени $\tau_{мл}$ — 1,5...2 мкс.

Адрес модуля памяти устанавливается на входах А13...А15 (рис. 4). С помощью сигналов ВК0, ВК1 выбирается младший и старший байты 16-разрядного слова памяти соответственно. К выводам ЗЧ и З3 подключаются параллельные RC-цепи, которые определяют задержку выда-

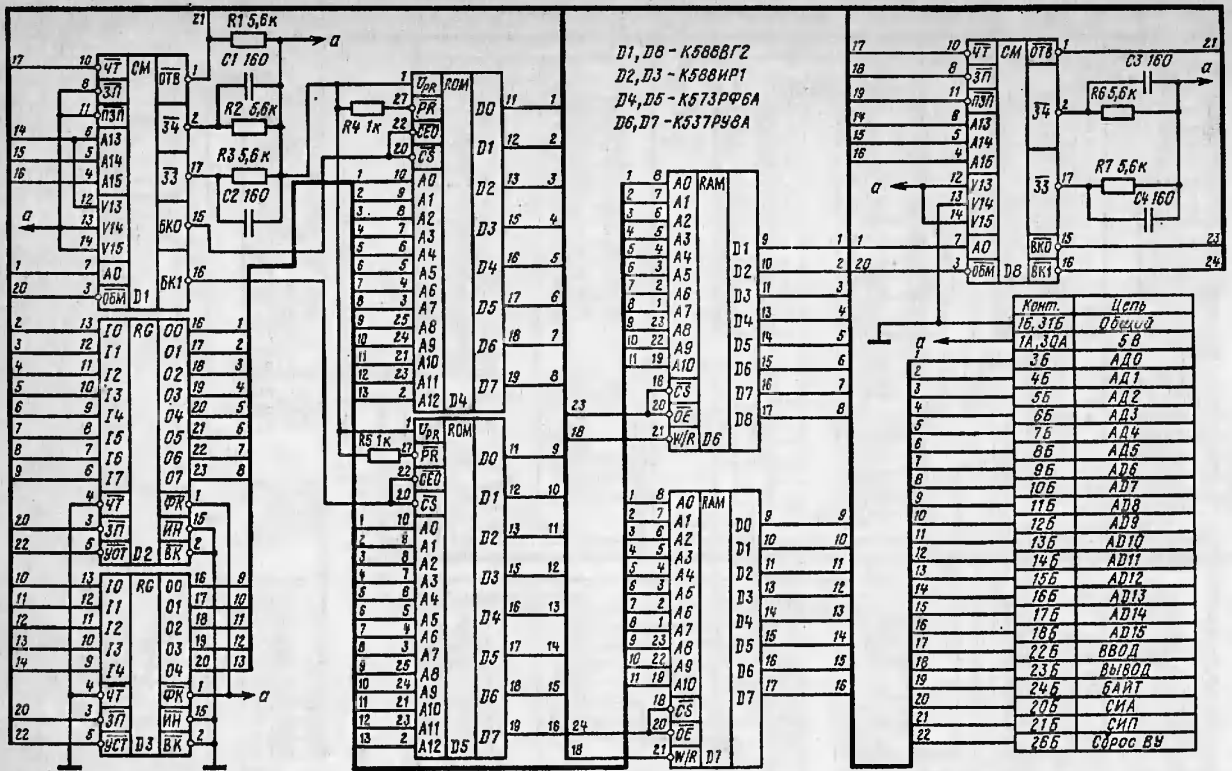


Рис. 4. Принципиальная схема модуля памяти

чи сигнала СИП относительно ВК0 и ВК1 при считывании и записи данных в память.

Для устойчивой работы ЦП с модулем памяти необходимо обеспечить соотношение $\tau_{\text{мп}}/\tau_{\text{ч/з}} > 1,5$, где $\tau_{\text{ч/з}}$ — постоянная, определяющая время выборки микросхем памяти.

Все внешние устройства, подключаемые к микроконтроллеру, должны быть совместимы по сигналам обмена с интерфейсом МПИ. Для магистральных приемопередатчиков нагрузка по току должна быть не более 8 мА для $I_{\text{вых}}^0$ ($U_{\text{вых}} = 0,4\text{В}$) и 0,4 мА для $I_{\text{вых}}^1$ ($U_{\text{вых}}^1 = 4,1\text{В}$).

Необходимо учесть, что БИС серии К588 критичны к емкости печатного монтажа и соединительных кабелей, длина которых не должна превышать 0,5 м. Для передачи информации на большие расстояния следует устанавливать интерфейсные БИС, организующие последовательный канал передачи данных [5].

Для настройки контроллера и подбора корректирующих цепей авторами использовался анализатор логических состояний TR-9588 производства ВНР, позволяющий записывать одновременно в 40 каналах глубиной до 1024 бит.

В составе автоматизированной системы к магистрали микроконтроллера были подключены дополнительные устройства: таймер, цифро-аналоговый и аналого-цифровой преобразователи, контроллер последовательного интерфейса. Эксплуатация показала высокую эффективность устрой-

ства при решении задач управления информацией и ее обработки.

Телефон 245-23-93, Ленинград

ЛИТЕРАТУРА

1. Фаулжер Р. Программирование встроенных микропроцессоров. — М.: Мир, 1985.
2. МикроЭВМ. Практическое пособие в 8 кн. Кн. 1. Семейство ЭВМ «Электроника 60» / Под ред. Л. Н. Преснухина. — М.: Высшая школа, 1988. С. 18—20.
3. Микропроцессоры. Архитектура и проектирование микроЭВМ. Учебное пособие для вузов в 3 кн. Кн. 1. Организация вычислительных процессов. / Под ред. Л. Н. Преснухина. — М.: Высшая школа, 1986. С. 321—327.
4. Микропроцессоры и микроЭВМ в системах автоматического управления / Под ред. С. Т. Хвоца. — Л.: Машиностроение, 1987. С. 208—255.
5. В. А. Бобков, Б. Н. Чернуха, В. С. Свиридович, В. П. Ключников. Расширенный микропроцессорный комплект БИС серии 588 // Микропроцессорные средства и системы. — 1987. — № 1. — С. 6—7.

Статья поступила 23.05.88

УДК 681.3.023

Е. А. Ходаковский

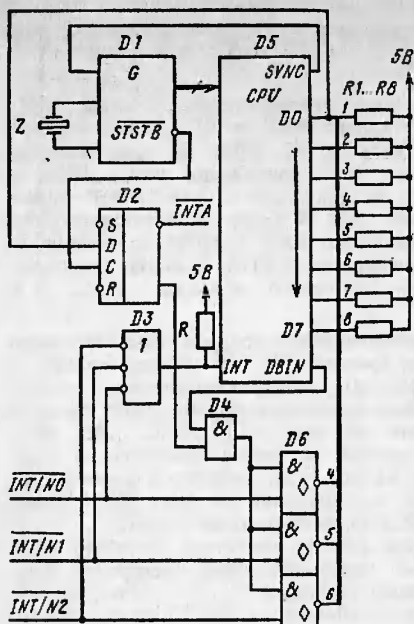
КОНТРОЛЛЕР ПРЕРЫВАНИЙ

Использование традиционных БИС для организации прерываний процессора (например, таких как К589ИК14, К580ВН59) в микросистемах больших

размеров затруднено (энергопотребление — критический параметр).

С помощью контроллера прерываний (рис. 1) можно значительно снизить

энергопотребление. Контроллер включает в себя следующие формирователи: кода рестарта (D6, R1...R8), сигнала INTA (D2), сигнала запроса прерыва-



D1 - KP580GF24; D2 - K533TM2; D3 - K533LA4; D4 - K533LI1; D5 - KP580BM00A; D6 - K533LA9

Рис. 1. Фрагмент центрального процессора с встроенным контроллером прерываний

ния, поступающего на микропроцессор (D3), и сигнала чтения кода рестарта (D4). Мощность, потребляемая контроллером от источника питания составляет около 70 мВт.

Работа контроллера. При появлении сигнала или группы сигналов на линиях INTIN0...INTIN2 D3 вырабатывает сигнал запроса прерывания (Лог. 1), который поступает на вход INT микропроцессора. По окончании выполнения текущей команды в первом машинном цикле по сигналу SYNC микропроцессор записывает в триггер D2 Лог. 1 и считывает код рестарта, поступающий с выходов D6 (при установке внутреннего триггера разрешения прерывания микропроцессора в единичное состояние по команде EI и D1). Значение вектора RST определяется состоянием линий D3...D5 шины данных (при этом на остальных линиях шины данных должны быть сигналы Лог. 1, обеспечиваемые резисторами R1...R8), поэтому при возникновении сигналов на линиях INTIN0...INTIN2 программа адресует на одну из ячеек памяти области рестарта микропроцессора. Адреса перехода зависят от состояний линий INTIN0...INTIN2 (см. таблицу). Следует учитывать, что при одновременной установке двух или трех запросов прерываний необходимо отметить приоритетность их обработки в ячейках рестарта, а также ввести операторы,

Состояние сигналов			Адрес ячейки рестарта программы
INTIN2	INTIN1	INTIN0	
H	H	H	—
H	H	L	0008H
H	L	H	0010H
H	L	L	0018H
L	H	H	0020H
L	H	L	0028H
L	L	H	0030H
L	L	L	0038H

ставящие «метку» в ячейку ОЗУ, сигнализирующую программе обработки прерываний об устройствах, запросивших прерывания.

Например, две программы обработки прерываний PROG0 и PROG1 реагируют на сигналы INTIN0 и INTIN1 соответственно (первая имеет высший приоритет по отношению к второй). При активизации одного из сигналов микропроцессор перейдет к считыванию ячеек 0008H или 0010H, поэтому в них необходимо записать

```

0008H  MVI A,INT0; загрузить метку INT0 в
        STA ADR; аккумуляторе
        JMP PROG0; запомнить метку в ячейке
        ; перейти к выполнению обработки прерываний
        ; программы PROG0
0010H  MVI A,INT1; загрузить метку INT1 в
        STA ADR; аккумуляторе
        JMP PROG1; запомнить метку в ячейке
        ; перейти к выполнению обработки прерываний
        ; программы PROG1

```

В ячейке ADR ОЗУ ставится метка INT0, ЛИБО INT1, указывающая запрос прерывания устройства по линии INTIN0 или INTIN1 соответственно.

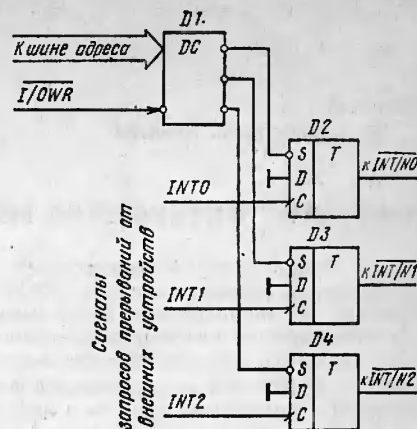
При одновременном установлении сигналов INTIN0 и INTIN1 микропроцессор переходит к считыванию ячейки 0018H, содержащей информацию о приоритетности. В данном случае в этом сегменте необходимо поместить

```

0018H  MVI A,INT01
        STA ADR
        JMP PROG0

```

Метка INT01 означает, что прерывание запросили одновременно два сигнала INTIN0 и INTIN1, а команда IMP PROG0 обслуживает в первую очередь программу PROG0, так как она имеет высший приоритет по отношению к PROG1. При выходе микропроцессора на программу обработки прерываний, последняя должна сбросить сигналы, запросившие прерывание, используя информацию ячейки ОЗУ ADR. Для снятия запроса на прерывание микропроцессор должен активизировать соответствующую линию сброса, воздей-



D1 - селектор адреса D2...D4 - K533TM2

Рис. 2. Фрагмент схемы внешнего устройства, формирующей сигналы запросов прерываний

ствующую на элемент, инициирующий запрос. На рис. 2 изображен фрагмент внешнего устройства, в котором в качестве элементов, формирующих запросы прерываний, используются D-триггеры (D2, D3, D4). Нарастающий фронт сигналов INT0...INT2, поступающих от внешних устройств, переводит триггеры в активное состояние. Сброс триггеров организован через селектор адреса D1. При этом микропроцессор должен выполнить команду записи в порт любого кода по адресу, активизирующему одну из трех линий селектора адреса D1. В нашем примере фрагмент программы обработки прерываний выглядит следующим образом:

```

PROG0: LDA ADR; загрузить метку в аккумуляторе
        CPI INT0; прерывание от сигнала INTIN0?
        CZ RES0; если да, вызвать сброс сигнала INTIN0
        CPI INT1; прерывание от сигнала INTIN1?
        CZ RES1; если да, вызвать сброс сигнала INTIN1
        CPI INT01; прерывание от двух сигналов INTIN0 и INTIN1?
        CZ RES01; если да, сбросить оба сигнала

EI; разрешить прерывание
RET; возврат в основную программу

RES0:  OUT ADRO; сброс сигнала INTIN0;
RES1:  OUT ADRI; сброс сигнала INTIN1;
RES01: OUT ADRO; сброс сигнала INTIN0
        OUT ADRI; сброс сигнала INTIN1

```

ADRO, ADRI — порты вывода, активизирующие линии установки триггеров D2, D3 в единичное состояние при записи в них любого кода.

Телефон 444-88-28, Киев

Статья поступила 23.05.88

ИМИТАТОР ИНТЕРФЕЙСА МПИ

Для обмена тестовой информацией с настраиваемым контроллером внешних устройств (КВУ) в режимах однократного или многократного ввода-вывода целесообразно применять простое в изготовлении вспомогательное устройство, автономно формирующее сигналы интерфейса МПИ (ОСТ11.105.903—80) и выполняющее функции пассивного абонента с изменяемым адресом и информацией.

Рассматриваемый имитатор (рис. 1 и 2) в отличие от описанного в [1] имитатора канала микроЭВМ «Электроника 60» выполняет функции не только активного, но и пассивного абонента на МПИ; реализован на микросхемах большей степени интеграции, что упрощает его наладку и повышает надежность работы.

Многokrратно повторяющиеся циклы ввода и вывода тестовой информации, формируемые имитатором, облегчают при наладке наблюдение на экране осциллографа или логического анализатора периодически повторяющихся сигналов МПИ, а также контроль прохождения этих сигналов в различных точках настраиваемого КВУ. Адрес и данные, используемые для обмена информацией с настраиваемым КВУ, набираются с помощью переключателей, расположенных на лицевой панели. Выводимые или вводимые информации (адреса и данные) контролируются с помощью светодиодов, сгруппированных в триады для упрощения пересчета двоичных кодов адреса (данных) в восьмеричные.

Для проверки и наладки активных устройств имитатор переключается в режим пассивного абонента канала МПИ, вводимые (выводимые в) из имитатора данные запоминаются на различных регистрах, на отдельном регистре устанавливается и адрес абонента, к которому обращается проверяемое устройство. Имитатор можно использовать в режиме прослушивания магистрали для установления адреса обращения или данных, которыми обменивается конкретный адресат.

В активном режиме на панели устанавливаются код выполняемой инструкции (Запись слова, Чтение слова и т. д.), адрес ведомого КВУ, данные для передачи в КВУ и запускается имитатор. ИМС К1802ВВ2 вырабатывает необходимые по заданной инструкции управляющие сигналы МПИ, сигналы управления ИМС К1802ВВ1, которые осуществляют передачу данных (установленных на ТРД) по адресу КВУ (заданному на ТРА) или считывание данных из КВУ, с адресом, заданным на ТРА.

В пассивном режиме по управляющим сигналам МПИ, поступающим от активного КВУ в РПА, записывается адрес по фронту сигнала К ОБМ Н, сравнивается с адресом имитатора (при совпадении кодов РПА и РВА ИМС К1802ВВ1 вырабатывает сигнал Свой В на выходе Г) и по сигналу Свой В данные или записываются в РПО, или считываются с РВД в МПИ по каналу Х.

В режиме прослушивания в РПА с МПИ записывается адрес, в РПД — данные по сигналам К ДЗП Н и К ДЧТ Н.

Имитатор работает следующим образом. При включении напряжения питания триггер (М1-1,2) устанавливается в нулевое состояние низким уровнем напряжения с кнопки S2, генератор тактовых импульсов вырабатывает сигналы ТИ В, поступающие на вход CLK ИМС (М8, М3). Для установки в нулевое состояние имитатора и КВУ используется кнопка S4, при нажатии которой формируется сигнал УСТ. «0» В, поступающий на вход CLR2 ИМС (М8), и сигнал К УСТ Н, поступающий в МПИ.

В активном режиме работы имитатора (тумблер S1 в положении Активный) код выполняемой инструкции устанавливается с помощью тумблеров S9...S12 (Инструкция). Имитатор как активный абонент на МПИ может работать в режимах однократного и многократного запусков.

Режим однократного запуска. Если тумблер S3 находится в положении «Однокр.», запуск инициируется нажатием и отпусканием кнопки S2. Импульсом с триггера (М1-1,2) ИМС (М8) обнуляется по входу CLR1, а триггер М2-1 устанавливается в «1» и совместно с триггером М2-2 формирует одиночный (задержанный синхронно с тактовой частотой) импульс записи инструкции, поступающий на вход WF ИМС (М8). В зависимости от заданной инструкции ИМС (М8) формирует все необходимые управляющие сигналы МПИ (К ОБМ Н, К ДЗП Н, К ДЧТ Н и др.), а также сигналы управления записью чтением ИМС (М4...М7), которые поступают через комбинационную схему (М3) на входы АХ0, АХ1, WХ, ЕСХ, RХ ИМС (М4...М7). Передаваемый в МПИ адрес КВУ набирается с помощью тумблеров S29...S44 ТРА и передается по каналу В в регистр (RG3) и в линии К АД0 Н... К АД15 Н МПИ через канал Х ИМС (М4...М7). При выполнении инструкции «Запись слова» данные устанавливаются на тумблерах S13...S28 ТРД, передаются в регистр задачи RG1 и в линии К АД0 Н... К АД15 Н через канал Х, сопровождаются управляющим сигналом К ДЗП Н. При выполнении инструкции Чтение слова ИМС (М8) вырабатывает сигнал на выходе RE1, по которому разрешается запись информации в регистр (RG2) из КВУ через канал Х ИМС (М4...М7). В режиме однократного запуска формируется только один цикл об-

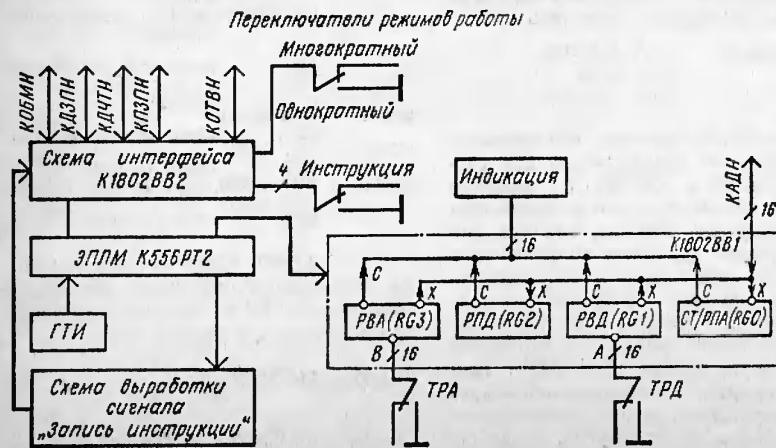


Рис. 1. Структурная электрическая схема имитатора

16-разрядные: ТРА, ТРД — тумблерные регистры адреса и данных; РВА — регистр выдачи адреса; РПД — регистр приема данных; РВД — регистр выдачи данных; РПА — регистр приема адреса (в активном режиме имитатора используется как таймер цикла СТ), реализованные на ИМС К1802ВВ1 [2]; К ОБМ Н, К ДЗП Н, К ДЧТ Н, К ПЗП Н, К ОТВ Н — канальные сигналы МПИ с низким уровнем активности, формируемые схемой интерфейса К1802ВВ2 [2]; ГТИ — генератор тактовых импульсов; ЭПЛИМ К556РТ2 [3] — комбинационная схема

Таблица истинности ЭПЛМ

Акт./Пас.																									Комментарии	
	ВУ Н	ОБМ В	Свой В	ДЗП В	ДЧТ В	Многокр./Одн.	Пуск Н	ПЗП В	16ТИ 65536ТИ	ТИ В	Запас	WE1 Н	WE2 Н	RE1 Н	RE2 Н	ТИ Н	Пуск Н	ОТВ В	АХ1 В	АХ0 В	WX Н	EX Н	RX Н			
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	B7	B6	B5	B4	B3	B2	B1	B0			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25		
1	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	1	Н	1	0	—	—	—	А	А	—	А	А	Разрешение чтения адреса с РА (RG3)
1	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	0	1	—	—	—	—	—	—	—	А	А	Разрешение чтения данных с РВ (RG1)
1	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	0	Н	1	1	—	—	—	А	—	А	А	—	—	Разрешение записи в РПД (RG2)
1	Н	Н	Н	Н	Н	Н	0	Н	Н	Н	Н	Н	Н	Н	Н	Н	—	А	—	—	—	—	—	—	—	Запуск ИМС (M8) от кнопки S2
1	Н	Н	Н	Н	Н	1	Н	Н	0	Н	Н	Н	Н	Н	Н	Н	—	А	—	—	—	—	—	—	—	Запуск ИМС (M8) многократный
1	Н	Н	Н	Н	Н	Н	Н	Н	Н	1	Н	Н	Н	Н	Н	Н	—	—	—	—	—	—	—	—	—	Формирование ТИ Н
0	0	1	Н	0	0	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	—	—	—	—	—	А	А	—	—	По сигналам К ВУ Н и К ОБМ Н адрес с МПИ записывается в RG0
0	1	1	1	1	0	0	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	—	—	А	А	—	А	А	—	—	По сигналу К ДВП Н данные записываются в RG2
0	1	1	1	0	1	0	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	—	—	А	—	А	—	А	А	—	По сигналу К ДЧТ Н данные считываются с RG1
0	1	1	1	1	0	1	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	—	—	—	А	—	А	А	—	—	Режим прослушивания при сигнале К ДЗП Н
0	1	1	1	0	1	1	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	—	—	—	А	8	А	А	—	—	Режим прослушивания при сигнале К ДЧТ Н

ЛИТЕРАТУРА

1. Комар Л. И., Якубович В. М., Усачев О. И. Радиотехника.— 1984.— Т. 39, № 8.
2. Микросхемы интегральные серии K1802. Руководство по применению. ОСТ11. 348.915—82.
3. Микросхемы интегральные серии K556. Руководство по применению. ОСТ11.340.915—82.

Статья поступила 12.12.87.

УДК 681.325

Е. Ф. Накалов, С. П. Тюлькин

УНИВЕРСАЛЬНЫЙ ИНТЕРФЕЙС ВВОДА-ВЫВОДА С ЭЛЕМЕНТАМИ САМОПРОВЕРКИ

Традиционный метод проверки портов ввода-вывода — внешняя коммутация выходных портов на входные. Предлагаемый интерфейс наряду с возможностью работы любого порта на ввод и на вывод может осуществлять самоконтроль этих портов и выходных буферов.

Для решения любой задачи проверки работоспособности аппаратуры необходим тестер (устройство, генерирующее тестовые воздействия и оценивающее результаты проверки). Эти функции в ЭВМ выполняет центральный процессор (ЦП).

Один из наиболее удобных методов самопроверки — метод с постепенным расширением зоны охвата. По этому методу узлы ЭВМ включаются в область саопроверки постепенно.

Этапы диагностики: ядро ЭВМ; работоспособность портов и элементов сопряжения с ВУ; стык ЭВМ — ВУ самого ВУ устройства.

Описываемое устройство наряду с сопряжением с ВУ проверяет работоспособность портов и элементов сопряжения. Предложенная схема интерфейса (рис. 1) позволяет: осуществить соединение с мощными входами-выходами ВУ;

использовать каждый порт на ввод или вывод, подключая выходы ВУ к входным контактам шинных формирователей (ШФ), либо входные контакты ВУ к выходным контактам ШФ;

тестировать работоспособность БИС КР580ВВ55 и ШФ их свободными контактами.

Эти функции делают интерфейс универсальным. Следует отметить, что ВУ защищены от воздействия тестовых последовательностей (при переключении ШФ на нерабочие входы или выходы рабочие выходы переводятся в высокоимпедансное состояние).

Пример универсального интерфейса ввода-вывода при настройке первого и второго портов на режим вывода в ВУ показан на рис. 2.

Проверка стыка ЭВМ-ВУ и самого ВУ без отключения этих входов может осуществляться функционально (на ВУ через стык ЭВМ-ВУ подаются тестовые воздействия; заключение о работоспособности делают по его ответам). Причину отказа стыка трудно точно идентифицировать, так как разные неисправности могут одинаково проявлять себя. Поэтому для устранения уже обнаруженного отказа на стыке ЭВМ-ВУ либо в самом ВУ можно использовать традиционный метод проверки портов ввода-вывода — отключить ВУ от ЭВМ и замкнуть выходные порты на входные. Однако, этот метод имеет ряд ограниче-

К центральному процессору

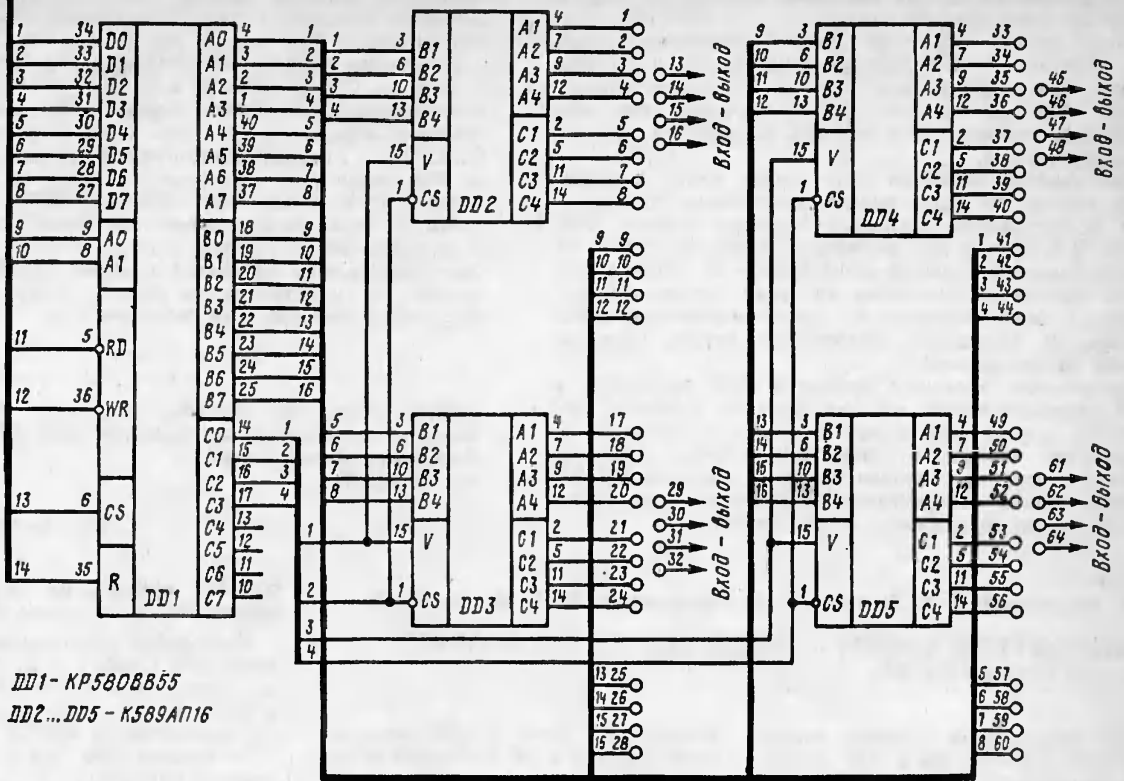


Рис. 1. Принципиальная схема универсального интерфейса

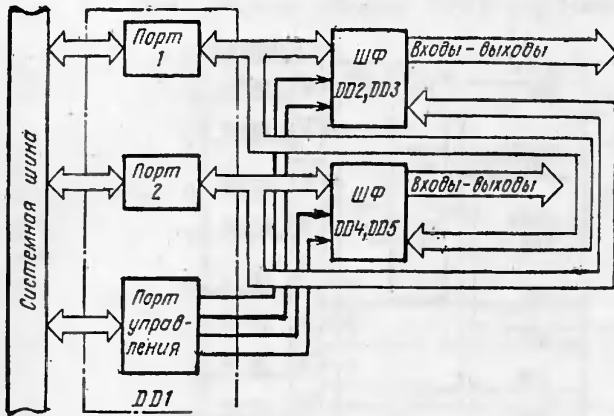


Рис. 2. Пример построения универсального интерфейса для режима первого и второго портов на вывод

ний: не всегда легко отключить ВУ от ЭВМ; могут отсутствовать входные, либо выходные порты.

Применение предложенного устройства облегчает традиционный метод тестирования портов ввода-вывода и позволяет создать эффективную систему самопроверки портов ввода-вывода ЭВМ. При наличии только ввода дополнительные входы и выходы ШФ могут использоваться как тестовые для проверки работоспособности стыка ЭВМ-ВУ.

290005, Львов, Ватутина, 5, ОКБ ЭИВТ ЛПИ;
тел. 72-88-35

Статья поступила 23.05.88

УДК 681.3.06

Л. А. Вознюк, А. Д. Горожин

ПРОГРАММНАЯ ЭМУЛЯЦИЯ ДИСПЛЕЯ ВТА 2000-15 НА БАЗЕ ДИСПЛЕЯ «СИМВОЛ»

В качестве консоли в составе микроЭВМ нередко используется дисплей «Символ» (отдельные комплексы СО-01, СО-04, СО-05). Однако некоторые его функциональные особенности препятствуют работе с такими программами, как экранный редактор Word Master, директор BEDITS, текстовый процессор Word Star, баз данных d BASE-II, игровые программы.

В дисплее «Символ» отсутствуют непосредственное позиционирование курсора в произвольное знакоместо экрана и свертка кадра, т. е. его перемещение вверх при достижении нижней границы экрана осуществляется на две строки; буфер экранной памяти, содержащий в дисплее, существенно превышает информационный объем экрана.

При разработке программного эмулятора (ПЭ) в качестве эмулируемого был выбран дисплей ВТА 2000-15. Этот дисплей имеет достаточно развитую систему команд, что облегчает настройку экранно-зависимых программ.

Эмуляция заключается в реализации программной обработки на дисплее «Символ» кодов управления курсором, очистки и реформирования текста дисплея ВТА 2000-15. Вывод символа сводится к непосредственному отображению его на экране дисплея. При выводе управляющих кодов эмулятор, используя управляющие коды дисплея «Символ», работает с курсором и содержимым экрана в соответствии с управляющими кодами.

Например, при поступлении последовательности управления курсором, состоящей из четырех кодов: авторегистра,

управления курсором, номера строки и номера позиции в строке, ПЭ устанавливает курсор в начало экрана, а затем перемещает его на требуемое число строк вниз и требуемое число позиций вправо.

Такой способ обработки кодов управления курсором обеспечивает ПЭ высокую устойчивость и помехозащищенность. Даже при сбое в программе пользователя ПЭ не требует реинициализации, так как ошибочная информация о текущих координатах курсора не накапливается.

Для свертки кадра на одну строку копия содержимого экрана по мере вывода информации запоминается в специально созданном экранном буфере ОЗУ микроЭВМ, причем при достижении последней строки на экране происходит прямой сдвиг буфера на строку вверх, и его содержимое выводится на экран. Таким образом, операция сверки выполняется с использованием экранного буфера, не затрагивая содержимого буфера экранной памяти самого дисплея.

Организация экранного буфера в ОЗУ микроЭВМ в виде последовательной цепочки символов упрощает вычисление адреса при выводе символа в произвольное знакомство экрана, а также выполнение операций очистки и перформирования текста. С помощью ПЭ без дополнительных аппаратных затрат реализован также режим вывода содержимого экрана на печать.

ПЭ написан на ассемблере МП К580ВМ80А, занимает вместе с экранным буфером 3,5 Кбайт оперативной памяти и работает в ОС СР/М-80 любой версии. Эмулятор выполнен в виде перемещаемого модуля, расположенного в ОЗУ «выше» или «ниже» ОС СР/М.

В начале работы ПЭ модифицирует три вектора перехода — CONST, CONIN и CONOUT базовой системы ввода-вывода BIOS. Таким образом обеспечивается перехват всех обращений к дисплею. Если ПЭ размещен ниже ОС СР/М, т. е. перед процессором консольных команд ССР, по ПЭ, кроме того, модифицирует адрес входа в BIOS (ячейки ОЗУ с адресами 0006H и 0007H) для того, чтобы не быть уничтоженным системными программами. К недостаткам ПЭ следует отнести некоторое снижение быстродействия по сравнению с дисплеем ВТА 2000-15, что, впрочем, не существенно при работе с программами типа экранных редакторов и им подобных.

252056, Киев, пр. Победы, 37, корп. 18, к. 509, Киевский политехнический институт, каф. ВТ, Горожину Александру Дмитриевичу; тел. 441-95-66

Статья поступила 4.04.88

УДК 681.325

В. Г. Бескровный, А. А. Урсатьев, С. Л. Сапожникова, М. А. Шестаковский

НАКОПИТЕЛЬ СМ5211 — ВНЕШНЕЕ ЗУ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Для подключения к каналу микроЭВМ «Электроника 60» и ДВК кассетных накопителей на магнитной ленте (КНМЛ) СМ5211 различных модификаций, выходящих на промежуточный интерфейс ИРПР [1] и ИРПР-ВТА,

разработаны блоки сопряжения — интерфейсы ВЗУ и ЭВМ. Первый из них выполнен в конструктиве Е2, устанавливается на свободное место в каркасе ВЗУ вместо блока СМ5211/730011, сопрягающего ВЗУ СМ5211.19 с ВТА

2000-10, второй — на полной плате микроЭВМ «Электроника 60».

Интерфейсы обеспечивают функционирование СМ5211 с КНМЛ СМ5206 под управлением ОС РАФОС, РАФОС 2 и ряда версий, включающих драйвер СТ накопителя на КНМЛ.

По емкости (320...400 блоков информации в зависимости от числа файлов) сторона кассеты практически эквивалентна 8-дюймовому гибкому диску при скорости движения ленты 38 см/с.

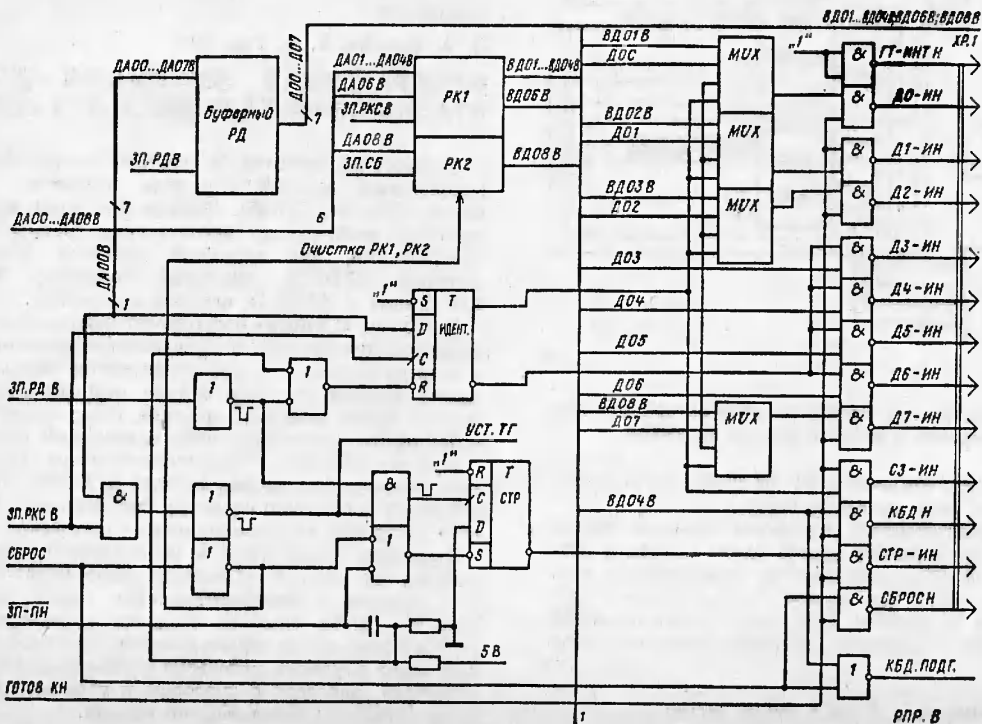


Рис. 1. Узел передачи команд и данных в ВЗУ

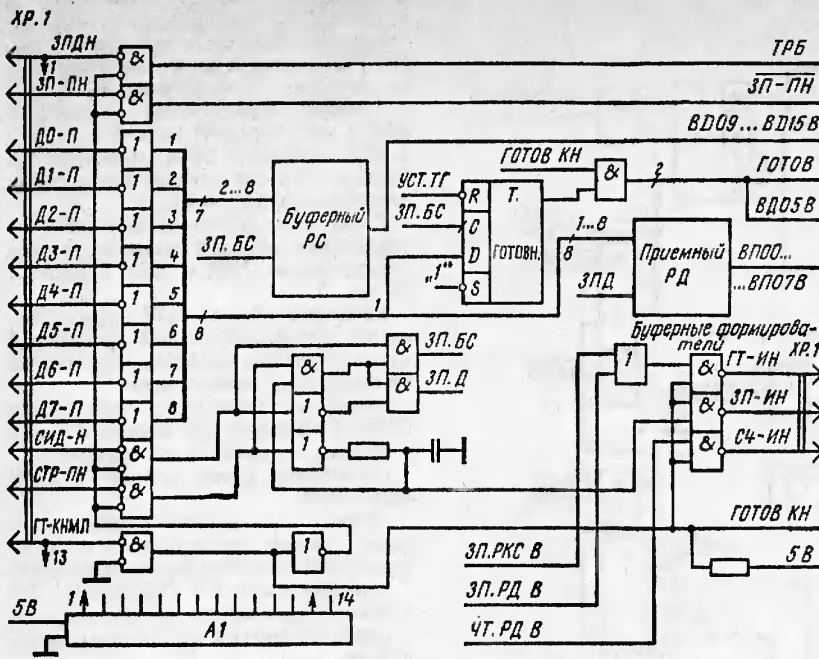


Рис. 2. Узел приема информации с ВЗУ

КНМЛ SM5206 фирмы Robotron допускают движение носителя со скоростью 19 см/с. Работа с КНМЛ в течение достаточно длительного времени (на цифровых кассетах типа 490 производства ГДР с лентой ORWO) показала высокую надежность функционирования и воспроизводимости записей на различных накопителях.

КНМЛ использовались в системе регистрации и экспресс-анализа статистических характеристик сигналов радиолокационных отражений [2] в качестве внешних долговременных ЗУ при организации архивов данных.

Интерфейс ВЗУ имеет два программно-доступных регистра: регистр команд и состояния (РКС) с адресом 177500₈ и регистр данных (РД) — 177502₈ (адрес вектора прерывания — 260₈).

В структуре интерфейса можно выделить четыре блока:

адресный селектор, каналные приемопередатчики и управляющая логика; узел передачи команд и данных (УПК и Д); узел приема информации (УПИ) с ВЗУ;

логика управления прерыванием (ЛУП).

УПК и Д (рис. 1) обеспечивает прием байта информации из канала ЭВМ и его передачу в интерфейс ВЗУ. В него входят: РД, регистр команд (РК), мультиплексоры команд-данных, буферные формирователи (передатчики) и логика управления, формирующая сигналы сопровождения передаваемого байта информации (СТР-ИН, СЗ-ИН), сигналы конца блока данных, передаваемых ВЗУ (КБД Н), сигналы выбора накопителя с номером «0» или

«1» и разрешения прерывания программы (РПР В).

Байт информации поступает в РД или РК (по шинам ДА 00...ДА 08 В) с каналных приемопередатчиков по фронту сигналов ЗП.РД В, ЗП.РКС В, ЗП.СБ. Стробирование на линиях интерфейсного соединителя происходит по срезу сигналов ЗП.РД В или ЗП.РКС В при установке в ноль триггера Т.СТР. По сигналу запроса интерфейса ВЗУ на прием информации от ЭВМ (ЗП-ПН) этот триггер переходит в состояние Лог. 1, сигнал СТР-ИН снимается.

Идентификация байта команды производится сигналами ЗП.РКС В и ДА 00 В; триггер Т.ИДЕНТ. принимает состояние ЛОГ.1 и мультиплексоры транслируют на вход буферных формирователей содержимое РК1, РК2 (при этом формируется сигнал СЗ-ИН). В исходное, нулевое, состояние триггер Т.ИДЕНТ. перебрасывается по срезу сигнала ЗП-ПН либо по фронту ЗП.РД.

УПИ с ВЗУ (рис. 2) обеспечивает прием байта состояния или данных с интерфейса ВЗУ и их дешифрацию в канал ЭВМ.

Узел содержит приемники информации с интерфейсного соединителя, буферный регистр состояния (РС), приемный регистр данных (ПРД) и логику управления, вырабатывающую сигналы записи байта данных в регистры РС или ПРД (ЗП.БС, ЗП.Д соответственно). Эти сигналы формируются по фронту строба СТР-ПН в соответствии с уровнем сигнала на шине СИД-Н. Сигнал запроса интерфейса ЭВМ на прием нового байта данных от ВЗУ ЗП-ИН формируется по срезу сигнала

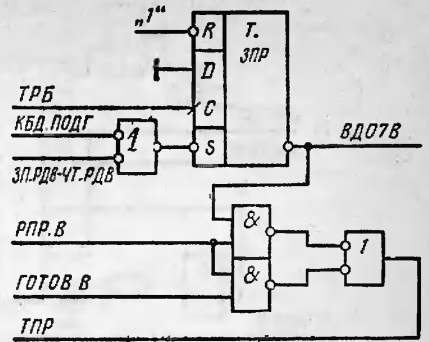


Рис. 3. Схема формирования сигнала требования прерывания

записи информации в один из регистров (РС или ПРД). В этом же блоке создается сигнал готовности интерфейса ЭВМ к передаче данных ВЗУ и сигнал подтверждения приема ЭВМ байта данных (ГТ-ИН и СЧ-ИН соответственно).

Нулевой разряд принимаемого с ВЗУ байта состояния дешифрируется триггером готовности (Т.ГОТОВН.) накопителя к работе с ЭВМ. Его установка в единицу свидетельствует о завершении команды КНМЛ. Триггер очищается сигналом УСТ.ТГ, вырабатываемым при совпадении сигналов ЗП.РКС и ПУСК. Следующие семь разрядов байта состояния (шины ВД01...ВД04 В; ВД06 В; ВД08 В) дешифрируются соответственно функциональному назначению разрядов РКС.

Содержимое РК и РС, а также триггера Т.ГОТОВН. (линии ВД 01...ВД15 В) считывается в канал ЭВМ. Канальные приемопередатчики стробируются сигналом ЧТ.РКС Н. Структура РКС полностью идентична структуре регистра команд и состояний устройства внешней памяти (УВП) на кассетной магнитной ленте УВК SM ЭВМ, например, УВП SM5208, SM3, SM4.

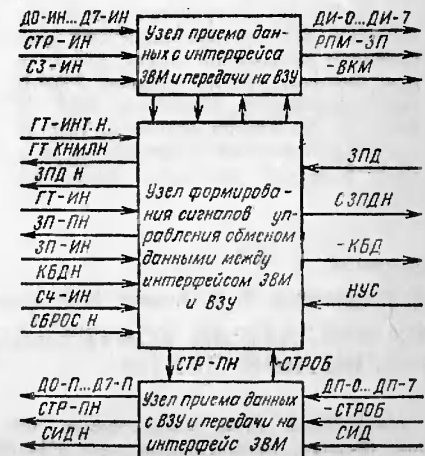


Рис. 4. Блок-схема интерфейса ВЗУ

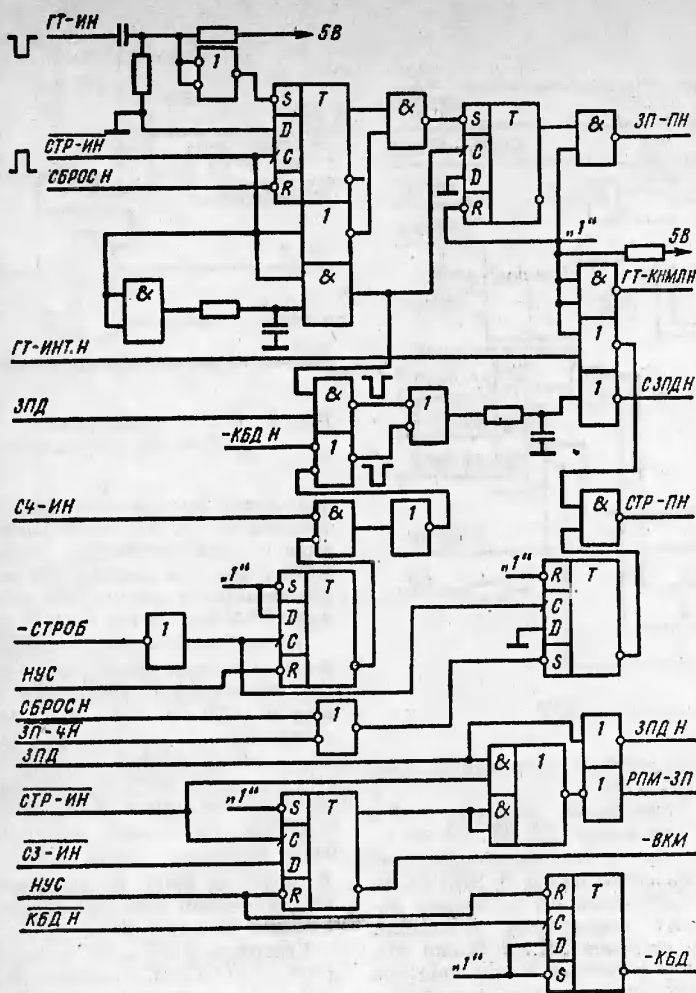


Рис. 5. Функциональная схема узла формирования сигналов управления интерфейса ВЗУ

Условия прерывания программы ЦП при работе с ВЗУ на кассетной МЛ: готовность ВЗУ к принятию данных с ЭВМ или их передаче в ЭВМ; завершение выполнения команды ВЗУ.

В первом случае триггер запроса прерывания (ЗПР) устанавливается в нуль сигналом требования байта (ТРБ) и при высоком уровне сигнала РПР.В ЛУП формирует сигнал К ТПР.Н (рис. 3). Во втором случае К ТПР.Н возбуждается высоким уровнем сигнала ГОТОВ.В (по окончании очередной

команды выбранным КНМЛ или по его готовности к работе) и идентифицируется нулевым разрядом байта состояния ВЗУ (пятый разряд РКС).

ЗПР переходит в исходное состояние по сигналам обращения к регистру данных (ЗП.РД.В и ЧТ.РД.В) и КБД. ПОДГ, формируемым при начальной установке устройств (сигнал СБРОС) и записи единицы в четвертый разряд РКС в ответ на $n+1$ ТРБ.

Блок-схема интерфейса ВЗУ приведена на рис. 4. В качестве приемников интерфейса и УПИ с ВЗУ используются

микросхемы К155ЛН1, К155ЛН1 и К155ЛЕ1. С помощью делителей напряжения (на стороне приемников) ИС ТТЛ согласуются с линией связи — соединителем интерфейсным (СИ).

Функции передатчиков интерфейса ВЗУ и узла передачи команд и данных в ВЗУ интерфейса ЭВМ выполняют буферные ИС с повышенной нагрузочной способностью (К155ЛА13).

Схема узла формирования сигналов управления обменом данными между интерфейсами ЭВМ и ВЗУ приведена на рис. 5.

Работоспособность ВЗУ проверяется специально разработанной программой «Тест устройства СМ5211». Она состоит из следующих подпрограмм:

проверка готовности устройства СМ5211 к выполнению команд, а также в режиме прерывания программы; считывание блока данных в ОЗУ микроЭВМ;

запись блока данных и обмен данными между устройством и ОЗУ микроЭВМ по опросу флага готовности; передача данных с ОЗУ микроЭВМ в режиме прерывания программы; вывод на дисплей необходимых сообщений;

обработка прерываний.

Разработана конструкторская документация на интерфейс ВЗУ и ЭВМ, а также комплект документации для изготовления последнего на интерфейсе пользователя (И5).

252207, Киев, пр. Академика Глушкова, 20, ИК АН УССР; тел. 265-30-35, 265-55-40

ЛИТЕРАТУРА

- Новиков Е. А., Чаадаев В. Г. Сопряжение кассетного накопителя на магнитной ленте СМ5211 с интерфейсом МПИ // Микропроцессорные средства и системы. — 1988. — № 2. — С. 76—78.
- Урсатьев А. А., Серебряников Ю. Л., Сапожникова С. Л., Тарасенко С. А. Комплекс программно-технических средств регистрации и анализа радиолокационных сигналов // Статистические методы в теории передачи и преобразования информационных сигналов: Тез. докл. Всесоюз. научно-техн. конф. — Киев: КНИГА, 1988. — С. 135.

Статья поступила 3.05.88

УДК 621.377

В. А. Давыдов, В. М. Козлов, В. Г. Попов

ОДНОПЛАТНЫЙ КОНТРОЛЛЕР МАГНИТНОЙ ЛЕНТЫ

МикроЭВМ типа «Электроника 60» комплектуются, как правило, накопителями типа СМ5300. Для них выпускается блочный контроллер 1588МЛ-10-001 и технически более совершенный контроллер МС2707.

Пользователям микроЭВМ «Электроника 60» известны и другие контроллеры, однако публикаций по ним нет. Технические характеристики этих контроллеров не лучше, чем у МС2707, за исключением компактности (есть полуплатные контроллеры).

Промышленностью начат выпуск НМЛ с повышенной скоростью обмена и улучшенными массогабаритными характеристиками, но для них пока нет быстродействующих контроллеров.

Предлагаемый контроллер (рис. 1) пригоден по скоростям обмена для использования с НМЛ серий СМ и ЕС.

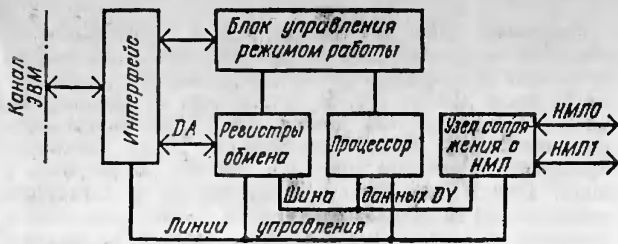


Рис. 1. Структурная схема контроллера

В нем предусмотрена удобная система отладки микропрограмм и аппаратной части*. Контроллер используется со стандартным драйвером и ТМОС.

Основные технические характеристики базовой модели

Тип накопителя	СМ5300.01, СМ5303, СМ5305, Н75.01, ЕС5002.03, ЕС5612
Тип интерфейса	МПИ
Потребляемая мощность, В·А, не более	25
Габаритные размеры, мм	294×262×12,5
Масса, г, не более	550
Метод записи	Без возврата к нулю (инверсный)
Скорость передачи данных, Кбайт/с	10...96
Число НМЛ	2

* А. С. 1288708 (СССР). Устройство для сопряжения ЦВМ с накопителями на магнитной ленте / В. А. Давыдов, В. Г. Попов, В. М. Козлов, Л. С. Чесалин — Опубл. 1987, Бюл. № 5, с. 206.

Физические адреса регистров обмена (РО), их структура и логика обмена полностью удовлетворяют стандартным драйверам и тестам для НМЛ. Исключением является регистр режима (РР) с адресом 172536(8) (табл. 1).

Таблица 1

Регистр	Адрес
Состояния (РС)	172 520
Команд (РК)	172 522
Счета байтов (РСБ)	172 524
Текущего адреса (РТА)	172 526
Ввода-вывода (РВВ)	172 530
Продольного контроля (РПК)	172 532
Режима (РР)	172 536

Связь контроллера с каналом ЭВМ обеспечивается с помощью схемы, приведенной на рис. 2. Приемопередатчики (ПР/ПЕР) и БИС обмена К1802ВВ2 принимают и выдают сигналы адреса-данных (16 разрядов) и управления. В программном режиме БИС обмена принимает сигналы К СИА Н, К ВВОД Н, которые в инвертированном виде выдаются на внутренние схемы контроллера. Кроме того, она транслирует сигналы ППР и ППД для других устройств. Дешифратор адреса (ДША) контроллера вырабатывает сигнал 1725ХХ при обращении по стандартным адресам регистров. Этот сигнал вместе с четырьмя младшими разрядами адреса записывается на время цикла канала в фиксатор адреса (ФА). Устройство управления адресом обмена (УРО), выполненное на ПЛМ, управляет регистрами обмена (РОМ — младший байт, РОС — старший байт). Чтение осуществляется 16-разрядным словом, а запись — байтами.

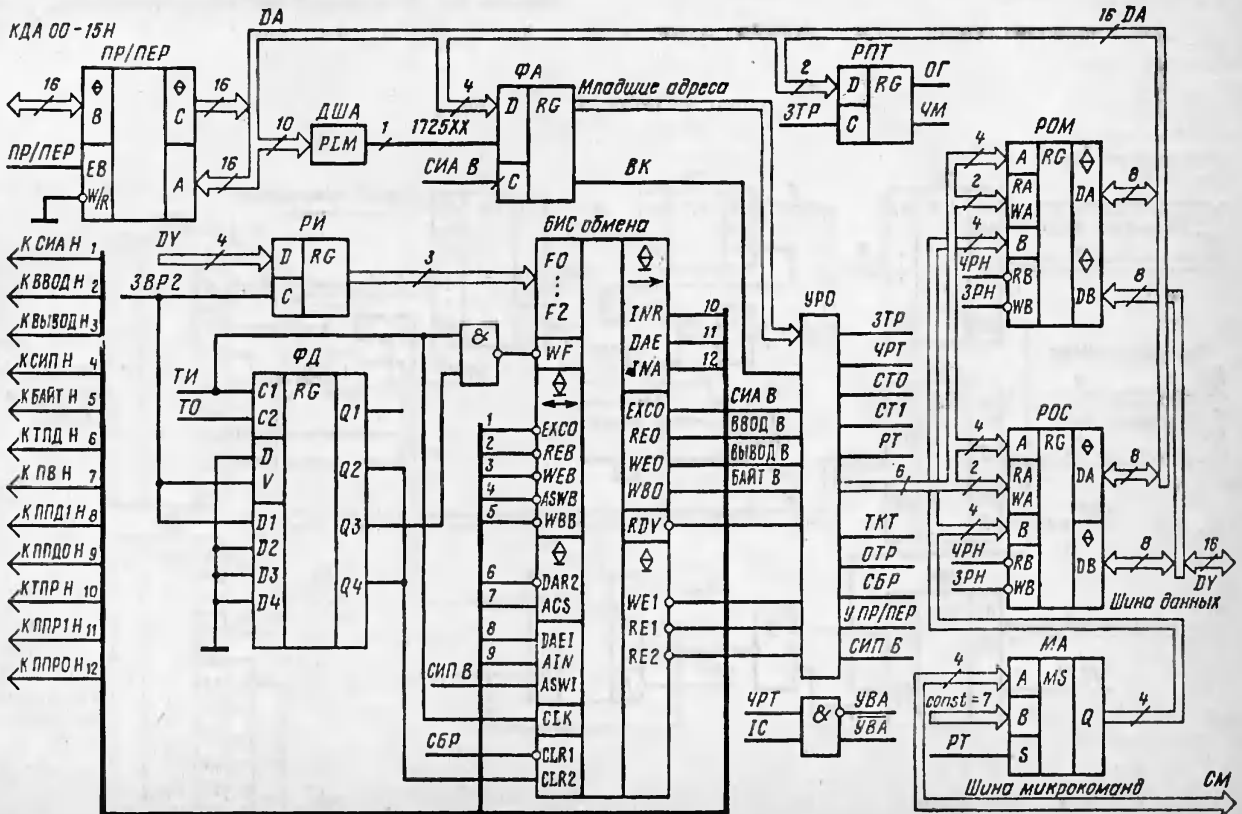


Рис. 2. Функциональная схема узла с каналом ЭВМ

Таблица 2

Процедура	F0	F1	F2
Чтение слова ПДП	0	0	0
Запись слова ПДП	0	1	1
Запрос прерывания	1	0	1
Сброс БИС обмена	1	1	1
Запись байта ПДП	1	0	0

В режимах прямого доступа (ПДП) или прерывания процессор исполняет микрокоманду (МК) вывода и вырабатывает импульс записи ЭВР2 в регистр инструкции (РИ). По шине данных ДУ передается код инструкции (три разряда). РИ необходим для согласования «быстрой» шины и «медленной» БИС обмена. Типы выполняемых процедур приведены в табл. 2.

Сигналом ЭВР2 инициируется работа регистра формирования диаграммы (ФД), с помощью которого очищается БИС обмена и записываются инструкции по сигналу WF. БИС обмена обрабатывает соответствующую диаграмму по тактам ТИ. Диаграммы циклов обмена полностью совпадают со стандартными для микроЭВМ «Электроника 60». БИС обмена кроме канальных сигналов вырабатывает сигналы для управления внутренними схемами контроллера, например сигналы RE1, RE2, WE1 для управления регистрами РОМ и РОС. В режиме записи слова или байта контроллер выставляет сигнал К ПД Н, принимает К ППД1 и при этом занимает канал (К ПВ Н, К СИА Н). По сигналу RE2 выбирается один из РО 172526(8), содержащий адрес ячейки оперативной памяти ЭВМ; по сигналу WE1 — данные из регистра 172530(8). Сигнал RE1 используется при запросе прерывания для выбора вектора прерывания, хранящегося в одном из РО с адресом 1000(2). Дешифрация сигналов RE1, RE2, WE1, RGY осуществляется в ПЛМ УРО.

Процессор (рис. 3). Длина МК и, соответственно, шина МК процессора СМ имеют разрядность 32. Процессор принимает 16-разрядные данные, обрабатывает и передает их по шине данных ДУ. В зависимости от типа применяемых микросхем ПЗУ память МК (ПМК) может быть расширена до 4096 32-разрядных слов. Процессор проверяет 32 внешних признака (из них 16 разрядов с шины ДУ) и выполняет арифметические и логические операции над 16-разрядными словами, циклические сдвиги, условные и безусловные переходы, переход на подпрограммы, ввод константы, обмен данными между внешними регистрами и регистрами процессора.

В состав процессора входят четыре секции АЛУ К1804ВС1 со схемами ускоренного переноса К1804ВР1, схема управления следующим адресом (СУСА) К1804ВУ3, формирователь адреса МК (ФАМ) на трех микросхемах К1804ВУ2, регистр МК (РМК) на восьми микросхемах К1804ИР1.

Неуправляемые выходы РМК образуют шину МК СМ. Управляемые (трестабильные) выходы соединены по 16 разрядов и подключаются к шине данных ДУ. Процессор организован таким образом, что любая МК с проверкой условия (наиболее длительная) будет закончена за один такт и результат проверки зафиксирован в триггере условий (ТУ). На адресные входы мультиплексора условий (МУ) с шины СМ подается 5-разрядный код одного из выбранных условий, а на схему ИСКЛ.ИЛИ — 27-й бит шины СМ — инвертор условий. Выход ТУ подключен ко входу ТST схемы управления следующим адресом, что и определяет вместе с четырьмя битами с шины СМ выбор с помощью ФАМ источника адреса МК. Параллельно СУСА включена микросхема, расширяющая возможности управления ФАМ (РУСА).

Типы МК, используемые в процессоре, приведены в табл. 3, обозначения полей показаны ниже:

10...18 — код команды АЛУ процессора;

A0...A3, B0...B3 — номера регистров процессора;

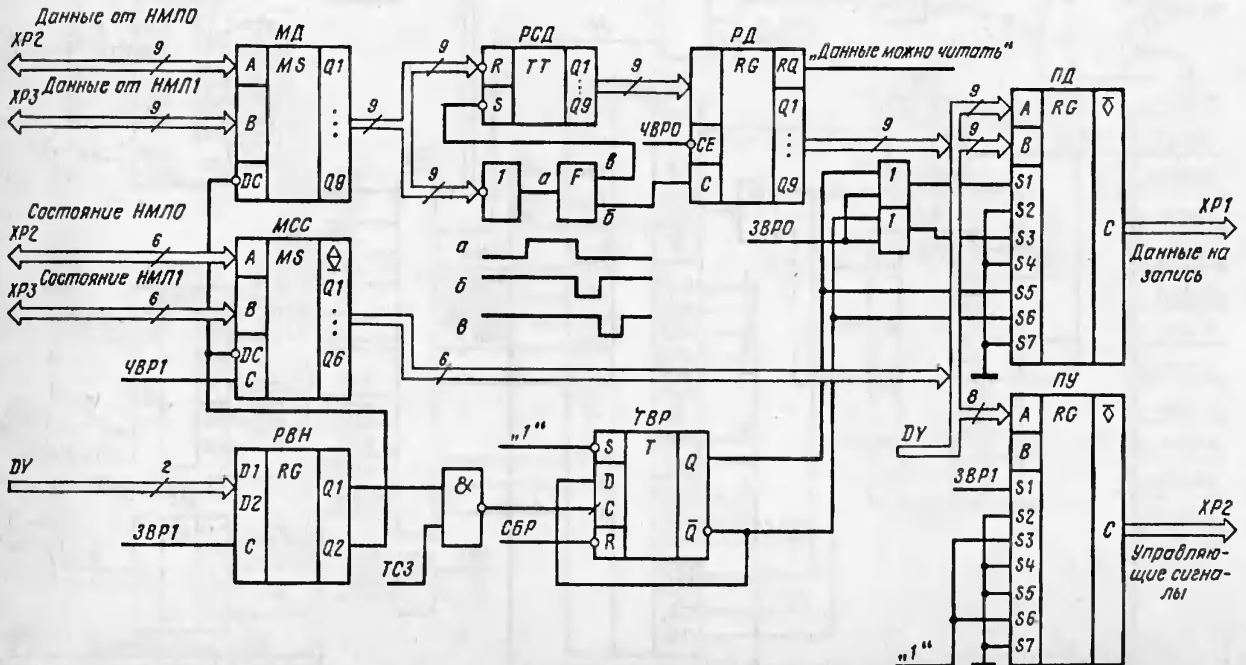


Рис. 3. Функциональная схема процессора контроллера

Команда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Процессорная	0	1	CO	B3	B2	B1	B0	18	17	16	15	14	13	12	11	10
Ввод	1	1	CO	B3	B2	B1	B0	18	17	16	15	14	13	12	11	10
Вывод	1	1	CO	B3	B2	B1	B0	18	17	16	15	14	13	12	11	10
Ввод константы	1	0	CO	B3	B2	B1	B0	18	17	16	15	14	13	12	11	10
Адрес перехода	0	0	—	—	W11	W10	W9	W8	W7	W6	W5	W4	W3	W2	W1	W0

Команда	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Процессорная	X3	X2	X1	X0	TST	Z4	Z3	Z2	Z1	Z0	SH1	SH0	A3	A2	A1	A0
Ввод	X3	X2	X1	X0	TST	Z4	Z3	Z2	Z1	Z0	1	RG	S3	S2	S1	S0
Вывод	X3	X2	X1	X0	TST	Z4	Z3	Z2	Z1	Z0	0	RG	S3	S2	S1	S0
Ввод константы	K15	K14	K13	K12	K11	K10	K9	K8	K7	K6	K5	K4	K3	K2	K1	K0
Адрес перехода	X3	X2	X1	X0	TST	Z4	Z3	Z2	Z1	Z0	1	—	—	—	—	—

SH0, SH1 — источник входного переноса (CO из МК или выходной перенос из последней секции АЛУ);
 TST — значение проверяемого условия (0 или 1);
 X0...X3 — команда схемы управления следующим адресом;
 CO — вход переноса АЛУ процессора;
 S0...S3 — адрес внешнего регистра;
 RG — тип внешнего регистра;
 K0...K15 — константа;
 W0...W11 — адрес перехода.

Разряды МК 21, 15 и 14 поступают на дешифратор команд (ДШК) и определяют тип МК: PR — процессорная, обеспечивает выполнение арифметических операций и сдвигов; IN — ввод данных в процессор из внешних регистров; OU — вывод данных из процессора во внешние регистры; IC — ввод констант из поля МК; AD — занесение адреса перехода в ФАМ.

частоты (ДЧ). Программируемый таймер (ПТ) управляется от процессора по шине DY. Основное назначение ПТ — задание периода записи на НМЛ и временных задержек при выполнении различных операций.

Сопряжение с НМЛ (рис. 4). Процессор выбирает нужный НМЛ, записывая 13-й разряд в регистр выбора НМЛ (РВН), который устанавливает направление мультиплексоров данных МД и состояний (МС). Сигналом ЧБР1 сигналы состояния, поступившие с НМЛ, читаются процессором по шине DY. Обработка девяти разрядов данных происходит иначе. Каждый бит подается на отдельный RS-триггер, где записывается на время приема одного байта. Кроме того, все девять разрядов подаются на схему ИЛИ, а затем на формирователь (Ф), который по срезу последнего пришедшего бита выдает собственно импульс записи данных в регистр данных

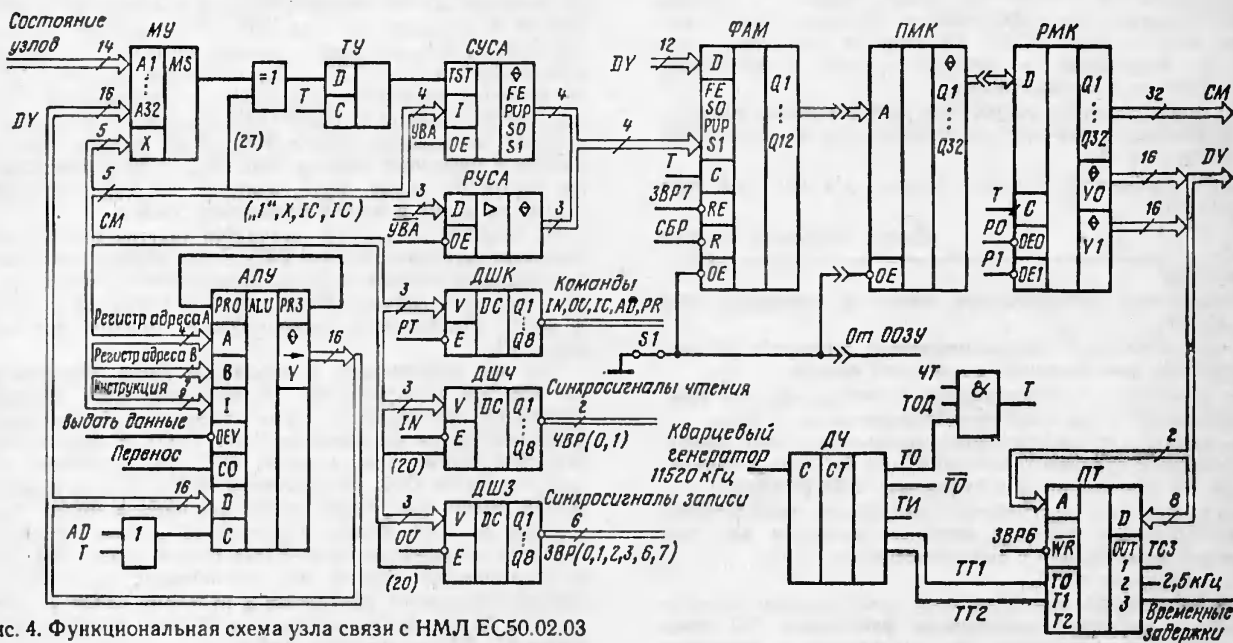
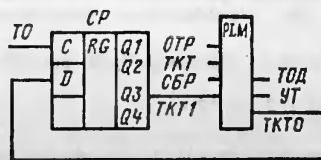


Рис. 4. Функциональная схема узла связи с НМЛ ЕС50.02.03

Дешифраторы чтения (ДШЧ) и записи (ДШЗ) формируют импульсы для чтения или записи различных регистров в контроллере. Синхронизация узлов процессора осуществляется подачей тактовых импульсов. Основную частоту задают кварцевый генератор 11,52 МГц и делитель



(РД) и после него — импульс установки RS-триггеров РСД в исходное состояние (в 1). Чтение РД происходит сигналом ЧВРО. Момент записи данных в РД процессор определяет по флагу, заведенному на МУ.

Для передачи на НМЛ сигналов управления и данных используются микросхема приемопередатчика с двумя буферными регистрами К583КП2. Через входы передатчика управляющих сигналов (ПУ) процессор заносит с шины DY необходимые команды для НМЛ, сопровождая их сигналом ЗВР1. Для передачи данных на НМЛ процессор предварительно устанавливает 11-й разряд в РВН, разрешая тем самым прохождение тактов сопровождения записи (ТСЗ) на триггер выбора регистра (ТВР). Последний по входам передатчика данных (ПД) S5 и S6 устанавливает один из внутренних регистров ПД на запись данных с шины DY (сигналом ЗВРО), другой — на выдачу данных, записанных в предыдущем такте, на НМЛ. Таким образом, процессор заносит в ПД данные в сравнительно произвольные моменты времени, а период выдачи на НМЛ остается постоянным.

Отладка контроллера. Переход в режим тестирования и отладки приводит к останову основных тактирующих импульсов (сигналов УТ), вырабатываются сопутствующие сигналы, обеспечивающие чтение двух слов МК. Пользователь с пульта ЭВМ или с помощью специальных тестовых программ может задавать любой адрес МК в ПМК (используя регистр 172536(8)), записывать МК в РМК и проверять срабатывание узлов контроллера в статике; считывать (за два цикла чтения) любую МК; запускать выполнение микропрограммы с любого адреса; работать в пошаговом режиме.

Контроллер отлаживается на микроЭВМ «Электроника 60» с использованием специального отладочного ОЗУ (ООЗУ), полностью имитирующего применяемые ПЗУ. ООЗУ доступно только по чтению со стороны контроллера (т. е. включается непосредственно в панельки ПЗУ). Формат считываемого слова совпадает с форматом МК. Плата ООЗУ вставляется в системный разъем микроЭВМ и имеет двунаправленный доступ со стороны канала к ячейкам ОЗУ. Микропрограмма или ее фрагмент загружается в ООЗУ из ЭВМ, затем с пульта ЭВМ к контроллеру подключаются адресные входы и выходы данных ячеек ООЗУ. По команде оператора контроллер включается в нужный режим и выполняет загруженную микропрограмму.

Для подготовки микропрограмм базовой модели контроллера создана кросс-система, работающая на микроЭВМ «Электроника 60».

249810, Калужская область, Таруса, а/я 50, СКБ КП; тел. 9-18-32

Статья поступила 30.03.88

РЕКЛАМА

Предлагаем лабораторные карты в стандарте IBM РС XT/AT:

- параллельный программируемый интерфейс (24 входа-выхода), совмещенный с макетной платой;
- 16-канальный 10-разрядный аналого-цифровой преобразователь с частотой преобразования до 1 МГц;
- сверхбыстродействующий аналого-цифровой преобразователь с программно-изменяемой частотой дискретизации (60 кГц—10 МГц) и буферным ОЗУ (1 Кбайт);
- усилитель с программно-изменяемым коэффициентом усиления с набором активных фильтров для совместной эксплуатации с вышеописанными АЦП;
- интерфейс ИРПС.

Все устройства комплектуются программным обеспечением с широкими сервисными функциями. ПО обеспечивает совместимость со стандартными пакетами обработки данных.

Разрабатываем нестандартные периферийные устройства по заказу.

Справки по тел. 250-56-91 с 9 до 12 ч.

103045, Москва, а/я 17.

УДК 681.32

В. И. Копылов, М. Ю. Меркулов

ЭМУЛЯТОР ДИСКА В СТАНДАРТЕ КАМАК

Известны разработки эмулятора диска (ЭД) на базе блока ОЗУ «Электроника 256К» и специального промежуточного интерфейса [1], а также на основе устройства памяти емкостью 1 Мбайт, выполненных в стандарте микроЭВМ «Электроника 60» [2]. В составе комплексов ДВК ЭД используется в качестве системного запоминающего устройства (ЗУ), поддерживающего операционную систему (ОС) RT-11, а накопитель на гибком магнитном диске является архивным устройством. ЭД существенно улучшает эксплуатационные показатели комплекса, повышает его надежность и увеличивает быстродействие системного ЗУ.

Эмулятор диска в каком-либо машинно-независимом интерфейсе позволил бы расширить указанные достоинства на комплексы, включающие ЭВМ, отличные по организации от микроЭВМ типа «Электроника 60». В современных системах автоматизации экспериментальных исследований широко распространен стандарт КАМАК, являющийся удобным средством связи ЭВМ с датчиками экспериментальной установки. Однако в нем используется принцип синхронного обмена данными, что вызывает определенные трудности при работе с модулями, быстродействие которых меньше цикла КАМАК. В частности, цикл КАМАК составляет около 1 мкс, тогда как среднее время обращения к ячейке блока ОЗУ «Электроника 256К» равно 1,7 мкс.

Разработанный модуль интерфейса блока ОЗУ «Электроника 256К» (рис. 1) в стандарте КАМАК обеспечивает обмен данными с блоком в асинхронном режиме. По команде записи NA(0)F(16) данные с шин W записываются в буферный регистр D34, D35 и взводится триггер D38(5), что вызывает установку сигнала CX3. Запись информации в блок ОЗУ сопровождается сигналом CX3, по которому снимается сигнал CX3 и сбрасывается триггер D38(5). Данные из блока ОЗУ читаются в два приема: сначала по команде чтения NA(1)F(0) данные записываются в буферный регистр D36, D37, а затем по команде N(0)F(0) могут быть считаны из буферного регистра на шины R-чтения магистрали крейта.

По команде NA(1)F(0) взводится триггер D38(9), что вызывает установку сигнала CX3. Когда данные для чтения подготовлены блоком ОЗУ, он вырабатывает сигнал CX1, по которому и записываются данные в буферный регистр, а также сбрасывается триггер D38(9) и снимается сигнал CX3.

Так как информация в эмуляторе диска представлена блоками размером 256 16-разрядных слов каждый, адресация производится двумя командами. Адрес блока устанавливается по команде NA(1)F(17) с шин записи W1...W12, причем код на шинах W11, W12 определяет номер устройства ОЗУ «Электроника 256К». Таким образом, общий объем программно доступной памяти может быть доведен до 1024 Кслов. Адрес слова в блоке устанавливается по команде NA(0)F(17) с шин записи W1...W8 и сбрасывается всякий раз по команде NA(1)F(17), чем обеспечивается обращение к нулевому слову в блоке данных. Адрес данных читается по команде NA(0)F(1) на шины R1...R20 магистрали крейта.

Драйвер ЭД разрабатывается для каждого типа ЭВМ и сопрягаемого с ней контроллера крейта КАМАК. В качестве примера ниже приводится текст драйвера (рис. 2), разработанного для конфигурации, включающей микроЭВМ «Электроника 60» и крейт-контроллер КАМАК типа СС-11. Драйвер включен в ОС RT-11 и имеет одну

D1, D32 - K155PA2; D2, D6... D9 - K155PH2; D3 - K155HD4; D4 - K155PE6; D5 - K155LE1;
 D10... D12, D33 - K155PH1; D13... D16 - K589AP26; D17... D21 - K155HE7; D22 - K155LA3;
 D23... D31 - K155PA13; D34... D37 - K155HP13; D38 - K155TM2; R1... R3 - 1кОм;
 R4 - 330 Ом; C1 - 5100 пФ; C2 - 47 мкФ; C3, C4 - 0,68 мкФ; DD1 - КД20ВА

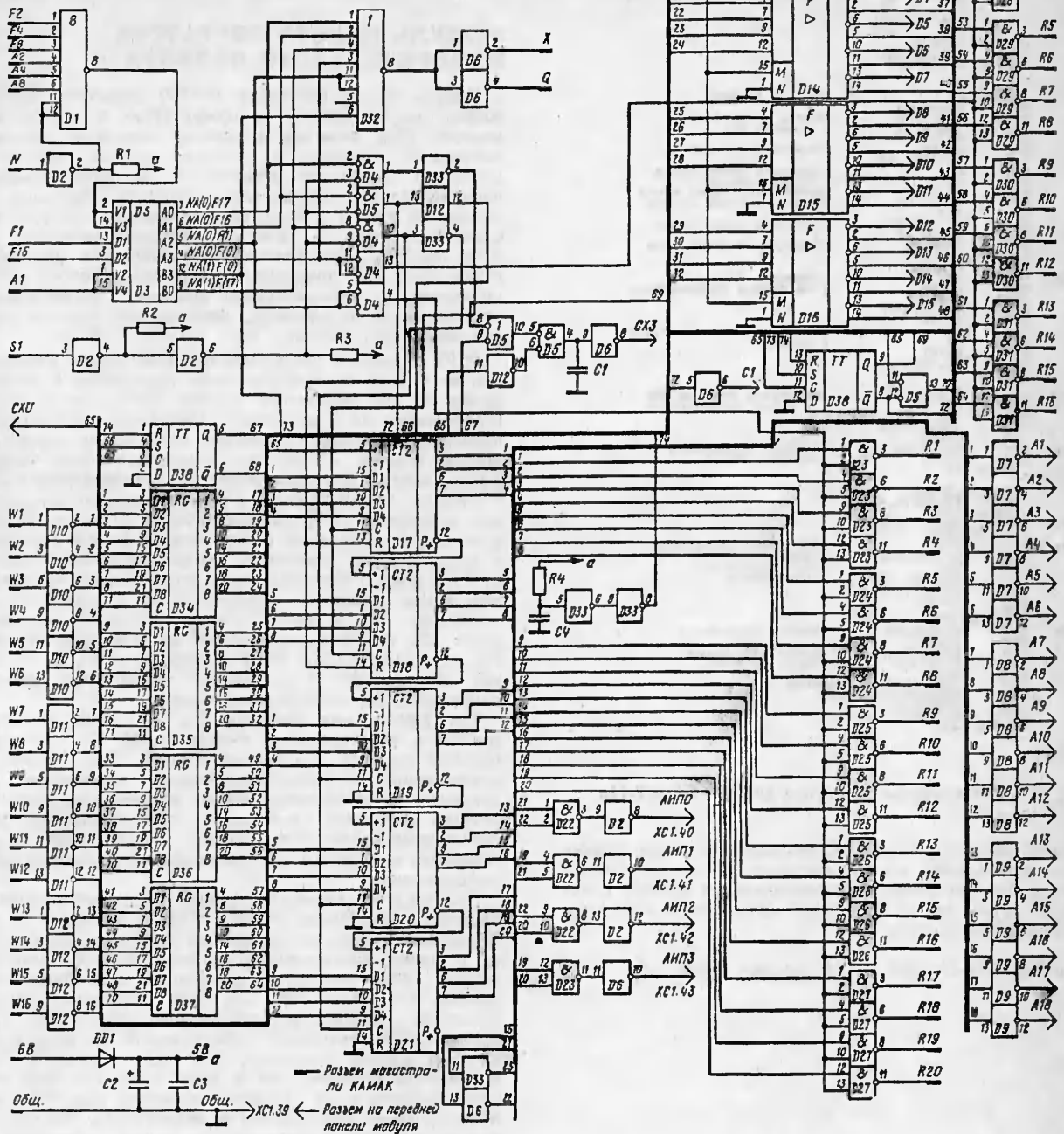


Рис. 1. Электрическая принципиальная схема интерфейса блока ОЗУ «Электроника 256К».

точку входа (за макрокомандой .DRWEG). При каждом обращении монитора к ЭД драйвер извлекает из очереди номер системного ЗУ, номер блока данных, адрес буфера пользователя и числа передаваемых слов, после чего определяет тип операции (чтение или запись). По оконча-

нии выбранной операции управление передается монитору ОС. При загрузке ОС с ЭД в ОЗУ ЭВМ копируется, а затем запускается системный загрузчик, записанный в блоках с адресами 2...5.

Разработанный ЭД в системе автоматизации, вклю-

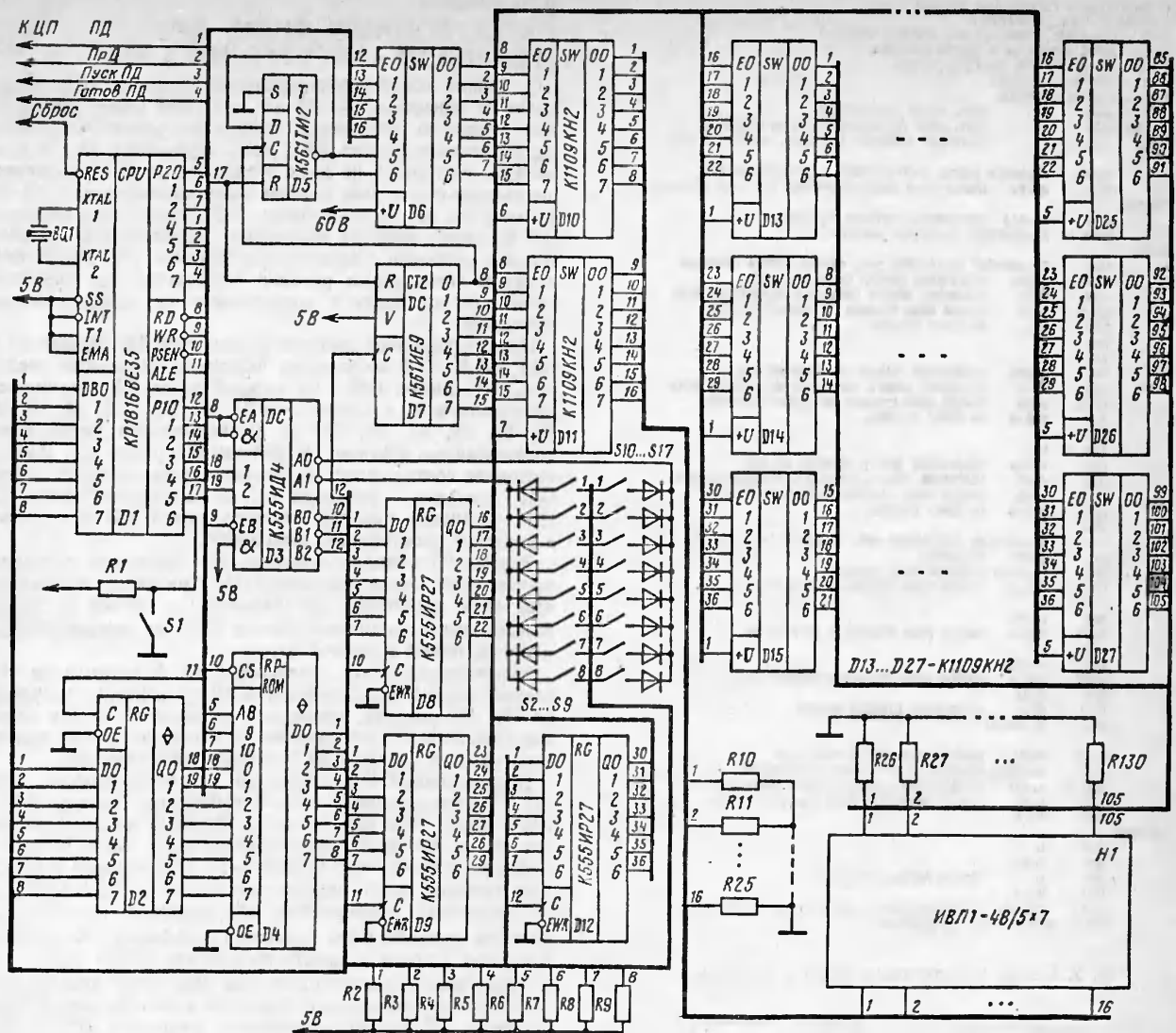


Рис. 1. Модуль пульта оператора

при выполнении ОЭВМ соответствующей операции чтения. Если обнаружена нажатая клавиша, то соответствующий ей код будет передан через последовательный интерфейс в ЦП. Дребезг контактов клавиатуры подавляется программно.

МПО обменивается информацией с ЦП прибора по последовательному интерфейсу, содержащему пять линий связи (две линии передачи данных и три линии управляющих сигналов) (рис. 3), что предпочтительнее, чем использование параллельного интерфейса, так как минимизируются аппаратные затраты на организацию интерфейса и число линий связи между ЦП и МПО, длина которых может быть значительной. Буферизация данных, передаваемых между ЦП и МПО, происходит в контроллере последовательного интерфейса, что в сочетании с использованием линий готовности и разрешения передачи позволяет организовать асинхронный режим обмена информацией между ЦП и МПО. ОЭВМ периодически проверяет готовность ЦП передать данные МПО (линия Готов ПД). При появлении готовности выполняется подпрограмма приема данных (разрешается передача данных из ЦП в МПО по линии Пуск ПД). Принятый байт преобразуется во внутреннее представление и

заносятся в буфер отображения. Выключатель S1 используется для перевода программы в режим диагностики, в котором на индикатор выводится набор символов, отображаемых МПО.

Возможно появление биений между частотами регенерации изображения на индикаторе (75 Гц) и тока накала (50 Гц), проявляющееся в дрожании изображения. Для уменьшения биений необходимо либо разности частоты регенерации и тока накала, либо синхронизировать программу регенерации с сетевой частотой. Кроме того, потенциал на катод индикатора подается через среднюю точку накальной обмотки, что также уменьшает амплитуду биений. Существует и другой способ устранения биений между частотами — питание цепи накала постоянным током. Возникающую при этом неравномерность яркости свечения из-за изменения потенциала катода вдоль его длины можно легко компенсировать программным путем, изменяя соответственно время свечения анодов.

Конструктивно МПО размещается на печатной плате размерами 220×233 мм. На одной стороне платы устанавливается индикатор, на другой — остальные элементы МПО. Клавиатура выполняется в виде отдельного

«ЭЛЕКТРОННЫЙ ДИСК» ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

В данной конструкции электронного диска (ЭД) в отличие от известных [1—4] все печатные платы стандартные. Функцию диспетчера памяти выполняет устройство параллельного обмена И2 (15КС—180—032) [5]. В микроЭВМ, состоящей из двух субблоков, можно разместить шесть дополнительных серийно выпускаемых плат ОЗУ ПЗ (размер ЭД составит 384 блока по 512 байт). Для создания ЭД большего размера необходимо использовать дополнительные субблоки с источниками питания [6]. МикроЭВМ с двумя источниками питания включается при поднятых клавишах «питание» и «программа» на дополнительном источнике.

Перед установкой дополнительных плат ПЗ движки S1.1, S3.1, S3.6...S3.8 необходимо включить, остальные отключить [7]. Вывод D34.3 на каждой плате ПЗ соединяется проводником 10 с контактами 51, 29, 33, 2, 1, 14, 10, 56, 46, 52, 19, 43, 54, 35, 48 (максимальное число плат, подключаемых к одному устройству И2, равно 15). Данное состояние соответствует адресации первых четырех банков ОЗУ, контакты — разрядам 01...15 выходного буфера И2. При установке разряда регистра плата ПЗ включается в адресное пространство микроЭВМ.

Разряд 00 выходного буфера И2 отключает с первого по четвертый банки основного ОЗУ. При работе с резидентными ОЗУ процессора устанавливаются пятый и шестой банки памяти, если резидентное ОЗУ не используется, — нулевой, пятый и шестой банки.

Управляемая плата основного ОЗУ (с первого по четвертый банки) подключается к 49-му контакту устройства И2 (00 разряд), проводник соединяется с 10-м выводом D38, дорожка печатной платы к контакту А : А3 канала перерезается (движок S2.3 должен быть включен).

Предложенный ЭД собран на 11 дополнительных платах ПЗ, размещенных в трех субблоках (размер ЭД — 704 блока). Его работа в ОС РАФОС поддерживается драйвером, который программируется на языке МакроасSEMBлера в позиционно-независимых кодах (после трансляции готовый файл переименовывается в DDS.SYS для SJ монитора). Информация об устройстве при загрузке системы автоматически заносится в таблицы резидентного монитора и затем в память по команде LOAD DD.

Временные характеристики ЭД описанной конструкции соответствуют данным, приведенным в [1]. Исходный текст драйвера ЭД на вычислительном комплексе ВУМС при размещении всех необходимых файлов на ЭД транслируется 18 с (при обычном способе — 45 с).

Телефон 26-42-72, Саратов

ЛИТЕРАТУРА

- Злотник Е. М., Стежко И. К., Анищенко В. В., Киркоров С. И. «Электронный диск» для вычислительных комплексов на базе микроЭВМ «Электроника 60» // Микропроцессорные средства и системы.— 1986.— № 5.— С. 90—92.
- Зеленко Г. В., Панов В. В., Попов С. Н. Электронный квазидиск для персональной ЭВМ // Микропроцессорные средства и системы.— 1984.— № 4.
- Сорокин Ю. Ю., Лаврентьев В. В., Максимьяк С. П., Субач В. В. «Электронный диск» для микроЭВМ «Электроника 60М» // Микропроцессорные средства и системы.— 1986.— № 5.— С. 92.
- Черняк А. Ю. Диспетчер памяти для микроЭВМ «Электроника 60» // Приборы и техника эксперимента.— 1985.— № 3.— С. 98—99.
- Устройство параллельного обмена И2 15КС—180—032. Техническое описание и инструкция по эксплуатации 3.858.383Т0.
- Центральный процессор М2. Эксплуатационные документы. 3.858.382Т0. С. 41—45.
- Запоминающее устройство ПЗ 15У30—16—004. Эксплуатационные документы. 3.858.355Т0, 33, ПЭЗ.

```

; ПОДПРОГРАММА РЕГЕНЕРАЦИИ ДИСКОВ
; РАСПРЕДЕЛЕНИЕ РЕГИСТРОВ ;
; R0 - УКАЗАТЕЛЬ СИМВОЛА В ОЗУ ДИСКЕТА (МАРКЕР)
; R1 - АДРЕС РЕГИСТРОВ И БУФЕРА СТОЛБЦОВ
; R3 - УКАЗАТЕЛЬ ЗОНЫ ЗНАКОГЕНЕРАТОРА
; R6 - СЧЕТЧИК ЗНАКОМЕСТ
; R7 - СЧЕТЧИК СТОЛБЦОВ
BUFFER EQU 16 ;НАЧ. АДРЕС ОЗУ ДИСКЕТА
SAVEAR EQU 8 ;НАЧ.АДРЕС РЕГИСТРОВ И БУФЕРА СТОЛБЦОВ
NUMY EQU 10 ;ПАРАМЕТР ЗАКЕРКИ ДЛЯ ТАКТ. ЧАСТОТЫ 6 МГц
REFR:
MOV R7,#0E0H ;АДРЕС 5-ГО СТОЛБЦА В ПОЗИЦИОННОМ КОДЕ
MOV R3,#0 ;ИНТЕР ЗОНЫ ЗНАКОГЕНЕРАТОРА ДЛЯ 6-ГО СТОЛБЦА
COLUMN:
MOV R6,#16 ;УСТАНОВКА СЧЕТЧИКА ЗНАКОМЕСТ
MOV R0,#BUFFER ;УСТАНОВКА МАРКЕРА
ROW3:
MOV R1,#SAVEAR ;УСТАНОВКА НАЧ. АДРЕСА БУФЕРА СТОЛБЦОВ
MOV A,#R0 ;ПОЛУЧЕНИЕ НОМЕРА СИМВОЛА ИЗ ОЗУ
ADD A,#R3 ;ИСКЛЮЧЕНИЕ НОМЕРА СИМВОЛА С УКАЗАТЕЛЕМ ЗОНЫ
MOV#3 A,#0A ;СИМВОЛ КОДА СТОЛБЦА ИЗ ЗНАКОГЕНЕРАТОРА
MOV OR1,A ;В БУФЕР СТОЛБЦА
INC R1
INC R0
MOV A,#R0 ;ПОЛУЧЕНИЕ НОМЕРА СИМВОЛА ИЗ ОЗУ
ADD A,#R3 ;ИСКЛЮЧЕНИЕ НОМЕРА СИМВОЛА С УКАЗАТЕЛЕМ ЗОНЫ
MOV#3 A,#0A ;СИМВОЛ КОДА СТОЛБЦА ИЗ ЗНАКОГЕНЕРАТОРА
MOV OR1,A ;В БУФЕР СТОЛБЦА
INC R1
INC R0
MOV A,#R0 ;ПОЛУЧЕНИЕ НОМЕРА СИМВОЛА ИЗ ОЗУ
ADD A,#R3 ;ИСКЛЮЧЕНИЕ НОМЕРА СИМВОЛА С УКАЗАТЕЛЕМ ЗОНЫ
MOV#3 A,#0A ;СИМВОЛ КОДА СТОЛБЦА ИЗ ЗНАКОГЕНЕРАТОРА
MOV OR1,A ;В БУФЕР СТОЛБЦА
INC R1
MOV R1,#SAVEAR ;УСТАНОВКА НАЧ. АДРЕСА РЕГИСТРОВ
MOV A,#R1 ;СТОЛБЦОВ
ORL P1,#7FH ;ИГНЕНИЕ ВСЕХ АНДОВ
MOV#A OR1,A ;СИМВОЛ КОДА СТОЛБЦА В РЕГИСТР D8
INC R1
MOV A,#R1 ;СИМВОЛ КОДА СТОЛБЦА В РЕГИСТР D9
MOV#A OR1,A ;СИМВОЛ КОДА СТОЛБЦА В РЕГИСТР D12
INC R1
MOV A,#R1 ;СИМВОЛ КОДА СТОЛБЦА В РЕГИСТР D12
MOV#A OR1,A ;СИМВОЛ КОДА СТОЛБЦА В РЕГИСТР D12
OUTL P1,A ;ВКЛЮЧЕНИЕ СТОЛБЦА АНДОВ
MOV R2,#NUMY
YL:
DJNZ R2,YL ;ЗАКЕРКА НА ВРЕМЯ ИЛИНКАЦИИ
DJNZ R6,ROW3 ;ПЕРЕХОД К СЛЕДУЮЩЕМУ ЗНАКОМЕСТУ
MOV A,#01 ;КОЛИЧЕСТВО СИМВОЛОВ ЗНАКОГЕНЕРАТОРА
ADD A,#R3 ;СДВИГ УКАЗАТЕЛЯ ЗОНЫ ЗНАКОГЕНЕРАТОРА
MOV R3,A
INPRST:
CLR C
MOV A,#R7
RR A ;МЕНА АДРЕСА СТОЛБЦА
MOV R7,A
JB7 COLUMN ;ПЕРЕХОД,ЕСЛИ НЕ ВСЕ СТОЛБЦЫ ЗАСВЕЧЕНЫ
;КОНЕЦ ПОДПРОГРАММЫ РЕГЕНЕРАЦИИ
    
```

Рис. 2. Схема подключения МПО к прибору

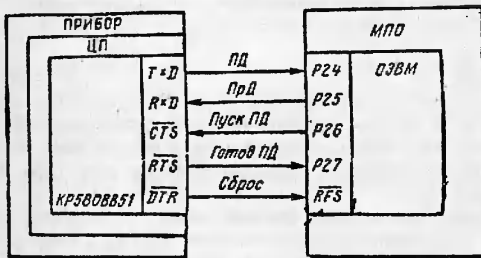


Рис. 3. Фрагмент программы, выполняющей регенерацию изображения на индикаторе

блока 4×4 клавиши и размещается на лицевой панели прибора рядом с платой МПО. При использовании в МПО ОЭВМ типа КР1816ВЕ48 вместо ОЭВМ КР1816ВЕ35 схема МПО может быть несколько упрощена за счет исключения ПЗУ (D4) и регистра адреса (D2).

Телефон 569-51-44, Ленинград

Статья поступила 14.12.87

УДК 681.322.1

А. В. Бедарев, В. С. Гравов

СОПРЯЖЕНИЕ ОДНОКРИСТАЛЬНОЙ ЭВМ КР1816ВЕ35 С ПЕРИФЕРИЙНЫМИ БИС СЕРИИ КР580

Часто при разработке аппаратуры на базе ОЭВМ КР1816ВЕ35 возникает необходимость расширить ее функциональные возможности, подключив к ней периферийные БИС серии КР580 [1, 2]. При этом порты периферийных БИС адресуются как ячейки внешнего ОЗУ ОЭВМ и возникает проблема частичной несовместимости временных характеристик сигналов ОЭВМ и периферийных БИС (рис. 1, 2).

Из диаграмм видно, что сигнал \bar{W} ОЭВМ становится активным до появления записываемого байта на шине данных ОЭВМ. Такой режим не соответствует рекомендованному из-

готовителями для части микросхем серии КР580 и приводит к неправильной работе некоторых из них, например КР580ВВ55 при включении по схеме, показанной на рис. 3 [2].

Для устранения несовместимости временных характеристик цикла записи использован узел, схема включения которого в микропроцессорную систему показана на рис. 4. Принцип его работы состоит в том, что фронт сигнала ОЭВМ задерживается на двух D-триггерах. Тактовым сигналом для этих триггеров служат импульсы, возникающие на выходе $T0$ ОЭВМ после вы-

полнения команды $EN\ T0\ CLK$. Фронт задерживается ровно на один период импульсов на выходе $T0$. Такое подключение в отличие от применения различных одновибраторов гарантирует работоспособность системы при любой тактовой частоте, а также при изменении тактовой частоты (замена кварца). Работу узла поясняют временные диаграммы, представленные на рис. 5. RC-цепь ($R1, C1$) в схеме на рис. 4 повышает устойчивость работы регистра-фиксатора адреса КР580ИР82.

ЛИТЕРАТУРА

1. 8080 Microcomputer systems user's manual. Santa Clara, Ca., USA, Intel Corp., 1975.
2. Каган Б. М., Сташин В. В. Основы проектирования микропроцессорных устройств автоматики.— М. Энергоатомиздат, 1987.

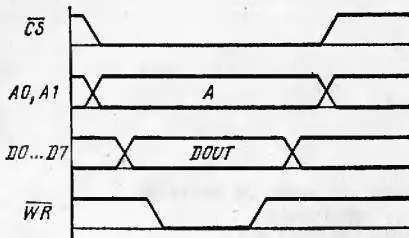


Рис. 1. Типовая временная диаграмма процесса записи байта в периферийные БИС КР580

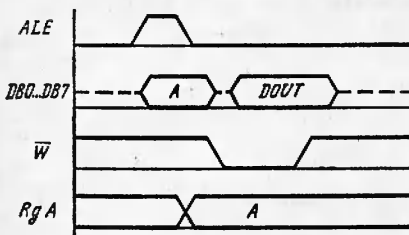


Рис. 2. Временная диаграмма процесса записи информации во внешнее ОЗУ ОЭВМ КР1816ВЕ35

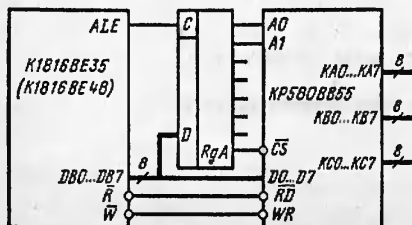


Рис. 3. Схема подключения КР580ВВ55 с адресацией портов микросхемы как ячеек внешнего ОЗУ ОЭВМ

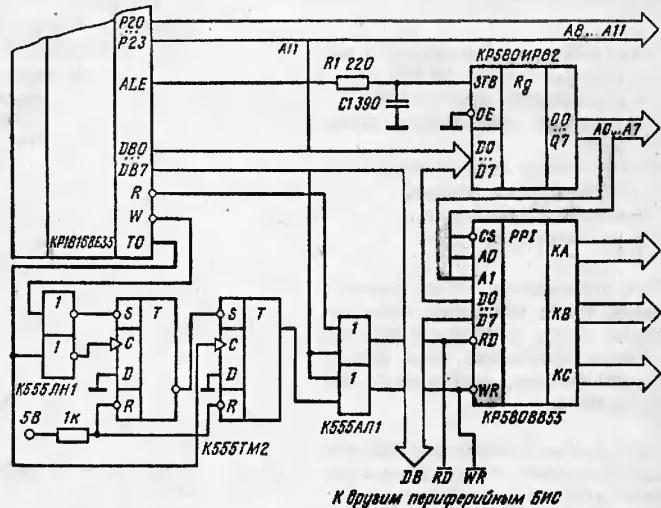


Рис. 4. Схема подключения КР580ВВ55 с укорочением импульса \bar{W} ОЭВМ

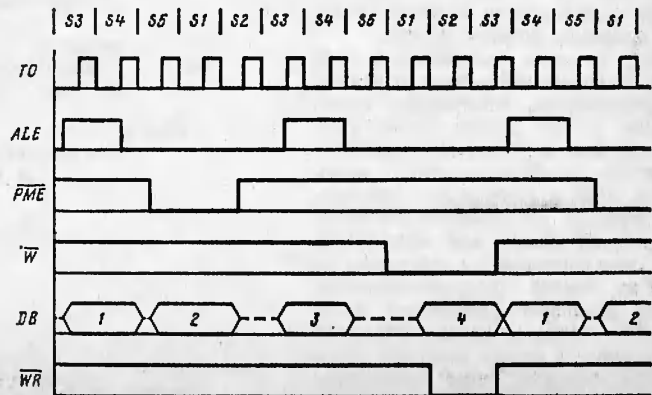


Рис. 5. Временные диаграммы работы узла сопряжения
Телефон 930-90-09, Москва

СКОРОЧТЕНИЕ И ЭВМ

Подход, связанный с применением ЭВМ к проблемам обучения скороочтению, мало известен. У нас в стране известные методы скороочтения можно пересчитать по пальцам [1—3]. Но эти методы можно отнести к научно обоснованным и фундаментальным. А метод, предложенный в данной статье, довольно простой как в подходе, так и в реализации.

Во всех методах [1—3] правильно выявлены основные недостатки традиционного способа чтения литературы и разработаны методы их устранения. Выделены такие основные недостатки при чтении: артикуляция, малое поле зрения, регрессии, отсутствие гибкой стратегии.

При обучении скороочтению ставится задача подавления этих недостатков. В условиях отсутствия каких-либо технических средств и педагогических кадров метод обучения скороочтению рассчитан на восемь недель:

- 1-я — отработка интегрального алгоритма чтения (ИАЧ),
- 2...3-я — подавление артикуляции,
- 4...5-я — развитие зрительного аппарата,
- 6-я — отработка дифференциального алгоритма чтения,
- 7-я — отработка внимания,
- 8-я — развитие памяти.

ИАЧ — это правило чтения текста с выделением таких моментов: *наименование книги, автор, источник и его данные, основное содержание, тема, фактографические данные, особенности материала, новизна.*

Под дифференциальным алгоритмом понимаются выделения *ключевых слов, смысловых рядов и цепи значений.*

Для каждого пункта разработан комплекс упражнений, и научиться быстро читать можно самостоятельно или под руководством опытных педагогов, например авторов метода.

Метод, который излагается в этой статье, позволяет избавиться от первых трех недостатков, отмеченных выше, примерно за 10 уроков (один урок по 30...40 мин в день). Для реализации метода необходимо иметь мало-мощную ЭВМ в комплекте с терминалом и хорошей операционной системой, как правило, мини- или микроЭВМ. После этого составляется программа на одном из языков программирования, которая реализует следующий алгоритм. Слова одно за другим выдаются на терминал в рамку шириной 5 см в центре экрана. Глаза читающего смотрят в одну и ту же точку. Все сло-

#include <stdio.h>

```
#define ESC 0x1B
#define clr printf("%c[2d;%dH",ESC,y,x)
#define move(x,y) printf("%c[2d;%dH",ESC,y,x)
#define cuwoff printf("%c[2d;0T",ESC)
#define curon printf("%c[2d;1T",ESC)
#define revon printf("%c[7m",ESC)
#define revoff printf("%c[27m",ESC)
#define bold printf("%c[1m",ESC)
#define red printf("%c[2m",ESC)
#define boldoff printf("%c[22m",ESC)
#define LPAR 1000
#define LW 40
#define HH 5
```

```
int speed[LPAR] = {1200,1600,1800,2400,1800,2000,2700,3400,4100,4800,0};
int raund[LPAR] = {120,120,60,600,300,300,300,300,300,300,0};
int *ps = speed;
int *pr = raund;
int mbox;
int fpar = 0;
long t1,t3;
int t2;
```

```
main(argc,argv)
char *argv[];
```

```
{
register char *p;
register i;
int c,nf;
FILE *fp;

nf = argc-1;
for (i=1; i<argc; i++) {
p = argv[i];
if( *p == '-' ) {
argv[i] = 0;
--nf;
for (++p; c = *p++; )
switch(c) {
case 'p':
case 'p':
if(++i >= argc || *p != 0)
exit(1);
getpar(argv[i]);
argv[i]=0;
--nf;
break;
default:
exit(1);
}
}
}
}
```

```
if (fpar == 0) getpar("SPRD.PAR");
cuwoff;
instr();
box();
if (nf == 0) {
if ((fp=fopen("SPRD.TEX","r")) == NULL)
exit(2);
setpar();
process(fp);
if (*ps != 0) { rewind(fp); goto m1; }
}
else {
for(i=1; i<argc; i++)
if (*(p=argv[i])) {
if((fp=fopen(p,"r")) == NULL) {
clr;
printf("не могу открыть %s",p);
exit();
}
process(fp);
fclose(fp);
}
}
curon; boldoff;
}
```



```

process(fp)
FILE *fp;
{
    char word[50][30];
    int wlen[50];
    register char *wptr;
    register int *wlenp;
    int wnum, snum;
    int c, n;
    register int j;
    long time();

    wnum=snum=n=0;
    wptr=&word[0][0];
    wlenp=wlen;
    while((c=getc(fp)) != EOF)
        if (c == ' ' || c == '\t' || c == '\n' ) {
            if (n == 0) continue;
            *wlenp++ = n;
            snum += n; n=0; wnum++;
            *wptr='\0'; wptr=&word[wnum][0];
        }
    else {
        *wptr++ = c;
        if(c==' ') ++n;
        if(c == '\n') {
            *wptr='\0'; *wlenp=n; snum += n;
            wnum++; n=0;
            t2=(t3*snum)/(1000L*wnum);
        loop: wlenp=wlen; wptr=&word[0][0]; j=wnum;
            for(; j-->0; wlenp++,wptr+=30) {
                move(40-*wlenp/2,mbox);
                puts(wptr); bdos(141,t2);
                move(21,mbox);
                puts(" ");
            }
            switch (c=bdosb(6,0xFF)) {
                case ' ': goto met;
                break;
                case 'a': exit(6);
                break;
                case 'p': par();
                break;
                case 'r': goto loop;
                break;
                case 'h': instr();
                box();
                default: break;
            }
            if(!i <= time())
                met: { ++ps; ++pr; setpar(); }
                wptr=&word[0][0]; snum=wnum=0;
                wlenp=wlen;
        }
    }
}

instr()
{
    FILE *fp;
    register i;

    clr;
    if ((fp=fopen("SPRD.INS", "r")) == NULL)
        exit(4);

    red;
    while((i=getc(fp)) != EOF) putchar(i);
    getch();
    bold; clr;
}

getpar(fn)
char *fn;
{
    register *pi;
    register *p2;
    FILE *fp;
}

```

ва выводятся только в рамке. Читатель по собственному усмотрению может увеличивать скорость воспроизведения текста на экране от 1200 до 4800 зн./мин с помощью специальных клавиш. При скорости 4800 зн./мин каждое слово появляется только на 0,1 с. Экспозиция остается постоянной независимо от количества букв в слове. Однако после высвечивания знаков препинания интервал между словами незначительно увеличивается. Фразы, например, отделяются друг от друга 0,2...0,3 с, чтобы подготовить читателя к следующему предложению.

В течение первых 5 мин машина выводит текст со скоростью 1200...1800 зн./мин, чтобы читатель привык к столь необычному методу чтения. Затем скорость возрастает до 2400 зн./мин. В следующие 5 мин — падает до 1800, через полчаса — возрастает до 4800. Обучаемые поставлены в такие условия, что они вынуждены читать с необычайно высокой для них скоростью, но приемов, с помощью которых можно было бы читать быстро и без ущерба для понимания, они еще не усвоили.

Вначале подобный «агрессивный» метод кажется крайне утомительным: слова одно за другим «выстреливаются» на экран, и читатель должен быть предельно собран — взгляд сфокусирован на рамку, и кажется невозможным, например, протереть глаза или откинуться в кресле и задуматься. Не остается ни мгновения, чтобы осмыслить только что прочитанное. Для этого необходимо выключить терминал. Однако чем чаще тренировки, тем привычнее чувствует себя такой читатель. Некоторые вскоре способны читать со скоростью до 9000 зн./мин без каких бы то ни было болезненных эффектов. При таком способе чтения восприятие крайне обострено. Специальные клавиши позволяют вновь воспроизвести только что прочитанный абзац или выдать копию печатного текста.

Однако при разработке этого метода ученые столкнулись с неожиданной проблемой: похоже, что пользующиеся подобным устройством, как и обыкновенные читатели, понимают только 70 % прочитанного, а запоминают еще меньше. У читателя иногда бывают провалы в памяти («За рубежом», 1980, № 24 (1041)).

Для реализации «агрессивного» метода были написаны программы для различных ЭВМ на языках БЕЙСИК-80, ассемблере и Си [4]. Пример одной из программ на языке Си для операционной системы МИКРОС-86: приведен на рисунке.

Для работы программы требуются файл с краткой инструкцией и файл с текстом (например, текст данной статьи, которая содержит 5779 символов, без учета пробелов и текста программы). Стандартные имена файлов SPRD.INS и SPRD.TEX. Все файлы должны находиться на одной дискете.

```

if((fp=fopen(fn,"r")) == NULL) return(0);
p1=sped; p2=raund;
while(fscanf(fp,"%d %d",p1,p2) != EOF) {
    p1++; p2++;
}
*ps=0;
fclose(fp);
fpar=1;
}

char lpar[80];
setpar()
{
    long time();

    if(*ps == 0) exits(6);
    t3 = (60L*50L*1000L)/(*ps);
    t1 = time() + *pr;
    sprintf(lpar,"скорость = %5d(сим/мин) раунд = %5d(сек)",*ps,*pr);
    par();
}

par()
{
    move(20,23);
    printf("%s%c",lpar,7,7);
    bdos(141,200);
    move(20,23);
    printf("%cEM",ESC);
}

long time()
{
    char tod[5];
    long t;

    bdos(156,tod);
    t=((tod[2] >> 4)*10 +(tod[2] & 15))*3600L;
    t+=((tod[3] >> 4)*10 +(tod[3] & 15))*60L;
    t+=((tod[4] >> 4)*10 +(tod[4] & 15));
    return t;
}

box()
{
    register i;
    int x,y;

    x=(80-WW)/2; y=(25-HH)/2;
    mbox=y+HH/2;
    move(x,y);
    revon;
    i=WW; while (i--) putchar(' ');
    move(x,y+HH-1);
    i=WW; while (i--) putchar(' ');
    for(i=y+1; i<y+HH-1; i++) {
        move(x,i); putchar(' ');
        move(x+WW-1,i); putchar(' ');
    }
    revoff;
}

exits(n)
int n;
{
    clr;
    switch (n) {
        case 1:
            puts("Вызов : spr [-p file1] [file2 ... fileN]\n");
            break;
        case 2:
            puts("Не могу открыть SPRD.TEX\n");
            break;
        case 4:
            puts("Не могу открыть SPRD.INB\n");
            break;
        default:
            break;
    }
    curon; boldoff;
    exit();
}

```

Имя файла с данными может задаваться произвольным, но тогда вызов программы происходит по другому: за именем программы надо набрать <имя. файла> с данными.

После вывода каждого предложения программа опрашивает состояние консоли. Если с консоли был введен символ, то программа его обрабатывает по следующим правилам:

- <пробел> — увеличить скорость,
- H — выдать инструкцию,
- R — повторить предложение,
- P — выдать временные характеристики,
- E — выйти из программы.

Программа может быть адаптирована профессиональным программистом практически для любой микроЭВМ примерно за 1...1,5 ч.

В заключение отметим, что «агрессивный» метод действительно позволяет повысить скорость чтения до 3600 знаков/мин и выше. Причем этот результат может быть достигнут за 2 урока. Но у этого метода есть один большой недостаток: научившись читать быстро, вам невероятно сложно будет вернуться обратно к медленному чтению и «смаковать», например, шедевры художественной литературы. На создание программ авторы затратили 96 ч, или около 8 ч на одну программу.

Сейчас авторы работают над созданием системы на основе персональной ЭВМ с цветным графическим монитором, которая будет реализовывать метод [1]. Затраты оцениваются примерно в 5 человеко-лет.

454067, Челябинск, ул. Ленина, 11, кв. 36, Зырянов А. Г.

ЛИТЕРАТУРА

1. Кузнецов О. А., Хромов Л. Н. Техника быстрого чтения.— М.: Книга.— 1983.— 175 с.
2. Лезер Ф. Рациональное чтение.— М. 1980.— 160 с.
3. Бородин В. А., Бородин С. М. Учим... читать: Уроки динамического чтения.— Л.: Лениздат, 1985.— 192 с.
4. Хротко Г. Языки программирования высокого уровня для микроЭВМ.— М.: МЦНТИ, 1985.

Статья поступила 20.11.87

РЕКЛАМА

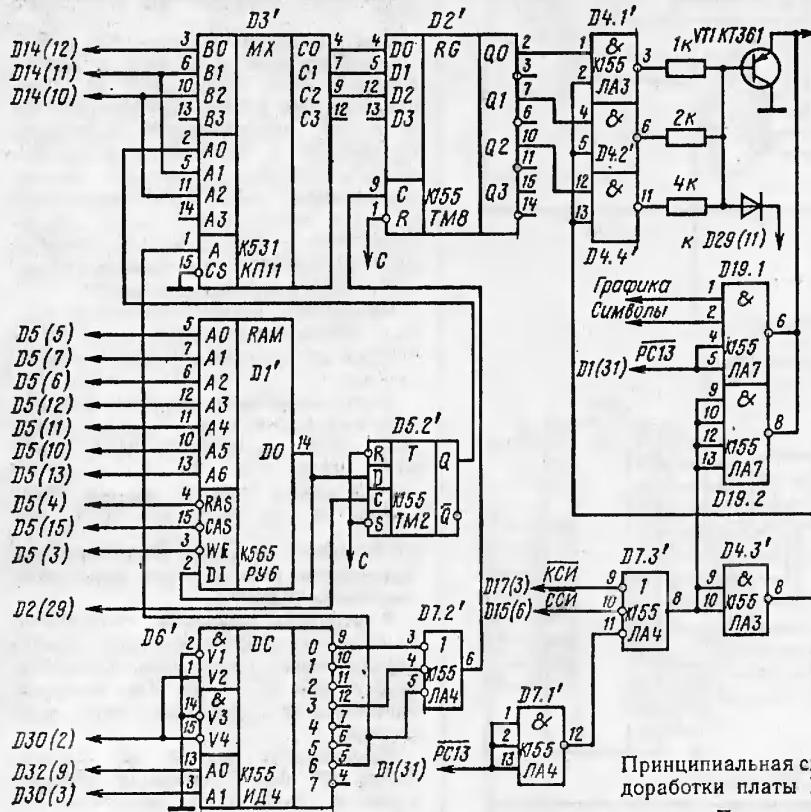
НТЦ «Дедал» предлагает услуги по оперативной разработке и изготовлению двух (многослойных) печатных плат. Длительность выполнения заказа (разработка и изготовление) — две недели. Воспользуемся услугами специалистов по разработке БМК.

С предложениями обращаться по адресу: 123362, Москва, ул. Свободы, 8/4. Телефон: 427-52-32, 491-51-72

РАШИРЕНИЕ ВОЗМОЖНОСТЕЙ ДВКЗ ДЛЯ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Доработка контроллера графического дисплея (КГД), входящего в состав

ДВКЗ, позволяет формировать изображение размером 286 строк по вертика-



Принципиальная схема доработки платы КГД

ли и 133 точки по горизонтали (см. рисунок). Число градаций яркости каждой точки равно восьми, что соответствует трем битам в ОЗУ КГД.

Переключение дисплея в полутоновый режим производится программно, для этого в 13-й разряд регистра управления (РУ) КГД засылается единица. Доработка не оказывает влияния на стандартные режимы дисплея, кроме того, возможна любая комбинация режимов из алфавитно-цифрового, графического и полутонового.

Для расширения длины слова ОЗУ КГД до девяти разрядов введен элемент D1', что позволяет хранить в одном слове три точки по три бита. Параллельный код яркости снимается со сдвигового регистра D14' и подается на порт В мультиплексора D3'. На порт А два младших разряда поступают с D14, а старший разряд — с триггера D5.2', который служит для хранения девятого разряда в течение цикла преобразования. Мультиплексор D3' управляется через дешифратор D6'. На вход D6' подаются три младших разряда с адресного счетчика. С выхода D3' код яркости подается на регистр D2', который строится сигналом, формируемым элементом D7.2'.

Код яркости преобразуется в видеосигнал на элементах D4.1', D4.2', D4.4' и транзисторе VT1. На элементах D7.1', D7.2', D4.3' собрана схема гашения видеосигнала на обратном ходе развертки. Для установки дополнительных корпусов на плате КГД предусмотрены свободные места.

Телефон 497-86-35, Москва

Статья поступила 22.02.88

Подпрограмма ввода данных

Операция	Адрес	Мнемокод	Комментарий
Пуск	0000	EI	Разрешение прерывания
	01	LX/SP, 07FFH	Начальное разрешение указателя стекла
Запись в ОЗУ	04	LX/D, 0400H	Инициализация D-пары
	07	HLT	Останов
	08	/N1	Ввод данных из УВВ в А
	0A	mov B, A	(A) → (B)
	0B	AN/, FOH	(A) <B2> → (A)
	0D	RRC	A _{m+1} → A _m , A0 → A7, A0 → (Π)
	0E	→	→
	0F	→	→
	10	→	→
	11	mov C, A	(A) → (C)
	12	ADDC	(A) + (C) → (A)
	13	→	→
	14	→	→
	15	→	→
	16	→	→
	17	mov C, A	(A) → (C)
	18	mov A, B	(B) → (A)
	19	SUBC	(A) - (C) → (A)
1A	STAXD	(A) → [(D) (E)]	
1C	INXD	(D) (E) + 1 → (D) (E)	
1D	EI	Разрешение прерывания	
1E	RET	Возврат из подпрограммы	

В. И. Ермакин

УСТРОЙСТВО ВВОДА ИНФОРМАЦИИ В МИКРОПРОЦЕССОРНУЮ СИСТЕМУ

Устройство ввода информации в микропроцессорную систему на базе МП КР580ИК80А включает в себя клавиатуру ввода данных, и клавиши управления «Запись в ОЗУ» и «Установка в 0», для которых разработана схема подавления дребезга контактов (рис. 1). Для нормального функционирования устройства необходимо нажать клавишу «Установка в 0» после включения питания.

Запись данных осуществляется последовательно в ОЗУ, начиная с восьмой разрядной ячейки 0400—07FF. Необходимо нажать две клавиши поочередно на клавиатуре ввода данных происходит запись двоично-десятичных данных в DD11). После нажатия

Государственная организация НТЦ «Мосворечье» гарантирует высокий уровень научно-технической продукции, своевременное и качественное выполнение заказов.

Предлагаем Вашему вниманию Универсальный программатор микросхем ПЗУ.

Программатор представляет собой малогабаритную приставку к персональной ЭВМ (типа ДВК и ЭВМ-совместимые с IBM/PC, Роботрон1715 и т. д.), позволяющую записывать созданную с помощью системных средств ПЭВМ (редакторы текста, трансляторы и т. д.) информацию в любой из известных типов микросхем ПЗУ с байтовой организацией (как однократно программируемые, так и перепрограммируемые).

Надежность записи достигается полным соблюдением требований ТУ и ГОСТов на программируемые микросхемы.

В настоящее время имеется библиотека подпрограмм для программирования ПЗУ серий К573, К556, К558, К1601, К155.

Наш адрес: 109017, Москва, Пятницкая, 36, тел. 231-29-72

РС-Комплект поможет Вам увереннее чувствовать себя в мире персональных компьютеров

В условиях дефицита технической информации эффективно использовать персональный компьютер, совместимый с IBM PC-XT/AT, Вам поможет РС-Комплект информационных материалов.

РС-Комплект состоит из четырех томов (21 книга) описания архитектуры и системного программного обеспечения.

РС-Комплект содержит информацию, необходимую для профессионального использования машин этого класса как в режиме эксплуатации готовых программных систем, так и в режиме разработки программного обеспечения широкого спектра прикладных задач — от локальных систем автоматизации инженерных и экономических расчетов до систем реального времени с непосредственным управлением оборудованием.

РС-Комплект разработан на базе литературы, существующей по данному вопросу, специально выполненных переводов фирменной документации по аппаратным и программным средствам, а также на основании собственного опыта специалистов-практиков со стажем работы с IBM PC-XT/AT не менее 10000 часов в режиме разработки программного обеспечения.

Подробный аннотированный перечень предлагаемых материалов, содержащий отдельные аннотации на каждую книгу, кооператив высылает наложенным платежом.

Заказы на аннотации и РС-Комплект присылайте по адресу: 220006, Минск, ул. Полевая, 26—19. Инженерно-производственный кооператив «Эф-фэкт». Тел. 21-09-55.

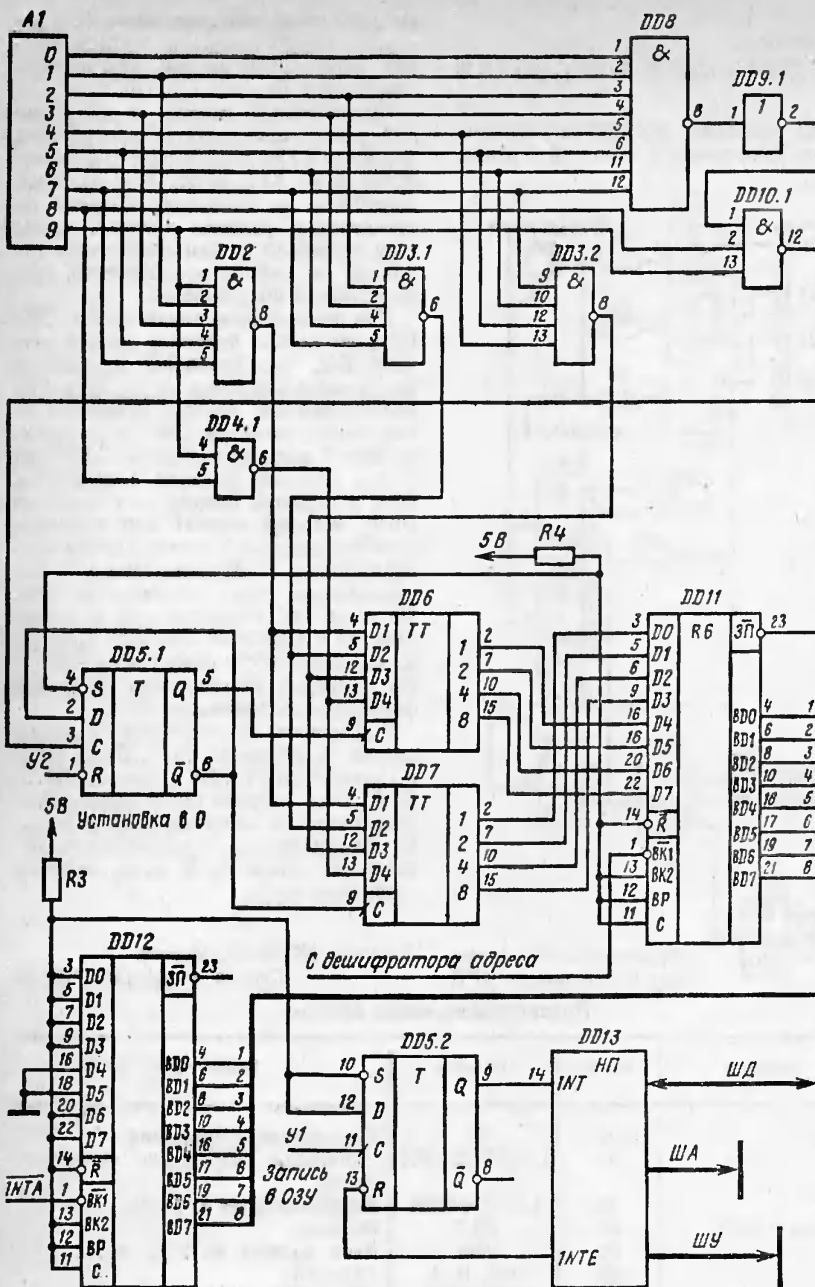
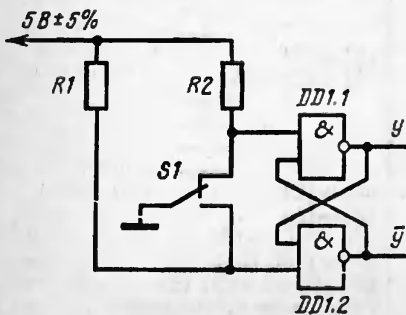


Рис. 1. Схема подавления дребезга контактов для клавиш управления



2. Устройство ввода информации в микропроцессорную систему

клавиши «Запись в ОЗУ» начинает выполняться подпрограмма прерывания (см. таблицу).

Предпочтительнее работать с данными в двоичном коде, поэтому подпрограмма прерывания предусматривает перевод кода в двоичный (2...10).

При применении устройства необходимо предусмотреть индикацию вводимых данных, адрес и коррекцию их ввода (рис. 2).

214004, Смоленск, 2-й Краснинский пер., д. 8, кв. 9, тел. 9-21-10 р.

УДК 681.326

Сивобород П. В., Юрченко И. В., Петренко Н. В., Ножницкий А. А., Ерошенко Н. И. **Универсальная 8-разрядная микроЭВМ КР1830ВЕ48 // Микропроцессорные средства и системы.**— 1990.— № 1.— С. 6.

Рассматриваемая микроЭВМ по архитектуре, структуре команд и языку ассемблера идентична микросхеме КМ1816ВЕ48. Имеет 97 базовых команд, в том числе программную и аппаратную установку режима микропотребления, когда работа задающего генератора блокируется, а содержимое регистра аккумулятора, регистра состояния и ОЗУ сохраняется.

Отличительная особенность микроЭВМ КР1830ВЕ48 — высокая помехозащищенность (гарантируется КМОП-технологией изготовления) и высокая экономичность: потребляемая мощность в рабочем режиме — не более 45 мВт, в режиме микропотребления — не более 0,45 мВт.

УДК 681.3.

Абрамов В. Г., Брябрин В. М., Горячая И. В., Щеглова Т. Н. **ЭРУДИТ — инструментальная система для поддержки обучения на базе ПЭВМ // Микропроцессорные средства и системы.**— 1990.— № 1.— С. 23.

Рассматривается конкретный подход к построению на базе ПЭВМ инструментальной системы для поддержки процесса обучения. В соответствии с развиваемым подходом средства подготовки учебного материала и его последующей интерпретации интегрированы в систему, обеспечивающую единый интерфейс с различными категориями пользователей (преподавателями и обучаемыми). Данный подход не предполагает специальной подготовки пользователя в области программирования и предусматривает простой способ использования специфических (графических, цветовых и др.) возможностей ПЭВМ. Приводятся особенности структурирования учебной информации и функциональные возможности системы ЭРУДИТ, реализованной на основе предложенного подхода.

УДК 681.324.06

Корнеев В. В., Монахов О. Г. **Параллельный Паскаль для вычислительных систем с программируемой структурой // Микропроцессорные средства и системы.**— 1990.— № 1.— С. 28.

Рассмотрено расширение языка Паскаль множеством операций над семафорами и процессами, которые позволяют организовать параллельную и распределенную обработку информации в микромашинных вычислительных системах с программируемой структурой.

УДК 681.3.06

Брусилловский Л. И., Скляр В. В. **Система файлового обмена KERMIT для организации межмашинной связи по терминальным линиям // Микропроцессорные средства и системы.**— 1990.— № 1.— С. 30.

Подробно описано сетевое программное обеспечение KERMIT, рассмотрены области его применения, типы используемых связей, режимы работы. Даны рекомендации по использованию системы.

УДК 681.3.06

Нафиков В. Р., Рабинович М. И. **Ядро Модула 2 ориентированной операционной системы для встраиваемых микропроцессорных систем // Микропроцессорные средства и системы.**— 1990.— № 1.— С. 37.

Изложена концепция и особенности реализации ядра ОС реального времени с помощью компилятора Модула 2, предназначенного для использования в составе программного обеспечения целевых систем, разработанных на основе языка Модула 2.

UDC 681.326

Sivoborod P. V., Yurchenko I. V., Petrenko N. V., Nozhnitsky A. A., Eroshenko N. I. **Universal 8-bit single-chip microcomputer type КР1830ВЕ48. // Microprocessor devices and systems.**— 1990.— N. 1.— P. 6.

The single-chip microcomputer type КР1830ВЕ48 is a CMOS version of КМ1816ВЕ48 (8048) LSI. It has 97 basic instructions, including hardware and software switching to low-consumption mode, when system clock is halted while RAM contents, as well as the accumulator and status word are retained. Excellent noise-immunity and low power consumption (45 mW when running, 0.45 mW in standby mode) guaranteed by CMOS technology are most important features of the new LSI.

UDC 681.3.

Abramov V. G., Bryabrin V. M., Goryachaya I. V., Sheglova T. N. **ERUDIT — the tutorial development system running on the PC. // Microprocessor devices and systems.**— 1990.— N. 1.— P. 23.

The practical approach to development system for tuition support using PC is described, when the tools for lesson preparation and its further interpretation means are integrated into single system. The system has uniform interface for different user categories (teachers and students). This method does not imply the necessity of any special skills in programming for the users and provides simple means for using specific features (graphics, color etc.) of the personal computer. The structure of lesson information and the operation modes of ERUDIT system, built using the described approach are illustrated.

UDC 681.324.06

Korneev V. V., Monachov O. G. **Concurrent Pascal for computers with parallel Structure // Microprocessor Devices and Systems.**— 1990.— N. 1.— P. 28.

The Pascal language extension, which gives semaphore and multiple processes support is described. The extensions provide means for parallel and distributed data processing in the microcomputer systems of the programmable architecture.

UDC 681.3.06

Brusilovskiy L. I., Sklyarov V. V. **KERMIT — the protocol for inter-computer file transfer using terminal data lines. // Microprocessor devices and systems.**— 1990.— N. 1.— P. 30.

The network software for data communication between various computer systems KERMIT is described. The suitable data lines and operation modes are specified. Some recommendations on the KERMIT system application are given.

UDC 681.3.06.

Nafikov V. R., Rabinovich M. I. **The MODULA-2-oriented operating system kernel for OEM microprocessor systems. // Microprocessor devices and systems.**— 1990.— N. 1.— P. 37.

The concept and practical implementation features of the real-time operating system kernel built using MODULA-2 compiler are described. The kernel is designed to support target system software, written in MODULA-2.

Мышкин И. Л., Заречнюк Г. О. Опыт использования языка Фортран-80 для разработки ПО встроенных микроЭВМ измерительных приборов // Микропроцессорные средства и системы.— 1990.— № 1.— С. 46.

Приводятся сравнение компиляторов для микропроцессора КР5801К80А с точки зрения размера исполняемой программы, показана эффективность применения транслятора «ФОРТРАН-80», дан пример разработки системы.

УДК 681.324

Назаров С. В., Луговец А. В. Исследование производительности микроЭВМ с магистральной архитектурой // Микропроцессорные средства и системы.— 1990.— № 1.— С. 49.

Рассматривается задача оценки производительности микроЭВМ в зависимости от интенсивности информационного обмена с периферийными устройствами. Предложены методика и средства проведения эксперимента по оценке производительности микро- и мини-ЭВМ, совместимых с микроЭВМ «Электроника 60». Приведены результаты экспериментов по оценке производительности микроЭВМ «Электроника 60М», «Электроника НЦ 80-01Д» и мини-ЭВМ СМ4.

УДК 681.325.5:181.4

Врублевский Ю. Н., Осинков Ю. П. Использование ТВ-приемников — в качестве мониторов ПЭВМ // Микропроцессорные средства и системы.— 1990.— № 1.— С. 54.

Рассмотрены возможные способы сопряжения ПЭВМ с ТВ-приемниками на примере бытового компьютера «Электроника БК-0010» и переносных телеприемников серии «Электроника».

УДК 681.327

Асанов Р. Ш., Никитин А. С. Имитатор интерфейса МПИ // Микропроцессорные средства и системы.— 1990.— № 1.— С. 74.

Рассматривается электрическая принципиальная схема имитатора, формирующего интерфейсные сигналы МПИ (ОСТ11.305.903—80) для нахождения неисправностей в контроллерах, работающих в активном и пассивном режимах; может использоваться как устройство прослушивания сигналов МПИ. Кратко излагается работа имитатора при обнаружении неисправностей в налаживаемом контроллере.

УДК 681.325

Накалов Е. Ф., Тюлькин С. П. Универсальный интерфейс ввода—вывода с элементами самопроверки // Микропроцессорные средства и системы.— 1990.— № 1.— С. 76.

Приводится способ тестирования портов ввода — вывода с возможностью работы порта на ввод и на вывод. Приведена принципиальная схема универсального интерфейса и рассмотрены режимы его работы.

УДК 681.325

Бескровный В. Г., Урсатьев А. А., Сапожникова С. Л., Шестаковский М. А. КНМЛ СМ 5211 — внешнее ЗУ микроЭВМ «Электроника 60» // Микропроцессорные средства и системы.— 1990.— № 1.— С. 78.

Описаны средства сопряжения накопителя на кассетной магнитной ленте СМ 5211.19 с микроЭВМ «Электроника 60». Интерфейс выполнен на микросхемах серии 155. Программные средства поддержки накопителя в ОС РАФОС — системный драйвер Ст.SYS.

Myshkin I. L., Zarechniuk G. O. Using FORTRAN-80 for built-in microcontrollers programming in measuring instruments. // Microprocessor devices and systems.— 1990.— N 1.— P. 46.

The executable code size comparison generated by various compilers for 8080 microprocessor is given. The FORTRAN-80 compiler was considered to be most effective. The sample target program design is illustrated.

UDC 681.324

Nazarov S. V., Lugovets A. V. The benchmark tests of bus-based microcomputers. // Microprocessor devices and systems.— 1990.— N 1.— P. 49.

The performance as a function of I/O data traffic for bus-based microcomputers is discussed. A method and tools for LSI-11-compatible microcomputer benchmarking are proposed. The practical benchmark test results for "Electronika 60M", "Electronika НЦ80 01Д" and СМ4 computers are given.

UDC 681.325.5:181.4

Vrublevsky Yu. N., Osinkov Yu. P. Home TV as a computer monitor. // Microprocessor devices and systems.— 1990.— N 1.— P. 54.

Some methods of personal computer interfacing to home TV are explained on the example of "Electronika" portable TV connection to «БК-0010» computer.

UDC 681.327

Asanov R. Sh., Nikitin A. S. Q-bus interface simulator. // Microprocessor devices and systems.— 1990.— N 1.— P. 74.

The circuit diagram of Q-bus interface simulator which generates control bus signals is shown. The simulator is designed for Q-bus controllers troubleshooting which may work in active and passive modes, and may be also used for Q-bus monitoring. The troubleshooting algorithm for a faulty controller is briefly described.

UDC 681.325

Nakalov E. F., Tulkin S. P. Universal I/O interface with self-diagnostics. // Microprocessor devices and systems.— 1990.— N 1.— P. 76.

The self-diagnostics method for I/O ports with bidirectional I/O capability is described. The circuit diagram of the universal port is shown and its modes of operation are explained.

UDC 681.325

Beskrovny V. G., Ursatiev A. A., Sapozhnikova S. L., Shestakovsky M. A. CM5211—peripheral of the "Electronika 60" microcomputer. // Microprocessor devices and systems.— 1990.— N 1.— P. 78.

The Q-bus adapter for CM5211 cassette tape drive was designed, using K155 ICs. The unit is supported by CT.SYS driver in RAFOS operating system.

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года

МПИ МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 2/1990 МОСКВА

**Общие вопросы
развития
информатики**

**МИКРОПРОЦЕССОРНАЯ
ТЕХНИКА**

**ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ**

Машинная графика

**Средства
отладки**

**ПРИМЕНЕНИЕ
МИКРОПРОЦЕССОРНЫХ
СРЕДСТВ
Средства
ввода-вывода**

**УЧЕБНЫЙ
ЦЕНТР**

Федукин А. К., Семиколенных А. Н.— Система комплексного централизованного обслуживания средств вычислительной техники	2
Захаров В. В.— Авторские права разработчиков программных средств	6
Душин-Бариковский В. Л., Терехин А. Т.— Нейронные сети и нейропроцессоры: тенденции развития исследований и разработок	12
Белоус А. И., Ковалевский В. К., Кособрюхов В. А., Пархомчук А. А., Чернуха Б. Н.— Универсальный временной преобразователь на основе БИС K512PC11	15
Петух А. М., Ободник Д. Т., Верховой В. П.— Генератор векторов на основе матричной БИС	17
Злотник Е. М., Андриченко В. В., Вокуть В. В.— Микропрограммируемый графический контроллер	19
Гулин Э. А., Гуслев В. Г., Налибоцкий Б. В., Петрусевич Ю. И.— Специальное программное обеспечение неоднородных локальных вычислительных сетей	26
Перепелов В. П., Русанов Д. С.— Динамическая загрузка программных сегментов в ОС РАФОС	28
Рябов А. Н., Ануфриев С. В.— Подпрограммы с переменным числом параметров на Фортране системы РАФОС	30
Ашкенази И. Д.— Программа распечатки таблиц с рубрикацией	33
Жуков А. А., Паскаль И. Ю.— Программа документирования графической информации	37
Курлянд В. М., Лагута В. В.— Графическая система для микроЭВМ ДВКЗ	40
Нифонтов Ю. В., Калита В. В., Бобков Г. М., Яфаев А. А.— Модули отображения УВК семейства СМ1800	42
Блох Е. М., Бодашков К. Б., Шляхтин В. В.— Графический контроллер для цветной графической системы	45
Горбунов А. Г., Мининко С. В., Мруга А. Д.— Отладочный модуль на основе генератора тестовых последовательностей	49
Малёжин О. Б., Верстаков В. А.— Программатор ПЛМ в стандарте микроЭВМ «Электроника 60»	53
Коваленко В. А., Мамедов М. Х., Черный Е. А., Шевкунов Е. Б.— Операционная система реального времени для микроЭВМ «Электроника НМС 12401»	56
Александров А. В., Кушнир М. И., Харламов А. В.— Символьный отладчик для языка MACRO-11	58
Харламов А. В., Кушнир М. И.— Средства управления тестированием и отладкой программ	60
Станоткин А. А.— Автоматизация построения тестов для программ по внешней спецификации	63
Бучнев А. Н., Васильев Н. П., Горовой В. Р., Крылатых Ю. П., Кушнир М. И.— Комплекс отладки программ микропроцессорной системы	67
Разенков С. Л., Тимофеев Е. Ю.— Метод увеличения протяженности магистрали интерфейса КОП	69
Арсенин В. П., Охрименко В. Р.— Интерфейсные модули для микроЭВМ «Электроника 60»	71
Иванов А. И., Кладов В. Е.— Организация интерфейса общей магистрали в мультипроцессорных системах	73
Донев В. С., Василевский А. К., Плужников А. В., Серых В. Я.— Микроконтроллер для океанологической измерительной аппаратуры с пониженным энергопотреблением	78
Медведев С. П., Абрамов В. Б., Печерская Р. М., Коршунов В. В.— Многоканальный интерфейс параллельного обмена	83
Джиган В. И., Щербаков С. В.— Аналого-цифровой преобразователь с унифицированным интерфейсом МПИ	85
Ионан Ю. Э.— Управление работой программных модулей	88
Ситников С. И.— Устройство контроля работы автоматизированного экспериментального комплекса	90

681.3

А. К. Федукин, А. Н. Семиколенных

СИСТЕМА КОМПЛЕКСНОГО ЦЕНТРАЛИЗОВАННОГО ОБСЛУЖИВАНИЯ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Одной из составляющих научно-технического прогресса общества является компьютеризация производственных процессов, переходящая в информатизацию различных сторон общественной жизни. Оба этих процесса должны поддерживаться системой предоставления услуг, обеспечивающей Пользователя возможностью выполнения работ, гарантирующих эффективное применение средств вычислительной техники (СВТ).

Система должна оказывать техническую помощь по всем проблемам, возникающим у Пользователя, а также способствовать эффективному применению приобретенных программно-технических средств.

Цель данной публикации — привлечь внимание Пользователя, разработчиков СВТ и специалистов по обслуживанию к проблемам, которые должны решаться совместно.

Широкое внедрение СВТ сопряжено с проблемой поддержания эксплуатационных свойств технических и программных средств. Эта проблема включает много раз обсуждавшихся, но пока не решенных вопросов:

- низкие эксплуатационные свойства СВТ;
- условия эксплуатации, не отвечающие установленным нормам;
- распределение работ между Пользователями и системой обслуживания;
- отсутствие сервисных услуг, обеспечивающих использование СВТ в конкретных технологических процессах обработки информации;
- недостаточная техническая оснащенность специалистов и т. д.

Существовавший ранее отраслевой принцип обслуживания СВТ вызывал у Пользователя вполне обоснованные претензии к организации обслуживания и качеству предоставляемых услуг. В связи с объединением на одном вычислительном центре (ВЦ) нескольких типов ЭВМ в одну вычислительную систему и для исключения ведомственного диктата производителя СВТ была разработана (в основном, на базе ведомственных предприятий) Государственная система комплексного централизованного обслуживания (КЦО) СВТ, которую можно характеризовать как организационно-управленческую систему, ориентированную на обслуживание всех типов СВТ (суперЭВМ, большие, средние, малые и персональные ЭВМ).

Система КЦО СВТ имеет региональные производственные или научно-производственные объединения (РПО). Реквизиты РПО приведены в приложении. Цель деятельности РПО — комплексный подход к решению проблемы удовлетворения потребности общества в услугах в своем регионе по информатизации региона на основе системного анализа данных объективной реальности и прогнозируемого развития информатизации региона.

Тенденции совершенствования обслуживания СВТ целесообразно определять на основе оценки современного состояния обслуживания, сложившихся отношений «система КЦО СВТ — Пользователь», состояния экономических организационных и технических проблем.

Оценка существующего состояния системы КЦО СВТ

К традиционным показателям деятельности системы КЦО относится объем производства в денежном и номенклатур-

ном выражениях. Эти показатели практически не связаны с конечными результатами производственной деятельности ВЦ. Более того, они находятся в противоречии с последними: чем больший объем работ по техобслуживанию и ремонту выполняют сервисные предприятия, тем лучше их показатели. Менее надежной технике соответствуют лучшие экономические условия работы сервисных предприятий и худшие условия работы обслуживаемых ВЦ.

Объект обслуживания (ЭВМ, отдельные устройства, вычислительные системы) для системы КЦО СВТ можно охарактеризовать уровнем надежности и совершенством эксплуатационных свойств. На настоящий период времени надежность СВТ и эксплуатационные свойства объекта обслуживания имеют следующие показатели: низкая по сравнению с СВТ, производимыми западными фирмами, средняя наработка на отказ (100...500 ч) и коэффициент технического использования менее 0,9. Требования к эксплуатационной технологичности (контролепригодности, диагностируемости, ремонтпригодности, дистанционному обслуживанию) ни Пользователем, ни системой КЦО СВТ не предъявлялись. В обеспечении ремонтно-восстановительных предприятий рабочими местами и ремонтной документацией заводы-изготовители и разработчики участия не принимали. Объем ЗИП, поставляемого с ЭВМ, составляет 5...11 % от стоимости ЭВМ.

Системой КЦО СВТ применяется регламентированный вид технического обслуживания, в котором заложена возможность только экстенсивного развития. Анализ работы системы КЦО СВТ выявил следующие отрицательные моменты в обслуживании:

- неоответствие технического уровня системы КЦО СВТ современному уровню развития СВТ;
 - информационная технология между звеньями управления базируется на избыточном документообороте;
 - автоматизация управления производственной деятельностью отсутствует;
 - недостаточная техническая оснащенность;
 - отсутствие механизма распределения ответственности между эксплуатационным персоналом Пользователя и специалистами системы КЦО СВТ;
 - при наличии ЗИПов у Пользователя и предприятий системы КЦО СВТ — сочетание хронического дефицита по ряду наименований с большими объемами неликвидов по другим наименованиям;
 - время реакции на запрос не удовлетворяет Пользователя;
 - предприятия системы КЦО СВТ представляют ограниченную номенклатуру услуг.
- Экономический механизм, действующий в системе КЦО СВТ, не определяет материальную заинтересованность исполнителя в конечном результате выполняемой работы. В подготовке производства системы КЦО СВТ разработчики и заводы-изготовители не принимают участия.

Тенденции развития системы КЦО СВТ

Планируемое увеличение объемов выпуска персональной вычислительной техники (ВТ) в стране, широкая автоматизация научной, управленческой и производственной деятельности, информатизация общества вовлекают растущие материальные и трудовые ресурсы в сферу эксплуатации, обслуживания и ремонта техники. В этой связи особенно возрастает роль КЦО, позволяющего за счет централизации материальных и трудовых ресурсов, применения индустриальных средств и методов проведения работ значительно сокращать приток новых сил и средств для обеспечения эксплуатации и обслуживания техники.

ного объединения (ПО), которая информационно связана с АИС инфраструктуры ПО. Наибольшая нагрузка по сбору и обработке информации приходится на РПО.

В структуре РПО должны быть АИС, АРМ бригадира, средства связи с ЭВМ производственного объединения. АИС должна выполнять функции распределения ресурсов, осуществлять подготовку и управление выполнением работ.

Схематически представленная на рис. 1 ИВС Пользователя является системой, обрабатывающей информацию по технологическому циклу, которая может включать локальные вычислительные сети (ЛВС), построенные на разных типах ЭВМ и имеющие различную топологию, а также средства, обеспечивающие работу ИВС в глобальной сети. Для поддержания эксплуатационных свойств ЛВС целесообразно в их составе иметь ЭВМ, выполняющую диагностическую функцию, и экспертную систему поиска отказа, а для дистанционного определения технического состояния ИВС — программное средство, обеспечивающее формирование необходимой информации.

На каждом уровне иерархии есть своя инфраструктура: склады, инженерно-технические сооружения, ремонтные предприятия, учебные центры, научные, социально-бытовые подразделения, транспорт, связь.

Организационно-технические решения

Кроме структурных изменений необходимо совершенствование организационно-технических мероприятий. В системе КЦО СВТ разработан и экспериментально опробован метод предоставления услуги с оценкой работы по конечному результату. Пока это осуществляется при выполнении только одной услуги — централизованного технического обслуживания. Предоставление услуги с оценкой работ по конечному результату — это способ перехода от затратного метода ценообразования к применению общестранно необходимых цен с учетом обеспечения на договорной основе предприятию-заказчику гарантированных показателей работоспособности эксплуатируемых ЭВМ, а сервисному предприятию-исполнителю — оплаты работ в зависимости от фактического уровня работоспособности

обслуживаемой техники. При заключении договора учитываются индивидуальные требования заказчика и условия, в которых эксплуатируется вычислительная система. Этот метод постепенно будет распространен на все виды услуг, предоставляемые системой КЦО СВТ.

В ближайшее время предполагается предложить следующие виды услуг: маркетинг, коммерческие услуги, услуги по созданию ИВС «под ключ», использованию, поддержке и модернизации ИВС, интеллектуальные услуги по созданию и введению баз данных, подготовка и переподготовка кадров.

Услуги сервиса как постоянный вид промышленных услуг находятся в стадии становления и развития. Именно в развитии этих услуг и заложен потенциал системы КЦО СВТ. Спрос на услуги определяется в процессе маркетинга.

Маркетинг — это действия, обеспечивающие комплексный подход к управлению производством и реализацией услуг, ориентируемые на учет спроса и требований Пользователя (рис. 2). Основными формами предоставления услуг могут быть комплексные и отдельные услуги, телесервис. Заключенный договор на комплексные услуги и телесервис обеспечивает предприятию стабильный объем работ. Отдельные услуги выполняются по разовым заявкам. Характерная особенность предлагаемых видов услуг — их наукоемкость. Известно, что наукоемкая продукция сопровождается использованием ресурсосберегающих технологий.

Стимул для применения ресурсосберегающих технологий — внедрение принципа оценки работ по конечному результату. Так, например, при централизованном техническом обслуживании ИВС ресурсосберегающей технологией является обслуживание по техническому состоянию. При этом во время цикла контроля технического состояния средств ИВС выполняется определенный объем обязательных работ и дополнительные работы, необходимость которых зависит от состояния технических средств, для чего целесообразно использовать программное средство, которое разработано и опробовано в эксплуатации.

Такая технология гарантирует поддержание эксплуатационных свойств на требуемом уровне. Ресурсосберегающая технология позволяет экономить трудовые ресурсы системы КЦО СВТ, Пользователя и вычислительные ресурсы ИВС.

Если применить системный подход и такой критерий оценки предоставляемой услуги как сокращение потерь машинного времени на ЭВМ Пользователя, то по всем видам услуг использование ресурсосберегающих технологий экономически целесообразно.

При переходе на предоставление новых видов услуг существенной проблемой является обеспечение системного подхода к подготовке производства. Применительно к системе КЦО СВТ он должен быть основан на взаимодействии трех основных участников (системы КЦО СВТ, Пользователя и изготовителя СВТ) выполнения комплекса мероприятий до начала процесса промышленного предоставления услуг. Предприятия, участвующие в изготовлении и поддержании эксплуатационных свойств изделий, должны иметь определенные обязательства друг перед другом. Для успешной подготовки производства необходимо определить оптимальные организационные связи и закрепить их экономическими решениями. Это будет обеспечивать техническую и технологическую подготовку производства, в том числе опережающую подготовку специалистов системы КЦО СВТ.

Система КЦО СВТ уже в настоящее время в состоянии поддерживать технологический процесс обработки информации и участвовать по заявке Пользователя в реализации возможных методов доступа к хранимой информации.

Система КЦО СВТ при реализации технологии обработки информации может участвовать также в создании рабочих мест на базе ПЭВМ, интегрированных межотраслевых, региональных и национальных баз данных, АРМ на разных типах ЭВМ.

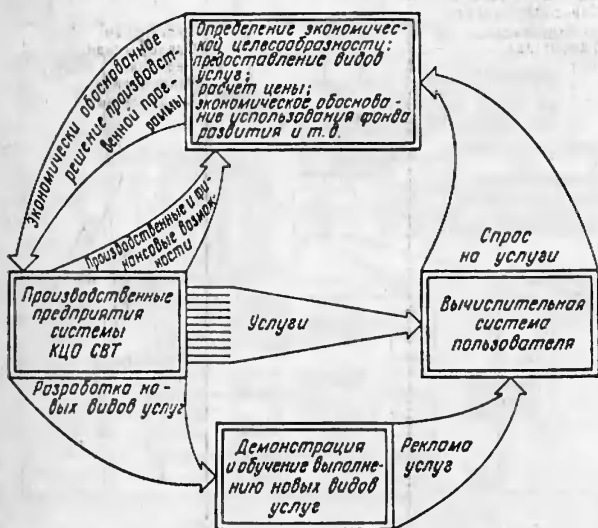


Рис. 2. Маркетинг в системе КЦО СВТ в новых экономических условиях

№ пп	ПО (НПО)		Зона деятельности			
	Наименование	Адрес: телефон	Союзная республика	АССР, автономный округ	Край	Область
	1	2	3	4	5	6
1	Владивостокское ПП ВТИ	690068, Владивосток, ул. Кирова, 23; тел. 6-41-68, 6-42-96			Приморский	
2	Воронежское ПО ВТИ	394012, Воронеж, ул. Тимирязева, 27; тел. 52-46-15				Белгородская, Воронежская, Липецкая, Тамбовская
3	Горьковское ПО ВТИ	603109, Горький, Краснофлотская ул., 56; тел. 34-03-98		Марийская, Мордовская, Чувашская		Горьковская, Кировская
4	Иркутское ПО ВТИ	664005, Иркутск, ул. Набережная Иркутта, 58; тел. 41-65-94		Бурятская, Тувинская, Якутская, Чукотский круг	Красноярский	Амурская, Иркутская, Камчатская, Магаданская, Сахалинская, Читинская
5	Казанское НПО ВТИ	420044, Казань, ул. Ямочева, 36; тел. 56-66-11		Татарская		
6	Куйбышевское ПО ВТИ	443090, Куйбышев, ул. Антонова-Овсеенко, 44; тел. 52-28-54, 52-20-05		Калмыцкая		Астраханская, Волгоградская, Куйбышевская, Пензенская
7	Ленинградское ПО ВТИ	195272, Ленинград, пр. Шаумяна, 18; тел. 528-97-01		Карельская, Коми		Саратовская, Ульяновская, Архангельская, Вологодская, Калининградская, Ленинградская, Мурманская, Новгородская, Псковская
8	Московское ПО ВТИ	127521, Москва, Октябрьская ул., 94; тел. 289-06-25				Брянская, Владимирская, Ивановская, Калужская, Калининская, Костромская, Московская, Орловская, Рязанская, Смоленская, Тульская, Ярославская
9	Новосибирское ПО ВТИ	630088, Новосибирск, 88, а/я 77, ул. Петухова, 16/1; тел. 42-98-25			Алтайский	Кемеровская, Новосибирская, Омская, Томская, Тюменская
10	Северо-Кавказская хозяйственная ассоциация ВТИ ИФОСЕРВИС	344017, Ростов-на-Дону, пр. Ленина, 91/1; тел. 32-22-87		Чечено-Ингушская, Кабардино-Балкарская, Дагестанская, Северо-Осетинская	Краснодарский	Ростовская
11	Свердловское ПО ВТИ	620028, Свердловск, ул. Нагорная, 12; тел. 72-79-39		Башкирская, Удмуртская		Курганская, Оренбургская, Пермская, Челябинская, Свердловская
12	Хабаровское ПП ВТИ	680038, Хабаровск-38, а/я 1184, ул. Пушкина, 31; тел. 38-92-29			Хабаровский	
13	Азербайджанское ПО ВТИ	3701232, Баку, ул. Степная, квартал 2578, д. 3; тел. 25-29-85	Азербайджанская ССР			
14	Армянское НПО ВТИ	375038, Ереван, ул. Абеяна, 6/1; тел. 35-01-31	Армянская ССР			
15	Белорусское ПО ВТИ	220004, Минск, ул. Островского, 12 а; тел. 23-58-19	Белорусская ССР			
16	Грузинское НПО ВТИ	380019, Тбилиси, ул. Церетели, 130; тел. 84-28-44	Грузинская ССР			
17	Казахское НПО ВТИ	480012, Алма-Ата, ул. Комсомольская, 101; тел. 69-17-24	Казахская ССР			
18	Киргизское ПП ВТИ	720009, Фрунзе, проспект 50 лет Киргизской ССР; тел. 29-72-34	Киргизская ССР			
19	Латвийское ПО ВТИ	226063, Рига, ул. Кенгарага, 10; тел. 26-07-00	Латвийская ССР			
20	Литовское ПО ВТИ	223041, Каунас, проспект Тайкос, 141; тел. 77-71-83	Литовская ССР			

№ пп	ПО (НПО)		Зона деятельности			
	Наименование	Адрес: телефон	Союзная республика	АССР, автономный округ	Край	Область
	1	2	3	4	5	6
21	Молдавское ПП ВТИ	277022, Кишинев, ул. Сортировочная, 15; тел. 62-65-11	Молдавская ССР			
22	Таджикское ПП ВТИ	734030, Душанбе, ул. Аяни, 259; тел. 25-25-95	Таджикская ССР			
23	Туркменское ПП ВТИ	744015, Ашхабад, ул. Рабочего класса, 38; тел. 2-19-91; 2-20-37	Туркменская ССР			
24	Украинская хозяйственная ассоциация ВТИ ИНКОМ	252155, Киев, ул. Ф. Пушиной, 30/32; тел. 444-80-00	Украинская ССР			
25	Узбекское ПО ВТИ	700096, Ташкент, ул. Мукими, 43; тел. 78-85-44, 76-71-85	Узбекская ССР			
26	Эстонское НПО ВТИ	200026, Таллинн, бульвар Кадака, 165; тел. 53-73-03	Эстонская ССР			

Телефон 245-25-96, Москва

Статья поступила 11.08.89.

УДК 34.002.6

В. В. Захаров

АВТОРСКИЕ ПРАВА РАЗРАБОТЧИКОВ ПРОГРАММНЫХ СРЕДСТВ

Введение. Несоответствие правовых норм регулирования отношений в создании и использовании программных средств вычислительной техники (ПС ВТ) потребностям повседневной хозяйственной деятельности признается в настоящее время фактором, сдерживающим развитие информатики даже в большей степени, чем недостаточный выпуск аппаратных средств. Отсутствие правовой охраны ПС ВТ препятствует созданию рынка ПС и тем самым затрудняет использование имеющейся ВТ. Это тот случай, когда экономические факторы для нормального функционирования и развития нуждаются в закреплении законом. Такое закрепление предопределено и является только вопросом времени и юридической техники. «Всякий раз, когда развитие промышленности и торговли создавало новые формы общения..., право вынуждено было их санкционировать...».

Как известно, правовые законы могут либо способствовать реализации объективных закономерностей экономического развития, либо препятствовать проявлению регулирующей роли этих закономерностей. Применительно к охране ПС ВТ эти общие положения нуждаются для своей реализации в решении конкретного вопроса: в какой правовой форме следует закрепить использование ПС ВТ, чтобы она соответствовала экономическим закономерностям и не навязывала такой охраны, которая препятствовала бы нормальному развитию отношений в этой сфере.

Вопросам правового режима создания и использования ПС ВТ в последнее время посвящено много публикаций. Судя по большинству из них, для ликвидации отставания правового регулирования общественных отношений в области информатики от уровня развития самих отношений недостаточно дополнений и усовершенствований действующего законодательства, так сказать эволюционных мер. Требуется создание новой отрасли права — права информатики [1], или программного права [2].

Многие организации каждодневно испытывают несовершенство правового регулирования в области использования программной продукции в хозяйственном обороте. Действующее хозяйственное законодательство располагает рядом документов, определяющих правовой режим ПС.

Первоначально появился документ [3], предусматривающий придание ПС статуса продукции производственно-технического назначения. Этот документ официально не отменен до сих пор, хотя постановлением ГКВТИ СССР от 29 июля 1988 был принят новый документ [4] по тому же вопросу и с тем же названием. В нем со ссылкой на [5] ПС подразделяются на два вида: научно-техническую продукцию и продукцию производственно-технического назначения. Однако развитие сети компьютерных центров, центров информатики, досуговых центров, где население за умеренную плату может пользоваться ВТ, а также наличие у населения значительного числа бытовых компьютеров поставило задачу насыщения рынка ПС, являющимися не продукцией производственно-технического назначения (что относится к средствам производства) и не научно-технической, а товарами народного потребления, как например канцтовары или средства оргтехники. Действующие нормативные акты такой возможности не предусматривают. Кроме того, ответственность за нарушение авторских прав не устанавливается [6].

В связи с этим в Минэлектронпроме разработан проект положения об авторском праве на ПС ВТ, основанный на использовании существующих и оправдавших себя правовых механизмов.

Возникновение авторских прав на ПС ВТ. Основанием возникновения первичных авторских прав является факт создания произведения (например, ПС ВТ), а вторичных — договор с автором. Соответственно моментами возникновения авторских прав будут даты создания произведения и заключения договора. Эти известные положения приходится повторять в связи с тем, что нет конца попыткам применить в авторском праве подходы, свойственные изобретательскому праву. Например, по изобретательскому праву лицо может быть признано (или не признано) автором изобретения только после того, как предложенное техническое решение пройдет государственную экспертизу на новизну и полезный эффект, а компетентные государственные органы зарегистрируют его и выдадут (или не

* Маркс К., Энгельс Ф. Соч.—2-е изд.—Т. 3.—С.64.

выдадут) соответствующий охранный документ — авторское свидетельство или патент. Таким образом, в изобретательском праве от момента создания технического решения до момента обретения авторских прав может пройти и, как правило, проходит несколько лет.

В авторском праве отказ издателей опубликовать произведение какого-либо автора никоим образом не ставит под сомнение его авторские права. И ответственность (реальную, экономическую) несет та организация, которая приняла решение сотрудничать или не сотрудничать с этим автором. Поэтому документ, определяющий авторские права разработчиков ПС ВТ, должен, на наш взгляд, содержать констатацию того факта, что «право авторства возникает в момент создания произведения и является неотчуждаемым» (т. е. первичное авторское право — «право авторства» принадлежит только создателю произведения, а остальные субъекты авторских правоотношений могут обладать только теми авторскими правомочиями, которые автор вправе уступить по договору, например право на опубликование, тиражирование, перевод).

Понятие ПС ВТ. С позиций авторского права и хозяйственного законодательства ПС ВТ — это объекты, обладающие совершенно разными свойствами. Если для хозяйственного законодательства важно наличие свойств, характеризующих полноценный товар (соблюдение ГОСТ, правил маркировки, соответствие категориям качества и проч.), то для авторского права важно видеть в ПС ВТ произведение интеллектуальной творческой деятельности, а выражено оно может быть в любой объективной форме. Поэтому, на наш взгляд, для обеспечения охраны авторских прав ПС ВТ достаточно определить как программу или комплекс программ на носителе данных либо как программный документ, содержащий текст программы (например, на исходном языке или в автокодах.) Все прочие программные документы, содержащие сведения, необходимые для создания, функционирования и эксплуатации ПС, могут быть признаны неотъемлемой частью конкретного ПС при выдаче технического задания на разработку или при заключении авторского договора.

Это нужно, во-первых, потому, что единая система программной документации предусматривает комплектование программного изделия соответствующей программной документацией.

Во-вторых, автор программы может написать не самую доходчивую и удобную программную документацию на разработанное им ПС ВТ.

В-третьих, программная документация с позиций гражданского законодательства является принадлежностью главной вещи — программного изделия. А принадлежность, т. е. вещь, предназначенная служить главной вещи и связанная с ней общим хозяйственным назначением, следует судьбе главной вещи, если законом или договором не установлено иное (ст. 139 ГК РСФСР). Таким образом, правовой режим, установленный для главной вещи (в том числе и режим объекта авторского права), распространяется и на принадлежность.

В-четвертых, авторским правом предусмотрено, что всякие вспомогательные материалы, например комментарии и какие бы то ни было пояснения, что прямо относится к эксплуатационной документации, могут сопровождать произведение только с согласия автора (ст. 480 ГК РСФСР). Поэтому представляется, что автор программы вправе определять (в договоре, заключаемом с организацией, использующей произведение), какую программную документацию он составит лично, а какая будет разработана другими лицами и с его разрешения войдет в состав ПС ВТ.

Базы данных (БД) в отличие от систем управления базами данных (СУБД), т. е. программ и их комплексов, программами не являются. Однако для лиц, использующих ВТ, в принципе безразлична «начинка» аппаратных средств. Для них важна возможность использовать аппаратные средства в своих целях. С этой точки зрения БД являются ПС не хуже прочих. Провести четкую грань между прикладной программой и БД, используемой сов-

местно с СУБД, по функциональным свойствам вряд ли возможно. Поэтому было бы справедливым отнести к ПС не только программы, но и БД на машинных носителях, если включенные в них данные подобраны и обработаны лично авторами — составителями. Единственное отличие, важное с рассматриваемой точки зрения, между программами и БД заключается в том, что программы рассматриваются как ПС ВТ на машинном носителе или в виде исходного текста в печатном издании, а БД — только на машинном носителе и в сочетании с конкретной СУБД. Аналогично должен решаться вопрос об отнесении к ПС ВТ баз знаний.

Авторское право и алгоритмы. Представляется естественным защищать правом как программы, так и алгоритмы, лежащие в их основе. Однако в рамках авторского права это неосуществимо, так как оно охраняет не идеи, а конкретное их воплощение. А это значит, что алгоритм, так сказать ядро, существо программы, охране авторским правом не подлежит. Невозможна защита авторским правом идеи, концепции, метода, принципа или открытия. Охране подлежит форма выражения этих идей. Общеизвестно, что на один и тот же сюжет у разных народов существуют сказки разных авторов. И каждая из них «охраноспособна» с позиций авторского права.

В мировой практике идея произведения интеллектуального творчества охраняется патентным (изобретательским) правом, а форма — авторским правом. С этой точки зрения права те юристы, которые считают, что «математическое обеспечение... целесообразно было бы охранять с помощью норм изобретательского права прежде всего потому, что в этом случае исключительное право возникло бы непосредственно на алгоритм (идею), а не на одну из частных форм его внешнего проявления в виде программы» [7, с. 25]. Однако пока Госкомизобретений СССР не рассматривает заявки на математическое обеспечение ЭВМ [8].

Авторы ПС ВТ. Следует, на наш взгляд, законодательно установить, что «Авторами ПС могут быть только физические лица: граждане СССР, иностранные граждане или лица без гражданства (апатриды)».

Такая формулировка может вызвать многочисленные вопросы и возражения, так как исключает из числа возможных авторов юридические лица — предприятия и организации, и недоумение потому, что последние десятилетия только и говорится о расширении прав предприятий и организаций. К тому же основная масса ПС разрабатывается в организациях в порядке выполнения служебного задания, «за зарплату», а по действующим правилам все, что произведено на предприятии, является его собственностью (точнее, собственностью собственника средств производства).

По нашему глубокому убеждению (совпадающему с требованиями действующего законодательства), организации могут иметь право на продукт не интеллектуальной, а производственной или организационной деятельности. В отдельных случаях (закрепленных законом) результат организационной деятельности имеет самостоятельную творческую ценность. Например, энциклопедия — гармоничная совокупность статей, авторское право на каждую из которых принадлежит автору статьи, а на всю энциклопедию в целом — организации, «организовавшей» этих авторов на создание этого качественно нового самостоятельного произведения.

Но случаев, когда результатом организационной деятельности является произведение, обладающее качественно новыми положительными свойствами по сравнению с теми произведениями, которые оно в себя включило, не так уж много, они все перечислены в законе, например авторское право организаций на кино- и телефильмы, радио- и телепередачи (ст. 486 ГК РСФСР).

Не исключено с этих позиций признание авторского права юридических лиц на сложные операционные системы, системы программирования и другие сложные комплексы программ, где организациям принадлежало бы первичное авторское право на совокупное произведение, включающее

части, имеющие самостоятельное значение, на которые закреплено авторское право конкретных лиц. Но пока этого нет, и нет, на наш взгляд, оснований горевать об этом: до сих пор расширение прав организаций проявлялось в расширении прав их должностных лиц, а не творческих работников.

Авторство на ПС, созданные в порядке выполнения служебного задания. Как при данном подходе охранять права организаций на произведения, созданные их сотрудниками в порядке выполнения служебных обязанностей? И вообще, выходит, что организация не имеет прав на такие произведения? На эти вопросы в законодательстве содержится ответ: если произведением является промышленная или сельскохозяйственная продукция, то такое произведение — полная собственность собственника средств производства, но если произведение — продукт творческого труда, то авторство закрепляется персонально за его создателем. Это определено, например, в ст. 483 ГК РСФСР: «Автору произведения, созданного в порядке выполнения служебного задания в научной или иной организации, принадлежит авторское право на это произведение».

Было бы неверно понимать изложенное как полное удовлетворение законодательным регулированием в этой сфере. Именно здесь требуется нормативно-правовое решение острейших практических вопросов. Во второй части только что упомянутой статьи говорится: «Порядок использования организацией такого произведения и случаи выплаты вознаграждения автору устанавливаются законодательством СССР или постановлениями Совета Министров СССР». Из этого можно сделать по крайней мере три вывода: 1) порядок использования таких произведений должен быть специально установлен, т. е. его нужно найти в законодательстве или создать (автор и организация не вправе установить его по своему соглашению); 2) вознаграждение за такие произведения выплачивается не во всех случаях; 3) решение этих вопросов не входит в компетенцию ни одного министерства или госкомитета, в том числе и ГКВТИ СССР. Это должен быть орган не ниже Совета Министров союзной республики, поэтому вопросы авторского права на ПС ВТ не могут быть решены ни самостоятельно ГКВТИ СССР, ни совместно министерствами и ведомствами.

Порядок использования организацией произведений, созданных в рамках выполнения служебного задания, на сегодня авторским правом не установлен. В литературе встречаются только общие рассуждения о том, что «организация не вправе без согласия автора осуществлять его публикацию... вносить в произведение изменения и т. п. А если автор не даст разрешения на опубликование произведения, не внесет в него поправки в соответствии с рекомендациями... администрации, произведение не может быть опубликовано, но автор будет нести ответственность по правилам трудового права» [9, с. 572, 573]. Далее приводятся рассуждения о том, означает ли наделение организации правом использования произведения одновременно лишение автора права аналогичных действий.

Рассуждения заканчиваются констатацией, что это «закон прямо не регламентировано» [9, с. 572, 573]. Тем не менее, предполагая, что права автора в этом случае все же ограничены, в Комментариях к ГК РСФСР записано: «Представляется, что... ограничение прав автора не должно действовать по истечении 3-х лет с момента завершения им работы». По нашему мнению все эти предположения, прямо на законе не основанные, свидетельствуют о серьезном пробеле в действующем законодательстве.

За произведения, созданные в порядке выполнения служебного задания, вознаграждение, помимо зарплаты, в принципе не выплачивается. В порядке исключения установлена выплата вознаграждения авторам за издание и переиздание учебников и учебных пособий после утверждения их соответствующими организациями [10], а также за второй и последующие тиражи произведений декоративно-прикладного искусства, хотя и со снижением на 50 % [11].

Такое законодательное решение вопроса представляется не соответствующим складывающемуся хозяйственному механизму. Признание произведений научно-технического творчества, результатов исследований и разработок товаром (товарной продукцией) [12] должно повлечь изменение и порядка вознаграждения за произведения, выполненные по служебному заданию. Конечно, можно требовать, чтобы автор отдал за зарплату (он и так бы ее получил) результат, который любая другая организация рада была бы приобрести на самых выгодных для него условиях. Но тогда призывы быть патриотом своего предприятия будут противоречить реальным имущественным интересам наиболее творческих его сотрудников.

Одно из средств, способных устранить такое противоречие — это, на наш взгляд, законодательное закрепление прав автора на долю от дохода за хозяйственное использование его произведения. Этот подход мог бы получить нормативное закрепление в следующей форме: «Порядок использования организацией произведений, созданных при выполнении служебного задания, а также порядок и размер выплаты вознаграждения авторам определяются в коллективном договоре, заключаемом профсоюзным комитетом от имени коллектива с администрацией предприятия. Порядок использования конкретного произведения и размер авторского вознаграждения за него должны быть установлены в пределах, определенных коллективным договором, в техническом задании на разработку ПС ВТ, выдаваемом перед началом работ». Такое стимулирование должно осуществляться не из средств госбюджета, а из средств предприятия. Соответственно его размер должен определяться самим предприятием с учетом реальной конъюнктуры на произведения каждого вида. Наиболее прибыльные для предприятия ПС будут стимулироваться сильнее, что вызовет более интенсивную и качественную их разработку. С другой стороны, предприятие, не создавшее для своих творческих работников достаточно благоприятного режима, рискует не только не получить наиболее эффективные разработки, но и растерять своих одаренных работников.

Соавторство. Законом установлено, что на произведение, созданное совместным творческим трудом двух или более лиц, авторское право принадлежит соавторам совместно независимо от того, образует ли такое произведение одно неразрывное целое или состоит из частей, каждая из которых имеет самостоятельное значение.

Вопросы соавторства достаточно полно урегулированы действующим авторским правом, за исключением одного момента. В законодательстве отсутствует охрана прав автора от принудительного «соавторства» должностных лиц. До сих пор этот вопрос был очень болезненным для изобретательского права. В авторском праве вопрос так остро не стоял, так как произведения, созданные в порядке выполнения служебного задания, составляют ничтожную долю по отношению ко всем охраняемым произведениям, да еще по ним не выплачивается вознаграждение. В изобретательской деятельности, где доля «служебных» изобретений высока, а вознаграждение, когда дело доходит до его выплаты, существенно, претендентов на соавторство специально искать не приходится. В изобретательском праве эту проблему пытались смягчить вознаграждением за содействие изобретательству и рационализации [13], а также изданием специального разъяснения, утвержденного Госкомизобретений «О соавторстве на изобретения, созданные в связи с выполнением служебного задания» [14], в котором предусматривается ряд мер, направленных на предотвращение «включения в число соавторов руководителей и других должностных лиц, не принимавших творческого участия в работе над созданием данного изобретения» [14, с. 302]. Необходимо введение мер аналогичного характера также и в авторском праве.

Предлагается законодательно закрепить, что должностное лицо, в подчинении которого находится автор ПС, созданного в организации в порядке выполнения служебного задания, не может быть признано соавтором (а соответственно не может обладать авторскими правами, в том

числе и правом на вознаграждение), если им осуществлялись только координирующие и распорядительные функции. Требования к разрабатываемому ПС должны быть определены в техническом задании на разработку до ее начала. Вмешательство должностных лиц в процесс создания ПС запрещается. Руководитель разработки (темы) обязан личным творческим трудом создать основную и имеющую самостоятельное значение часть ПС. Лица, осуществлявшие консультирование и дававшие рекомендации, могут быть включены разработчиками в число соавторов по окончании разработки.

Авторство лиц, использующих чужое ПС для создания нового. В соответствии с положениями авторского права (ст. 494 ГК РСФСР) самостоятельное авторское право принадлежит автору нового произведения, созданного с использованием чужого произведения в порядке, предусмотренном законом. Это право не прелятствует другим лицам использовать в том же порядке любое из этих произведений для создания нового. Для использования опубликованного произведения, его перевода или перелдки в произведение другого вида (например, повествовательного в драматическое или сценарий и наоборот) требуется согласие автора оригинального произведения, уплата ему соответствующего вознаграждения.

Без согласия автора и без уплаты авторского вознаграждения, но с указанием фамилии автора и источника заимствования допускается использование чужого изданного произведения для создания нового, творчески самостоятельного произведения. Это, вообще-то, очевидные положения: хочешь перевести на русский язык сказку К. Коллоди «Пиноккио», получи разрешение и выплати вознаграждение, а можешь сочинить на ее основе своего «Буратино» — сочиняй, нет препятствий. Кроме того, авторское право знает случаи использования произведений без согласия автора, но с выплатой авторского вознаграждения. Например, использование композитором изданных литературных произведений для создания музыкальных произведений с текстом. Но, увы, этот случай не применим к заимствованию опубликованных ПС ВТ для создания новых.

Таким образом, действующее авторское право на использование чужого опубликованного произведения для создания нового в одних случаях сохраняет только право на имя, в других, — дополнительно, на вознаграждение, а в-третьих, кроме того, требует предварительного согласия автора произведения. Такой подход полностью применим при регулировании порядка использования опубликованных ПС ВТ для создания на их основе новых ПС. Более того, первый и третий случаи прямо предусмотрены действующим законодательством (ст. 488 и ч. 1 ст. 492 ГК РСФСР). А вот второй, промежуточный, вариант, когда требовать получения предварительного разрешения было бы излишне, а не выплатить вознаграждение было бы несправедливо, к ПС ВТ не относится (ст. 495 ГК РСФСР содержит исчерпывающий перечень таких случаев). Однако, на наш взгляд, следует закрепить возможность использовать опубликованное ПС ВТ для существенного повышения его потребительских свойств без согласия автора оригинала (программы-прообраза), но с выплатой ему вознаграждения. Сюда мы относим случаи, так сказать, «последовательного соавторства», когда есть возможность более полно и оперативно удовлетворить общественную потребность путем модернизации ПС.

В связи с этим предлагается закрепить следующие возможности использования чужих ПС ВТ для создания новых: «Автору адаптации или перевода, созданного по договору с автором оригинала, принадлежит авторское право на созданное им произведение. Без согласия автора прототипа, но с указанием его имени и с выплатой ему причитающейся доли вознаграждения допускается использовать чужое опубликованное ПС для создания нового, существенно усовершенствованного. Существенным усовершенствованием ПС следует считать внесение изменений, позволяющих повысить быстродействие, снизить объем занимаемой памяти или повысить надежность работы системы, если это значительно повышает потребительские

свойства ПС на вычислительных машинах того класса, на которые оно ориентировано; расширить или существенно изменить круг задач, решаемых с его применением; повысить производительность и комфортность труда пользователей ПС ВТ для конкретного круга задач.

Допускается без согласия автора и без уплаты авторского вознаграждения, но с обязательным указанием имени автора и источника заимствования использование чужого опубликованного ПС для создания нового, творчески самостоятельного произведения.

В случаях, когда какая-либо программа (базовая) предоставляет возможность для создания целого поколения программ (нисходящих) с широким диапазоном использования и с новыми потребительскими свойствами (т. е. когда базовые и нисходящие программы образуют один функциональный комплекс), разработчики базовой и нисходящей программ имеют авторские права на свои программы. Возможность совместной публикации комплекса программ определяется распространителем (издателем) по согласованию с авторами.

Размер вознаграждения, на наш взгляд, должен определяться соглашением между автором нового ПС и организацией-издателем с уведомлением автора использованного произведения о размере его доли. Если он сочтет выделенную ему долю вознаграждения не соответствующей его творческому вкладу во вновь созданное ПС, он вправе оспорить решение через суд.

Программистский фольклор. Практическое отсутствие режима правовой охраны авторских прав разработчиков ПС ВТ, а также реальных хозяйственных путей использования ПС ВТ привело к хождению по рукам большого числа ПС, авторы которых зачастую неизвестны. Предприятия, которые захотели бы тиражировать такие пользующиеся спросом и прошедшие практическую проверку ПС ВТ, встали бы перед проблемой, как удовлетворить общественную потребность (спрос) и не нарушить права неизвестного автора и прежде всего на вознаграждение. Использование в хозяйственной деятельности (т. е. для получения прибыли) этого программистского фольклора представляет актуальную правовую и нравственную проблемы. Варианты их решения дают международные конвенции по авторскому праву. Так, Бернская конвенция об охране литературных и художественных произведений [15, с. 109] в п. 4 ст. 15 предусматривает, что «для неопубликованных произведений, автор которых неизвестен, но в отношении которых есть все основания предполагать, что он является гражданином страны... за законодательством этой страны сохраняется право назначить компетентный орган, представляющий автора и правомочный защищать его права и обеспечивать их осуществление...».

Исползовать этот механизм для защиты прав авторов ПС, утративших контроль над своими произведениями, можно было бы, установив следующее правило: «Для неопубликованных ПС, авторы которых неизвестны, но есть основания предполагать, что они граждане СССР, полномочным представителем авторов-разработчиков по просьбе заинтересованных организаций может выступать ГосФАП в лице организации, ведущей соответствующий фонд алгоритмов и программ (ФАП), которая вправе заключать договоры об использовании указанных ПС и аккумулировать суммы вознаграждения, причитающегося авторам этих произведений. Гражданин, доказавший свое авторство на такое ПС, становится стороной по всем договорам, заключенным ранее организацией, ведущей ФАП, по поводу использования этого ПС, и правомочен впрямь пользоваться авторскими правами по своему усмотрению. Средства, аккумулированные за счет использования созданного им произведения (за вычетом понесенных расходов), выдаются (перечисляются) ему в месячный срок со дня признания его автором произведения (по его заявлению)».

Подготавливая такое предложение, мы не рассчитываем, что выполнение этих функций возьмет на себя Всесоюзное агентство по авторским правам (ВААП). В соответствии со сложившейся практикой регистрацию алгоритмов и программ в стране осуществляют органы ГосФАП. Даже если

эти функции будут возложены на создаваемую в настоящее время Государственную систему программного обеспечения (ГСПО), они все равно не должны дублироваться ВААПом. К тому же ВААП осуществляет регистрацию и учет только тех произведений, «за использование которых выплата авторского вознаграждения осуществляется через ВААП» (п. 1 «о» Устава ВААП [16, с. 307]). Поэтому охрану фольклора предлагается возложить на организации, ведущие учет и регистрацию ПС, т. е. ГосФАП или ГСПО.

Личные немущественные права авторов-разработчиков ПС ВТ. Предлагается выделить семь основных немущественных авторских правомочий. Ниже дана краткая характеристика шести из них, седьмое — право на использование чужого опубликованного произведения для создания нового — уже рассматривалось.

Право авторства. Лицу, творческим трудом которого создано ПС, принадлежит право авторства на это ПС, т. е. право на признание другими лицами факта создания им данного произведения. Право авторства возникает в момент создания произведения, не отчуждается и не передается по какому-либо основанию. Отказ от него недействителен.

Право на имя. Автору ПС принадлежит право на использование своего произведения под своим именем, под условным (но не чужим) именем (псевдонимом) или без обозначения имени (анонимно). Право на имя не передается.

Право на опубликование и воспроизведение. Автору ПС принадлежит право на опубликование своего произведения, в том числе разрешение опубликования любым образом и в любой форме. ПС считается опубликованным (выпущенным в свет), если оно издано (т. е. опубликовано в печатной форме), распространено каким-либо иным способом, сообщено неопределенному кругу лиц. ПС не считается опубликованным, если ознакомиться с ним могли только специально допущенные лица.

Не считается опубликованием (выпуском в свет) информация о ПС с изложением его возможностей и основного принципа работы, так как объектом охраны является не алгоритм, не ставившиеся цели и общие методы их достижения, а конкретная реализация алгоритма, конкретный способ достижения поставленных целей.

Организация, в порядке выполнения задания которой создано ПС, не вправе без согласия автора опубликовать его, так как автору принадлежит право принимать решение о завершенности произведения и его пригодности к публикации.

Автору ПС принадлежит право на воспроизведение своего произведения, включая право разрешать воспроизведение любым образом и в любой форме. Воспроизведение (дублирование, тиражирование или иное размножение) ПС — это повторное (или неоднократное) придание ПС объективной формы или размножение материальных носителей этого ПС. Воспроизведение и опубликование ПС с коммерческой целью может осуществляться только юридическими лицами (государственной, кооперативной организацией и проч.) и только по договору с автором (с разрешения автора). Допускается без согласия автора и без уплаты авторского вознаграждения воспроизведение полученных на законном основании ПС для удовлетворения личных потребностей (без извлечения прибыли), а также для целей архивного хранения. Право на воспроизведение и опубликование может быть уступлено автором по договору какой-либо организации для коммерческого использования.

Право на неприкосновенность. Автору ПС принадлежит право на неприкосновенность произведения, т. е. право требовать признания произведения целостным и завершенным. В связи с этим воспрещается без согласия автора вносить какие бы то ни было изменения в само произведение, его название, имя автора, а также снабжать его иллюстрациями, предисловиями, комментариями и какими бы то ни было пояснениями, в том числе эксплуатационной документацией. Не является нарушением указанного требования использование в законном порядке чужого опубликованного ПС для создания нового. Право на непри-

косновенность не передается. Отказ от него не действителен.

Право на переделку, адаптацию и перевод. Автор ПС пользуется правом переделывать свое произведение для получения аналогичного или сходного ПС, а также правом адаптировать, переводить его или разрешать внесение изменений, адаптацию и перевод. Переделкой ПС считается такое его изменение, которое не делает его новым, творчески самостоятельным произведением, т. е. без существенных изменений, дополнений или сокращений. Адаптация ПС — изменение с целью сделать возможным применение ПС в другой среде. Например, адаптация к ВТ другого типа или с другим набором аппаратных средств, для работы в другой операционной или программной среде, к другому кругу пользователей. Перевод ПС — его адаптация путем изложения на другом языке программирования, иностранном языке или языке народов СССР.

Имущественные права авторов-разработчиков ПС ВТ. Многие немущественные права не передаются, автор не вправе их уступить, и не будет иметь правового значения его отказ от прав. Так, не передаются права авторства, на имя, неприкосновенность, использование чужого произведения для создания нового. Но два комплекса правомочий могут быть уступлены, причем за вознаграждение. Это право на переделку, адаптацию и перевод и право на опубликование и воспроизведение.

Относительно переделки, адаптации и перевода у автора есть выбор: перевести (переделывать, адаптировать) свое произведение самому или уступить эту возможность кому-либо за вознаграждение. А вот опубликование и воспроизведение ПС с коммерческой целью, т. е. для получения дохода отдельными гражданами в виде индивидуальной трудовой деятельности не разрешено. Здесь автор может выбирать, какой именно организации он уступит это свое право, но только в случае, если ПС разрабатывалось не в порядке выполнения служебного задания.

Кроме того, автор имеет в отдельных случаях право на вознаграждение за использование его произведения для создания нового.

Авторское вознаграждение должно выплачиваться организациями-издателями ПС по каждому из названных оснований. Например, вознаграждение за издание исходного текста программы в виде печатного изделия и вознаграждение за воспроизведение программы на машинных носителях должны выплачиваться независимо друг от друга.

Вознаграждение за коммерческое использование ПС должно определяться, на наш взгляд, таким образом, чтобы возмещались общественно необходимые затраты на разработку и стимулировалось удовлетворение спроса, т. е. общественной потребности в данном ПС. Для этого в авторском договоре следует предусматривать выплату единовременного вознаграждения в сумме, пропорциональной трудозатратам на разработку ПС; потиражного вознаграждения в сумме, пропорциональной спросу на данное ПС (в процентах от цены или в твердом размере с каждого изготовленного изделия или с каждой сделки по его переписи, прокату).

Размер вознаграждения должен определяться соглашением сторон (т. е. автора и организации, которой автор уступает права на коммерческое использование ПС, например организации-издателю) в пределах действующих ставок авторского вознаграждения, а при отсутствии утвержденных вышестоящими органами ставок размер авторского вознаграждения должен согласовываться сторонами с учетом конъюнктуры рынка ПС ВТ и возможностей сторон.

При функциональном использовании ПС, созданного в порядке выполнения служебного задания, только внутри организации, выдавшей задание на его разработку, авторское вознаграждение выплачиваться не должно.

При коммерческом использовании ПС организацией, выдавшей задание на его разработку, единовременное вознаграждение выплачиваться не должно, а потиражное может выплачиваться в порядке и пределах, устанавливаемых в коллективном договоре. Однако следует нормативно закрепить, что потиражное вознаграждение в этом случае

выплачивается в доле, устанавливаемой в техническом задании на разработку ПС, но не менее, например, 10 % и не более 100 % от сумм, подлежащих выплате за аналогичную разработку, выполненную не по служебному заданию. Остальная часть потиражного вознаграждения перечисляется в фонд материального поощрения этой организации.

Размер единовременного авторского вознаграждения можно рассчитать, применяя Укрупненные нормы времени на разработку ПС ВТ, утвержденные Постановлением Госкомтруда СССР и ВЦСПС от 24 сентября 1986, № 358/22-20.

Представляется, что нет необходимости устанавливать общесоюзные ставки авторского потиражного вознаграждения, а также ограничивать его минимальный или максимальный размер. Решение этого вопроса следует закрепить за сторонами (автором и использующей организацией), чтобы размер вознаграждения был взаимовыгоден в конкретных условиях.

Ответственность за неправомерное использование ПС. Переход предприятий и организаций на хозяйственный расчет, резкое увеличение числа производственных кооперативов привели к расширению круга субъектов, для которых доходность производства — основной критерий успешности деятельности. В этих условиях важно удержать коммерческую предприимчивость в берегах законности, для чего требуется создать такой правовой режим, при котором правомерная предприимчивость обеспечивала бы получение высоких прибылей, а нарушение закона или договорных обязательств было бы чревато экономическим крахом. К сожалению, в настоящее время такой механизм ответственности в хозяйственных отношениях фактически отсутствует: предприятия могут нарушать договорные обязательства, а их работники будут по-прежнему получать премии. Аналогичная ситуация наблюдается и в авторском праве. Так, действующим законодательством (ст. 499 и 500 ГК РСФСР) предусмотрено, что в случае бездоговорного использования произведения или с нарушением условий договора автор имеет право требовать: восстановления нарушенного права (например, публикации в печати о допущенном нарушении); запрещения выпуска произведения в свет или прекращения его распространения; возмещения убытков.

Если учесть, что нарушение авторских прав, например плагиат, есть по существу крах интеллектуальной собственности, то напрашивается аналогия из области уголовного права. Представьте себе, что вас обокрали, а закон охраняет ваше право на публикацию этого факта в печати, право требовать запрещения жулику пользоваться краденым и право требовать возврата похищенного. Думается, что вряд ли таким образом можно пресечь воровство. Вряд ли, жулика испугает необходимость вернуть похищенное. Более серьезных мер ответственности действующее авторское право не предусматривает. Если учесть, что судебный порядок применения мер такой ответственности связан с значительными затратами времени и сил, то неудивительно, что к нему прибегают только в случае крупного нарушения имущественных прав.

В странах с более развитыми рыночными отношениями ответственность за нарушение авторских прав значительна, а уличение в сознательном нарушении закона ставит нарушителя на грань банкротства. Так, по американскому федеральному закону об охране авторских прав потерпевшая сторона вправе по суду претендовать на получение компенсации в размере 50 тыс. долл. за каждую программу, с которой была снята копия. В [17] есть сообщения о практике применения этого закона. Кроме того, международное авторское право (ст. 16 Бернской конвенции) предусматривает арест (конфискацию) контрафактных (незаконно изготовленных) экземпляров [15]. Действующее отечественное гражданское законодательство предусматривает взыскание в доход государства (конфискацию) имущества по сделкам или действиям, совершенным с целью, противной интересам государства и общества (ст. 49 ГК РСФСР). Однако в случаях нарушения авторских прав это положение не применяется, как будто кража интеллектуальной

собственности не противна интересам государства и общества.

Для действенной охраны авторских прав разработчиков следует, на наш взгляд, установить имущественную ответственность за их нарушение в таком размере, чтобы это имело не только компенсационное, но и большое общепредупредительное значение.

Прежде всего в соответствии с принципами гражданского права виновная сторона должна возместить потерпевшей стороне все убытки, которые в таких случаях будут в основном включать в себя «упущенную выгоду», т. е. те не полученные доходы, которые потерпевшая сторона получила бы, если бы закон или договор не был нарушен (ч. 2 ст. 219 ГК РСФСР). На сегодня практика взыскания «упущенной выгоды» в хозяйственных взаимоотношениях не распространена, а арбитражи на удовлетворение таких исков идут крайне неохотно. Обычно недопоставка kolejных комплектующих срывает выпуск дорогостоящих изделий, но виновная сторона отделяется легким испугом (8...12 % от стоимости поставляемых комплектующих). Представляется очевидным, что перенос этой практики (вопреки требованиям действующего законодательства) в процессуальную деятельность авторского права может фактически парализовать правовое регулирование.

Размер убытков, на наш взгляд, следует определять с учетом договоров, заключенных на тот же вид использования того же ПС с другими сторонами; действующих ставок авторского вознаграждения, как утвержденных, так и конъюнктурных, применяемых для аналогичных произведений и способов их использования; экспертных оценок качества и товарной ценности ПС; других материалов, обосновывающих размер причиненных убытков.

Таким образом, предлагается определить размер ответственности за нарушение авторских прав пропорционально причиненным убыткам (упущенной выгоде). Однако ограничиться только упущенной выгодой — все равно, что отобрать у вора похищенное. В этом еще нет санкции, нет заслуженных неблагоприятных последствий. Не соблюдение авторских прав должно наказываться неустойкой (штрафом). Причем убытки могут взыскиваться в полной сумме сверх штрафа (ч. 2 ст. 189 ГК РСФСР). Что касается штрафа, то он взыскивается не только в пользу потерпевшей стороны (ст. 187 ГК РСФСР), но и в доход государства (предпочтительно местного бюджета). Размер каждого вида штрафа (потерпевшему и в бюджет) следует установить не ниже размера причиненных убытков.

Повторное нарушение авторских прав должно наказываться строже (например, размеры подлежащих возмещению убытков и уплачиваемых штрафов содержат суммы ранее возмещенных убытков и ранее уплаченных штрафов).

Следует, наверно, закрепить право суда снизить размер ответственности с учетом материального положения виновной стороны в пределах сумм, подлежащих взысканию в доход государства.

Необходимо вероятно, предусмотреть и возможность компенсации убытков и выплаты штрафа в добровольном порядке. Например, при бездоговорном использовании ПС или несоблюдении условий договора по заявлению потерпевшей стороны виновная сторона должна иметь возможность добровольно в месячный срок выплатить сумму штрафа и причиненных убытков. В этом случае виновная сторона избавляла бы себя от штрафа в доход государства и оплаты судебных расходов.

Реальное обеспечение авторских прав разработчиков ПС ВТ позволит сформироваться рынку ПС, гибко и оперативно удовлетворяющему общественные потребности. Широкое обсуждение накопившихся проблем и предлагаемых различных способов их решения поможет подготовить документы, учитывающие как интересы сторон в отношениях создания и использования ПС ВТ, так и реалии складывающегося хозяйственного механизма.

Телефон 166-45-36, Москва

1. Карась И. З. Правовое регулирование общественных отношений в сфере информатики // Советское государство и право. — 1987. — № 3.
2. Масляев А. И. Проблемы правовой охраны программного обеспечения ЭВМ и объектов биотехнологии. Обзор выступлений участников научно-практической конференции // Вопросы изобретательства. — 1986. — № 5.
3. Положение о порядке разработки, производства, поставки и использования программных средств вычислительной техники, а также автоматизированных систем и систем обработки информации: Утв. Постановлением ГКНТ СССР 13 февраля 1984, № 41 // Правовая работа в народном хозяйстве: Сб. нормативных актов / Под ред. Б. В. Кравцова — М.: Юридическая литература, 1987.
4. Положение о порядке разработки, производства, поставки и использования программных средств вычислительной техники и информатики: Утв. Постановлением ГКВТИ СССР от 29 июля 1988. — М.: ГКВТИ, 1988.
5. Положение о правовом статусе субъектов отношений в области программных средств вычислительной техники и информатики: Утв. Постановлением ГКВТИ СССР от 29 июля 1988. — М.: ГКВТИ, 1988.
6. Положение об учете и охране авторских прав разработчиков программных средств вычислительной техники и информатики: Введ. в действие Постановлением ГКВТИ СССР от 30 ноября 1988, № 28. — М.: ГКВТИ, 1988.
7. Жуков В. И. Проблемы правовой охраны программного обеспечения ЭВМ и объектов биотехнологии. Обзор выступлений участников научно-практической конференции // Вопросы изобретательства. — 1986. — № 5.
8. О признании и изобретениями объектов вычислительной техники, характеризуемых математическим обеспечением ЭВМ. Разъяснение Госкомизобретений СССР от 13 мая 1975, № 4. — В сб.: Законодательство СССР по изобретательству. — М.: ВНИИПИ, 1981. — Т. 1.
9. Комментарии к Гражданскому кодексу РСФСР. — М.: Юридическая литература, 1982.
10. О ставках авторского вознаграждения за издание произведений науки, литературы и искусства. Постановление Совета Министров СССР РСФСР от 22 апреля 1975, № 243, разд. 8. // СП РСФСР. — 1975. — № 9.
11. Об авторском вознаграждении за использование в промышленности произведений декоративно-прикладного искусства. Постановление Совета Министров РСФСР от 2 июня 1960. № 793. // СП РСФСР. — 1960. — № 23. — ст. 104.
12. Положение о договорах на создание (передачу) научно-технической продукции (Утв. Постановлением ГКНТ СССР от 19 ноября 1987, № 435).
13. Положение о порядке премирования за содействие изобретательству и рационализации: Утв. Постановлением Госкомтруда СССР от 23 июня 1983, № 134 // Вопросы изобретательства — 1983. — № 12.
14. О соавторстве на изобретения, созданные в связи с выполнением служебного задания. Разъяснение Госкомизобретений СССР от 28 мая 1971, № 1 (29). — В сб.: Законодательство СССР по изобретательству. — М.: ВНИИПИ, 1981. — Т. 1.
15. Международные конвенции об авторском праве. Комментарии. — М.: Прогресс, 1982.
16. Устав Всесоюзного агентства по авторским правам (утвержден 20 сентября 1973 г.) // Авторское право: Сб. нормативных актов. — М.: Юридическая литература, 1985.
17. Стивен У. Филдс. Приговор суда — предупреждение использующим чужое программное обеспечение без разрешения // Электроника. — 1982. — № 20.

В. Л. Дунин-Барковский, А. Т. Терехин

НЕЙРОННЫЕ СЕТИ И НЕЙРОПРОЦЕССОРЫ: ТЕНДЕНЦИИ РАЗВИТИЯ ИССЛЕДОВАНИЙ И РАЗРАБОТОК

За последние три года в мире резко возрос интерес к анализу механизмов работы нейронных систем и построению их технических аналогов [1, 2]. Ведущие электронные компании и правительственные органы по субсидированию научных разработок в США, Японии и Западной Европе подключились к работам в данном направлении. Появившееся 20 лет назад в качестве анекдотической описки словосочетание «нейронная бомба»* сейчас всерьез употребляется некоторыми представителями ведомств, ответственных за перспективные военные разработки [6, 7]. Уверенно вошло в научно-техническую терминологию слово «нейрокомпьютер», впервые, по-видимому, употребленное Робертом Хехт-Нилсеном в названии своей фирмы по производству средств моделирования нейронных сетей [8].

Можно сказать, что ситуация в области нейронных сетей имеет характер научно-технического «бума», масштаб которого по порядку величин не уступает возмущению, вызванному открытием высокотемпературной сверхпроводимости. Кратко охарактеризуем основные особенности этого бума, практически не касаясь существа проблемы, а описывая лишь ее «наукометрический» аспект.

Можно выделить три основные причины резкого повышения интереса к нейропроцессам и «нейропроцессорам».

Во-первых, размеры активных функциональных элементов электронных схем сейчас сравнялись с размерами синапсов — элементарных активных составляющих нервной системы, в то время как качество переработки информации в нейронных системах во многих случаях превосходит качество работы электронных систем (и это несмотря на то, что быстрдействие электронных схем на шесть порядков выше).

Во-вторых, успехи систем искусственного интеллекта, основанных на формально-логическом дедуктивном подходе, оказались пока более скромными, чем это ожидалось. Поэтому преимущество «математически строгого» подхода к созданию систем искусственного интеллекта перед «эмпирическим» подходом, основанным на повторении того, что уже реализовано в природе, сейчас представляется сомнительным.

В-третьих, внутренняя логика развития вычислительной техники ведет к появлению систем со всевозрастающей степенью параллелизации, а в нейросистемах представлены уже отработанные тысячелетиями параллельные системы вычислений и переработки информации, обладающие к тому же высокой защищенностью по отношению к повреждению элементов и «ошибкам при монтаже».

Несколько слов об основных этапах развития данной отрасли.

Началом теоретического исследования нейронных сетей можно считать опубликованную в 1943 г. работу У. Маккаллока и У. Питтса [9], в которой вводится понятие формального нейрона как устройства, имеющего два устойчивых состояния и переходящего из одного состояния в другое в зависимости от состояний связанных с ним других нейронов. Важную роль в развитии представлений о механизмах функционирования нейронных сетей сыграла опубликованная в 1949 г. монография Д. Хебба [10], где была высказана гипотеза о том, что в процессе запоминания и обучения изменяются интенсивности меж-

* Журналы «Успехи физических наук» и «Письма в ЖЭТФ», публикуя работы по проблемам нейросистем, упорно вставляют букву «т» в слово «нейрон» [3, 4, 5].

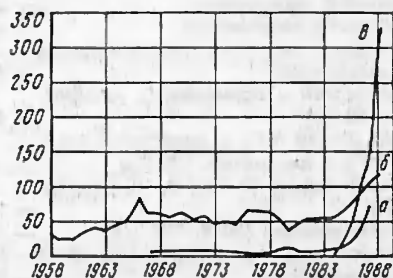
Статья поступила 14.03.88

нейронных связей. Ф. Розенблаттом [11] в конце 50-х годов введено понятие перцептрона — нейронной сети специального вида, способной обучаться распознаванию образов. В конце 60-х годов М. Минским и С. Пейпертом [12] проведен теоретический анализ перцептронов и, в частности, показана ограниченность вычислительных возможностей перцептронов простейшего типа (однослойных). С последним фактом связывают уменьшение интереса к исследованию перцептронов и, как следствие, к исследованиям нейронных сетей вообще (см. дискуссию на эту тему в [13]). Тем не менее, ряд исследователей продолжает заниматься нейронными сетями, и в 70-х — начале 80-х годов закладывается фундамент теории нейронных сетей [14—19]. Значительный рост активности исследований в этой области связывают с появлением в 1982 г. работ Д. Хопфилда [20, 21], в которых нейронная сеть рассматривается не как логическое вычислительное устройство, а как динамическая самоорганизующаяся система.

Мы попытались получить объективную количественную оценку тенденций развития исследований в области нейронных сетей. Для этого был использован библиографический справочник «Science Citation Index», издаваемый Институтом научной информации США. Прежде всего была прослежена динамика встречаемости термина «нейронные сети» (neural networks) в заголовках статей из примерно 4000 журналов, сканируемых этим справочником, за последние 20 лет. С 1986 г. начинается быстрый рост числа таких статей: 31 и 74 в 1986 и 1987 гг. против 11 и 17 в 1984 и 1985 гг. (кривая *a* на рисунке). Можно оценить этот рост как экспоненциальный с периодом удвоения примерно в один год. Была также прослежена динамика цитирования некоторых наиболее известных работ, связанных с исследованием нейронных сетей. Кривая *b* отражает динамику цитируемости монографии Д. Хебба, начиная с 1957 г. Можно отметить три периода повышения интереса к этой работе: 1966 г. (86 ссылок), 1976 г. (70 ссылок), и 1985—1988 гг. (71, 86, 109 и 120 ссылок соответственно), т. е. очевиден рост интереса в последние годы к этой, в общем-то, довольно старой и малодоступной работе.

Наиболее впечатляющей выглядит динамика цитируемости работ Д. Хопфилда, главным образом упоминавшейся выше статьи 1982 г. (кривая *в*). Число ссылок в 1984—1988 гг. составляет соответственно 7, 38, 87, 148 и 334, что соответствует экспоненциальному росту с периодом удвоения менее одного года (можно даже сделать вывод о росте более быстром, чем экспоненциальный).

Была также проанализирована тематика примерно 700 работ по нейронным сетям. Эти работы, как правило, имеют либо биологическую, либо вычислительную направленность. В первом случае нейронная сеть рассматривается как модель для описания функционирования центральной нервной системы или другой биологической системы, во втором — как основа архитектуры вычислительного устройства. Это, конечно, не исключает того, что работы, ориентированные на решение биологических задач, могут быть очень полезными при решении вычислительных задач, и наоборот.



Динамика некоторых характеристик активности исследований по нейронным сетям

Работы вычислительной направленности, составляющие более двух третей от общего числа, в свою очередь, можно разделить на теоретические, в которых аналитически или путем машинного моделирования исследуются вычислительные свойства сетей различного типа, и прикладные, описывающие способы практического создания процессоров с нейронно-сетевой архитектурой. Наблюдающийся в последние три — четыре года резкий рост числа публикаций по нейронным сетям происходит прежде всего за счет увеличения числа теоретических работ. Лидируют здесь Калифорнийский технологический институт и Бэлл лаборатория (США). Работы, посвященные технической (в том числе микроэлектронной) реализации нейронно-сетевых устройств переработки информации, составляют примерно 10 % всех работ вычислительного направления. В остальных публикациях предлагаются варианты оптической реализации нейропроцессора. Основные авторы — сотрудники Калтеха и Пенсильванского университета (США).

Оптические процессоры потенциально могут обладать более высокими рабочими характеристиками по сравнению с микроэлектронными, однако пока они имеются лишь в виде лабораторных образцов. Что же касается нейропроцессоров на микроэлектронной базе, то уже выпущены коммерческие образцы таких процессоров, содержащие порядка 10^8 нейронов и 10^8 программируемых связей [22].

Общий вывод состоит в том, что в ближайшие годы можно с большой уверенностью ожидать продолжения начавшегося в 1985—86 гг. информационного взрыва в исследованиях по нейронным сетям и нейропроцессорам. Дополнительным стимулом к исследованиям в этой области станет доступность нейропроцессоров, что расширит возможности их теоретического анализа и практических разработок (особенно в области изображений и речи)

ЛИТЕРАТУРА

- 1 IEEE First International Conference on Neural Networks.— San Diego, California, June 21—24, 1987 (Vol. 1—234 p.; Vol. 2, 820 p.; Vol. 3—790 p.; Vol. 4 846 p.).
- 2 IEEE International Conference on Neural Networks.— San Diego, California, July 24—27, 1988 (Vol. 1—700 p.; Vol. 2—660 p.).
- 3 Веденов А. А., Левченко Е. Б. Об одном классе нелинейных систем с памятью // Письма в ЖЭТФ.— 1985.— Т. 41.— Вып. 8.— С. 328—331.
- 4 Дунин-Барковский В. Л. Многонейронные структуры: теория и эксперимент // УФН.— 1986.— Т. 150.— Вып. 2.— С. 321—323.
- 5 Фейгельман М. В., Иоффе Л. Б. Спиновые стекла и модели памяти // — УФН.— 1986.— Т. 150.— Вып. 2.— С. 323—325.
- 6 Johnson R. C. DARPA's heural net programm: \$ 390,000,000 over eight jears.— PC AI.— 1988.— Vol. 2.— p. 49.
- 7 Castela z P. F. Neural networks in defense applications.— IEEE Int. Conf. on Neural Networks. — 1988.— Vol. 2.— P. 473—480.
- 8 Stubbs D. F. Editorial.— Neurocomputers.— 1987. Vol. 1.— P. 1.
- 9 McCulloch W., Pitts W. A. A logical calculus of the ideas imminent in nervous activity.— Bull. Math. Biophysics.— Vol. 5.— P. 115—137
- 10 Hebb D. O. The organization of behavior N Y., 1949
- 11 Rosenblatt F. The perceptron: Aprobabilistic model for information storage and organization in the brain.— Psychol. Rev.— 1958.— Vol. 65.— P 386—408.
- 12 Minsky M., Papert S. Perceptrons.— Cambridge, Massachusetts, 1969.
- 13 Daedalus // Proc. of the Amer Acad. of Arts and Sci.— 1988.— Vol 117 N 1 (Special ISSUE on Artificial Intelligence).
- 14 Anderson J. A. A simple neural net work generating interactive memori.— Math. Bioscience.— 1972. Vol 14.— P 197—220.

15. Grossberg S. Studies of mind and brain: Neural principles of learning, perception, development, cognition and motor control.— Boston: Reidel, 1982.
16. Kohonen T. Self-organization and associative memory.— Berlin: Springer, 1984.
17. Dunin-Barkovski V. L., Larionova V. B. An information learning and its recall in the Marr's memori unit.— Biol. Cybernetics.— 1985.— Vol. 51.— P. 407—415.
18. Neural networks for computing.— AIP Conf. Proceedings.— 1986.— Vol. 151.
19. Parallel distributed processing. Vol. 1—2.— Cambridge: MIT Press, 1987.
20. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities. // Proc. of Nat. Acad. of Sci.— 1982.— Vol. 79.— P. 2554—2558.
21. Тэнк Д. У., Хопфильд Д. Д. Коллективные вычисления в нейронподобных электронных схемах // В мире науки.— 1988.— № 2.— С. 44—53.
22. Маныкин Э. А., Сурина И. И. Нейронные сети и их оптические реализации М.: ЦНИИАтоминформ, 1988.

ПИСЬМО В РЕДАКЦИЮ

Навязчивый сервис

О существовании службы сервиса ЭВМ, в частности центров вычислительных услуг (ЦВУ) Московского ПО ВТИ, покупатель ПЭВМ узнает из вкладыша в формуляр, запрещающего первое самостоятельное включение машины без письменного разрешения этой службы под угрозой потери гарантий. Наш «ненавязчивый» сервис — дело знакомое, поэтому требование направить гарантийное письмо о дополнительной оплате ввода ПЭВМ в эксплуатацию не вызывает удивления, как, впрочем, и неопределенные сроки начала работ по плану ЦВУ — до конца текущего квартала.

Практически представители центра появляются в самом конце квартала и, если ПЭВМ сразу не заработала, долго исчезают (поскольку план все равно не выполнен), захватив с собой неисправный блок для передачи в ремонт и обещая вернуть его дней через десять. Далее выясняется, что десять дней — это мантра при неизвестной характеристике, и начинаются длительные и нудные переговоры с обычным перепасовыванием и отговорками об отсутствии стенов, запасных частей и т. д.

Замороченному «услугами» ЦВУ пользователю и в голову не приходит обратиться непосредственно на предприятие-изготовитель ПЭВМ, не имея на руках «технически обоснованного» акта рекламации. Составить такой актспециалист не может, а служба сервиса не хочет, поскольку это не входит в перечень предоставляемых услуг.

Энергичные переговоры с ЦВУ в лучшем случае могут увенчаться получением технического акта, констатирующего неисправность ПЭВМ. И все.

Но есть альтернативный вариант: руководствуемся «Положением о поставках продукции производственно-технического назначения», утвержденным Постановлением СМ СССР от 25 июля 1988, № 888 и закрепляющим некоторые права пользователя ПЭВМ. Тогда оказывается, что для обращения на предприятие-изготовитель не обязательно обладать актом рекламации и что в недельный срок может прибыть наладчик и выполнить все необходимые работы. Наладчик действительно прибывает, оказывается знающим свое дело специалистом (очевидно, потому, что «приближен» к производству), и машина наконец работает.

Но вот вопрос: как в дальнейшем обойтись без ПО ВТИ и многочисленных его ЦВУ и решать проблемы сервиса непосредственно с изготовителем ПЭВМ?

М. Гордон, Москва

УНИВЕРСАЛЬНЫЙ ПРОГРАММИРУЕМЫЙ МИКРОКОНТРОЛЛЕР

Предлагаемый микроконтроллер представляет собой универсальное измерительно-управляющее устройство, способное кроме непосредственно измерения различных электрических и неэлектрических величин (при применении соответствующих датчиков) и отображения результата измерений на встроенном цифро-аналоговом табло управлять различными устройствами по заданной программе, а также передавать результаты измерений в центральную ЭВМ для статистической или иной обработки и документирования результатов измерений.

Область применения предлагаемого устройства — системы допускового контроля и настройки различных узлов и блоков радиоэлектронной аппаратуры, локальные системы автоматического регулирования на предприятиях радиопромышленности, химических предприятиях и т. п.

Основными достоинствами предлагаемого устройства по сравнению с имеющимися являются:

возможность измерения импульсных параметров входного сигнала в произвольные, заранее заданные моменты времени с дискретностью отсчета времени 0,5 мкс;

возможность измерения временных характеристик входных сигналов (частоты, длительности импульсов, временных задержек);

встроенная схема индикации результатов измерения как на семисегментных индикаторах, так и на шкальном светодиодном индикаторе с выделением зоны допустимых значений;

наличие программных средств, позволяющих пользователю-непрограммисту (например, технологу) изменять как состав измеряемых параметров, так и допуски на измеряемые параметры в процессе контроля;

малая стоимость контроллера.

Основные технические данные:

Количество входов для измерения	
постоянного напряжения	— 8;
переменного напряжения	— 4;
амплитудных значений (максимальных и минимальных)	— 4
длительностей и периодов (с дискретностью 50 нс)	— 4
частоты (0—40 МГц с дискретом 1кГц, 0—10 МГц с дискретом 100 Гц)	— 1;
Аналоговый выход (0—30 В, 16-разрядный ЦАП)	— 1;
Дискретные выходы (30 В, 300 мА)	— 16;
Интерфейс для связи с центральной ЭВМ	— RS-232;
Точность обработки	— 8 разрядов;

Процессор — КР1816ВЕ35.

Адрес: 109017, Москва, Пятницкая, 36.

Телефон 231-04-74. Телекс 412104 SPERO SU.

УДК 621.3.049.771.14

А. И. Белоус, В. К. Ковалевский, В. А. Кособрюхов, А. А. Пархомчук, Б. Н. Чернуха УНИВЕРСАЛЬНЫЙ ВРЕМЕННОЙ ПРЕОБРАЗОВАТЕЛЬ НА ОСНОВЕ БИС К512ПС11

КМОП БИС К512ПС11 преобразователя частота-код (ПЧК) предназначена для преобразования аналоговой информации частотных датчиков в 16-разрядный двоичный или двоично-десятичный код и последующей цифровой обработки. Однако функциональные возможности БИС позволяют эффективно использовать ее в качестве универсального временного преобразователя.

Условное графическое изображение микросхемы дано на рис. 1, структурная схема — на рис. 2. Блок управления контролирует работу всех остальных блоков БИС. Шинный интерфейс служит для задания режимов и выдачи информации, что обеспечивается нали-

чием входов-выходов с тремя состояниями. Вычитающий счетчик периодов совместно с управляющим блоком определяет длительность временного интервала измерения $T_{изм}$ в зависимости от числа заданных периодов. Суммирующий счетчик импульсов подсчитывает импульсы в течение $T_{изм}$. Двоичный или двоично-десятичный код, образованный в результате счета, является дискретным отображением измеряемой частоты. Входы счетчиков периодов и импульсов коммутируются со входами FR1, FR2, БИС ПЧК с помощью блока выбора счетчика. БИС ПЧК в основном режиме работает либо как преобразователь частоты, либо как частотомер. В первом случае на счетчик периодов подается измеряемая частота, на счетчик импульсов — эталонная. Во втором случае — наоборот. В Рг. 1, 2 хранятся константы начальной установки счетчиков периодов и импульсов соответственно; в Рг. 3 — данные, сформированные счетчиком импульсов в результате преобразования частоты.

Блок определения дробной части повышает точность преобразования частоты при незначительном увеличении времени преобразования. Для этого определяются логические значения частоты $F_{в1}$, подаваемой на счетчик импульсов в начале ($F_{в1}$) и в конце ($F_{в2}$) периода $T_{изм}$. Например, для числа K_0 уточненное значение кода вычисляется по формуле $K = K_0 + \Delta K$, где ΔK — поправка, величина которой зависит от $F_{в1}$, $F_{в2}$:

$F_{в1}$	0	0	1	1
$F_{в2}$	0	1	0	1
ΔK	0	-0,5	-0,5	-1,0

Значение ΔK выражается в единицах младшего разряда кода K_0 . Константы $F_{в1}$, $K_{в2}$ хранятся в Рг. 4

Основные технические характеристики ПЧК

Тактовая частота, МГц, не менее
 счетчика периодов
 счетчика импульсов
 Напряжение источника питания, В $5 \pm 10\%$
 Выходное напряжение, В
 низкого уровня ($I_{OH} = 1,6$ мА) $0,4 \pm 0,1$
 высокого уровня ($I_{OH} = -0,4$ мА) $U_{CC} - 0,1$
 Входное напряжение, В
 низкого уровня $0,8$
 высокого уровня $U_{CC} - 0,1$

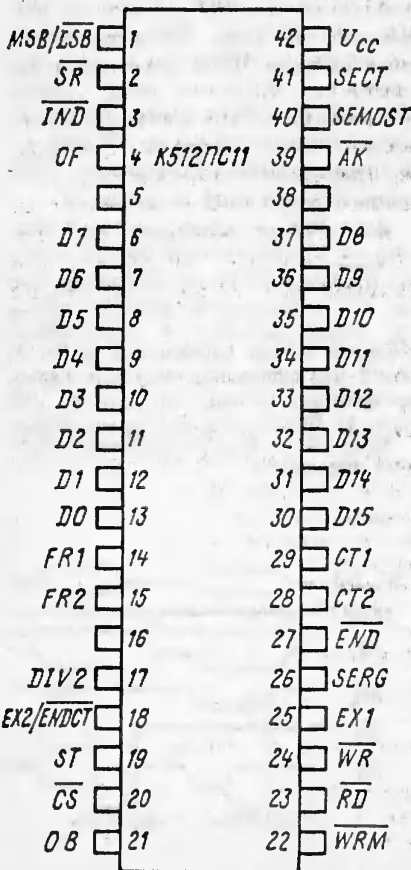


Рис. 1. Условное графическое изображение БИС К512ПС11

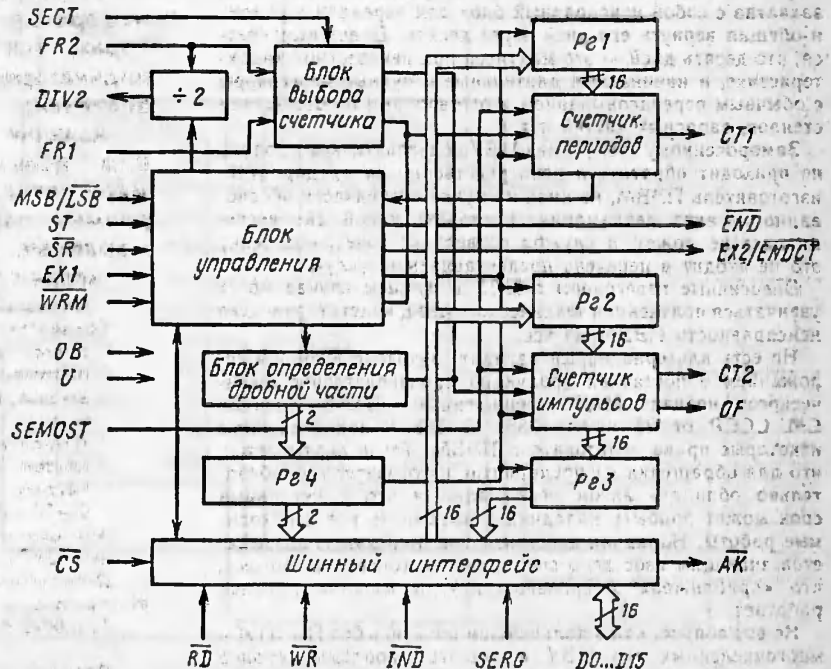


Рис. 2. Структурная схема БИС К512ПС11

Ток утечки, мА
на входах
на входах-выходах в третьем со-
стоянии 15
30
Потребляемый ток
статический, мА 50
динамический, мА 10

НАЗНАЧЕНИЕ ВЫВОДОВ БИС ПЧК (см. рис. 1)

Вход MSB/LSB — старший-младший. Несколько БИС ПЧК могут работать в сборке. Тогда та из них, на которую подаются частоты F1, F2, называется младшей, все остальные — старшими. На вход MSB/LSB младшей БИС подается нуль, на входы старших — единица. При подаче единицы схема выделения периода измерения $T_{изм}$, входящая в схему управления, обеспечивает принудительное прохождение импульсов со входов FR1, FR2 на тактовые входы счетчика импульсов и счетчика периодов.

Вход SR — общий сброс; активный низкий. При подаче нуля обнуляются все триггеры, счетчики и регистры ПЧК.

Вход IND — инверсия данных; активный низкий. При подаче нуля микропроцессор и ПЧК обмениваются данными в инверсном коде.

Выход OF — переполнение; активный высокий.

Появление единицы указывает на однократное или многократное переполнение счетчика импульсов. Сигнал переполнения сбрасывается: по сигналу SR; при записи данных в регистры 1, 2 ПЧК; по положительному

фронту сигнала на входе RD при наличии нуля на входе WRM.

Входы-выходы D0...D15 — ввод-вывод данных. Эти выводы подключаются к 16-разрядной шине данных. Представляют собой входы-выходы на три состояния.

Входы FR1, FR2 — входы частот F1, F2.

Выход DIV2 — выход частоты FR2, деленной на два.

Выход EX2/ENDCT — выход расширения-конец счета, управляемый входом MSB/LSB; EX2 — выход старших БИС сборки ПЧК, ENDCT — младших.

Выходы EX2 и EX1 служат для расширения счетчика периодов (старший выход EX2 подключается к младшему входу EX1) и позволяют добиться независимости предельной частоты, подаваемой на вход счетчика периодов, от разрядности ПЧК.

Единица на выходе EX2 устанавливается при обнулении счетчика периодов и единице на входе EX1. Сигнал ENDCT (активный низкий) фиксирует факт завершения первого преобразования частоты в код после подачи на ПЧК сигнала ST. Сбрасывается ENDCT аналогично сигналу OF.

Вход ST — пуск; активный высокий.

По положительному фронту сигнала на этом входе данные переписываются из регистров начальной установки в счетчики. При обработке сигналов WRM и WR — блокируется. Если на вход ST постоянно подается единица,

то после обработки сигналов WRM или WR ПЧК автоматически перезапускается.

Вход CS — выбор кристалла; активный низкий. Подача нуля разрешает запись-чтение данных.

Вход OB — общий вывод.

Вход WRM — запись в память; активный низкий. По отрицательному фронту сигнала WRM информация из счетчика импульсов переписывается в Рг. 3, а константы F_{в1}, F_{в2} заносятся в Рг. 4. Сигналы WRM и RD блокируют друг друга, поэтому, если в момент прихода отрицательного фронта сигнала WRM на вход RD подается нуль, начало обработки сигнала WRM задерживается до окончания сигнала RD.

Конечная длительность импульса перезаписи ограничивает число микросхем, работающих в сборке ПЧК в непрерывном режиме, до двух (рис. 3). Число N микросхем в сборке ПЧК можно неограниченно увеличивать, если использовать ПЧК в режиме однократного преобразования частоты. При этом сигнал WRM должен подаваться через время $\tau \geq 0,5$ мкс после появления сигнала на выходе END или ENDCT.

Вход RD — чтение; активный низкий. По сигналу RD при наличии разрешения на входе CS и отсутствии нуля на входе WRM информация выдается из ПЧК на шину данных D0...D15. Этот факт должен подтверждаться сигналом на выходе АК (рис. 4). Блокировка выхода RD сигналом WRM снимается при нуле на входе ST.

Вход WR — запись; активный низкий. По отрицательному фронту сигнала WR при наличии разрешения на входе CS открываются входные ключи на входах D0...D15 и генерируется импульс записи информации в Рг. 1, Рг. 2. По окончании импульса записи появляется сигнал на выходе АК (рис. 4). Время задержки $\tau_{WR/RD} \leq 5$ мкс.

Вход EX1 — вход расширения; активный высокий.

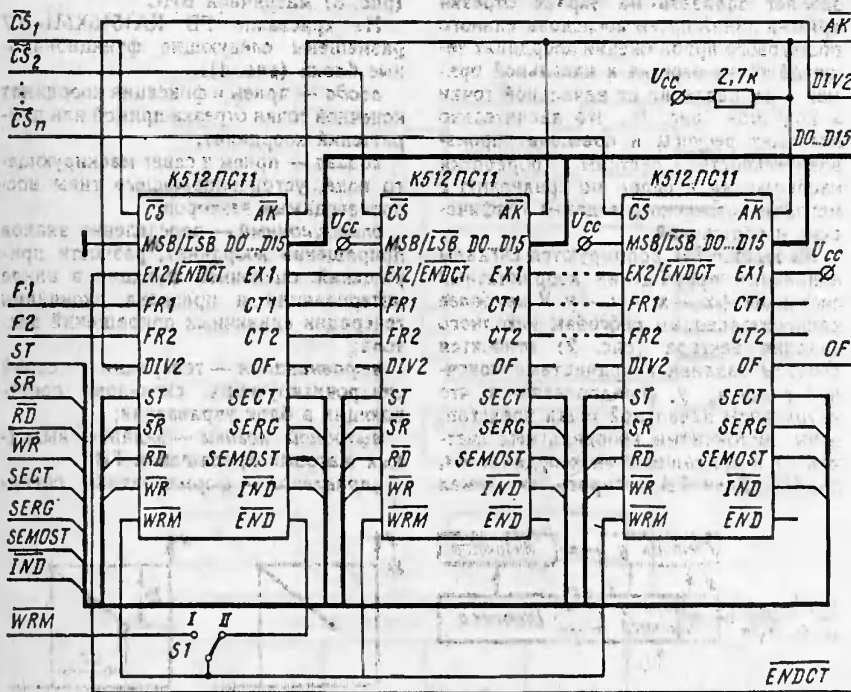


Рис. 3. Схема 16×N-разрядного преобразователя для режимов однократного (I) и непрерывного (II) преобразований частоты

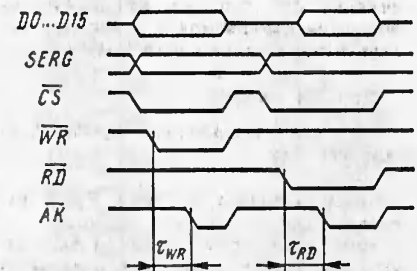


Рис. 4. Временные диаграммы сигналов записи-чтения

Вход SERG — выбор регистра. Запись $SERG=1$ — Рг. 1, $SERG=0$ — Рг. 2.

Чтение $SERG=1$ — Рг. 3, $SERG=0$ — на D7 подается сигнал $ENDCT$. На D14, D15 из Рг. 4 подаются значения F_{B1} , F_{B2} , на D0...D6, D8...D13 — единицы.

Выход END — конец; активный низкий, импульсный. Отрицательный фронт импульса END формируется по окончании периода $T_{изм}$; положительный — по окончании импульса перезапуска. Нуль на выходе END появляется при общем сбросе.

Выход СТ2 — инверсный выход старшего счетчика импульсов. В сборке ПЧК подключается к следующему входу $FR2$.

Выход СТ1 — выход старшего разряда счетчика периодов. В сборке ПЧК подключается к следующему старшему входу $FR1$.

Выход АК — сигнал исполнения пассивного; активный низкий.

Подтверждает факт записи-чтения данных при обмене данными между ПЧК и процессором (рис. 4).

Вход SEMOST — выбор системы счисления. Счетчики импульсов и периодов работают в двоичной ($SEMOST=1$) или двоично-десятичной ($SEMOST=0$) системе счисления.

Вход SECT — выбор счетчика. При $SECT=0$ вход $FR1$ коммутируется с тактовым входом счетчика периодов, вход $FR2$ — с тактовым входом счетчика импульсов; при $SECT=1$ $FR1$ — с тактовым входом счетчика импульсов, $FR2$ — с тактовым входом счетчика периодов. Оба счетчика переключаются по положительному фронту тактовых частот, подаваемых на входы $FR1$, $FR2$ ПЧК.

Выход U_{сс} — вывод питания от источника напряжения.

РЕЖИМЫ РАБОТЫ ПЧК

Преобразование «частота-код»

В этом режиме работы различают два подрежима: однократного и непрерывного преобразования частоты в код. В обоих подрежимах микросхема может работать либо как преобразователь, либо как частотомер в зависимости от того, на какой счетчик подаются измеряемая и эталонная частоты (назначение входа $SECT$).

Порядок работы:

предустановка входов 1, 3, 25, 40, 41 (см. рис. 1);

сброс; загрузка регистров Рг. 1, Рг. 2 начальной предустановки счетчика; пуск. На вход ST подается положительный фронт либо положительный импульс длительностью не менее 0,5 мкс;

при появлении сигнала на выходе $ENDCT$ в режиме непрерывного пре-

образования считываются данные; в режиме однократного преобразования перед считыванием необходимо подать отрицательный импульс WRM ;

в режиме непрерывного преобразования после очередного появления сигнала $ENDCT$ можно повторно считать данные; в режиме однократного преобразования необходимо осуществить перезапуск подачей сигнала на вход ST .

Преобразование «код-время»

Этот режим аналогичен предыдущему, за исключением того, что используются только счетчик периодов и выход $ENDCT$ или END .

Преобразование «код-частота»

Режим аналогичен режиму «частота-код» в подрежиме непрерывного преобразования частоты. Используются только счетчик периодов и выход END . Коэффициент деления частоты по выходу END $K=N_0+1$, где $N \geq 1$ —

УДК 681.323

А. М. Петух, Д. Т. Ободник, В. П. Верховой

ГЕНЕРАТОР ВЕКТОРОВ НА ОСНОВЕ МАТРИЧНОЙ БИС

Применение генератора векторов (ГВ) в устройствах отображения графической информации (УОГИ) позволяет задавать на экране отрезки прямых линий путем последовательного пошагового приближения координат текущей точки отрезка к идеальной прямой в направлении от начальной точки к конечной (рис. 1), что значительно экономит ресурсы и повышает производительность системы обработки изображений в целом по сравнению с методом поточечного задания графических изображений.

На выходе ГВ формируются сигналы единичных приращений координатных счетчиков $+x$, $-x$, $+y$, $-y$. К наиболее распространенным способам исходного задания вектора (рис. 2) относятся способы задания координатами конечной точки x_k , y_k (предполагается, что координаты начальной точки представлены состояниями координатных счетчиков) и приращениями координат Δx , Δy [1]. Если ГВ собирать на схемах

число, записанно в Рг. 1. Максимальная частота, при которой не нарушается приведенное соотношение для коэффициента деления, не превышает 300 кГц. При дальнейшем увеличении частоты значение K увеличивается из-за внутренних задержек в БИС.

Преобразование «время-код»

В этом режиме на вход ST постоянно подается нуль. Используются счетчик импульсов и вход MSB/LSB . В течение длительности положительного импульса, подаваемого на вход MSB/LSB , разрешается прохождение частоты на тактовый вход счетчика импульсов. После завершения преобразования необходимо подать на вход WRM сигнал перезаписи из счетчика импульсов в Рг. 3, после чего можно считать данные.

Телефон 77-68-53, Минск

Статья поступила 7.03.89

средней степени интеграции, то понадобится 20...50 микросхем [2]. Но можно его выполнить и на основе одной (рис. 3) матричной БИС.

На кристалле ГВ КА1515ХМ1-107 размещены следующие функциональные блоки (рис. 4):

ввода — прием и фиксация координат конечной точки отрезка прямой или приращений координат;

сдвига — прием и сдвиг маскирующего кода, устанавливающего типы воспроизводимых векторов;

операционный — определение знаков приращений координат, разности приращений, оценочной функции в цикле интерполяции и признака окончания генерации единичных приращений вектора;

выравнивания — генерация серий синхронизирующих сигналов, поступающих в блок управления;

выходной логики — задание выходных шаговых приращений ГВ;

управления — формирование сигнала



Рис. 1. Вариант структуры устройства отображения графической информации с использованием генератора векторов: f_0 — опорная импульсная последовательность

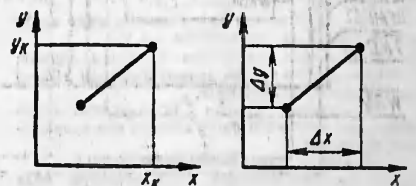


Рис. 2. Основные способы исходного задания вектора

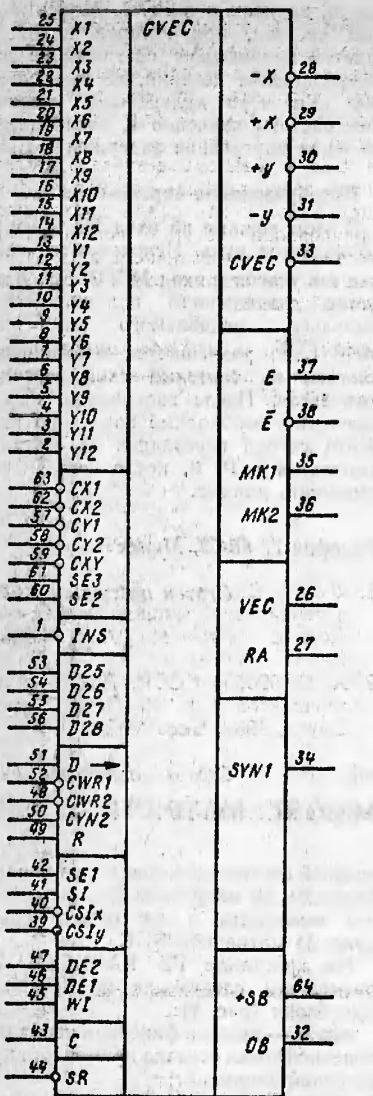


Рис. 3. Условное графическое обозначение генератора векторов KA1515XM1-107

лов управления другими блоками УОГИ.

Назначение выводов ГВ

X1...X12 — входы координаты x или ее приращение;
 Y1...Y12 — входы координаты y или ее приращение;
 CX1, CX2 — стробы младшего и старшего байтов данных по координате x;
 CY1, CY2 — стробы младшего и старшего байтов данных по координате y;
 CX, CY — строб пятого байта данных двух младших разрядов по координатам x и y;
 SE3 — сигнал управления форматом данных;
 SE2 — сигнал приема данных в прямом SE2=0 или обратном SE2=1 коде;
 INS — сигнал приема 12-разрядных кодов координат начальной точки вектора или приращений координат;
 D25...D28 — входы схемы совпадения;
 D — последовательный вход регистра кода маски;
 CWR1, CWR2 — стробы восьми младших X1...X8 и четырех старших X9...X12 разрядов кода маски.

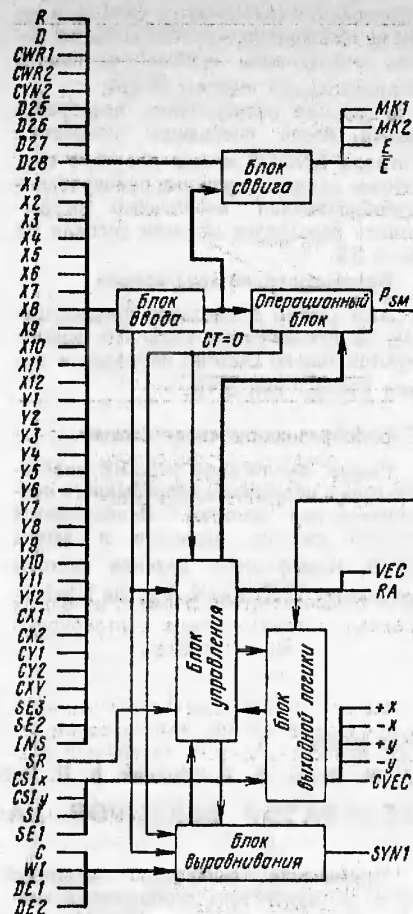


Рис. 4. Структурная схема генератора векторов

CYN2 — тактовый вход регистра кода маски, сдвиг содержимого в сторону старших разрядов на один разряд;
 SR — сигнал установки нулевого состояния регистра кода маски;
 SE1 — сигнал задания вектора;
 S1 — информационный вход знака приращений координат;

CS1x, CS1y — стробы знака приращения по координатам x, y;
 DE1, DE2 — сигналы определения частоты выдачи единичных приращений (DE1=1, DE2=0 — частота C/3; DE1=0, DE2=1 — частота C/4; DE1=0, DE2=0 — частота определяется наклоном вектора);
 W1 — сигнал управления приостановкой генерации векторов;
 C — тактовый вход;
 SR — вход начальной установки;
 +x, -x, +y, -y — выходы единичных приращений координат вектора;
 CVEC — строб выходов приращений вектора;
 E, E-bar — прямой и инверсный выходы схемы совпадения;
 MK1, MK2 — выходы пятого и двенадцатого разрядов регистра кода маски;
 VEC — сигнал окончания выдачи выходов приращений;
 RA — сигнал готовности к приему задания вектора (RA=1 — готов);
 SYN1 — выход синхронизации внешних устройств.

Способ задания вектора определяется сигналом SE1: при SE1=0 — координатами конечной точки, при SE1=1 — приращениями координат.

Двоичные 12-разрядные коды координат и приращений подаются на информационные входы X1...X12, Y1...Y12 блока ввода и сопровождаются стробирующими сигналами CX1, CX2, CY1, CY2, CX, CY (рис. 5). Последним должен поступать сигнал CX1: переход его в Лог. 1 выводит блок управления из нулевого состояния. Формат принимаемых данных определяется сигналом SE3 (рис. 6).

Если отрезок задается приращениями координат, то значение знака приращений подается на вход S1 (0 соответствует знаку +, а 1 — знаку -) и проводится стробирующим сигналом CS1x для Δx и CS1y для Δy. Активный уровень стробирующего сигнала — нулевой. В основу организации ГВ положен метод модифицированной оценочной функции, в соответствии с которым по координате с большим приращением единичный шаг выполняется в каждый тактовый момент времени, а по координате с меньшим приращением — в зависимости от знака вычисляемой в каждом такте оценочной функции. Признаком большего приращения — знак разности Δ=Δx-Δy. Начальное значение оценочной функции принимается равным половине большего приращения. В дальнейшем оценочная функция определяется в операционном блоке ГВ согласно выражениям:

$$O\Phi_i = O\Phi_{i-1} + \Delta \text{ при } O\Phi_{i-1} < 0;$$

$$O\Phi_i = O\Phi_{i-1} - M\Phi \text{ при } O\Phi_{i-1} \geq 0,$$

где OΦ_i — значение оценочной функции на i-м шаге; MΦ — меньшее приращение. Приращения координат могут быть и отрицательными, поэтому выходные единичные шаги по координатам выполняются как с увеличением, так и уменьшением координат. Выходные приращения +x, -x, -y, -y будут зависеть от знаков приращений координат S1Δx, S1Δy и знака оценочной функции S1OΦ (см. таблицу). Единичные значения сигналов +x, -x, +y, -y обозначают наличие прира-

Таблица истинности выходов приращений

S1Δx	S1Δy	S1Δ	S1OΦ	+x	-x	+y	-y
0	0	0	0	1	0	0	0
0	0	0	1	1	0	1	0
0	0	1	0	0	0	1	0
0	0	1	1	1	0	1	0
0	1	0	0	1	0	0	0
0	1	0	1	1	0	0	1
0	1	1	0	0	0	0	1
0	1	1	1	1	0	0	1
1	0	0	0	0	1	0	0
1	0	0	1	0	1	1	0
1	0	1	0	0	0	1	0
1	0	1	1	0	0	1	1
1	1	0	0	0	1	0	0
1	1	0	1	0	1	0	1
1	1	1	0	0	0	0	1
1	1	1	1	0	1	0	1

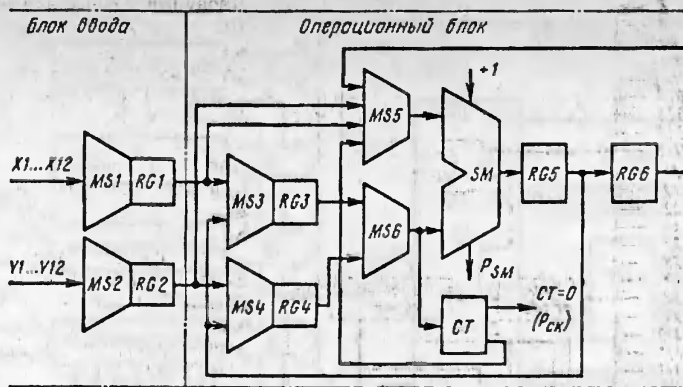


Рис. 5. Функциональная схема блоков ввода и операционного:

MS1...MS6 — мультиплексоры, RG1...RG6 — регистры; SM — комбинационный сумматор; CT — счетчик

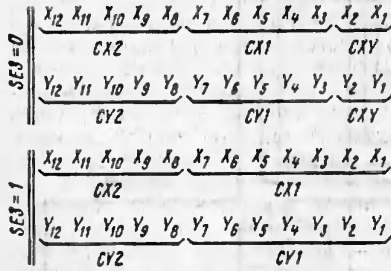


Рис. 6. Соответствие формата исходных данных разрядам кодов и стробирующим сигналам

шения, а нулевые — отсутствие (в БИС ГВ активному значению сигналов $+x$, $-x$, $+y$, $-y$ соответствует отрицательный импульс).

В итоге БИС ГВ формирует единичные приращения координат вектора, заданного приращениями координат; принимает начальную координату и формирует единичные приращения координат вектора, заданного координатами конечной точки; принимает и смещает код маски с выдачей в последовательном коде; выравнивает скорость формирования в зависимости от наклона вектора; управляет частотой выдачи единичных приращений координат вектора; принимает координаты или их приращение в различных форматах, в прямом или обратном коде; приостанавливает генерацию вектора.

нататы или их приращение в различных форматах, в прямом или обратном коде; приостанавливает генерацию вектора.

ЛИТЕРАТУРА

1. Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики / Пер. с англ. — М.: Мир, 1985. — Кн. 1.
2. А. С. 920636 СССР. Линейный интерполятор / А. М. Петух и др. — Оpubл. 1982, Бюл. № 25.

Статья поступила 7.02.89

Основные параметры БИС ГВ КА1815ХМ1-107

	мин.	макс.
Напряжение источника питания, В	4,5	6,5
Входное напряжение, В	0	U _{CC}
низкого уровня		
высокого уровня		
Выходное напряжение, В		0,4
низкого уровня	4,0	
высокого уровня		
Время удержания информации по информационным входам относительно сигналов синхровходов, нс	100	
Период следования синхримпульсов по входу С, нс	200	
Разрядность, дв. разр.		12
Потребляемый ток в статическом режиме, А		1,0
Наработка на отказ, ч	50 000	

УДК 681.325.5:681.385

Е. М. Злотник, В. В. Анищенко, В. В. Бокуть

МИКРОПРОГРАММИРУЕМЫЙ ГРАФИЧЕСКИЙ КОНТРОЛЛЕР

Графический контроллер (ГК) представляет собой микропроцессорную систему повышенного быстродействия на основе МПК БИС серии КР1804 с архитектурой, ориентированной на обслуживание растровых средств машинной графики. Контроллер предназначен для взаимодействия с видеопамятью (ВП) и видеопроцессором растровой графической подсистемы, подключаемой к ПЭВМ ЕС1840/1841 [1]. Обеспечивая микропрограммную реализацию таких процедур, как отсечение изображения, попадающего за пределы области вывода на экране, генерирование векторов, заполнение многоугольников цветом, штриховкой, шаблонами, обработка растровых массивов и т. д., ГК значительно повышает быстродействие графической системы в целом.

В состав ГК входят узлы операционный и управления, блоки сопряжения с графической и системной шинами

(рис. 1) Операционный узел включает БИС процессорного элемента (ПЭ) К1804ВМ1, внешний мультиплексор адреса регистров общего назначения (МХ) и сверхоперативное запоминающее устройство (СОЗУ) с регистром-счетчиком адреса.

Структура и система команд 16-разрядной микропрограммно управляемой БИС К1804ВМ1 ориентированы на построение быстродействующих контроллеров для управления периферийными устройствами в реальном масштабе времени, предварительной обработки передаваемых данных и контроля состояния объекта. Используемые в структуре ПЭ схемотехнические решения подчеркивают его преимущества перед известными ПЭ [2] именно для реализации графических функций.

Среди них более 30 базовых одно- и двухоперандных логических и арифметических инструкций, ориентированных на обработку битовой, байтовой и 16-разрядной информации; инструкций обработки в одном такте трех операндов одновременно, благодаря чему поле разрядов выбирается из двух операндов данных с помощью маскирующего операнда за один такт. Тактовая частота — 5 МГц.

При выполнении байтовых операций обрабатываются восемь младших разрядов. Циклический сдвиг 16-разрядных данных на 1...15 разрядов выполняется за один такт.

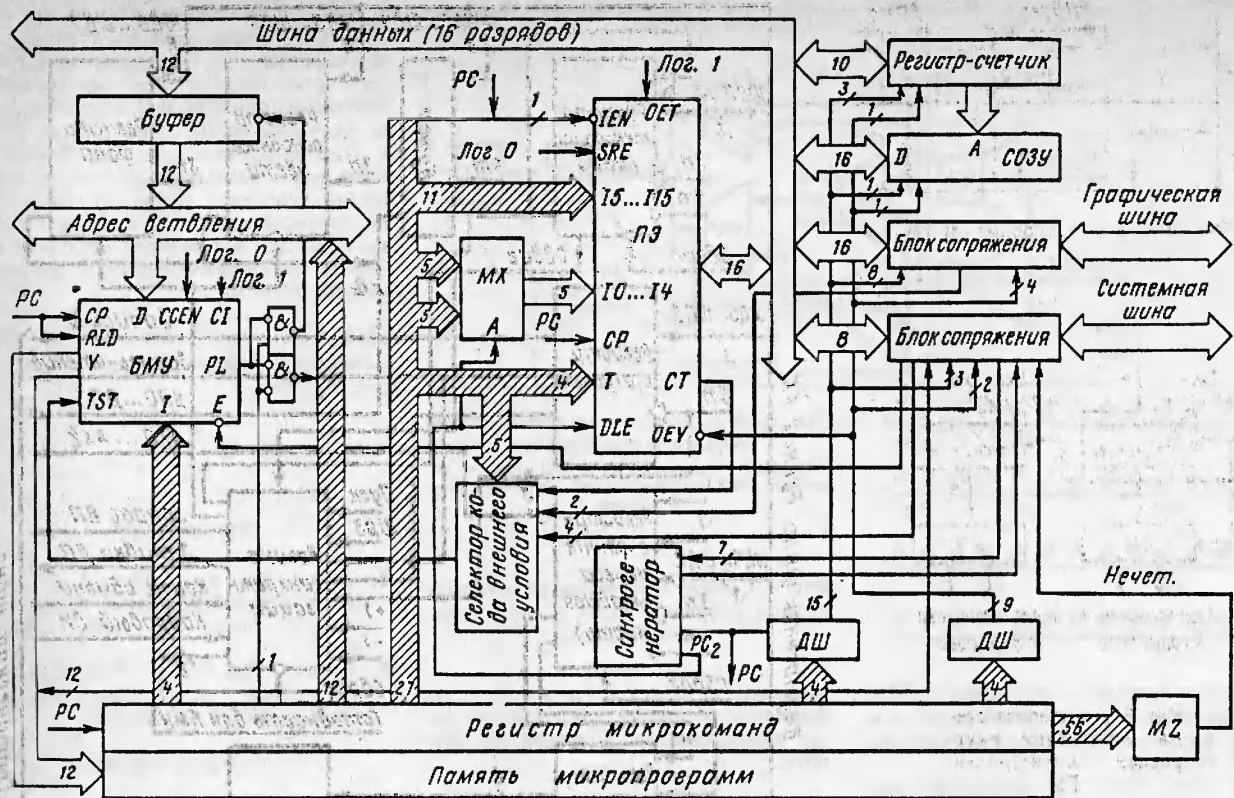


Рис. 1. Структурная схема ГК

В составе ПЭ насчитывается 32 16-разрядных регистра общего назначения (РОН) с одноадресной структурой. Двухадресная система выборки организована с помощью МХ.

Микрокоманда содержит два 5-разрядных поля адреса РОН: одно для задания адреса чтения, второе — записи. При выполнении любой операции в первой половине такта на вход 10...14 ПЭ поступает адрес А одного из операндов, во второй половине же такта — адрес В, по которому записывается результат операции. На управляющий вход МХ подается сигнал РС₂ с выхода генератора. Такая схема требует для выполнения операций, не использующих двухадресную выборку, одинаковых значений полей А и В, что обеспечивается автоматическим копированием содержимого поля А в В.

Для увеличения объема регистровой памяти в контроллере используется СОЗУ емкостью 1К 16-разрядных слов на микросхемах КР541РУ2. Адрес СОЗУ хранится в регистре адреса, представляющем собой реверсивный счетчик, что позволяет СОЗУ работать в трех режимах адресации. В режиме обычного обращения адрес СОЗУ предварительно заносится в регистр адреса, в режиме инкрементирования увеличивается а в режиме декрементирования уменьшается на 1. Все режимы реализованы аппаратно, что позволяет использовать СОЗУ в качестве памяти или буфера при передаче или приеме массивов данных.

Узел управления включает БИС блока микропрограммного управления (БМУ), селектор кода внешнего условия, память микропрограмм, регистр микрокоманд, схему контроля на Нечет памяти микропрограмм (МЗ) и дешифраторы (ДШ) для управления внутренней шиной данных ГК.

Основу узла управления составляют БИС БМУ КР1804ВУ4 с фиксированной разрядностью адреса [3]. В зависимости от значения инструкции на входе ДШ источником адреса следующей микрокоманды может быть

внутренний счетчик микрокоманд или внешний вход адреса ветвления Д.

Источник адреса перехода на входе Д задается полем микрокоманды либо на шине данных через трехстабильный буфер сигналом РЛ и специальным разрядом микрокоманды.

Селектор кода внешнего условия позволяет выбирать для текущей микрокоманды один из восьми источников внешнего условия или условия с выхода ПЭ. Дополнительный бит полярности в микрокоманде позволяет изменить содержимое источника условия (например, условие «равно 0» заменить на условие «не равно 0»).

12-разрядный адрес следующей микрокоманды формируется БИС БМУ или ПЭ посредством блока сопряжения с системной магистралью. Для хранения микрокоманд используется память микропрограмм объемом 2К 56-разрядных слов (семь микросхем КР56РТ18), что достаточно для реализации всех функций ГК.

Синхрогенератор КР1804ГГ1 работает в следующих режимах, задаваемых блоком сопряжения с системной шиной: «Приостановка», «Пошаговый», «Ожидание», «Нормальная работа» — и формирует две серии синхронимпульсов: РС и РС₂ (рис. 2). Тактовым синхронимпульсом РС микрокоманда, выбранная из памяти микропрограмм, записывается в регистр микрокоманд, содержимое которого при этом контролируется с помощью схем контроля четности (нечетности).

Компоненты ГК взаимодействуют через быстродействующую внутреннюю магистраль (рис. 3), состоящую из шин данных и адреса. Шина данных (16 двунаправленных цепей) предназначена для обмена 16-разрядными словами или байтами (по восьми младшим разрядам). Под шину адреса в микрокоманде выделено по четыре разряда на два поля приемника и передатчика информации соответственно.

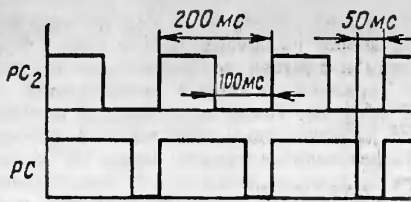


Рис. 2. Серии синхронимпульсов

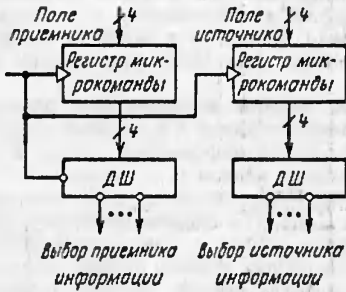
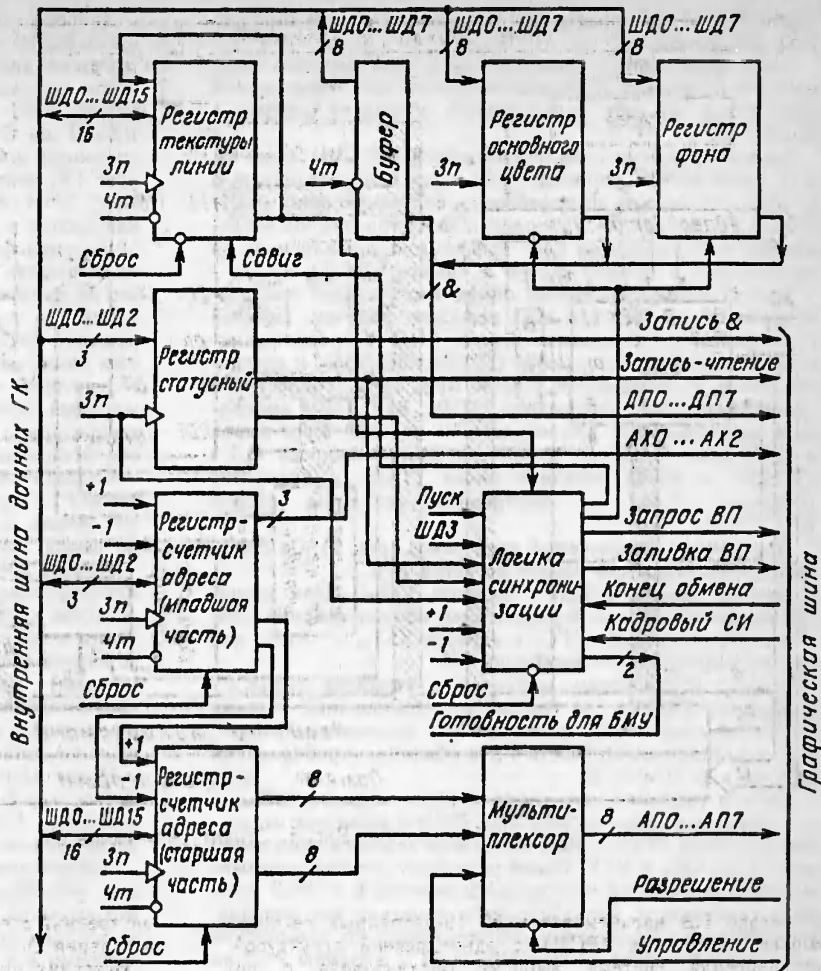


Рис. 3. Логическая схема управления внутренней магистралью ГК

Рис. 4. Функциональная схема блока сопряжения с графической шиной



Блок сопряжения с графической шиной (БСГШ) обеспечивает обмен и управление ВП и видеопроцессором, режимы записи и чтения ВП, формирование адресов при обращении ВП, задание цвета (индекса цвета) основного и фоновое; формирование различных типов линий и т. д. (рис. 4). Структурно ВП состоит из восьми плоскостей форматом 1024×512 , т. е. один элемент отображения (ЭО) кодируется восьмью битами. Функции ВП: запись или чтение отдельных ЭО, одновременная запись восьми ЭО, расположенных в строку; заполнение памяти фоновым изображением (заливка) в течение одного кадра. ВП собрана на БИС ОЗУ КР556РУ5. Каждая плоскость ВП содержит восемь БИС, селективируемых с помощью 3-разрядного адреса.

Для выбора ячейки внутри БИС ОЗУ используется 16-разрядный адрес, мультиплексируемый побайтно. Видеопроектор запрограммирован таким образом, что при выводе изображения на экран считываются сразу восемь ЭО каждой плоскости. Координаты ЭО задаются парой (X, Y), где $X=1...1024$, а $Y=1...512$. Линейная организация ВП требует 19-разрядного адреса, который делится на две части: 16-разрядную старшую и 3-разрядную младшую.

В качестве регистров адреса применяются каскадно связанные счетчики К555ИЕ7, позволяющие минимизировать число тактов ГК для вычисления адреса при линейной интерполяции.

Режимы обращения к ВП устанавливаются 4-разрядным статусным регистром. Запись единицы в старший разряд инициирует заливку фона по всей ВП за один цикл кадровый синхронимпульс. Два младших разряда задают режимы записи-чтения одного ЭО или записи восьми ЭО одновременно. В режиме записи в ВП в зависимости

от значений третьего разряда статусного регистра и младшего разряда регистра типа линии, работающего в режиме циклического сдвига, автоматически формируются требуемые текстура и цвет линии. ЭО принимает значение основного цвета или фона либо не изменяется. В последнем случае обмен не инициируется.

Логика синхронизации (ЛС) управляет выработкой сигналов, обеспечивающих взаимодействие ГК с ВП и видеопроцессором. Цепочка «узел управления (УУ) — ЛС — ВП» синхронизируется четырьмя парами сигналов (рис. 5). Первая пара сигналов, инициирующих обычный обмен или заливку, вырабатывается в узле управления для ЛС. Вторую пару на основе первой вырабатывает ЛС для ВП — это запрос на доступ к одному (восьми) ЭО либо запрос на заливку фоном. По окончании обмена ВП вырабатывает сигнал «конец обмена». Для заливки с этой целью используется кадровый синхронимпульс. И, наконец, ЛС формирует два условия для селектора кода внешних



Рис. 5. Синхронизация компонентов ГК

условий БМУ: готовность БСШ к новому обмену и кадровый синхронизульс.

Возможности этого аппаратно-микропрограммного «конвейера» повышают производительность всей графической подсистемы за счет параллельной обработки данных в узлах.

Блок сопряжения с системной шиной (БССШ) (рис. 6). ЦП КР1810ВМ86 управляет ГК через централизованную систему управления интерфейсом — системную шину (СШ) ПЭВМ ЕС1840/41 [4]. ГК поддерживает три способа ввода-вывода информации на СШ: программно управляемый, по прерыванию и прямой доступ к памяти (ПДП).

Программно управляемый обмен информацией с прерываниями используется в двух режимах: рабочем, который поддерживается посредством первой БИС ППИ программируемого параллельного интерфейса (ППИ) К580ВВ55, и диагностическим — посредством второй БИС ППИ совместно с триггером запроса прерывания (ТПР). БИС ППИ выбирается дешифратором при обращении по СШ. Ввод-вывод с ПДП обеспечивается двумя триггерами: запроса ПДП (ТПДП) и конца передачи блока (ТКБ) и буферным регистром (один из внутренних регистров первой БИС ППИ).

Программно управляемый ввод-вывод (рис. 7). Со стороны СШ доступны все четыре регистра БИС:

регистр данных (РГД) с адресом ХХ0₁₆ — для двустороннего байтного (8-разрядного) обмена информацией; регистр статусный (РГС) с адресом ХХ1₁₆ — только для чтения информации о состоянии ГК;

регистр управляющих сигналов и разрядов (РГУСР) с адресом ХХ2₁₆ — для приема и формирования (чтения и записи) сигналов, сопровождающих асинхронный обмен, т. е. для анализа готовности ГК к обмену и стробирования данных в РГД и РГС;

регистр управляющего слова (РГУ) с адресом ХХ3₁₆ — для задания режима работы БИС ППИ и установки отдельных разрядов РГУСР. Рабочий режим задается занесением константы С6₁₆ в РГУ. После установки режима содержимое всех регистров и триггеров обнуляется. В РГУСР можно устанавливать или сбрасывать отдельно три разряда, разрешающие прерывания соответственно по завершении записи с внутренней шины в РГС и РГД и чтению РГД на внутреннюю шину ГК.

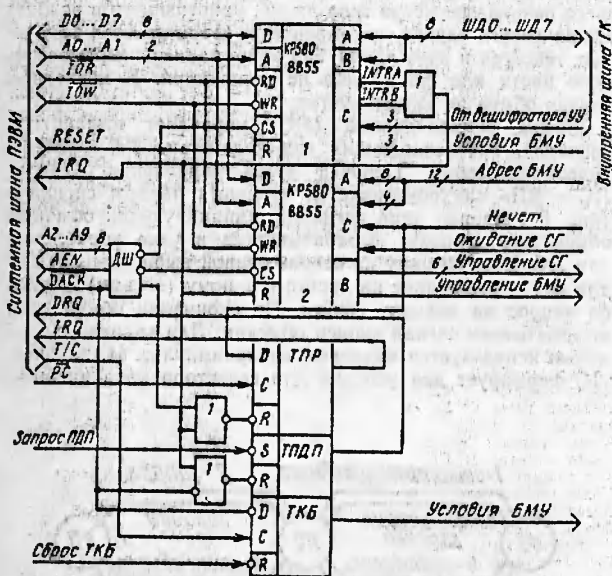


Рис. 6. Функциональная схема блока сопряжения с системной шиной

Запросы на прерывания INTR_A и INTR_B по внешней схеме ИЛИ выдают запрос на прерывание на СШ — IRQ.

Режим диагностики и запуска осуществляется посредством программно управляемых портов ввода-вывода на второй БИС ППИ (рис. 8). После прохождения сигнала RESET по СШ все регистры устанавливаются в нулевое состояние, а каналы переводятся в режим ввода. ЦП тестирует ГК, используя следующие возможности подключения БИС ППИ. Канал А и старшая половина канала С подключаются к выходу БИС БМУ, на котором формируется 12-разрядный адрес следующей микрокоманды, что позволяет считать адрес текущей микрокоманды либо задавать его принудительно из ЦП. Адресный выход БИС БМУ переводится в высокоимпедансное состояние сигналом «разрешение БМУ» на старшей линии канала В. Шесть младших линий канала В используются для управления синхронизатором (одна линия не задействована). С помощью младшей половины канала С считается состояние синхронизатора и сигнал контроля на Нечет. (две линии не задействованы).

Следовательно, порты должны настраиваться посредством РГУ следующим образом: канал А и старшая половина канала С — на ввод-вывод информации, канал В — на вывод, младшая половина канала С — на ввод.

Прямой доступ к памяти. Перед передачей в ГК команды, инициирующей обмен с ПДП, ЦП должен установить все средства соответствующего канала в контроллере ПДП. К ним относятся регистры: адреса страниц памяти; текущего адреса (начального адреса); текущего счетчика (размер массива); режима (запись или считывание из памяти, увеличение или уменьшение текущего адреса, одиночный или блочный режим передачи); маски запросов ПДП (разрешение работы по соответствующему каналу).

После загрузки этих регистров управление ПДП передается ГК. В свою очередь, в ГК выполняется несложный алгоритм. Например, для чтения (записи) информации из памяти он состоит из пяти шагов (рис. 9).

Заключение. В ГК используется горизонтальный принцип микропрограммирования с конвейерным регистром микрокоманд. Микрокоманда разделяется на поля, каждое из которых управляет отдельным ресурсом (ПЭ, БМУ, регистрами и т. д.) [6]. Плата ГК размерами 200×240 мм (конструктив ПЭВМ ЕС1840/1841) работает в составе растровой графической подсистемы, размещенной в блоке расширения ЕС1841 [4]. Для настройки на архитектуру, генерации языка микропрограммирования и отладки ГК применялась САПР микропроцессорных устройств МЕТАМИКРО [7].

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

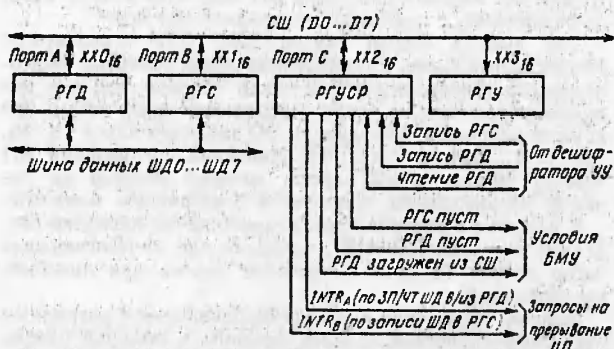


Рис. 7. Включение БИС ППИ в режиме программно управляемого обмена

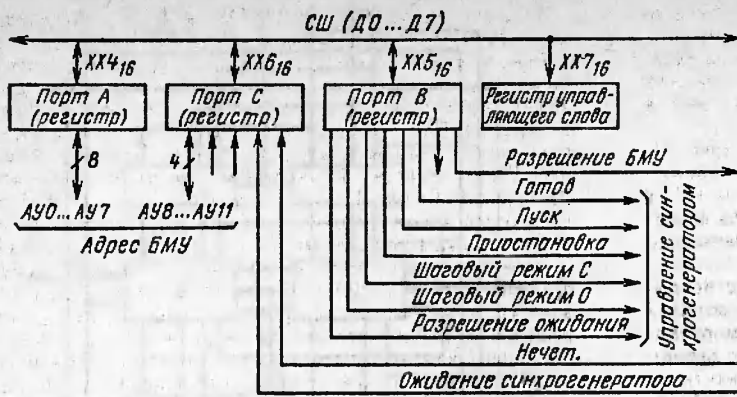


Рис. 8. Включение БИС ППИ в режиме диагностики и запуска

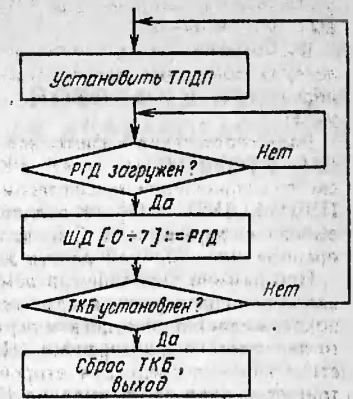


Рис. 9. Алгоритм работы ГК при чтении байта из ОЗУ в режиме ПДП

ЛИТЕРАТУРА

1. Злотник Е. М., Киркоров С. И., Стежко И. К. Графический адаптер для ПЭВМ ЕС1840 // Микропроцессорные средства и системы.— 1988.— № 4.— С. 25—27.
2. Злотник Е. М., Анищенко В. В., Левин Е. М., Пашковская В. Ф. Графический контроллер. Архитектура и система диагностики.— Препринт. Ин-т техн. кибернетики АН БССР.— Минск.— 1988.— № 28.
3. Злотник Е. М. Секционированные микропроцессоры.— Минск: Наука и техника, 1984.— С. 68—69.
4. Машина на вычислительная электронная цифровая персональная профессиональная ЕС1841: Техническое описание Е11.700.012 ТО, 1986.— 264 с.

5. Алексенко А. Г., Галицин А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М: Радио и связь, 1984.— С. 21—30.
6. Злотник Е. Н., Анищенко В. В., Левин Е. М., Пашковская В. Ф. Графический контроллер. Микропрограммирование.— Препринт Ин-т техн. кибернетики АН БССР.— Минск.— 1988.— № 37.— 30 с.
7. Семенов О. И., Бокуть В. В., Гриншпан Л. А., Малюш Я. Т., Ткачев Е. Е., Шерлинг Д. Р. Система автоматизированного проектирования мультипроцессорных устройств на секционированных микропроцессорах // Микропроцессорные средства и системы.— 1988.— № 4.— С. 66—69.

Статья поступила 14.02.89

УДК 681.325

ПРОФЕССИОНАЛЬНЫЕ ОДНОПЛАТНЫЕ МИКРОЭВМ

МикроЭВМ и мМС1300 — набор одноплатных компьютеров (см. табл.), выполненных в стандарте Евромеханики на плате Е2 с системной шиной, совместимой с И41. Внутриплатные ресурсы микроЭВМ могут быть расширены внешними средствами. Встроенные 16-цветные видеоконтроллеры имеют выход на стандартный IRGB-интерфейс.

В семействе два класса микроЭВМ: 8-разрядные с дисковой ОС ДОС1810 (ISIS-II) и СР/М на базе МП КР580ВМ80А или КМ1821ВМ85 и 16-разрядные с ОС MS DOS на МП К1810ВМ88, совместимые с IBM PC XT. Системное и инструментальное ПО обеспечивает разработку и программирование прикладных микросистем на основе ОЭВМ К1816ВЕ48, К1816ВЕ51 и МП КР580ВМ80А, КМ1810ВМ85, К1810ВМ88, К1810ВМ86, i80188, i80186, i80286, i80386 и их модификаций. ПО включает программные средства обработки текстов.

Разработаны импульсные источники питания* и крейты с различными электрическими и конструктивными характеристиками, обеспечивающие построение микровычислительных комплексов со встроенными приводами ГМД и ЖМД.

Система мМС1361ХТ — старшая модель семейства одноплатных микроЭВМ, совместимая с персональными компьютерами типа IBM PC XT. Однако в отличие от них выпол-

нена в конструктиве Европлаты Е2 с размерами 233××220 мм и содержит на плате: ЦП, основную память, контроллеры клавиатуры, ГМД, ЖМД и цветной графики (CGA). Программа BIOS полностью совместима с ОС MS DOS, которая загружается с ГМД или ЖМД. Системный интерфейс микроЭВМ удовлетворяет стандарту И41 с соединителем типа СНП59-96 без арбитра шины.

На базе микроЭВМ можно построить: персональные компьютеры с ОС MS DOS профессионального назначения; инструментальные микросистемы и АРМ разработчика микропроцессорных устройств; деловые компьютеры для текстовой и графической обработки и баз данных; учебные, бытовые и игровые системы.

Технические характеристики микроЭВМ

ЦП	К1810ВМ88
ОС	MS DOS
Объем, Кбайт	
ОЗУ	256
ПЗУ	64
Контроллер ГМД	КМ1821ВГ72А
Число приводов ГМД	2
Диаметр диска ГМД, мм	133
Контроллер ЖМД	i82062
Число приводов ЖМД	1
Число каналов ИРПС	1
Контроллер клавиатуры	K531IP24
Подсистема звука	XT
Видеоконтроллер	МС6845
Объем видеобуфера, Кбайт	16
Системный интерфейс	И41
Тип соединителя	СНП59-96
Тип платы	Европлата Е2
Габаритные размеры, мм	233×220
Потребляемый ток, А	5
Потребляемая мощность, Вт	25

* Дианов А. П., Шелкунов Н. Н. Малогабаритные источники питания для микросистем // Микропроцессорные средства и системы.— 1987.— № 3.— С. 73—76.

141700, Долгоспрудный-1 Моск. обл., аб/я 46, ЦНТТМ «Физтех»; тел. 485-44-77, телегайн 346187 «Наука»

М. Ф. Созыкин

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ НА ОСНОВЕ БИС КР572ПВ1

Модуль АЦП позволяет обрабатывать данные в виде аналоговых напряжений с помощью микроЭВМ «Электроника 60» или ВУМС-28-025С. Использование в составе модуля БИС КР572ПВ1 (рис. 1) дает возможность получить большое разрешение, реализовать режим поразрядного уравнивания, уменьшить размеры платы по сравнению с серийно выпускаемым модулем аналогового ввода 15КА-60/8-010.

Электрическая схема модуля (рис. 2) построена с учетом специфики микроЭВМ «Электроника 60». Модуль и ЭВМ обмениваются данными через интерфейс И2. Число входных разрядов И2 — 16, модуля — 12, поэтому данные считываются за один такт. В качестве согласующих буферных регистров применяется микросхема КР580ИР82. Управляющие слова в эти регистры записываются по адресам с приходом сигналов «ввод» или «вывод», что обеспечивает синхронность работы модуля и микроЭВМ.

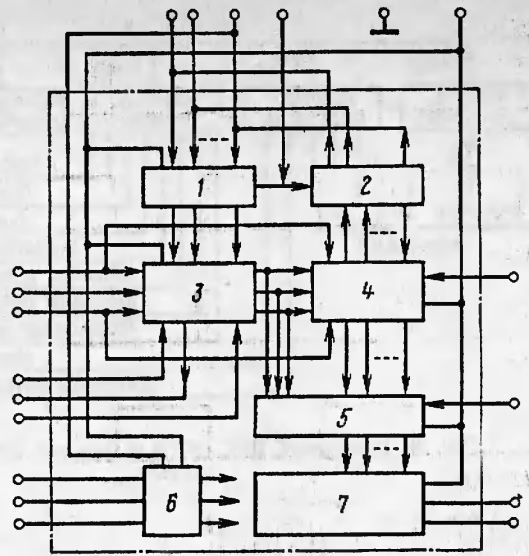


Рис. 1. Структурная электрическая схема БИС АЦП КР572ПВ1:

1, 2 — входная и выходная логика; 3 — регистр сдвига; 4 — регистр последовательного приближения; 5 — регистр АЦП; 6 — схема управления; 7 — АЦП

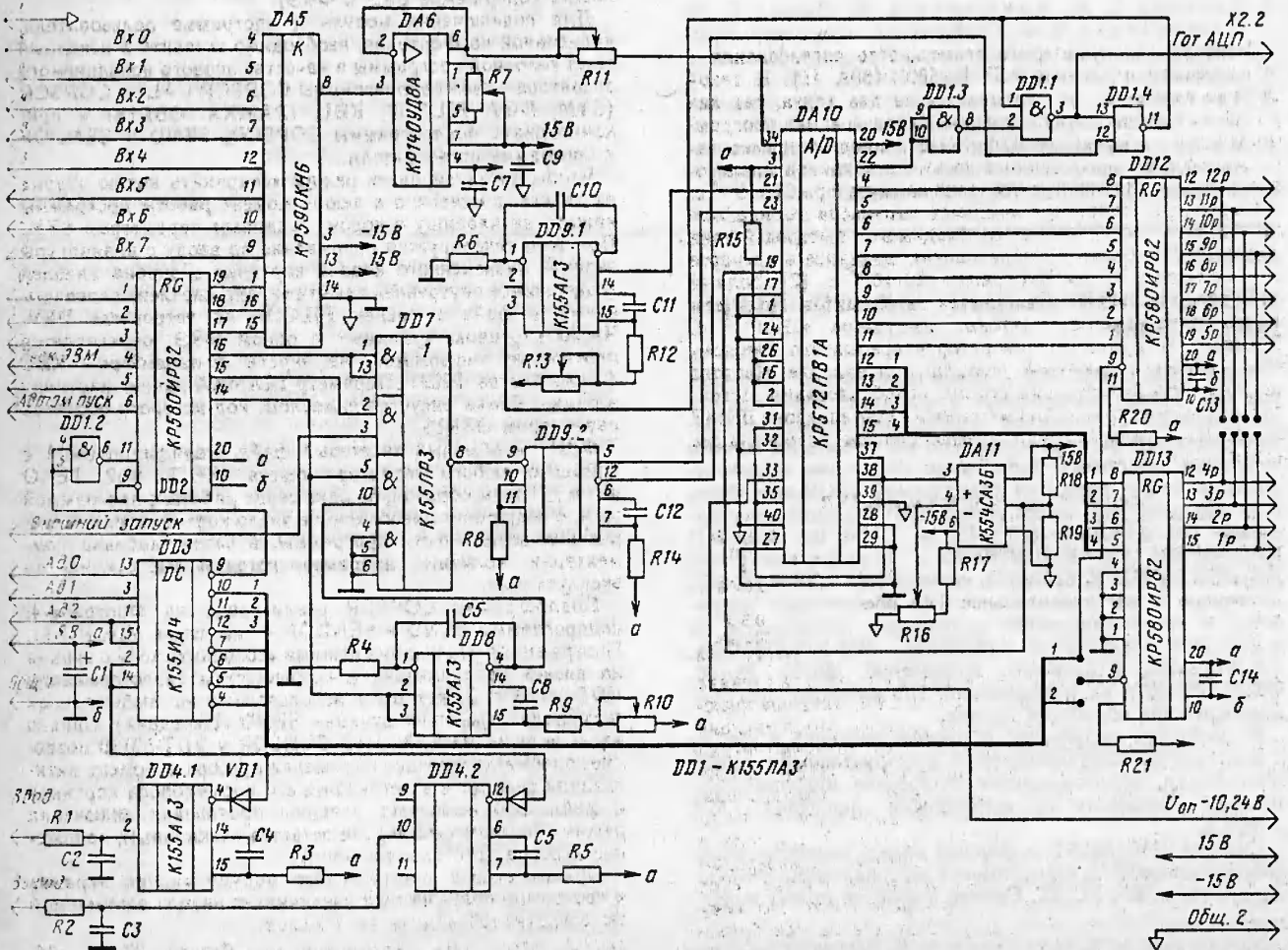


Рис. 2. Принципиальная схема модуля АЦП

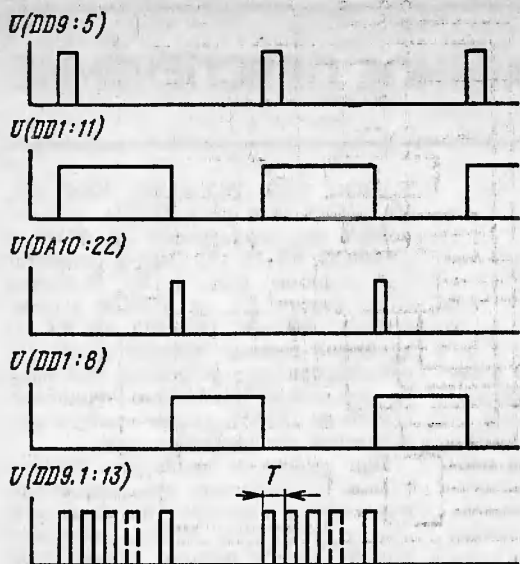


Рис. 3. Временные диаграммы работы модуля АЦП

В модуле предусмотрена возможность согласования с МП-системой на основе МП КР580ИК80А [1]. В такой системе данные будут считываться за два такта, так как у МП 8-разрядная шина данных. Использование программируемого коммутатора КР590КН6 позволяет последовательно обрабатывать данные восьми источников аналоговых сигналов. Время переключения коммутатора $0,3 \cdot 10^{-6}$ с.

Параметры генератора тактовых импульсов выбираются из условия обеспечения необходимого быстродействия модуля АЦП. Время преобразования, заданное в паспорте на микросхему КР572ПВ1, равно $120 \cdot 10^{-6}$ с. [2], полный цикл преобразования совершается за 28 тактов. Получаем период генерируемых тактовых импульсов $4,3 \cdot 10^{-6}$ с. В модуле предусмотрен генератор непрерывного автономного запуска, создающий удобства при наладке. Частота запуска должна быть ниже частоты преобразования. Источник опорного напряжения выполнен на стабилитроне Д818Е с высоким температурным коэффициентом стабилизации напряжения $K_{ст} = \pm 0,001 \% / K$.

Техническая характеристика модуля АЦП

Число входных двоичных разрядов	12
Диапазон входных напряжений, В	0...120, 24
Время преобразования, мкс, не более	120
Число аналоговых входов	8
Дискретность преобразования, В	0,0025
Потребляемый ток от источников питания, не более	0,5 А
5 В	100 мА
15 В	—120 мА
15 В	—120 мА

Цикл преобразования АЦП начинается с приходом сигнала запуска на вход S триггера DD1.1, DD1.4; устанавливающего на прямом выходе единичный уровень, разрешающий работу тактового генератора (рис. 3). Одновременно сигнал запуска поступает на вход микросхемы КР572ПВ1, обеспечивая синхронную работу регистров и компаратора. По окончании цикла преобразования формируется сигнал «Гот» — готовность, поступающий на вход R триггера и запрещающий работу тактового генератора. На инверсном выходе триггера синхронизации появится сигнал «Гот АЦП», который поступает на интерфейс ЭВМ. По нему ЭВМ должна начать считывание результатов преобразования.

Телефон 62-92-54, Горький

ЛИТЕРАТУРА

1. Архитектура и проектирование микроЭВМ. Организация вычислительных процессов / Под ред. Л. Н. Преснухина. — М.: Высш. шк., 1986. — Т2.
2. Федорков Б. Г., Телец В. А., Дегтяренко В. И. Микроэлектронные цифро-аналоговые преобразователи. Массовая библиотека инженера «Электроника». — М.: Радио и связь. — 1984. — Вып. 41.

Статья поступила 24.02.89

КРАТКОЕ СООБЩЕНИЕ

УДК 681.3.06

А. А. Тараненко

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РЕЖИМА «КОПИЯ ЭКРАНА» ДЛЯ СИСТЕМЫ РАЗДЕЛЕНИЯ ВРЕМЕНИ

Разработан унифицированный программный модуль «копия экрана» для вывода содержимого внутренней памяти экрана видеотерминала VDT-52130 в файл на устройстве прямого доступа. Модуль реализован в системе разделения времени РАФОС/ТС ЭВМ СМ1420 и состоит из трех подпрограмм: COPSCR (заказать файл копии экрана — ФКЭ), SEND (считать содержимое внутренней памяти экрана дисплея в буфер копии экрана — БЭК) и SENDOF (переписать содержимое БЭК в ФКЭ).

Для подключения модуля к программе пользователя, написанной на Фортране, необходимо вставить в исходный текст головной программы в качестве первого исполняемого оператора вызов подпрограммы COPSCR CALL COPSCR (SYM, DEV, FILCOP, KBL, IRAMKA, OBRIS) и прикомпоновать подпрограммы COPSCR, SEND и SENDOF к программе пользователя.

Чтобы активизировать режим «сохранить копию экрана на диске», достаточно в любой момент работы программы нажать на клавишу с кодом, заданным параметром SYM. При этом генерируется прерывание по вводу с клавиатуры заранее назначенного кода и картинка с экрана дисплея (содержимое внутренней памяти экрана дисплея) переписывается в файл с именем FILCOP на устройстве DEV. Число картинок, хранимых в одном ФКЭ, определяется плотностью информации на экране и параметром KBL (число блоков ФКЭ). Параметр IRAMKA задает картинку в рамке. Рамка рисуется символом, код которого задается параметром OBRIS.

ФКЭ — обычный текстовый файл, модифицируемый с помощью любого редактора текста (EDIT, K52, TECO и т. п.). Таким образом, за один сеанс работы с программой можно подготовить необходимое число картинок, иллюстрирующих возможности программы, а затем, добавив комментарии, получить, например, готовую инструкцию по эксплуатации.

Подпрограмма COPSCR реализована на Фортране-4, подпрограммы SEND и SENDOF — на языке MACRO-11. Программная реализация режима «сохранить копию экрана на диске» не привязана к особенностям видеотерминала VDT-52130 и может быть использована на любых типах терминалов, имеющих функцию SEND (Передача). Однако наличие функции SENSING CURSOR у VDT-52130 позволяет запомнить текущее положение курсора в момент активизации режима и восстановить его после сброса картинки в файл. Это исключает внешние проявления включения режима (содержимое экрана остается неизменным, положение курсора восстанавливается).

Прикомпоновка программного модуля «копия экрана» к программе пользователя увеличивает размер занимаемой ею оперативной памяти на 2 Кслов.

140100, Моск. обл., Раменское, ул. Лесная, 27, кв. 34; тел. 556-60-12

Сообщение поступило 22.08.88

УДК 681.06

Э. А. Гунин, В. Г. Гуслев, Б. В. Налибоцкий, Ю. И. Петрусевич

СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НЕОДНОРОДНЫХ ЛОКАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Специальное программное обеспечение (СПО) разработано для объединения в единую сеть ЭВМ различной архитектуры (ЕС1840/41/42, «Искра 1030», «Нейрон», IBM PC/XT/AT, SM4, SM1420, SM1300, SM1600, SM1505, SM1700, ДКВ, LSI-11, PDP-11, VAX-11), работающих под управлением различных ОС (Альфа-ДОС, MS DOS, ОС РВ, RSX-11M/M-PLUS, РАФОС, RT-11, P/OS, МОС ВР, VAX/VMS).

В процессе разработки описанных ниже версий СПО мы стремились к тому, чтобы при работе с этим ПО пользователь испытывал как можно меньше затруднений, неизбежно связанных с внедрением нового программного продукта. Именно поэтому постарались сделать так, чтобы доступ из одних ЭВМ к различным устройствам других ничем не отличался от способа доступа к этим же устройствам, подсоединенным непосредственно к удаленной ЭВМ.

При такой организации локальной вычислительной сети (ЛВС) различные ЭВМ с ограниченным набором периферийных устройств, именуемые в дальнейшем периферийные ЭВМ (ПЭВМ), подключаются к одной или нескольким центральным ЭВМ (ЦЭВМ), обладающим более широкими возможностями по сравнению с ПЭВМ (накопителя на магнитных дисках большой емкости, накопителя на магнитных лентах, высокоскоростные печатающие устройства и т. д.). При таком подключении мощность ПЭВМ возрастает за счет доступа к ресурсам ЦЭВМ, а ЦЭВМ, в свою очередь, переходит на более рациональный и эффективный режим работы, поскольку появляется возможность часть ее заданий распределить между ПЭВМ, что особенно ощутимо, когда ПЭВМ работают в режиме редактирования текстов или сбора данных.

Этот, далеко не полный, список преимуществ неоднородной ЛВС мы хотим дополнить еще одним, на наш взгляд, наиболее важным — возможностью обмена информацией между программно и аппаратно несовместимыми ЭВМ, входящими в состав ЛВС, поскольку крайне неудобной альтернативой указанной возможности является перенос исходных текстов программ и другой информации вручную.

Все разработанное нами СПО в зависимости от того, где оно функционирует, можно разделить на две части — СПО ЦЭВМ и СПО ПЭВМ. СПО ЦЭВМ состоит из многозадачной многопользовательской ОС и задач-диспетчеров, поддерживающих взаимодействие между ЭВМ согласно разработанному протоколу. СПО ПЭВМ, в свою очередь, содержит в своем составе ОС на ПЭВМ, набор сетевых драйверов, взаимодействующих с задачами-диспетчерами ЦЭВМ, и программы поддержки удаленного терминала ЦЭВМ на ПЭВМ и изменения конфигурации.

Основа СПО на ПЭВМ — драйвер сетевого диска, с помощью которого пользователь обращается к дисковому устройству или файлу ЦЭВМ как к диску, подключенному непосредственно к ПЭВМ.

Кроме драйвера сетевого диска на ПЭВМ могут быть драйверы сетевого накопителя на магнитной ленте и сетевого записи-ориентированного устройства (принтера, терминала, графопостроителя и т. п.). В таком случае эти устройства также становятся доступными с ПЭВМ, причем обращение пользователя к ним такое же, как к устройствам, подключенным непосредственно к ПЭВМ.

СПО следует рассматривать не как альтернативу существующим и широко распространенным сетевым средствам, а лишь как их дополнение, поскольку при их совместной работе не происходит каких-либо конфликтов.

Все версии специального ПО, описанные ниже, — оригинальные программные продукты, разработанные авторами. Они не содержат обращений к стандартным сетевым средствам, поставляемым с упоминаемыми ОС. К настоящему времени разработаны и находятся в стадии внедрения шесть версий СПО для организации неоднородных ЛВС. Различия между ними обусловлены разными в операционных системах ЭВМ, входящими в состав ЛВС.

Специальное программное обеспечение (версии 2.1, 2.2, 4.1)

В СПО версий 2.1, 2.2, 4.1 в роли ЦЭВМ выступает ЭВМ, работающая под управлением одной из следующих ОС: ОС РВ, RSX-11M/M-PLUS,

РАФОС, NTS, TSX-PLUS, МОС ВР, VAX/VMS, а в роли ПЭВМ — микроЭВМ под управлением ОС ФОДОС (РАФОС, RT-11, ОС ДВК и совместимых с ними) (рис. 1—3). В состав СПО версии 2.1 на ПЭВМ входят: ФОДОС, РАФОС, ОС ДВК или RT-11, драйверы сетевых блочно- и записи-ориентированных устройств; программа поддержки удаленного терминала ЦЭВМ на ПЭВМ; задачи-утилиты для изменения конфигурации сети.

При работе с драйвером сетевых дисков пользователю предоставляется возможность доступа к некоторым дискам и файлам ЦЭВМ, имеющим формат записи данных, совместимый с форматом данных ОС РАФОС (ОС ДВК, RT-11). Обращение к таким дискам, с точки зрения пользователя, не отличается от обращения к дискам ПЭВМ, поскольку все функции поддержки протокола возложены на драйвер и соответствующую задачу-диспетчер в ЦЭВМ. Аналогично выглядит обращение к печатающему устройству ЦЭВМ. Для экономии пространства на дисковых устройствах ЦЭВМ всем ПЭВМ выделяется один системный диск, доступный только для чтения,

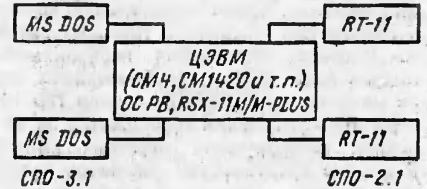


Рис. 1.

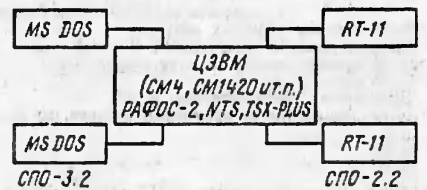


Рис. 2.



Рис. 3.

а также по семь рабочих дисков на каждого пользователя. Размеры этих дисков могут быть различными и легко меняться в процессе работы.

Драйвер сетевых устройств, ориентированных на запись, предоставляет возможность ПЭВМ обращаться ко внешним устройствам последовательного доступа ПЭВМ (принтеры, терминалы и т. п.).

Задачи-утилиты для изменения конфигурации сети позволяют динамически присоединять и переназначать (в среде ОС ПЭВМ) различные устройства ЦЭВМ и получать информацию о текущих назначениях сетевых устройств.

СПО версий 2.1, 2.2, 4.1 на ЦЭВМ включает в себя одну из указанных выше ОС, задач-диспетчеров и систем виртуальных дисков (разного вида драйверы виртуальных или логических дисков, имеющиеся в большинстве ОС). Система виртуальных дисков позволяет назначать любое дисковое устройство или файл в его качестве на ЦЭВМ. В функции задач-диспетчеров входит: поддержка сетевого протокола на ЦЭВМ; прием информации из ПЭВМ и распределение ее между различными периферийными устройствами ЦЭВМ; считывание данных с периферийных устройств ЦЭВМ и передача их на обслуживаемую ПЭВМ; обеспечение межзадачного обмена данными.

Для обслуживания каждой ПЭВМ на ЦЭВМ выделена одна задача-диспетчер. В СПО версий 2.1, 2.2, 4.1 входят обычно два вида задач-диспетчеров — загружающие и поддерживающие. Различие между ними состоит в том, что при запуске загружающей задачи-диспетчера происходит загрузка ОС с соответствующего диска ЦЭВМ в ПЭВМ, затем эта задача-диспетчер переходит в режим поддержки протокола сети. Использование загружающих задач особенно актуально при работе ПЭВМ в бездискковой конфигурации.

СПО версий 2.1, 2.2, 4.1 предоставляет пользователю следующие возможности:

- работать на ПЭВМ в режиме удаленного терминала ЦЭВМ;
- загружать ОС ФОДОС (РАФОС, ОС ДВК, RT-11) в ПЭВМ из ЦЭВМ;
- работать с ОС ФОДОС на ПЭВМ как в дисковой, так и в бездискковой конфигурации;
- доступа для обмена информацией (файлами) ко всем периферийным устройствам ЦЭВМ.

Размер задачи-диспетчера для обслуживания одной линии связи не превышает 1 Кбайт в памяти ЦЭВМ, что позволяет работать с ЦЭВМ одновременно большому числу ПЭВМ. Поскольку общий размер задач-диспетчеров в оперативной памяти ЦЭВМ, незначителен даже при большом числе ПЭВМ, ЦЭВМ может достаточно эффективно поддерживать многотерминальную работу.

В СПО версий 2.1, 4.1 обмен файлами

между ЦЭВМ и ПЭВМ, а также между самими ПЭВМ осуществляется двумя способами: при первом способе используются утилиты FLX ОС ЦЭВМ, так как системные и пользовательские диски ПЭВМ (сетевые, разумеется) выступают как ее (ЦЭВМ) собственные; при втором — программный эмулятор ОС RT-11 под управлением ОС РВ или МОС ВП. Файловые системы на ПЭВМ и ЦЭВМ совпадают. Применение этих способов позволяет производить быстрый обмен файлами в сети, причем текстовые и объектные файлы подвергаются преобразованию в соответствии с форматами представления данных, принятыми в различных ОС.

В процессе генерации СПО версий 2.1, 2.2, 4.1 легко адаптируются для работы с различными устройствами межмашинной связи (стандартными и нестандартными). В роли таких устройств могут выступать устройства последовательного обмена (УПО) с интерфейсами ИРПС, Стык С2, RS-232С, мультиплексоры передачи данных, СПИ-15. Наиболее пригодны устройства СПИ-15 на основе интерфейса DL-КИ/СИ, производимые Ленинградским НПО «Электронмаш». Они характеризуются высокой надежностью и быстродействием. Кроме того, в их состав входит аппаратно-реализованная и записанная в ПЗУ программа, поддерживающая удачную, на наш взгляд, протокол, принятый в СПО Алиса. Именно этот протокол используется в начальный момент работы СПО версий 2.1, 2.2, 4.1 для загрузки ОС в ПЭВМ. Применяя СПИ-15, авторам удалось на ПЭВМ при работе с сетевыми дисками добиться скорости обмена 26 Кбайт/с на верхнем уровне.

Длительный опыт работы с СПО версий 2.1, 2.2, 4.1 показал, что не только по простоте устройству, но и по надежности и емкости работа с сетевыми дисками предпочтительнее работы с НГМД, входящими в состав ПЭВМ. Более того, доступ к накопителям на магнитной ленте ЦЭВМ предоставляет уникальную, в определенном смысле, возможность хранить файлы ОС ПЭВМ на магнитной ленте, что позволило почти полностью отказаться от использования для этих целей ГМД.

Специальное программное обеспечение (версии 3.1, 3.2, 4.2)

В СПО версий 3.1, 3.2, 4.2 в роли ЦЭВМ (СМ4, СМ1420, СМ1600, «Электроника 100/25» и т. д.) выступают те же машины, что и в СПО версий 2.1, 2.2, 4.1, а в роли периферийных (ЕС1840/41/42, IBM PC/AT/XT, «Искра 1030» и другие, совместимые с ними программно) — ЭВМ, работающие под управлением ОС MSDOS или PC DOS (см. рис. 1—3). Состав СПО версий

3.1, 3.2, 4.2 на ЦЭВМ и ПЭВМ совпадает с составом СПО версий 2.1, 2.2, 4.1.

При работе с СПО версий 3.1, 3.2, 4.1 пользователь может ввести дополнительное дисковое устройство, размер которого физически ограничивается лишь размером дискового пространства, выделенного для данной ПЭВМ на ЦЭВМ. Например: ПЭВМ ЕС1840, имеющая в своем комплекте два НГМД, соединена с ЭВМ СМ1420, в состав которой входят накопители на магнитных дисках типа СМ5408. Если один из этих накопителей выделен для работы ЕС1840 в составе СПО-3.1, то в этом случае в ПЭВМ вводится дополнительный диск размером около 15 Мбайт. Согласно принятому в ОС MS DOS (PC DOS) соглашению все дисковые устройства именуруются последовательно буквами латинского алфавита. Следовательно, имя сетевого диска в такой конфигурации ПЭВМ будет «С» («А» и «В» принадлежат двум приводам НГМД). Во время работы пользователь ПЭВМ может динамически присоединять для работы то или иное устройство ЦЭВМ. Драйверы сетевых устройств устанавливаются в процессе загрузки ОС на ЦЭВМ. Для этого в файл конфигурации системы включаются дополнительные строки, которые кроме имен файлов, хранящих код драйверов, могут содержать информацию, необходимую для настройки каналов связи (например, скорость обмена или номер канала). Все системное и прикладное ПО работает с драйвером сетевого диска без каких-либо изменений. Существует лишь ограничение для применения программ, содержащих обращения непосредственно к базовой системе ввода-вывода BIOS для обслуживания дисковых устройств. Это позволяет использовать большое по объему ПО на ПЭВМ, не имеющих в своем составе дисков большой емкости.

Для обмена файлами между устройствами ЦЭВМ и сетевыми дисками ПЭВМ (расположенными на ЦЭВМ) разработана специальная программа MSF, в функции которой кроме обмена информацией входит преобразование форматов текстовых файлов. Эта программа, работающая под управлением ОС ЦЭВМ, предоставляет возможность выполнять в режиме эмуляции большинство файловых команд монитора MS DOS, таких как DIR, COPY, DEL, PRINT, CD, MD и т. п., а также две специальные команды: TRANSFER — перенос файлов между дисками с различной файловой структурой и INIT — отделенный от форматирования процесс создания файловой структуры MS DOS на дисках ЦЭВМ. С помощью этой программы файлы, принадлежащие пользователю ПЭВМ, перемещаются на устройства ЦЭВМ и обратно. После такого обмена с ними можно проводить следующие операции: записать их на магнитную ленту для хранения и пере-

носа на другие ЭВМ; произвести обмен данными с сетевыми дисками других ПЭВМ, в том числе работающих под управлением ОС РАФОС (ОС ДВК, RT-11, ОС РВ, МОС ВП и т. д.).

Возможность последней операции обусловлена тем, что на одной ЦЭВМ могут параллельно работать СПО версий 2.1, 2.2, 4.1 и 3.1, 3.2, 4.2. При работе в составе СПО версий 3.1, 3.2, 4.2 на ЦЭВМ используются задачидиспетчеры поддерживающего типа, поскольку загрузка ПЭВМ осуществляется обычно не из ЦЭВМ, а с их собственных дисковых устройств. Для работы СПО версий 3.1, 3.2, 4.2 со стороны ЦЭВМ требуется та же аппаратура, что и для СПО версий 2.1, 2.2, 4.1. Можно применять последовательные коммуникационные каналы, имеющиеся почти во всех моделях персональных ЭВМ.

Кроме того, для ЕС1840, ЕС1841 можно дополнительно использовать следующие устройства:

интерфейс типа DL-КИ/СИ, разработанный авторами. Это устройство позволяет присоединить ЕС1840, ЕС1841 к ЦЭВМ через аппаратуру СПИ-15 и в случае необходимости может быть относительно легко переделано под стандарт других ЭВМ (например, «Искра 1030» или IBM PC/AT/XT);

устройство сопряжения системной шины ЕС1840, ЕС1841 с шиной микроЭВМ «Электроника 60», ДВК и т. п. (МПИ или Q-шина), позволяющее подсоединить к ЕС1840, ЕС1841 различную аппаратуру, разработанную для микроЭВМ «Электроника 60» и ДВК, в частности СПИ-15.

В СПО не существует каких-либо ограничений для подключения одной ПЭВМ одновременно к нескольким ЦЭВМ, работающим под управлением различных ОС, что позволяет обмениваться информацией между самими ЦЭВМ. Это обстоятельство делает возможным создание сетевых комплексов сложной архитектуры на базе СПО, максимально способствующих решению задач пользователя.

Специальное программное обеспечение может быть легко перенесено на другие типы ОС и ЭВМ, например ДЕМОС, ИНМОС, UNIX, PICK, OS/2 и т. д.

220036, Минск-36, ул. Куприянова, 17—37, Гукин Э. А.; тел. 56-30-46

УДК 681.3.08

В. П. Перепелов, Д. С. Русанов

ДИНАМИЧЕСКАЯ ЗАГРУЗКА ПРОГРАММНЫХ СЕГМЕНТОВ В ОС РАФОС

При решении больших задач на ЭВМ типа СМ4 часто возникают трудности в связи с малым размером виртуального адресного пространства программ (менее 65 Кбайт). Практически во всех ОС для СМ4 реализованы два основных механизма экономии адресного пространства:

организация цепочек программ, при которой решаемая задача делится на фазы, реализуемые отдельными программами, выполняемые последовательно друг за другом. Данные между фазами при этом передаются преимущественно через файловую систему;

организация оверлейных программ, что позволяет использовать один и тот же участок виртуального адресного пространства для размещения различных программных сегментов, причем замена их производится автоматически при вызове отсутствующего в данный момент в памяти сегмента.

Однако для ряда задач применение механизма цепочек или оверлеев вызывает значительные накладные расходы. Рассмотрим, например, программу интерпретативного имитационного моделирования дискретных систем, в которой модели компонент системы представляются рядом подпрограмм, структура связей между компонентами задается входными данными, а алгоритм моделирования вместе с передачей данных между моделями реализуется главной программой. Если при этом число компонент велико и оперативная память не может вместить все возможные модели одновременно, то очевидна необходимость применения какого-либо способа экономии памяти. Очевидно также, что не каждая из моделируемых систем включает в себя все возможные компоненты и набор компонент для моделирования каждой конкретной системы мог бы поместиться в оперативную память, если туда не загружать неиспользуемые в конкретном случае модели.

Единственный способ реализации такой программы в условиях дефицита адресного пространства — использование механизма оверлеев. Однако число обращений к диску для загрузки оверлейных сегментов можно минимизировать, лишь зная структуру конкретной моделируемой системы. В общем же случае при любом выбранном размещении подпрограмм-моделей по оверлеям всегда найдется такая структура моделируемой системы, при которой число перезагрузок оверлеев будет значительно, что приведет к резкому падению скорости работы программы. На рис. 1 нетрудно видеть, что, если моделируемая в данный момент система состоит из компонент M1, M2 и M3,

расположенных в различных сегментах одного уровня, любые вызовы подпрограмм из фрагментов M1, M2, M3 сопровождаются считыванием соответствующего сегмента с диска. И несмотря на то, что реально нужен лишь сравнительно небольшой фрагмент кода сегмента, в память считывается весь сегмент целиком. Это означает, что большую часть оперативной памяти будут занимать коды неиспользуемых в данной конфигурации компонент.

Ряд ОС, таких, например, как ОС ЕС, MS DOS, позволяет загрузить и вызвать программы динамически. С помощью этого механизма необходимое множество подпрограмм загружается в оперативную память, после чего произвольно передается управление любой из них. При этом исключаются лишние обращения к дискам, экономится память, обеспечивается возврат в главную программу в точку вызова (рис. 2).

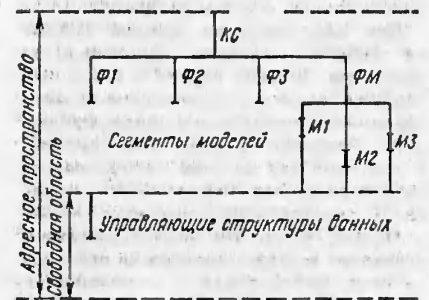


Рис. 1. Структура оверлейной программы.

КС — корневой сегмент; Ф1...Ф3 — фазы некоторой предварительной обработки; ФМ — фаза моделирования, в которой прикомпонованы три сегмента моделей M1...M3 (в виде сегментов следующего уровня)

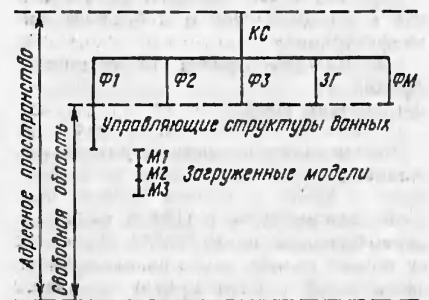


Рис. 2. Структура программы с динамическими сегментами.

ЗГ — модуль, реализующий механизм динамической загрузки сегментов

Статья поступила 21.11.88

В то же время семейство ОС, совместимых с ОС РАФОС и ОС РВ, работающих на ЭВМ типа СМ4, не включает в себя средства динамической загрузки. В этих системах настройка на конкретные адреса в виртуальном адресном пространстве осуществляется статически на этапе компоновки и размещение какой-либо части программы в произвольные адреса невозможно.

В процессе работы над системой логического моделирования цифровых устройств интегрирующего типа на ЭВМ СМ4 под управлением ТS-монитора ОС РАФОС авторы статьи столкнулись с проблемой чрезмерных накладных расходов при попытке использования механизма оверлеев в моделирующей программе (см. рис. 1). Буквально на первых же прогонах выяснилось, что скорость процесса моделирования даже для не очень сложных схем фактически определялась быстродействием дискового устройства, которое при этом также чрезмерно загружалось, что недопустимо для систем с коллективным доступом. Кроме того, наличие в оперативной памяти неиспользуемых в каждой конкретной схеме моделей привело к значительному перерасходу памяти, что снижало размерность моделируемой схемы.

Указанные затруднения привели к необходимости разработки механизма динамической загрузки программных сегментов в ОС РАФОС. При этом структура программы приняла вид, показанный на рис. 2. Для ясности дальнейшего изложения напомним функции программы-компоновщика, собирающего выполняемую программу из нескольких объектных модулей: планирование размещения программных секций в памяти; определение значений глобальных символов; настройка адресов в зависимости от конкретного размещения программы в оперативной памяти.

На выходе компоновщик генерирует три типа файлов: программы — SAV, REL или LDA; таблицы символов, содержащие информацию относительно размещения глобальных символов программы в виртуальном адресном пространстве, — STB; карты памяти — MAP. Динамическим сегментом будем называть фрагмент исполняемого машинного кода программы, хранящийся в отдельном файле на магнитном диске. Он может быть считан прикладной программой в любой свободный участок оперативной памяти достаточного размера.

В основе реализации метода динамических сегментов лежит идея, что процесс разрешения внешних ссылок и настройки перемещаемых адресов программы может быть с этапа компоновки на этап выполнения. Предлагаются два типа сегментов: динамически перемещаемый — ДПС и динамически загружаемый — ДЗС. Различие

между двумя типами сегментов: в ДПС все ссылки на глобальные символы, находящиеся вне сегмента, должны быть разрешены с помощью файла таблицы символов основной программы, а в ДЗС — нет. Из этого следует, что ДПС может использоваться только одной конкретной основной программой, а ДЗС — любой программой, содержащей необходимые процедуры загрузки динамических сегментов.

В процессе реализации описанного выше подхода были разработаны структура файлов динамических сегментов, алгоритм загрузки и вызова подпрограмм и программа создания динамических сегментов из отдельных объектных модулей. Структура динамического сегмента была выполнена с учетом того, чтобы как можно большая часть работы по компоновке сегмента, а также по его связыванию с главной программой могла быть выполнена статически, т. е. на этапе компоновки самого сегмента, а до этапа выполнения отложено только то, что не может быть выполнено статически.

В формате файла динамического сегмента (рис. 3) первые два слова содержат имя динамического сегмента в коде RADIX-50, далее два слова — имя главной программы в RADIX-50. Для ДЗС это поле содержит нули. Следующее слово — длина программного кода в байтах, после которого следует собственно код. После программного кода идет таблица перемещений в стандартном формате ОС РАФОС. Каждое слово таблицы перемещений представляет собой значение адреса, содержимое которого, деленное на 2, должно быть модифицировано. Знаковый разряд этого слова показывает, каким образом нужно модифицировать указанную ячейку. Если он содержит единицу, то в модифицируемое слово следует записать разность между его содержимым и текущим значением счетчика команд, в противном случае — их сумму. Признаком конца таблицы перемещений служит слово, содержащее

Имя модуля	Имя ГП	Длина кода
Исполнимый код		
Таблица перемещений		-2
Таблица точек входа		-2
Таблица внешних ссылок		-2

Рис. 3. Формат файла динамического сегмента

значение —2. За таблицей перемещений следует таблица точек входа сегмента, состоящая из записей по три слова, первые два из которых представляют имя точки входа в коде RADIX-50, а третье — смещение этой точки относительно начала кода. Конец таблицы отмечается знаком —2. Если данный файл — ДЗС, то далее идет таблица внешних ссылок, представляющая собой множество записей переменной длины. Каждая запись включает имя глобального символа в RADIX-50, счетчик числа ссылок на данный символ и по числу ссылок — адреса, в которые нужно занести значения смещений до данного глобального символа. Признаком конца таблицы служит слово, содержащее значение —2.

Алгоритм загрузки динамического сегмента следующий: загрузить код в память; если до ДЗС, то в соответствии с таблицей внешних ссылок и таблицей символов основной программы вычислить смещения до неопределенных в нем глобальных символов и занести их в соответствующие адреса кода динамического сегмента; записать в память таблицу точек входа динамического сегмента. Тогда обращение к подпрограмме с именем SUBR из уже загруженного сегмента может быть выполнено по алгоритму: найти в таблице точек входа элемент, соответствующий SUBR и определить адрес точки входа; передать управление по найденному адресу.

Поскольку компоновщик ОС РАФОС (программа LINK) не выдает информацию, необходимую для построения динамических сегментов, была разработана специальная программа — генератор динамических сегментов GDS, которая выполняет те же функции, что и LINK, но на выходе дает файл динамического сегмента, построенного из одного или нескольких объектов файлов. Программа запускается командой монитора R или RUN и имеет командную строку в формате CSI:

спвхфхф [Ключ] = спвхф1...спвхф6

Входных файлов может быть не более шести. Ключ принимает одно из двух значений: /R — создается файл ДПС; /L — создается файл ДЗС. Входные файлы по умолчанию имеют расширение OBJ, выходной файл в зависимости от заданного режима — расширение DLS (ключ L) или DRS (ключ R).

Программа полностью написана на языке Паскаль. Ввиду идентичности структур объектных файлов в ОС РАФОС и ОС РВ данная схема динамической загрузки может быть практически без изменений применена и в ОС РВ.

Основные достоинства метода динамических сегментов:

в оперативной памяти находятся только те подпрограммы, которые необходимы в данный момент. Высвободив-

шаяся память может быть использована в других целях;

нет обращений к дискам для перезагрузки оверлейных сегментов;

возможно неограниченное наращивание множества динамических сегментов, что значительно увеличивает функциональную гибкость программного обеспечения.

Кроме того, механизм динамических сегментов обладает и рядом ограничений и недостатков, связанных с тем, что в некоторых моментах он противоречит принятой в указанных ОС концепции статической привязки адресов: программная секция, объявленная в динамическом сегменте с атрибутом OVR, на самом деле не будет перекры-

ваться с одноименной программной секцией в основной программе, так как в момент построения динамического сегмента известен лишь файл таблицы символов основной программы, а в нем не содержится информации о том, как размещена та или иная программная секция;

динамический сегмент не может содержать непосредственно ссылки на глобальные символы, не определенные в нем самом или в основной программе; глобальные символы, определенные в динамическом сегменте, не становятся известными основной программе, т. е. она также не может прямо на них ссылаться. Доступ к ним возможен

лишь посредством смещений относительно точек входа, адреса которых хранятся в специальной таблице внутри основной программы. При этом нужно четко знать структуру кода динамического сегмента.

Из перечисленных ограничений следует, что использование динамических сегментов связано с определенными трудностями и будет эффективно лишь в случае, когда другие подходы будут явно уступать методу динамических сегментов по накладным расходам.

Телефон 158-42-01, Москва

Статья поступила 1.11.88

УДК 681.3.08

А. Н. Рябов, С. В. Ануфриев

ПОДПРОГРАММЫ С ПЕРЕМЕННЫМ ЧИСЛОМ ПАРАМЕТРОВ НА ФОРТРАНЕ СИСТЕМЫ РАФОС

Общие сведения о передаче параметров

Для передачи параметров между программными единицами языка Фортран требуется соответствие числа и типов формальных и фактических параметров. Это не дает возможность составлять на Фортране подпрограммы или подпрограммы-функции, обрабатывающие переменное число фактических параметров.

В работе [1] предлагается подход, позволяющий расширить возможность передачи параметров между программными единицами. В частности, допускалось обращение к подпрограмме с переменным числом фактических параметров, которое не могло превышать число формальных параметров, указанных в заголовке вызываемой программной единицы. В случае передачи меньшего числа фактических параметров соответствующим формальным параметрам присваивались некоторые значения по умолчанию, задаваемые пользователем в явном виде в вызываемой подпрограмме. Для реализации такой возможности использовалась служебная подпрограмма на ассемблере.

В настоящей работе предлагается более общий подход к передаче параметров, реализуемый введением дополнительных служебных подпрограмм на Макроассемблере [2—4].

Работа с адресами на Фортране

Операционная система РАФОС имеет ряд служебных подпрограмм и подпрограмм-функций, позволяющих работать с адресами на Фортране СМ.

Функция **IPEEK** возвращает содержимое слова, находящегося по указанному абсолютному 16-битовому адресу памяти. Обращение к функции:

I=IPEEK (ADDR),

где **ADDR** — проверяемый абсолютный адрес слова. При нечетном значении параметра происходит прерывание; **I** — результат выполнения функции.

Функция **IPEEKВ** возвращает содержимое байта, находящегося по указанному абсолютному 16-битовому адресу памяти. Адрес может быть четным и нечетным. Обращение к функции:

I=IPEEKВ (ADDR),

где **ADDR** — проверяемый абсолютный адрес байта, допускаются нечетные адреса; **I** — результат выполнения функции.

Подпрограмма **IPOKE** записывает 16-битовую целую величину в заданную абсолютным адресом ячейку памяти.

Обращение к подпрограмме:

CALL IPOKE (ADDR, IVALUE),

где **ADDR** — модифицируемый абсолютный адрес, при нечетном значении параметра происходит прерывание; **IVALUE** — целое значение, записываемое по указанному адресу.

Функция **IPOKEВ** записывает определенную 8-битовую величину в указанный байт. Адрес может быть как четный, так и нечетный. Обращение к функции:

I=IPOKEВ (ADDR, VALUE),

где **ADDR** — целая спецификация абсолютного адреса, подлежащего изменению, допускается нечетная адресация; **VALUE** — целое значение, которое будет храниться в данном адресе.

Функция **IADDR** возвращает 16-битовый абсолютный адрес ячейки памяти, по которому размещается ее параметр, в виде целого значения функции. Обращение к функции:

I—IADDR (ARG),

где **ARG** — переменная, константа или выражение, адрес ячейки памяти которого нужно получить.

Аппарат передачи параметров и соглашения о межпрограммных связях

В компиляторах на языке Фортран для передачи управления подпрограмме и возврата из нее используются команды **JSP** и **RTS**:

JSR RN, SUB.

Команда выполняется следующим образом. Сначала содержимое регистра **RN** заносится в стек. После этого содержимое регистра **PC** перемещается в **RN**. В **PC** в этот момент находится адрес следующей команды вызываемой программы. На этот адрес должно возвращаться управление из подпрограммы после ее завершения. Затем в **PC** заносится содержимое приемника и управление передается подпрограмме. Команда перехода к подпрограмме **JSR RN, SUB** «моделируется» следующими командами:

MOV RN, —(SP) — занесение **RN** в стек,
MOV PC, RN — сохранение адреса возврата,
JMP SUB — переход п/п **SUB**.

Для передачи управления обратно вызываемой программе применяется команда возврата из подпрограммы **PTS RN**. Во время выполнения команды **RTS** в **PC** помещается содержимое регистра **RN**, после чего выбирается элемент из вершины стека и помещается в регистр **RN**. Команда возврата из подпрограммы **RTS RN** «моделируется» следующими командами:

MOV RN, PC — занесение адреса возврата,
MOV (SP)+, RN — восстановление **RN**.

В Фортране СМ в качестве регистра **RN** задается седьмой регистр **PC**. В этом случае при выполнении команды **JSR PC, SUB** адрес возврата находится в стеке, где **SUB** — глобальный символ, точка входа в подпрограмму. Ре-

регистр R5 содержит адрес списка аргументов, имеющего следующий формат:

Не определено	Число аргументов
Адрес первого аргумента	
Адрес второго аргумента	
.	
.	
Адрес N-го аргумента	

Адрес младшего байта первого слова помещается в RS. Значение -1 используется как адрес любых опущенных аргументов, которые отмечаются в операторе CALL запятыми. Например: CALL SUB (A,,B)

Структура данных при передаче параметров и использование стека

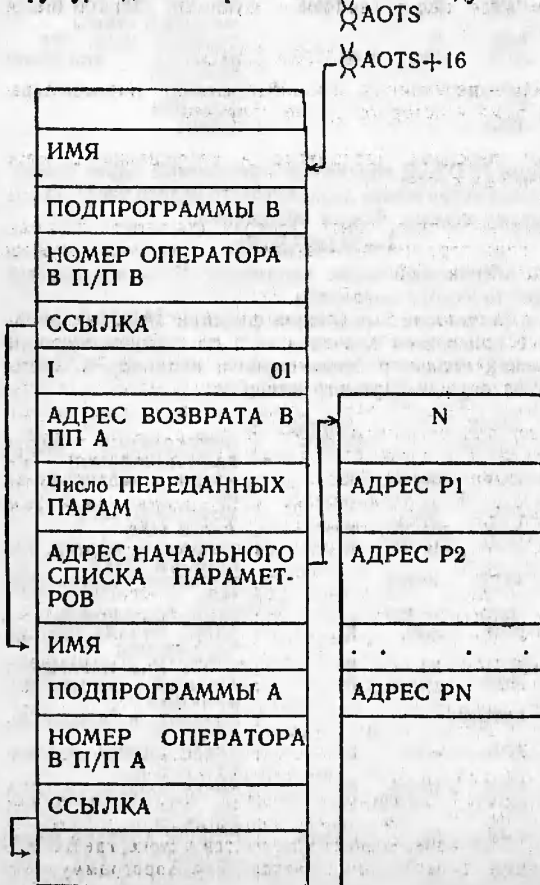
Компилятор транслирует программы на Фортране в один из двух типов объектных программ: с прямым или модульным кодом. В зависимости от типа кода для передачи параметров используются стеки различного строения.

Пусть из подпрограммы А вызывается подпрограмма В: CALL В (P1, P2, ..., PN).

Структура данных в стеке в момент входа в подпрограмму (п/п) В приведена в табл. 1—3.

Таблица 1

Структура данных при использовании модульного кода



Структура и назначение программы OTIS

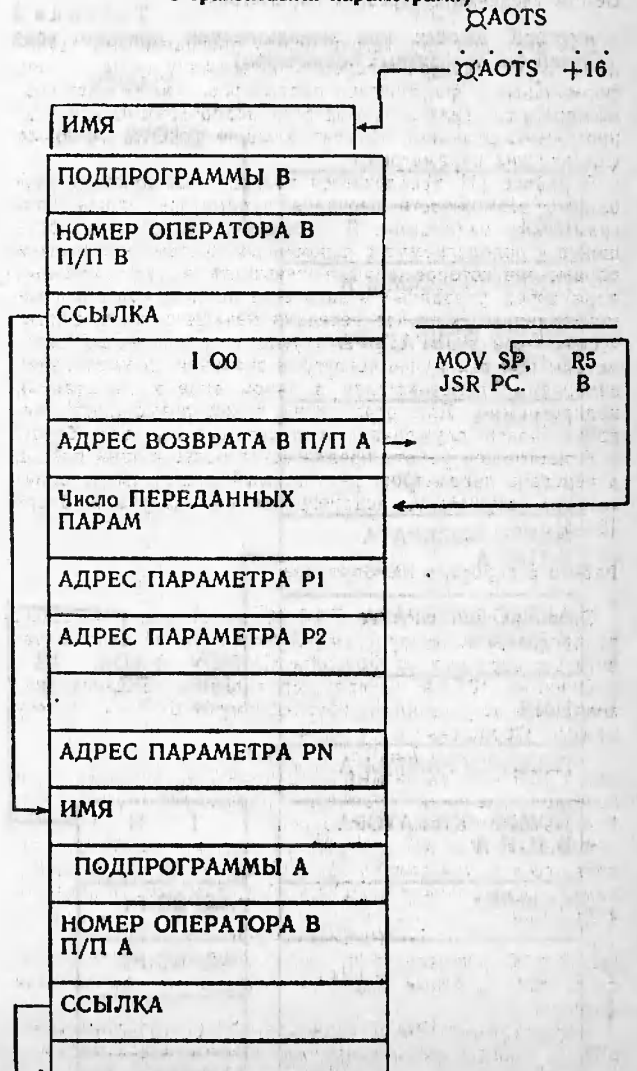
OTIS — служебная программа, организующая связь между подпрограммами. В зависимости от генерируемого объектного кода используются разные точки входа в программу: в коде THR — α OTIS, а в FIS, EIS, EAE — $\alpha\alpha$ OTIS.

Текст программы OTIS, полученный в результате дизассемблирования:

α OTIS::	PSECT OTIS	α 1			
	.GLOBL	α AOTS			
	MOV	α 0,R1			: ВХОД В α OTIS
	--2				: УКАЗАТЕЛЬ НА 2-Е СЛОВО
$\alpha\alpha$ OTIS:	CLR	R1			: ВХОД В $\alpha\alpha$ OTIS
	CMP	(SP)+,R0			: ПРОДВИНУЛИ УКАЗАТЕЛЬ СТЕКА
	MOV	(R4)+,R0			: АДРЕС ВХОДА В П/П
	MOV	α AOTS,R3			: АДРЕС α AOTS
	MOVB	164(R3),-(SP)			: СТАРЫЙ ТИП КОДА (БАЙТ)
	MOV	16(R3),-(SP)			: СТАРОЕ SP
	MOV	(R3),-(SP)			: НОМЕР ОПЕРАТОРА→СТЕК
	MOV	(R0)+,-(SP)			: ИМЯ
	MOV	(R0)+,-(SP)			: ПОДПРОГРАММА→СТЕК
	CLR	(R3)			: ОЧИСТКА α AOTS

Таблица 2

Структура данных при использовании прямого кода (случай с транзитными параметрами)



```

MOV SP, 16(R3) ; УКАЗАТЕЛЬ СТЕКА
MOV R1, -(SP) ; ТИП КОДА (FIS ИЛИ THR)
MOV (R5)+, R1 ; ЧИСЛО ПАРАМЕТРОВ
BEQ M1 ; ПЕРЕХОД, ЕСЛИ ПАРАМЕТРОВ, 0

M2: MOV (R0)+, R2 ; ПЕ
BEQ M1 ; PE
MOV (R0), R0 ; СЫЛ
DEC (R5)+, (R0)+ ; КА
BEQ M1 ; ПА
DEC R1 ; РА
DEC R2 ; МЕ
BNE M2 ; ТОВ

M1: MOV (SP)+, 164(R3) ; ПРОВЕРКА ТИПА КОДА
BEQ M3 ; ПЕРЕХОД, ЕСЛИ 0 (Т.Е. FIS)
JMP (R4)+ ; УХОД, ЕСЛИ THR
M3: CALL (R4) ; УХОД, ЕСЛИ FIS
CMP (SP)+, (SP)+ ; ПРОПУСТИЛИ ИМЯ
MOV (AOTS), R4 ; АДРЕС AOTS
MOV (SP)+, (R4) ; НОМЕР ОПЕРАТОРА →
; → AOTS
MOV (SP)+, 16(R4) ; ССЫЛКА → AOTS
MOV (SP)+, 164(R4) ; ТИП КОДА → AOTS
RETURN ; ВОЗВРАТ В ВЫЗЫВАЮЩУЮ П/П
.END ; КОНЕЦ

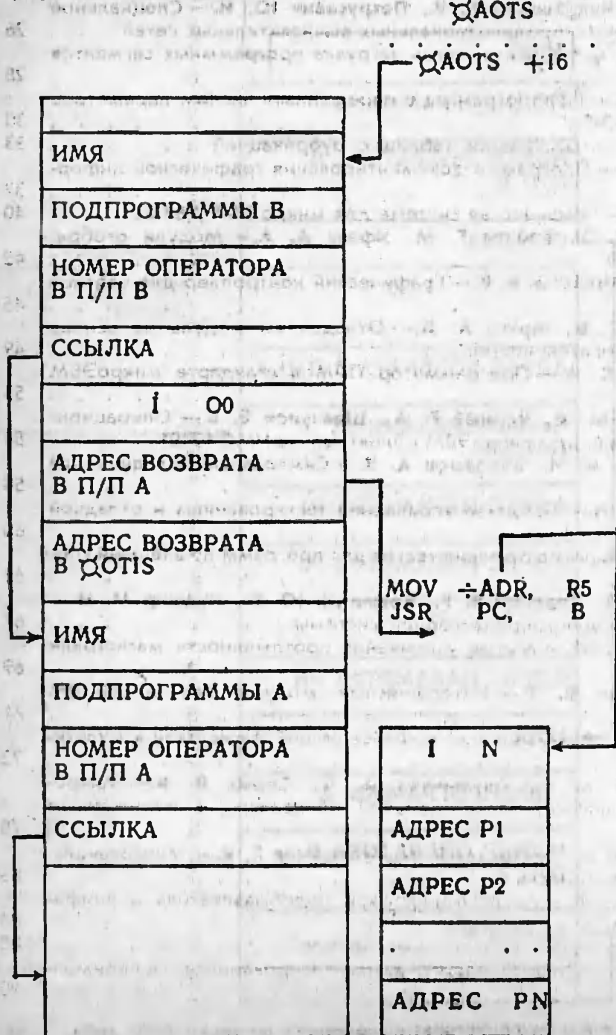
```

```

.TITLE JNOW
.GLOBL JNOW, AOTS ; ОПРЕДЕЛЕНИЕ AOTS КАК ГЛОБАЛЬНОЙ
JNOW: MOV AOTS, R1 ; АДРЕС AOTS
MOV 16(R1), R1 ; УКАЗАТЕЛЬ НАЧАЛА ЦЕПОЧКИ В СТЕКЕ
TSTB 10(R1) ; ПРОВЕРКА КОДА (FIS, THR)
BEQ FIS ; УХОД, ЕСЛИ КОД FIS
THR: ADD #20, R1 ; АДРЕС НАЧАЛА СПИСКА ПАРАМЕТРОВ

```

Таблица 3
Структура данных при использовании прямого кода (случай без транзитных параметров)



Обработка переменного числа параметров

Общая схема построения подпрограмм, обрабатывающих переменное число параметров, следующая:
 использование оператора SUBROUTINE без списка параметров;
 определение числа переданных фактических параметров (подпрограмма JNOW);
 определение адресов фактических параметров (подпрограмма JADPAR);
 выборка значений фактических параметров (подпрограммы IADDR, MOVVAL);
 необходимая обработка переданных параметров;
 засылка необходимых значений в соответствующие выходные фактические параметры (подпрограммы IADDR, MOVVAL);
 выход из подпрограммы с помощью RETURN.
 Данная схема может варьироваться, в частности не исключается использование формальных параметров (например, какие-то из параметров могут быть обязательными).

Функция JNOW позволяет определить число фактических параметров, переданных данной программной единице. Форма обращения:

$$I = JNOW()$$

где I — имя переменной, которой присваивается число переданных параметров. Ошибки при выполнении отсутствуют.

```

M: MOV (R1)+, R0 ; ЗАСЫЛКА ЗНАЧЕНИЯ ФУНКЦИИ
RETURN ; ВОЗВРАТ В ВЫЗЫВАЮЩУЮ П/П
FIS: ADD #12, R1 ; АДРЕС АДРЕСА ВОЗВРАТА В П/П А
MOV (R1)+, R2 ; АДРЕС ВОЗВРАТА В П/П А
MOV -6(R2), R0 ; ТО, ЧТО СТОИТ ПЕРЕД JSR PC, B
CMP R0, +10605 ; СПИСОК АДРЕСОВ ПАРАМЕТРОВ В СТЕКЕ?
BEQ M ; ПЕРЕХОД, ЕСЛИ НЕТ
MOV (R0), R0 ; ЗАСЫЛКА ЗНАЧЕНИЯ ФУНКЦИИ
RETURN ; ВОЗВРАТ В ВЫЗЫВАЮЩУЮ П/П
.END ; КОНЕЦ

```

Функция JADPAR определяет абсолютный адрес фактического параметра в виде двухбайтового целого числа по его порядковому номеру. Форма обращения:

$$I = JADPAR(K),$$

где I — абсолютный адрес параметра; K — порядковый номер фактического параметра.

Если в результате выполнения функции JADPAR, переменной I присвоится значение -1, то соответствующий фактический параметр будет опущен, например в CALL A(P1, P3) опущен параметр номер 2.

```

.TITLE JADPAR
.GLOBL JADPAR, AOTS ; ОПРЕДЕЛЕНИЕ AOTS КАК ГЛОБАЛЬНОЙ
JADPAR: MOV #2(R5), R3 ; НОМЕР ПАРАМЕТРА В R3
ASL R3 ; УМНОЖИЛИ НА 2
MOV AOTS, R1 ; АДРЕС AOTS
MOV 16(R1), R1 ; УКАЗАТЕЛЬ НАЧАЛА ЦЕПОЧКИ В СТЕКЕ
TSTB 10(R1) ; ПРОВЕРКА КОДА (FIS, THR)
BEQ FIS ; УХОД, ЕСЛИ КОД FIS
THR: ADD #20, R1 ; АДРЕС НАЧАЛА СПИСКА ПАРАМЕТРОВ
M: ADD R3, R1 ; АДРЕС N-го ПАРАМЕТРА
MOV (R1)+, R0 ; ЗАСЫЛКА ЗНАЧЕНИЯ ФУНКЦИИ
RETURN ; ВОЗВРАТ В ВЫЗЫВАЮЩУЮ П/П
FIS: ADD #12, R1 ; АДРЕС АДРЕСА ВОЗВРАТА В П/П А
MOV (R1)+, R2 ; АДРЕС ВОЗВРАТА В П/П А
MOV -6(R2), R0 ; ТО, ЧТО СТОИТ ПЕРЕД JSR PC, B
CMP R0, #10605 ; СПИСОК АДРЕСОВ ПАРАМЕТРОВ В СТЕКЕ?
BEQ M ; ПЕРЕХОД, ЕСЛИ НЕТ

```

```

ADD R3 R1 ; АДРЕС N-ГО ПАРАМЕТРА
MOV (R0), R0 ; ЗАСЫЛКА ЗНАЧЕНИЯ
                ФУНКЦИИ
RETURN ; ВОЗВРАТ В ВЫЗЫВАЮ-
        ЩЮ П/П
.END ; КОНЕЦ

```

Подпрограмма **MOVVAL** пересылает значение, источник и приемник которого заданы абсолютными адресами. Форма обращения:

```
CALL MOVVAL (J1, L, JP),
```

где **J1** — абсолютный адрес источника пересылаемого значения; **L** — длина пересылаемого значения в байтах; **JP** — абсолютный адрес приемника значения.

При использовании этой подпрограммы могут возникнуть ошибки. Например, обращение

```
CALL MOVVAL (IADDR(1), 2, IADDR(2))
```

приведет к изменению в данной программной единице значения целой константы 2 на 1, и во всех дальнейших вычислениях будет использоваться новое значение константы — 2.

```

TITLE MOVVAL
MOVVAL: MOV C2(R5), R0 ; АДРЕС J1 В R0
MOV C4(R5), R1 ; L В R1
MOV C6(R5), R2 ; АДРЕС JP В R2
1x: MOVB (R0)+, (R2)+ ; ПЕРЕСЫЛКА БАЙТА
DEC R1 ; УМЕНЬШИЛИ L НА 1
BNE IX ; ПЕРЕХОД, ЕСЛИ НА 0
RETURN ; ВОЗВРАТ В ВЫЗЫВАЮ-
        ЩЮ П/П
.END ; КОНЕЦ

```

В качестве примера рассмотрим подпрограмму, выполняющую поэлементное суммирование произвольного числа массивов типа **REAL** с одинаковым числом элементов. Обращение к подпрограмме:

```
CALL MATSUM (L, A, B1, B2, ..., BN),
```

где **L** — число элементов в каждом массиве; **A** — массив, в который помещается результат поэлементного суммирования ($A+B1+B2+\dots+BN$, $B1, B2, \dots, BN$) — суммируемые массивы типа **REAL**.

Данный программный комплекс проверялся во всех версиях системы **РАФОС**: **РАФОС-1**, **РАФОС-2**, систем **NTS**, **ФОДОС**:

```

C SUBROUTINE MATSUM
C ОПРОС И ПРОВЕРКА ЧИСЛА ФАКТИЧЕСКИХ ПАРАМЕТРОВ
C N-JHOW ( )
C IF (N.LE.2) RETURN
C
C ОПРОС И ПРОВЕРКА ДЛИНЫ МАТРИЦ
C
C CALL MOVVAL (JADPAR (1), 2, IADDR (L1))
C IF (L.LT.1) RETURN
C
C СУММИРОВАНИЕ ЭЛЕМЕНТОВ МАТРИЦ
C
C DO 20 K=1, L
C S=0
C DO 10J=3,N
C CALL MOVVAL (JADPAR (J)+ 4*(K-1), 4, IADDR (SB))
C S=S+SB
10 C ANTINUE
C
C ЗАСЫЛКА ОЧЕРЕДНОГО ЭЛЕМЕНТА МАТРИЦЫ A
C
C CALL MOVVAL (IADDR (S), 4, JADPAR (2) +4*(K-1))
C CONTINUE
C RETURN
C END

```

Телефон 56-14-28, Горький

ЛИТЕРАТУРА

- Gujar U. G., Fellow D. M. Fortran Routines with Optional Arguments.— Software // Proctice and Experience.— 1981.— Vol. 11, N 2.— P. 187—193.
- Операционная система **СМ ЭВМ РАФОС**:
2 Микропроцессорные средства № 2

Справочник / Л. И. Валикова, Г. В. Вигдорчик, А. Ю. Воробьев, А. А. Лукин; Под общ. ред. В. П. Семика.— М.: Финансы и статистика, 1984.— 207 с.

- Вигдорчик Г. В., Воробьев А. Ю., Праченко В. Д. Программирование на ассемблере для **СМ ЭВМ**.— М.: Финансы и статистика, 1983.
- Сичановский Дж. У. Программирование на **Макро-11** и организация **PDP-11**.— М.: Радио и связь, 1985.

Статья поступила 1.11.88

УДК 681.3.06

И. Д. Ашкенази

ПРОГРАММА РАСПЕЧАТКИ ТАБЛИЦ С РУБРИКАЦИЕЙ

Общие сведения. Реляционные (табличные) базы данных — наиболее универсальные для хранения любого типа информации и простые для малоподготовленных пользователей. **РБД-микро*** — самая популярная среди баз данных, реализованных в рамках **ОС RT-11 (РАФОС, ФОДОС)** для микроЭВМ типа «Электроника 60», **ДВК2, ДВК3, «Электроника 85»** и мини-ЭВМ типа **СМ4**.

Программа предназначена для создания иерархически организованных документов на основе **РБД-микро**. Эти документы имеют вид таблиц с рубрикацией, содержащих информацию как о самих объектах, так и о частях, разделах, подразделах, к которым относятся объекты. Эта информация вносится в таблицы в виде внутренних подзаголовков.

ОФАП МЭП СССР КАТАЛОГ ПРОГРАММНЫХ СРЕДСТВ ОФАП С. 1

ШИФР	НАЗВАНИЕ	КРАТКОЕ ОПИСАНИЕ	ОБЪЕМ	ГДЕ ИЛИ КЕМ РАЗРАБОТАНО
------	----------	------------------	-------	-------------------------

1. ПРОГРАММНЫЕ СРЕДСТВА ПЕРСОНАЛЬНЫХ ЭВМ

1.1. ПРОГРАММНЫЕ СРЕДСТВА ОС RT-11 1.1.1. БАЗЫ ДАННЫХ

1.1.1.1.	РБД	РЕЛЯЦИОННАЯ СУБД С УДОБНЫМИ ЭКРАННЫМ ИНТЕРФЕЙСОМ	1000 Кбайт	г. КИЕВ
1.1.1.2.	DBSMNJ	РЕЛЯЦИОННАЯ СУБД С РАЗВИТЫМИ СРЕДСТВАМИ ПОИСКА И СОРТИРОВКИ	2000 Кбайт	Фирма DECUS

Входные данные. Необходимая для создания документа информация хранится в двух связанных между собой базах данных: «Объекты» и «Классификатор». «Объекты» содержат табличные данные, «Классификатор» — наименования частей, разделов, подразделов и т. п. Связующим столбцом обеих баз является классификационный код, который, с одной стороны, ставится в соответствие каждому объекту, с другой, — каждому наименованию рубрики.

Классификационный код — набор (1...6 значений) целых чисел, разделенных символом «.» (точка). Эти числа — номера частей, разделов, подразделов и т. д., по которым классифицируются объекты. Глубина деления на разделы, подразделы и т. д. может колебаться в различных частях таблицы от 1 до 5 уровней. В базе «Данные» классификационный код имеет еще один, более глубокий, внутренний уровень классификации, не находящийся отражения в названиях рубрик. Изменение этого уровня позволяет регулировать вывод на печать столбцов документа, имеющих признак повторяемости (см. ниже описание шаблона документа). Классификация от внешнего уровня к внутреннему меняется слева направо. Записи баз данных предполагаются лексико-графически упорядоченными относительно классификационного кода.

В некоторых документах какие-либо из уровней классификации объектов временно оказываются неизвестными (не путать с несуществующими!). В этом случае в классификационном коде объекта неизвестный код подраздела вместе с разделителем-точкой заменяется просто точкой. Программа вынесет объект в отдельную подгруппу, не предписывая ей заголовка. Примеры классификационных кодов: 2.5.1.15., 3.1., 2..1.

Для увеличения скорости работы программы и облегчения поиска записи, с которой начинается документ, вводятся ограничения: столбец «классификационный код» должен быть первым в базе данных, вторым — столбец, содержащий символическую информацию, индивидуальную для каждой записи базы. Они классифицируются по четвертому типу. Для корректного переноса не следует вносить информацию в базу, разрывая слова между строками.

Приметы баз данных «Данные» и «Классификатор»:

НО-МЕР	ШИФР	НАЗ-ВАНИЕ	КРАТКОЕ ОПИСАНИЕ	ОБЪЕМ	ГДЕ ИЛИ КЕМ РАЗРАБОТАНО
0	1	2	3	4	5

1	1.1.1.1.	РБД	РЕЛЯЦИОННАЯ СУБД С УДОБНЫМ ЭКРАННЫМ ИНТЕРФЕЙСОМ	1000 Кбайт	г. КИЕВ
2	1.1.1.2.	DBSMNJ	РЕЛЯЦИОННАЯ СУБД С РАЗВИТЫМИ СРЕДСТВАМИ ПОИСКА И СОРТИРОВКИ	2000 Кбайт	ФИР-МА DECUS

НОМЕР	ШИФР ПРОГРАММЫ	НАИМЕНОВАНИЕ РУБРИКИ
0	1	2
1	1.	ПРОГРАММНЫЕ СРЕДСТВА ПЕРСОНАЛЬНЫХ ЭВМ
2	1.1.	ПРОГРАММНЫЕ СРЕДСТВА ОС RT-11
3	1.1.1.	БАЗЫ ДАННЫХ

Возможности программы. Программа печати таблиц с рубрикой позволяет создавать большой класс документов в точном соответствии с требованиями ГОСТ 1.5 от 1985 г.: печатать разнообразные выходные формы, задаваемые файлами шаблона документов; выводить документ на печатающее устройство, терминал или диск; регулировать ширину полей выходного документа; печатать

документ в одну или (если ширина документа меньше 76 позиций) две страницы; располагать стандартную страницу вдоль или поперек листа; нумеровать страницы документа, начиная с произвольно заданного номера, или отказаться от нумерации; нумеровать выводимые на печать записи, начиная с произвольного номера, или отказаться от нумерации; печатать документ, начиная с любой записи, входящей в базу «Данные» и имеющей отличие от других значение (ния записи) второго реквизита, или с одного заданного номера записи по другой; печатать любое требуемое число страниц документа; печатать документ, используя последовательно различные базы данных и классификаторы, в частности находящиеся на сменяемых магнитных носителях; грамматически корректно переносить текст со строки на строку; переносить текст на следующую строку по неотображаемому символу X ; вставлять заголовок документа в шапку первой страницы; производить записи на последней странице документа, позволяющие окончательно оформить его.

Благодаря дополнительным возможностям, предоставляемым программой, окончательный вид печати выгодно отличается от осуществляемого программой СУБД РБД-микро RETRDB, в том числе и от печати по шаблону документа. Принципиальным отличием, помимо печати рубрик, является корректный грамматический перенос слов, единиц измерения, математических формул. По-другому печатаются числовые данные: числа выравниваются по левому краю и обрезаются строго по формату, что обеспечивает сохранение вида таблицы.

Для обеспечения соответствия с ГОСТ 1.5 от 1985 г. в программу введены следующие соглашения: номера нечетных страниц печатаются в правом верхнем углу листа, четных — в левом; вводится признак повторяемости значения в столбце таблицы, т. е. при наличии в шаблоне документа признака повторяемости для столбца таблицы в зависимости от вида этого признака повторяющиеся значения в столбце либо опускаются, либо второе из них заменяется символами «То же», а последующие — кавычками (см. ниже).

Шаблон документа. Все разнообразие выходных форм, печатаемых программой, задается с помощью файлов шаблонов документов, которые можно создать любым стандартным текстовым экраным редактором.

Примечание. Редактор шаблонов документов EDDOC, входящий в СУБД РБД-микро, нестандартный. Текстовый файл, создаваемый редактором EDDOC, содержит дополнительные неотображаемые служебные символы. Признаком конца строки в этом файле служит символ «PC» код (12), в то время как конца строки стандартного текстового файла — «BK», «PC» ((15), (12)). Для преобразования текстового файла, создаваемого редактором EDDOC, в стандартный следует обработать его программой RETRDB (команда «печать документа»).

В качестве редактора, создающего документы произвольной ширины, рекомендуется широкоформатная версия EDIK (версии 2.7 и выше), позволяющая проследить с помощью ключа /A наличие посторонних символов в файле шаблона. При ширине документа, не превышающей 80 символов, в работе с дисплеем 151Э-00—13 наиболее удобен редактор ED, входящий в систему РТК-микро.

Файл шаблона документа состоит из трех независимых частей: шапки таблицы; командной строки, следующей за шапкой таблицы (признаком конца шапки таблицы и начала командной строки служит символ «.», знаком конца командной строки — «.»); комментария, предназначенного для облегчения создания и идентификации шаблона.

Для экономии оперативной памяти на шапку таблицы и выходной документ накладываются следующие ограничения: ширина шапки и ширина документа не должны превышать 158 символов; шапка таблицы в файле шаблона документа не должна содержать верхнюю, нижнюю и правую обрамляющую черты. Они дорисовываются программно. Левая обрамляющая черта шапки таблицы должна присутствовать обязательно во избежание ошибков по вине исполняющей системы PASCAL-ДЕК; объем шапки таблицы с учетом описанных требований не

УДК 681.3.06

А. А. Жуков, И. Ю. Паскаль

БЕЙСИК/РАФОС С ВСТРОЕННЫМИ ГРАФИЧЕСКИМИ ПРОГРАММАМИ

Версия БЕЙСИК/РАФОС создана путем включения ассемблерных программ пользователя в состав транслятора на этапе его генерации [1]. В этом случае работа с ассемблерными программами осуществляется через CALL-интерфейс [2]. С точки зрения пользователя, такой способ добавления программ к БЕЙСИКу выглядит как внесение в состав интерпретатора дополнительных операторов, поскольку ключевое слово CALL при обращении к программам может отсутствовать. В этом случае программа пользователя вызывается по имени с указанием списка передаваемых параметров. Добавление в БЕЙСИК ассемблерных программ графического вывода увеличивает размеры интерпретатора на 3 Кбайт.

При создании ассемблерных программ графического вывода за основу были взяты подпрограммы первого уровня пакета машинной графики ТЕВУС [3]. Все программы * работают в системе координат экрана, определяемой контроллером графического дисплея. Для восьмицветных контроллеров «Электроника 4702» и АЭ.410.10 поле вывода составляет 256×256 точек с началом координат в левом верхнем углу экрана. Цвета кодируются цифрами 0—7.

Одна из программ — DIRECT (N) — устанавливает направление вывода символов, фигур и спрайтов (элементов изображений), которые строятся программами DCHAR, ASCII, РАМКА, SORD, TREY, SPRIT (рис. 1). Точ-

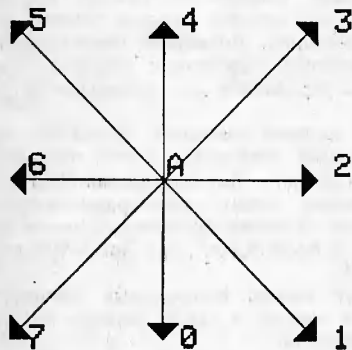


Рис. 1. Возможные направления вывода фигур, символов и спрайтов

ка А соответствует левому верхнему углу поля выводимого символа, фигуры, спрайта. Выбранное направление указывает положение грани, которая обозначает левую границу поля для вывода объекта в направлении 0. Положение букв для некоторых направлений приведено на рис. 2.

БЕЙСИК позволяет контролировать тип и число параметров, передаваемых ассемблерной программе [4]. Используя эти преимущества, можно контролировать типы входных данных и преобразовывать вещественные данные в целые. Преобразование имеет важное значение, так как вещественные и целые записываются в БЕЙСИК/РАФОС по-разному: после целых констант и переменных ставится символ %. Возможность контроля и преобразования из вещественного типа в целый в ассемблерных программах освобождает пользователя от необходимости многократного набора символа % и устраняет ошибки, связанные со смещением типов входных данных.

Если в результате контроля числа входных данных оказывается, что их недостаточно для работы программы, то она не выполняется и на терминал выдается сообщение МАЛО ПАРАМЕТРОВ.

Преобразование типов входных данных выполняется с помощью стандартных подпрограмм IR , RI из библиотеки BSCLLB, входящей в дистрибутив БЕЙСИК. Для вывода сообщений из ассемблерных программ на терминал и работы со строковыми переменными (в программе ASCII) используются внутренние программы интерпретатора БЕЙСИК: ARGER , MSG , FIND .

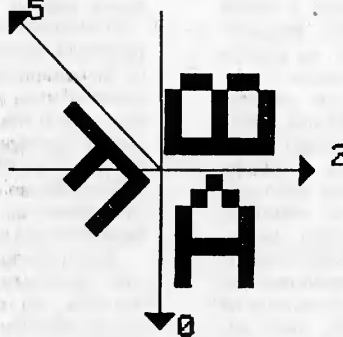


Рис. 2. Положение букв для различных направлений:

положению буквы А соответствует направление 0, В — 2, Г — 5

```
10 ERASE (0) \ POINT(0,0,0)
20 N=255 \ X=0 \ Y=0
30 FOR I=0 TO N STEP 16
40 VECTR(1,X+N,Y+1) \ VECTR(1,X+N-1,Y+N)
50 VECTR(1,X,Y+N-1) \ VECTR(1,X+1,Y)
60 NEXT I
70 DIRECT (0) \ FRMT(2,2)
80 POINT (0,87,119) \ ASCII(4, BASIC )
```

Рис. 3. Программа на БЕЙСИК/РАФОС с графическим выводом

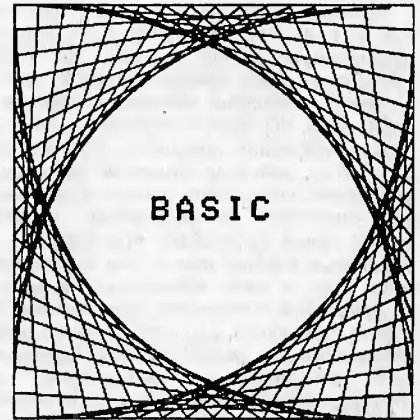


Рис. 4. Результат работы программы, приведенной на рис. 3

БЕЙСИК позволяет работать с графическими программами как в программном режиме, так и в режиме калькулятора, что очень удобно на этапе начального знакомства с машинной графикой. В качестве входных данных графических программ можно использовать кроме переменных и констант еще и выражения.

Текст программы для вывода на графический дисплей тестового изображения приведен на рис. 3, результат работы этой программы — на рис. 4. Программы COLOR и SPRIT позволяют наращивать программные средства графического вывода (разработана программа графического редактора на языке БЕЙСИК, которая существенно упрощает процесс рисования на графическом дисплее). Создана библиотека данных для программы SPRIT и программа на БЕЙСИКе для составления и редактирования спрайтов.

634050, Томск, пр. Ленина, 36. Томский государственный университет, радиотехнический ф-т; тел. 90-97-95

* Программы с комментариями приведены в письме авторов, следующем за статьей (Прим. ред.)

ЛИТЕРАТУРА

1. Программное обеспечение СМ ЭВМ, операционная система с разделением функций РАФОС. БЕЙСИК. Генерация системы. Руководство системного программиста. Т. 5. Кн. 3. Ч. 1.— 16 с.
2. Доломанов В. Г., Канофьев

ев А. В. Структурные особенности интерпретирующей системы БЕЙСИК/РАФОС // Микропроцессорные средства и системы.— 1987.— № 4.— С. 25—28.

3. Гилев К. А., Луцевич Л. В. Пакет машинной графики ТЕВУС // Автоматизированные системы научных исследований, обучения и управления в вузах // Межвуз. сб. науч.

тр.— Новосибирск, 1985.— С. 110—117.

4. Программное обеспечение СМ ЭВМ, операционная система с разделением функций РАФОС. БЕЙСИК. Руководство программиста. Т. 5. Кн. 3. Ч. 2.— 36 с.

Статья поступила 21.02.82

УДК 681.3.022

А. А. Жуков, И. Ю. Паскаль

ПРОГРАММА ДОКУМЕНТИРОВАНИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Программа GDP (GRAPHIC — DISK — PRINTER), оформленная как стандартная утилита ОС RT-11, позволяет запоминать изображение с графического дисплея в упакованном формате в виде файла на любом носителе, воспроизводить изображение из файла на графическом дисплее и получать распечатку изображения на любом принтере, имеющем режим графической печати (УВВПЧ-30, Роботрон 6329, D-100 и т. п.). Объем программы 5 Кбайт.

Благодаря специальному алгоритму упаковки изображения средней сложности размерами 256×256 точек с трехбитовой кодировкой цвета точка занимает 2...10 Кбайт (в худшем случае 32 Кбайт). Для сравнения отметим, что страница текста на листе стандартного формата составляет 3...5 Кбайт.

Для микроЭВМ типа «Электроника 60» серийно выпускается ряд контроллеров графических дисплеев («Электроника 4702», АЭ.410.10, ЦДР-Э и т. п.), имеющих поле вывода размерами 256×256 точек и позволяющих изображать точки разных цветов. Ввод-вывод программ на эти контроллеры осуществляется через драйвер GD. Дисплей представляется в данном случае в виде устройства прямого доступа блочной бескаталоговой структуры, каждый байт которого несет информацию о цвете отдельной точки, а номер блока и порядковый номер байта в блоке определяют координаты точки.

Сложнее обеспечить совместимость с печатающими устройствами, поскольку способы реализации графического режима в различных типах принтеров существенно различаются. Разработаны три версии программы, каждая из которых может работать только с одним типом принтера: УВВПЧ-30, Роботрон 6329 и D-100. Исходный текст программы на языке ассемблера содержит фрагменты для работы с каждым из указанных устройств, поэтому на конкретный тип принтера она настраивается на этапе трансляции с соответствующим значением параметра генерации.

Вызывается программа GDP аналогично команде монитора COPY в ОС RT11 V5.2:

- GDP GD: FILNAM — запись изображения с дисплея в файл;
 GDP FILNAM GD: — воспроизведение изображения из файла на дисплей;
 GDP FILNAM LP: — распечатка изображения из файла на принтер;
 GDP GD: LP: — распечатка изображения с дисплея на принтер.

Изображение на дисплее может содержать до восьми цветов; для печати каждой точки используется растр 2×2, причем печатаются не все точки раstra. Изменяя сочетания печатаемых точек в растре, можно по-разному изображать различные цвета на распечатке (рис. 1).

Программа предоставляет некоторые дополнительные возможности, реализуемые с помощью ключей:

1. Пауза перед выводом изображения (ключ /W). Этот ключ позволяет создавать простые «слайдофильмы». Предварительно составляется командный файл, в котором каждая команда, сопровождаемая ключом /W, выводит на дисплей один «слайд» из серии рисунков:

```
GDP RIS1 GD: /W
GDP RIS2 GD: /W
GDP RIS3 GD: /W
```

Перед выводом каждого рисунка программа будет ожидать нажатия клавиши (BK) ((CP)) на терминале. давая возможность демонстратору сопроводить рисунок соответствующим пояснением.

Разновидность этой возможности — ключ /W:N с параметром, где N — задержка в секундах перед выводом рисунка (программа ожидает не нажатия клавиши, а просто истечения заданного временного интервала).

Например:

```
GDP RIS1 GD          вывести рис. 1
GDP RIS2 GD: /W:10  через 10 с вывести рис. 2
```

2. Перекраска изображения (ключ /R).

С помощью этого ключа можно изменять цвета воспроизводимого изображения (цвета кодируются цифрами 0...7). Например, команда GDP RIS1 GD/R:1:4 при выводе на дисплей рис. 1 заменит точки красного цвета (код 1) исходного изображения на точки синего цвета (код 4). Этот ключ можно использовать для «закрашивания» отдельных элементов изображения. Так, если цвет фона рисунка черный (код 0), то команда GDP RIS1 GD/R:1:0:2:0:3:0 вызовет элементы рисунка с кодами 4, 5, 6, 7 (синий, фиолетовый, голубой, белый). Элементы с кодами 1, 2, 3 (красный, желтый, зеленый) сольются с фоном.

3. Задание печатных и непечатных точек раstra при выводе на принтер (ключ /S).

Этот ключ позволяет выбрать вид раstra:

```
GDP RIS1 LP: /S:1:1111:2:1010:3:0011! для цвета 1 — растр ::, для цвета 2 — растр :, для цвета 3 — растр..
```

GRAPHIC-DISK-PRINTER V05.02 ТОМСК ТГУ РФФ ЖУКОВ А.А., ПАСКАЛЬ И.Ю.
 LP: *NCOL

■	ЧЕРНЫЙ	- РАСТР: ■
■	КРАСНЫЙ	- РАСТР: ■
■	ЗЕЛЕНый	- РАСТР: ■
■	ЖЕЛТЫЙ	- РАСТР: ■
■	СИНИЙ	- РАСТР: ■
■	ФИОЛЕТОВЫЙ	- РАСТР: ■
■	ГОЛУБОЙ	- РАСТР: ■
□	БЕЛЫЙ	- РАСТР: ■

Рис. 1. Соответствие цветов точкам в растре

В программе предусмотрены три стандартных набора растров, которые можно выбрать с помощью ключа /C:N, где N — номер набора (0...2). Если в команде нет ключей /C и /S, то автоматически устанавливается первый стандартный набор. Ключи /C:3 и /C без параметров выбирают набор, заданный пользователем с помощью ключа /S. Эти ключи также позволяют скрыть отдельные элементы изображения при выводе его на печать. Так, если выбран набор, задаваемый пользователем, а растр для соответствующего цвета не задан, все позиции растра считаются непечатными.

4. Вывод справочной информации (ключ /H).

По этому ключу программа выводит на терминал краткую справку о своих возможностях с описанием ключей.

Кроме того драйвер графического дисплея GD (рис. 2) позволяет выводить изображение на дисплей повернутым на 90, 180, 270° и в зеркальном виде в соответствии с номером устройства GD. Для кодировки цвета используются три бита байта, а оставшиеся пять битов — для маскирования цветов изображения, уже выведенного на дисплей.

```

;**** GD - ДРАЙВЕР ГРАФИЧЕСКОГО ДИСПЛЕЯ ****
.MCALL .DRDEF
.DRDEF GD,360,0,0,177620,170

.DRDEF GD
MOV #14, @GDHCSR+2
MOV GDCC, R3
MOV (R3), R3 ;НОМЕР БЛОКА
SWAB R3
ASL R3 ;АДРЕС НА ДИСПЛЕЕ
MOVB QHUNIT(R5), R4
BIC #17770, R4 ;НОМЕР УСТРОЙСТВА
MOV QHBUFF(R5), R0 ;АДРЕС БУФЕРА
MOV QHWCNT(R5), R1 ;СЧЕТЧИК СЛОВ
BPL READ
WRITE: NEG R1 ;ЗАПИСЬ
ASL R1
LOOPW: MOV R3, R2
BIT #4, R4
BEQ 1H
COMB R2 ;ЗЕРКАЛЬНОЕ ИЗОБРАЖЕНИЕ
BIT #2, R4
BEQ 2H
COM R2 ;ПОВОРОТ НА 180°
BIT #1, R4
BEQ 3H
COMB R2 ;ПОВОРОТ НА 90° (НА 180+90)
SWAB R2
MOV R2, @GDHCSR
MOVB (R0), R2
ASR R2
ASR R2
ASR R2
COM R2
BICB R2, @GDHCSR+3 ;УБРАТЬ НЕМАСКИРУЕМЫЙ ЦВЕТ
BIS (R0)+, @GDHCSR+3; ЗАПИСАТЬ ЦВЕТ
INC R3
SOB R1, LOOPW
BR FIN
READ: ASL R1 ;ЧТЕНИЕ
TST END
BEQ LOOPR
MOV #EOFH, @-(R5)
CLR END
BR FIN
LOOPR: MOV R3, @GDHCSR
MOVB @GDHCSR+3, (R0)
BICB #370, (R0)+
INC R3
SOB R1, LOOPR
TST R3
BNE FIN
INC END
GDINT:
FIN: .DRFIN GD
END: .WORD 0
.DREND GD
.END

```

Рис. 2. Драйвер графического дисплея для контроллера «Электроника 4702»

634050, Томск, пр. Ленина, 36. Томский государственный университет, радиофизический ф-т; тел. 90-97-95

Статья поступила 21.02.89

ПИСЬМО В РЕДАКЦИЮ

Уважаемая редакция!

За время подготовки статей к публикации нами завершена работа по созданию варианта БЕЙСИКА для ДВКЗ. От описанного в статье «БЕЙСИК / РАФОС со встроенными графическими программами» он отличается двумя особенностями, связанными с тем, что в ДВКЗ графический и алфавитно-цифровой экраны совмещены. В связи с этим введены дополнительная функция GRAF с двумя параметрами, позволяющая включать-выключать отдельно отображение графического и алфавитно-цифрового экранов дисплея ДВКЗ, и автоматическое включение отображения алфавитно-цифрового экрана при выходе из программы независимо от типа выхода: нормальный, по ошибке, по прерыванию с клавиатуры. Все остальные графические функции имеют тот же формат, что и в БЕЙСИКЕ, описанном в статье (для совместимости программ). При разработке графических процедур основным требованием к ним было максимальное быстродействие и удобство обращения для пользователя, имеющего минимальный опыт работы с БЕЙСИКом.

В вашем журнале (1988, № 4) опубликована статья Кулаичева А. П. с описанием одного из возможных вариантов графического БЕЙСИКА для ДВКЗ. Поэтому мы сочли необходимым провести небольшой сравнительный анализ двух реализаций БЕЙСИКА (нашего и опубликованного).

К несомненным достоинствам опубликованного варианта следует отнести следующие возможности: выделение графического окна, что особенно важно при одновременном выводе графической и текстовой информации; заполнение штриховкой замкнутого контура; получение на принтере копии с графического экрана.

Недостаток данного БЕЙСИКА в том, что он плохо ориентирован на конечного пользователя, особенно слабо подготовленного. Некоторые функции (MODE, GCOL, FILL) требуют аргументов, смысл которых сложен для понимания (попробуйте запомнить, какой режим экрана соответствует какому аргументу команды MODE). Судя по листингам приведенным в статье программ параметры графических подпрограмм должны быть целыми. Обычно при написании программ используются операнды вещественного типа, а в данной реализации подстановка таких операндов в качестве параметров (по незнанию или по ошибке) приведет к непредсказуемым результатам. В предлагаемой версии БЕЙСИКА есть средства, позволяющие встроенной подпрограмме определить тип параметра и при необходимости преобразовать его в нужный тип (интерфейс между БЕЙСИКом и графическими подпрограммами занимает порядка 50 слов).

Кроме того, в статье ничего не сказано о ситуации, когда в ходе выполнения программы запрещен текстовый вывод, а программа завершается по ошибке или прерывается оператором. Если при этом не работала команда MODE, разрешающая текстовый вывод, то ситуация может оказаться труднопреодолимой, особенно для начинающего программиста.

Мы считаем, что в программах, ориентированных на массовое использование, всегда целесообразно предусматривать возможность получения NLP-текста (краткой справочной информации). При желании этот текст можно вывести в оверлейную область, что практически не уменьшит объем свободной памяти, но значительно облегчит работу с программой.

В команду MODE целесообразно ввести два параметра, один из которых будет отвечать за разрешение-запрещение текстового, другой — графического высвдов; цвет фона желательно устанавливать отдельной командой; команду WINDOW дополнить формированием окна в мировых координатах (необязательно положительных целых); команду RPING — возможностью изменять масштаб символов и направление строк (хотя бы вертикальное и горизонтальное).

Отметим, что команда CLS — приятное, но необязательное дополнение (то же самое можно сделать средствами БЕЙСИКА, выдав на печать строку SHR\$(27)+'H'+CNR\$(27)+'J'); команды CIRCLE и ELLIPS (судя по их параметрам) долго выполняются и плохо сопрягаются с другими элементами рисунка (гораздо полезнее было бы строить дуги по трем точкам); команда DUMP описана очень кратко — без указания, на какие типы принтеров она рассчитана (скорее всего не каждый сможет ею воспользоваться из-за отсутствия нужного принтера). Все это отнюдь не умаляет достоинств данной реализации, которая предоставляет достаточный набор графических средств для широкого класса задач.

В нашем варианте БЕЙСИКА из-за особенностей преобразования координат текущей точки при отрисовке изображений (чтобы обеспечить максимальное быстродействие) некоторые алгоритмы (задание окна, штриховка контура, построение дуги) отсутствуют. Кроме того, поскольку различные принтеры по-разному реализуют графический режим, пришлось отказаться от включения в БЕЙСИК подпрограммы получения твердой копии графического экрана.

Эти недостатки частично восполняются наличием функций COLOR и POINT, позволяющих на уровне пользовательских программ реализовать алгоритмы штриховки контура, построения дуги, вывода графической информации в файл и на принтер (хотя быстроедействие этих программ не всегда удовлетворительное). Правда, опыт показывает, что далеко не каждый рисунок «заслуживает» вывода на печать, поэтому, если такая необходимость возникнет, можно выйти из БЕЙСИКА и воспользоваться другой программой (например, описанной в нашей статье утилитой GDP) для вывода изображения в файл или на принтер.

К основным достоинствам нашего варианта можно отнести возможность использования в качестве параметров операндов любого типа; большой набор стандартных графических примитивов (точка, отрезок, прямоугольник, треугольник, рамка, окружность, круг); автоматическое разрешение текстового вывода при выходе из программы; возможность получения краткой справки о графических подпрограммах. Кроме того, подпрограмма вывода спрайтов позволяет масштабировать символы и спрайты, поворачивать их на определенный угол, что дает возможность использовать БЕЙСИК не только для научно-технических расчетов, но и для написания игровых, демонстрационных и обучающих программ. Последнее слово о достоинствах, недостатках, удобстве различных реализаций БЕЙСИКА бесспорно должно принадлежать пользователям, и мы были бы рады выслушать мнения как опытных, так и начинающих программистов.

С уважением А. Жуков, И. Паскаль

P. S. Если какие-то моменты, касающиеся нашего варианта БЕЙСИКА для ДВКЗ, окажутся непонятными, мы сочли возможным приложить к письму его краткое описание.

Графический БЕЙСИК для ДВКЗ.

Графический БЕЙСИК представляет собой БЕЙСИК / РАФОС со встроенными графическими подпрограммами, которые позволяют:

- включать-выключать отображение графического и (или) алфавитно-цифрового экранов дисплея;
- закрашивать графический экран заданным цветом;
- выводить графические примитивы: точку, отрезок, прямоугольник, треугольник, рамку, окружность, круг; отдельные символы и строки ASCII; спрайты (произвольные изображения 16×16 точек);

- изменять размеры символов и спрайтов, ориентировать их под разными углами (с шагом 45°);
- читать цвет заданной точки на экране.

Для отрисовки графических примитивов используются быстрые линейные алгоритмы. Все подпрограммы могут

вызываться как из программы, так и непосредственно в режиме команд. Использование ключевого слова CALL необязательно. Параметрами подпрограмм могут быть константы, переменные и выражения как целого, так и вещественного типов. Применение в качестве параметров целых констант, переменных и выражений ускоряет вызов подпрограмм.

Встроенные подпрограммы

HELP	— выводит краткую информацию о графических подпрограммах;
GRAF (K)	— включает (K=1) и выключает (K=0) отображение графического экрана;
GRAF (K, K1)	— аналогична предыдущей. K1 (1/0) включает-выключает отображение алфавитно-цифрового экрана.

Если в ходе выполнения программы алфавитно-цифровой экран выключен, то при выходе из программы (как при нормальном завершении, так и при аварийном по `CV / C`) он в любом случае будет включен автоматически. Соответственно, в режиме команд алфавитно-цифровой экран не выключается при любом значении K1.

В следующих подпрограммах `C` означает цвет (0 — черный, 1 — белый). Для указания цвета разрешается использовать любые числа, не равные 0 или 1, но при этом учитывается только младший двоичный разряд.

ERASE (C)	— закрашка экрана одним цветом;
POINT (C,X,Y)	— вывод точки с координатами X,Y;
VECT1 (C,X0,Y0,X,Y)	— вывод отрезка, соединяющего точки с координатами X0,Y0 и X,Y;
VECTR (C,X,Y)	— аналогична VECT1; координаты начальной точки — конец предыдущего отрезка или точки — конец предыдущего отрезка; выводимая подпрограммой POINT;
	— прямоугольник размерами LXXLY, с координатами левого верхнего угла X, Y;
	— рамка (параметры такие же, как в SQRD);
	— равнобедренный прямоугольный треугольник с координатами вершины X, Y и стороной L;
OCR(C,X,Y,R)	— окружность с координатами центра X и радиусом R;
CIRCL (C,X,Y,R)	— круг (параметры такие же, как в OCR);
DCHAR (C,I)	— вывод символа ASCII; I — код символа, координаты левого верхнего угла задаются в предыдущей операции: ERASE — 0, POINT, VECT1, VECTR — X,Y, DCHAR, ASCII — правый верхний угол последнего символа;
ASCII (C,'TEXT')	— вывод строки текста 'TEXT' (координаты левого верхнего угла первого символа, как в DCNAR);
sprit (CA % (%))	— вывод спрайта. A % (0) — целый (обязательно массив из 16 элементов); каждый элемент массива соответствует одной строке точек изображения 16×16; отдельный бит означает одну точку в строке спрайта; крайние левой точке строки соответствует старший бит элемента массива; положение спрайта на экране определяется так же, как для символа;
FRMT (MX,MY)	— масштаб символов и спрайтов. Первоначальный размер символа 8×8, спрайта 16×16. Подпрограмма увеличивает их размеры по X в MX раз, по Y в MY раз;
DIRECT (D)	— ориентация символов (DCHAR), строк (ASCII), спрайтов (SPRIT), прямоугольников (SQRD), рамки (RAMKA) и треугольников (TREY). D — одно из восьми направлений;
COLOR (X,Y,C)	— чтение цвета точки с координатами X, Y; C — выходной параметр (не может быть константой или выражением).

Телефон 90-97-95, Томск

ГРАФИЧЕСКАЯ СИСТЕМА ДЛЯ МИКРОЭВМ ДВКЗ

Предлагаемая графическая система предназначена для использования в научных расчетах и системах автоматизации научного эксперимента и обеспечивает вывод различных данных на устройства графического вывода (дисплеи, графопостроители и т. п.) под управлением программ пользователя, написанной на языках Фортран, Паскаль, МАКРО 11. Графический монитор — резидентная часть графической системы — занимает 1,7 Кслов памяти комплекса и функционирует как часть ОС РАФОС (ФОДОС, ОС ДВК, SJ-, FB-мониторы). В FB-мониторе два выполняющихся задания могут одновременно использовать разные устройства графического вывода без перезагрузки графической системы. Инициализируется система по команде GP.SYSON.

Структура графической системы имеет четыре уровня: программа пользователя, графические сегменты, графический монитор, драйвер графических устройств.

В текст программы пользователя включаются операторы графического языка, которые представляют собой вызовы функций графического монитора. Этот язык может быть расширен за счет написания программистом графических сегментов, т. е. программных модулей, содержащих фрагменты программ высокого уровня для реализации стандартных графических операций с помощью многократных обращений к монитору. Программист может использовать при написании сегментов языка МАКРО 11 и Паскаль. Особенность выполнения программ, размещенных в сегментах, — их динамическая загрузка в память в момент первого обращения к любой подпрограмме данного сегмента и возможность освобождения занимаемой сегментом памяти по команде из прикладной программы. Механизм распределения памяти определяется исполняющей системой языка, на котором написана прикладная программа. Всего можно сформировать 26 разных сегментов, в каждом из которых размещается до 32 подпрограмм.

Благодаря динамической загрузке сегментов программа пользователя формирует требуемую конфигурацию графической системы в процессе работы и, таким образом, экономит память. В отличие от организации оверлейных модулей в системе РАФОС механизм распределения памяти для сегментов позволяет использовать память, освобождающуюся после удаления сегмента, не только для другого сегмента (как в случае оверлеев РАФОС), но и для данных программы.

Для облегчения написания сегментов на языке МАКРО 11 разработана макробибблиотека GP.MAC.MLB, а для программирования их на языке Паскаль — модуль SEGREP.PAS.

Графический монитор. В его функцию входит дешифрирование и выполнение основного набора графических операций (установка режимов, перемещение пера, формирование текстовых строк и т. д.), а также распознавание запросов к графическим сегментам, их загрузка и вызов. Монитор содержит подпрограммы генераторов векторов и символов, которые включают в работу, если устройство вывода не имеет этих возможностей. Информация о параметрах и функциях устройства вывода передается монитору графическим драйвером.

Генератор векторов предназначен для непрерывного выполнения цепочки точек, соединяющих текущую и заданную точки. Он включается в работу при каждом вызове запроса MOVE, независимо от того, поднято перо или опущено. При этом для рисования каждой из генерируемых точек обращаются к драйверу, а для рисования отрезков прямых применен алгоритм Брезенхэма [1]. Если устройство вывода содержит встроенный генератор векторов, то обращаются к драйверу один раз с указанием конечной точки пере-

мещения (начальной точкой считается текущее положение пера).

Работа с генератором векторов аналогична работе с генератором символов. При выполнении запроса STRING к драйверу устройства обращаются многократно для перемещения в точку, координаты которой и состояние пера задаются программой генератора символов. Если устройство само способно изображать символы, программный генератор отключается и обращаются к драйверу один раз.

При работе графической системы с программируемым генератором символов использован принцип динамического размещения в памяти. Наборы символов (шрифты) хранятся на диске в специальных файлах и загружаются в память по запросу программы пользователя. Символы кодируются с помощью специальной программы FONT, обеспечивающей экранный режим редактирования старого или создания нового шрифта. Кодирование символа, точечное в матрице переменного размера, аналогично алфавитно-цифровым дисплеям или устройствам печати. Размер матрицы для одного набора постоянный — до 16×16 символов, но может отличаться для разных наборов. Аналогично удалению сегмента набор символов можно уничтожить, а занимаемую им память вернуть программе.

В многозадачной ОС в каждом задании может быть использовано свое устройство вывода независимо от других заданий и устройств. При этом каждое задание имеет свой набор параметров, однозначно описывающих состояние графической системы и устройства вывода, свои сегменты и свой шрифт.

Графический монитор обеспечивает работу прикладной программы с двумя независимыми системами координат: устройства (физические координаты) и пользователя (математические координаты). Эти два режима переключаются специальным запросом к монитору, после чего во всех командах, в которых в качестве параметров выступают координаты, они преобразуются в соответствующую систему.

В некоторых алгоритмах растровой графики можно считать содержимое буфера регенерации изображения. Однако графопостроители и многие простые дисплеи не имеют такой возможности. Предлагаемая графическая система позволяет организовать вывод графической информации с помощью внутреннего буфера изображения, в котором информация доступна для записи и чтения. После того как режим включен, вся выводимая информация одновременно заносится в буфер в виде черно-белого эквивалента изображения, причем цвет, совпадающий с цветом фона, кодируется 0, а любой не совпадающий — 1.

При работе с устройствами не растровой структуры буфер позволяет реализовать два дополнительных режима.

Режим бланкирования. Если в точке, в которую должно переместиться перо, уже имеется какое-то изображение, отличное от фона, то перо поднимается и новое изображение не накладывается на старое. Таким способом выполняется временной приоритет выводимых изображений по принципу: раньше нарисована — приоритет выше.

Режим псевдовывода. Изображение заносится в буфер, но к устройству вывода обращений нет, т. е. не формируется реальное изображение на бумаге или на экране дисплея. В результате изображение в буфере можно дополнить некоторыми деталями, вывод которых искажает картину, а сами детали необходимы для работы алгоритма рисования некоторого сложного объекта. Этот режим полезен в сочетании с бланкированием для удаления невидимых линий в пространственных изображениях при работе с графопостроителем.

Принцип удаления невидимых линий состоит в упорядочении выводимых элементов рисунка, чтобы каждый последующий элемент находился дальше от зрителя, чем все предыдущие. Далее в буфере организуется экран, состоящий из всех уже нарисованных к данному моменту элементов изображения и, таким образом, накрывающий все последующие элементы. Новый элемент изображения выводится в те участки рисунка, которые не закрыты экраном,

после чего экран модифицируется. Если выводимый объект представлен контуром, а на самом деле он сплошной и должен закрывать собой задний план рисунка, нужно закрасить его в режиме псевдовывода (рис. 1).

При работе с растровым дисплеем в принципе можно упростить алгоритм удаления невидимых линий, изменив порядок вывода элементов изображения на обратный и закрасив контурные элементы цветом фона. В этом случае буфер не понадобится, но такой модифицированный алгоритм не будет работать на устройствах типа графопостроителя.

Содержимое ячейки буфера, соответствующей текущей позиции пера, программа может прочитать с помощью запроса RDCOLR. Если устройство вывода само реализует возможность чтения содержимого буфера регенерации (информация об этом передается монитору из драйвера), то запрос будет работать без включения буфера, но не будет работать в режиме бланкирования и псевдовывода.

Некоторые микроЭВМ оборудованы графическим дисплеем, аппаратно совмещенным с основным системным терминалом, причем в одно и то же время можно получить на его экране или только алфавитно-цифровую или только графическую информацию. Пример такой конфигурации — комплекс ДВКЗ. Для повышения эффективности работы оператора с таким дисплеем графическая система предоставляет возможность управления его работой с помощью символов, вводимых с клавиатуры:

- <CY/T> — включает режим системного терминала;
- <CY/G> — включает режим отображения графической информации;
- <CY/P> — запускает копирование буфера регенерации изображения дисплея на печатающее устройство;
- <CY/V> — дополнительная функция.

Те же самые операции могут быть выполнены программно с помощью запроса MODE с соответствующими аргументами. Клавиатурные функции активизируются только после инициализации графической системы.

Как уже отмечалось, графическая система имеет модульную структуру, причем ряд ее компонентов динамически считывается в память микроЭВМ с диска по мере необходимости. При этом должны выполняться некоторые условия:

файлы графических сегментов и шрифтов должны находиться на устройстве;

чтобы монитор графической системы мог работать с ними, нужно предварительно инициализировать его. При этом в его внутренние таблицы записываются размеры и начальные блоки файлов, имеющих тип GSF и находящихся на устройстве GF. Компонента ОС USSR не используется, а ее состояние (SWAP/NOSWAP) не имеет значения. При работе с диском используется канал I4.

Структура предлагаемой графической системы существенно отличается от общепринятой организации графических пакетов в виде библиотеки процедур. Монитор

оформлен в виде драйвера псевдоустройства GP, что позволило отказаться от громоздкой библиотеки, содержащей объектные модули компонент графического пакета. Работа с такой библиотекой в условиях малой емкости диска и ОЗУ очень неудобна. Основная часть графической системы является общей для всех прикладных программ, использующих ее, поэтому в каждую такую программу включается модуль минимального размера (~150 слов). Наконец, иерархическая организация графической системы позволяет легко перенастраивать ее для работы с другими устройствами вывода. Для этого в системе выделена аппаратно-зависимая часть — графический драйвер, работа остальных компонент системы и прикладных программ не зависит от устройств вывода.

Драйвер графических устройств — программа на ассемблере в позиционно независимом коде, при написании которой используются макроопределения из специальной библиотеки GDMAC.MLB. Трудоемкость его написания не больше, чем любого драйвера устройства ввода-вывода в ОС РАФОС [2].

При разработке интерфейса между монитором и драйвером графической системы принимались определенные меры для уменьшения затрат времени на передачу параметров и управления ими, поэтому способ обращения к драйверу отличается от принятого в ОС. Одна из его важных особенностей — возможность доступа к таблице параметров, описывающих состояние графической системы (например, текущий цвет пера и фона, состояние пера: опущено — поднято и т. д.). Эти параметры не передаются драйверу при каждом обращении к нему, а используются им наравне с монитором по принципу общих областей. Кроме того, драйвер поддерживает работу системы с несколькими устройствами одновременно.

Информация о характеристиках устройства хранится в драйвере в служебной области (например, наличие встроенных генераторов векторов и символов).

Графическая система для микроЭВМ ДВК состоит из следующих составных частей:

- графический монитор (файл SY:GP.SYS),
- драйвер устройств вывода (файл SY:GD.SYS),
- графические сегменты (файлы GF:SEGX.GSF, где X=A..Z),
- наборы символов (файлы GF:FNTX.GSF, где X=A..Z), объектные модули интерпретатора процедур для языков, высокого уровня (файлы GRIP.OBJ — Паскаль, GRIF.OBJ — Фортран).

Предоставляются также библиотеки макроопределений для программирования графических сегментов (GPMAC.MLB) и графического драйвера (GDMAC.MLB), модуль SEGDEF.PAS для программирования сегментов на Паскале, модуль GRDEF.PAS для использования в прикладных программах, работающих с графической системой, а также программы для генерации шрифтов FONT.SAV и текстов TEXT.SAV.

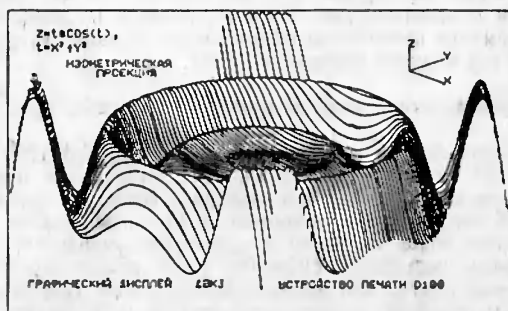


Рис. 1

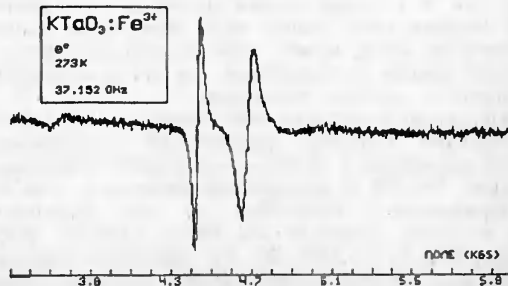


Рис. 2

```

TYPE BUFFER=ARRAY[1..3000] OF INTEGER;
PROCEDURE SPECTR(FBEG,FEND:REAL;
VAR SOURCE:BUFFER;
N:INTEGER);
VAR SPEMIN,SPEMAX:INTEGER;
FLD,PSTEP:REAL;
I,J:INTEGER;
BEGIN
SPEMIN:=SOURCE[1]; SPEMAX:=SOURCE[1];
FOR I:=2 TO N DO BEGIN
IF SOURCE[I]>SPEMAX THEN SPEMAX:=SOURCE[I];
IF SOURCE[I]<SPEMIN THEN SPEMIN:=SOURCE[I];
END;
MODE(SETDEV,TT); BOUND(0,0,350,350);
MODE(FISMAT,MAT); BOUND(FBEG,SPEMIN,FEND,SPEMAX);
FLD:=FBEG;
MOVE(FLD,SOURCE[1]);
MODE(ONOFF,ON); MODE(SETPEN,WHITE);
PSTEP:=(FEND-FBEG)/(N-1);
FOR I:=2 TO N DO BEGIN
FLD:=FLD+PSTEP; MOVE(FLD,SOURCE[I]);
END;
XAXIS(SPEMIN,"ПОЛЕ (KGS)",10,0,1);
HEAD;
END;

```

Данная графическая система используется авторами в составе программного обеспечения автоматизированного комплекса для исследования твердых тел методом электронного парамагнитного резонанса (ЭПР). (см. рис. 2 и 3).

Телефон 444-15-40, Киев

ЛИТЕРАТУРА

1. Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики. Т. 2.— М.: Мир, 1985.— С. 140.
2. Валикова Л. И., Вигдорчик Г. В. и др. Операционная система СМ ЭВМ РАФОС.— М.: Финансы и статистика, 1984.— С. 171—187.

Статья поступила 11.07.88

УДК 681.3.

Д. В. Степаненко, Ю. И. Бударин, Л. М. Белкин

ПАКЕТ ПРОГРАММ ДЛЯ РАБОТЫ С ГРАФИЧЕСКИМИ ТЕРМИНАЛАМИ СМ6404

Автоматизированные рабочие места типа АРМ 2.01 (и им подобные) на базе СМ1420 комплектуются устройствами ввода графической информации СМ6404, работающими под управлением микроЭВМ СМ1800. Обычно СМ6404 применяются только для ввода графической информации с помощью специальных программ, что неэффективно как с точки зрения использования микроЭВМ СМ1800 (ее возможности практически не используются), так и с точки зрения загрузки центральной машины, которая из-за малого числа терминалов, входящих в комплект АРМ, может одновременно обслуживать существенно меньше пользователей, чем это ей позволяют объем памяти и мощность процессора.

В НИИПТмаше разработан пакет программ для работы с устройствами СМ6404, позволяющий использовать названные устройства в качестве терминалов операционной системы ОС РВ и существенно расширять возможности терминальной микроЭВМ за счет включения СМ6404 в состав устройств ОС РВ в качестве обычного терминала, с которого ОС РВ принимает команды оператора и с которым обмениваются данными любые задачи, с сохранением возможности работы микроЭВМ в автономном режиме и передачи файлов центральной машинной терминальной микроЭВМ с автоматическим пре-

образованием файлов из формата ОС РВ в формат ОС СМ1800 и обратно. В состав пакета входят следующие компоненты:

эмулятор терминала для СМ1800, обеспечивающий передачу информации между дисплеем и центральной ЭВМ и управляющий операциями с дискетами СМ1800; привилегированная задача приема потока данных с СМ6404, функционирующая на центральной ЭВМ «невидимо» для пользователя;

библиотека макроопределений, с помощью которых работающие на центральной ЭВМ программы пользователя могут выполнять операции ввода-вывода на дискетах микроЭВМ;

псевдодрайвер — привилегированная задача, реализующая выполнение макрокоманд библиотеки;

диалоговая задача, осуществляющая передачу файлов между СМ6404 и центральной ЭВМ по командам оператора.

Для полного использования возможностей пакета центральной ЭВМ должна иметь диспетчер памяти.

Преимущества данной разработки: существенное расширение возможностей АРМ и их центральной машины СМ1420, увеличение числа обслуживаемых пользователей и степени загрузки машины, создание дополнительных удобств при использовании устройств СМ6404. Экономический эффект данной разработки (не менее 5,0 тыс. руб. на один АРМ) — только за счет увеличения полезного фонда машинного времени не менее чем на 50 %.

Пакет внедрен в НИИПТмаше в ноябре 1988 г. По данной разработке имеется комплект эксплуатационной документации.

Телефон 3-94-50, Краматорск Донецкой обл.

Сообщение поступило 22.06.88

УДК 681.3.022

Ю. В. Нифонтов, В. В. Калита, Г. М. Бобков, А. А. Яфаев

МОДУЛИ ОТОБРАЖЕНИЯ УВК СЕМЕЙСТВА СМ1800

В состав большинства УВК СМ1800 входят алфавитно-цифровые видеотерминалы, работающие в качестве системной консоли: ВТА 2000-30, ВТА 2000-15М, СМ7209 (производства ПНР) и другие с аналогичными характеристиками. Консольный видеотерминал обеспечивает отображение только алфавитно-цифровых текстов на монохромном экране.

Для отображения символьной и графической информации на многоцветных и монохромных экранах разработаны и освоены в производстве четыре модуля, каждый из которых выполнен в виде одного или двух блоков элементов на платах типа Е2 (один блок элементов имеет размеры 248×240×16 мм). Модули устанавливаются непосредственно в монтажный блок УВК и через магистраль системного интерфейса И41 обеспечивают обмен информацией с процессорами. Преобразованная в видеосигнал информация подается на вход видеоконтрольных устройств (ВКУ) и модулей индикации (МИ).

Характеристики модулей отображения (табл. 1)

Модуль вывода символьной информации СМ 1800.7003 (МВСТ) обеспечивает отображение текстовой и псевдографической информации в различных форматах (табл. 2). МСВТ состоит из формирователя ТВ-сигнала; ОЗУ для хранения кодов символов и признаков выделения; ОЗУ программ построения символов (ОЗУ знакогенератора); основных портов для взаимодействия с ЦП (порты формата, начальной строки, координат X и Y, кодов символа и признака, знакогенератора).

ЦП обращается к ОЗУ через порты, т. е. МСВТ по отношению к ЦП представлен как массив адресуемых пор-

Характеристика модулей отображения SM1810

Характеристика	SM1800.7003 (МВСТ)	SM1800.7004 (МВГТ)	SM1810.7005 (ВГК)			SM1810.7006 (ВКЦ)	
			01	02	03	Символьный режим	Графический режим
1	2	3	4	5	6	7	8
Год освоения	1983	1985	1988				1988
Число символов в строке	64,32	—	—	—	—	80,40	—
Число строк символов	32, 16, 8	—	—	—	—	25	—
Число точек							
по горизонтали	—	384	720	640	1024	—	640,320
по вертикали	—	256	512	512	768	—	200
Число цветов							
символов	8	—	—	—	—	16	—
фона	8	—	—	—	—	8	—
точек	—	8	16 (из палитры 4096)			—	2,4
Число символов алфавита	256	—	—	—	—	256	—
Матрица знакоместа	8×8, 8×16	—	—	8×8	—	8×8	—
Число команд для построения графического изображения	—	—	14			—	—
Синхронизация							

Таблица 2

тов, которые размещаются в поле адресов ВУ, доступных ЦП. Формат вывода выбирается ЦП путем записи кода формата в порт формата. Коды и признаки символов записываются МСВТ и читаются в ЦП по значениям портов координат X и Y, соответствующим номерам символа и строки.

Содержимое этих портов меняется по команде записи кода символа из процессора. В ОЗУ кодов символов и признаков записывается информация о цвете и фоне символа, признак маркера или подчеркивания через порт кода признака. Адрес ячейки памяти задается аналогично с помощью портов X и Y.

Для чтения-записи ОЗУ знакогенератора в порт X выводится номер строки матрицы разложения изображения символа, в порт Y — код символа.

Для сдвига информации на экране ЦП заносит в порт начальной строки номер строки символов, которая первой должна индентифицироваться на экране. Номера следующих строк автоматически формируются относительно начальной. Соответствующие аппаратные средства существенно ускоряют эту операцию.

Модуль вывода растровой графической информации SM1800.7004 (МВГТ) предназначен для вывода на экран 384×256 точек, каждая из которых представлена тремя информационными битами, что позволяет иметь восемь уровней яркости на черно-белом экране или восемь цветов на цветном. По отношению к ЦП МВГТ представляет собой три части ОЗУ по 16К 8-разрядных слов, занимающих одно и то же поле адресов для хранения информации красного зеленого и синего цветов.

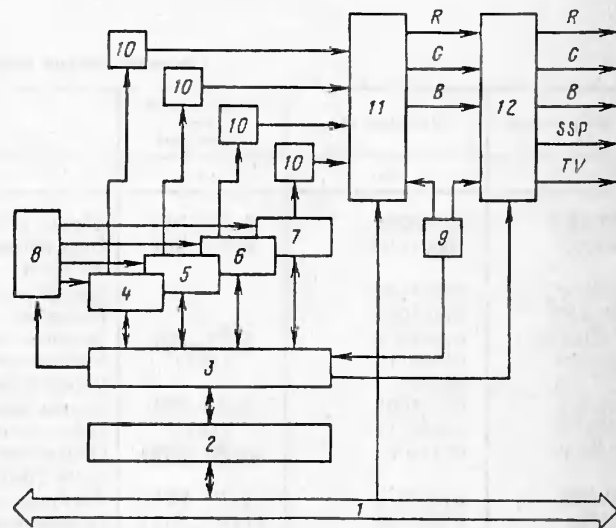
Память может программно включаться (выключаться) ЦП в адресное поле памяти УВК с помощью специального порта. Если в этот порт предварительно записана информация, выбирающая одну из частей памяти МВГТ, то записывается или читается именно эта часть, МВГТ формирует сигналы запрета INH1/, INH2/, по которым отличается основная память УВК (оперативная и постоянная), зона адресов которой совпадает с зоной адресов оперативной памяти МВГТ.

Зона адресов оперативной памяти МВГТ и адрес порта задаются переключками наборных полей. Построением графических рисунков на МВГТ занимается непосредственно ЦП.

Видеографический контроллер SM1810.7005 (ВГК) представляет собой устройство вывода графической информации на телевизионный индикатор (см. рис.). ВГК построен

Форматы вывода данных на экран

Код формата	Число символов		Число строк символов	Число элементарных точек в матрице изображения символа	
	на экране	в строке		по горизонтали	по вертикали
00	2048	64	32	8	8
01	512	32	16	16	16
10	1024	64	16	8	16
11	256	32	8	16	32



Структурная схема видеографического контроллера SM1810.7005:

1 — магистраль ИА1; 2 — шинные формирователи; 3 — графический контроллер; 4...7 — блоки 0...3 видеопамати (64К×16); 8 — дешифратор; 9 — генератор тактовых импульсов; 10 — сдвиговый регистр; 11 — схема управления цветом; 12 — формирователь телевизионного сигнала

на базе микросхемы K1809ВГ4, имеющей систему команд, позволяющую строить рисунки графическим способом с помощью точки, линии, дуги, прямоугольника, графического символа, заполненной зоны, наклонного графического символа и наклонной заполненной зоны.

ВГК позволяет создавать полиэкранные изображения: экран разбивается на два поля со своими начальными адресами длиной, выраженной числом строк. Изображение на полях строится и редактируется независимо.

Число адресуемых точек, информация о которых хранится в видеопамяти, составляет 1024×1024 , поэтому не все они видны на экране. ВГК позволяет панорамировать изображение, изменяя начальный адрес поля, т. е. просматривать на экране все адресуемые точки. Панорамирование возможно по вертикали с дискретностью в одну строку, по горизонтали — в одно слово (16 точек).

С помощью ВГК можно масштабировать, т. е. увеличивать размеры изображения. Коэффициент увеличения графического символа при записи может изменяться от двух до шестнадцати. Каждая точка повторяется в видеопамяти по горизонтали и по вертикали определенное число раз, заданное коэффициентом увеличения.

Изображение на экране может быть увеличено в два или четыре раза. Для этого по вертикали каждая строка повторяется, а по горизонтали увеличивается время высвечивания каждой точки в соответствии с коэффициентом увеличения изображения. Кроме того, предусмотрена возможность увеличения изображения в два раза только по горизонтали (для индикаторов с низкой разрешающей способностью).

Графический контроллер (ГК) управляет видеопамтью объемом 256К 16-разрядных слов, разделенной на четыре блока по 1024×1024 бит (см. рисунок).

В процессе воспроизведения информация считывается одновременно из всех блоков, что обеспечивает 4 бит на один элемент изображения, и подается на сдвиговые регистры, где преобразуется в последовательный видеокод, поступающий на схемы управления цветом. Выбор 16 цветов из палитры 4096 осуществляется ЦП путем записи соответствующей информации в три порта вывода схемы управления цветом: номера цвета, яркости красного цвета, яркости синего и зеленого цветов.

Регистры яркости красного, синего и зеленого цветов представляют собой три быстродействующих запоми-

нающих устройства (БЗУ) емкостью 16×4 бит каждое. В процессе воспроизведения изображения информация из видеопамяти поступает на адресные входы этих БЗУ. По этим адресам считывается информация о яркости цветов отдельной точки. Каждый разряд данных БЗУ имеет свой вес в определении яркости цвета.

Рисунок строится ГК по командам ЦП, представляющим собой байт, за которым следует последовательность байтов параметров команды, необходимых для определения деталей этой команды (табл. 3). Место изменения содержимого видеопамяти определяется курсором, адрес которого задается командой CURS. За изменением адреса курсора следит сам ГК. Изображение курсора на экран не выводится.

Внутренний 16-разрядный регистр маски ГК определяет, содержимое какого разряда или разрядов в выбранном для обращения слове видеопамяти должно быть изменено. Регистр маски должен быть загружен командой MASK.

Для подготовки ГК к построению рисунка необходимо загрузить соответствующие команды и параметры. Командой CURS задается адрес рабочего слова и точка в слове, с которой начинается построение рисунка.

Восьмой и девятый байты, загружаемые командой PRAM, обеспечивают построение рисунка заданным типом линий (сплошная, штриховая, штрих-пунктирная и т. д.). Для вычерчивания графических символов и заполненных зон необходимо загрузить в запоминающее устройство ГК 8 байт матрицы изображения символа. С помощью команды FIGS выводятся последние параметры построения рисунка.

Видеоконтроллер цветной SM1810.7006 (ВКЦ) обеспечивает отображение алфавитно-цифровой или графической информации на модулях индикации M32Ц11/2, A543/13. Используется для построения консоли в УВК SM1810.53 и SM1810.13*. Модуль может работать в графическом или символьном режиме по командам ЦП, в графическом режиме разрешение составляет 640×200 точек для монохромного изображения или 320×200 для цветного (четыре цвета, одновременно индицируемых на экране).

* Бобков Г. М. Видеоконтроллер SM1810.7006 // Микропроцессорные средства и системы. — 1989. — № 4. — С. 67

Таблица 3

Система команд графического контроллера

Обозначение	Двоичный код	Максимальное число параметров	Выполняемая функция
1	2	3	4
RESET	00000000	8(P1...P8)	Сброс, установка режима простоя
SYNC	00001111 DE	8(P1...P8)	Определение формата изображения: DE=1 — экран засвечен; DE=0 — погашен
VSYNC	011011M	—	Выбор режима работы: M=1 — ведущий; M=0 — ведомый
START	01101011	—	Выход из режима простоя и погашенного экрана
CCHAR	01001011	3(P1...P3)	Задание характеристик курсора
ZOOM	01000110	1(P1)	Увеличение изображения
BCTRL	000011	—	Управление гашением экрана: DE=0 — экран погашен; DE=1 — засвечен
CURS	01001001	3(P1...P3)	Запись адреса курсора (рабочего слова)
PITCH	01000111	1(P1)	Определение размера видеопамяти по горизонтали
PRAM	01111SA	16(P0...P15)	Определение начальных адресов, длин полей экрана и матрицы графического символа: A — адрес начального параметра (0...15)
MASK	01001010	2(P1, P2)	Загрузка регистра маски
FIGS	01001100	11(P1...P11)	Определение параметров для построения рисунка
FIGD	01101100	—	Построение рисунка, определенного командой FIGS
GCHRD	01101000	—	Построение графического символа или заполненной зоны, определенных параметрами команды FIGS
CURD	11100000	5(P1...P5)	Чтение адреса курсора
WDAT	001TYPEOMOD	—	Запись данных в видеопамять
RDAT	101TYPEOMOD	—	Чтение данных из видеопамяти

ВКЦ по функциям и системе команд полностью соответствует графическому адаптеру CGA IBM PC. Отличие заключается в том, что он может обмениваться с ЦП И41 не только байтами, но и 16-разрядными словами.

Особенности применения модулей

При построении систем отображения с применением описываемых модулей отображения и УВК семейства SM1800 должны быть выбраны средства отображения и режимы их работы с учетом особенностей конкретного объекта. Вывод информации на экран через модуль отображения требует затрат машинного времени на пересылку команд и параметров.

В случае вывода алфавитно-цифровой или псевдографической информации это время будет минимальным при применении модулей ВКЦ или МВСТ.

Если программируемый знакогенератор МВСТ загружен программами построения символов, то для вывода одного символа требуется вывести четыре байта информации. Практически смена полного экрана при работе с ВКЦ или МВСТ воспринимается наблюдателем, как будто она происходит «мгновенно».

Если ВКЦ и МВСТ используются в графическом режиме, то изображение строится по точкам. За одно обращение при пересылке первого байта в ВКЦ передается информация о цвете четырех точек (для 4-цветного режима) и яркости восьми точек одного цвета.

На вывод одного символа с помощью ВГК затрачивается больше времени (1...3 мс в зависимости от цвета символа и фона). Зато графическая информация выводится значительно быстрее, так как ВГК работает без участия ЦП и выхода на магистраль И41. Время построения графического примитива самим ВГК может быть рассчитано исходя из того, что один элемент изображения выводится

в один блок видеопамати за четыре такта. Например, диагональ прямоугольника 1024×1024 точки в варианте 03 ВГК строится за 1,3 мс.

Часто требуется выделить отдельные отображаемые элементы мерцанием. В ВКЦ это делается аппаратными средствами; в МВСТ или МВГТ информация перезаписывается в память модулей из ЦП с частотой мерцания; в ВГК выделяется один из блоков видеопамати для записи в него признака мерцания точки (в этом случае требуется перезапись с частотой мерцания только одной ячейки в таблице цветов схемы управления цветом). Число цветов на экране при этом сокращается до восьми.

Модули МВСТ, МВГТ и ВГК позволяют удалять ВКЦ или МИ, на котором воспроизводится изображение, на расстояние до 200 м от ЭВМ. Помехи 50 Гц ЭВМ и внешние магнитные поля вызывают колебания изображения и модуляцию по яркости. Для снижения их влияния в перечисленных трех модулях предусмотрена синхронизация кадров изображения с сетью 50 Гц: на специальный вход модулей через разделительный понижающий трансформатор должно быть подано напряжение 6,3 В по сети 50 Гц. В модуле ВКЦ не предусмотрены аналогичные меры, поэтому МИ, подключаемый к ВКЦ, должен заземляться в той же точке, что и УВК.

Модули МВСТ и МВГТ обеспечены тестовыми программами для SM1800 (МВСТ имеет драйвер для работы с DOS1800); ВГК — тестами и драйверами для SM1810 с BOS1810 и DOS1810.

ВКЦ с клавиатурой и МИ используется в модификациях УВК SM1810, поддерживающих МДОС1810, совместимую с PC DOS фирмы IBM. При установке в такие УВК модуля ВГК и второго МИ поддерживается ряд известных пакетов профессиональной графики.

Телефон 455-57-41, Москва

Статья поступила 15.06.88

УДК 681.327.11.06

Е. М. Блох, К. Б. Бодашков, В. В. Шляхтин

ГРАФИЧЕСКИЙ КОНТРОЛЛЕР ДЛЯ ЦВЕТНОЙ ГРАФИЧЕСКОЙ СИСТЕМЫ

Одноплатный графический контроллер обеспечивает формирование цветного изображения (64 цвета) в шести плоскостях памяти отображения размерами 1024×512 точек (частота строчной развертки ТВ-монитора 15,625 кГц). Кроме того, изображение может иметь две независимые 8-цветные поверхности с возможностью их наложения по заданному приоритету. На видеоконтрольном устройстве (ВКУ) обеспечивается разрешение 640×512 пиксел (цветных элементов отображения) в режиме чересстрочной и 640×256 пиксел — в режиме прогрессивной разверток. В последнем случае объем памяти отображения позволяет иметь два независимых кадра (двойная буферизация).

Использование высокочастотного телевизионного стандарта (частота строчной развертки 31,5 кГц) позволяет формировать графическое изображение в трех плоскостях памяти отображения размерами 1024×1024 пиксел. При этом на ВКУ одновременно отображается восемь цветов с разрешением до 1024×1024 пиксел в режиме чересстрочной и 640×522 пиксел в режиме прогрессивной разверток.

Контроллер построен на базе трех

БИС графических сопроцессоров (ГСП) K1809ВГ4 (рис. 1) с банком памяти отображения емкостью 128 Кбайт каждый на БИС K565РУ5. Использование трех ГСП обеспечивает параллельную работу в плоскостях 1—3 группы 1 или в плоскостях 4—6 группы 2 (рис. 2). При формировании изображения в шести плоскостях (64 цве-

та) работа выполняется последовательно в плоскостях групп 1 и 2.

Линии, дуги, прямоугольники, графические символы вычеркиваются со скоростью порядка 0,8 млн. пиксел/с. Графический символ при записи в память и изображение в памяти при выводе на экран увеличиваются с коэффициентом 1...16. Аппаратное панорами-

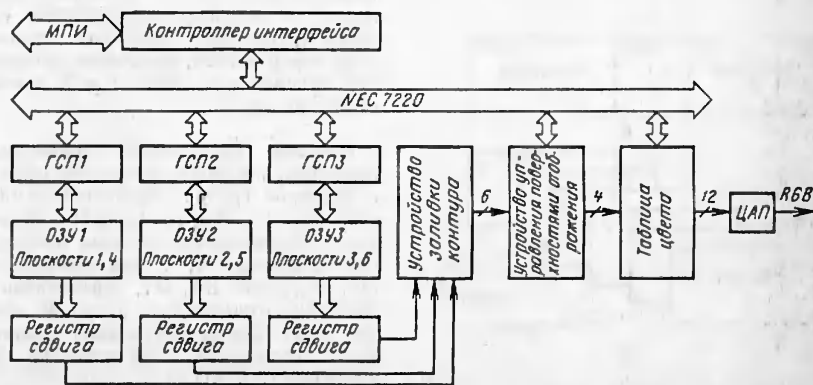


Рис. 1. Структурная схема графического контроллера

рование в горизонтальном направлении осуществляется с дискретностью 16 пиксел, в вертикальном — с точностью до линии раstra.

В контроллере реализованы аппаратные средства «заливки» контура горизонтальными линиями, соединяющими пиксели одного цвета. Устройство управления поверхностями отображения позволяет изменять режим работы контроллера. Можно, например, определить одну поверхность, имеющую шесть слоев памяти и 64 цвета одновременно (изображения в слоях формируется за два прохода), или две поверхности отображения по три слоя, соответствующие группам 1 и 2. При этом для каждой поверхности можно задать свою гамму из восьми цветов. При работе с двумя поверхностями задается приоритет их наложения, т. е. указывается поверхность, изображение в которой заслоняет изображение в другой поверхности.

Цветовая гамма изображения в контроллере изменяется с помощью таблицы цветности и цифро-аналоговых преобразователей (таблица объемом 16×12 бит на трех БИС K530PY2 обеспечивает выбор цветов из гаммы 4096 оттенков).

Контроллер имеет интерфейс шины NEC7220, который через интерфейсную плату подключается к МПИ или И41. Для повышения скорости обмена ЦП используется контроллер интерфейса на основе вентиляльной матрицы серии K1801ВП-1. Он обеспечивает параллельную загрузку 8- и 16-разрядными данными трех ГСП и реализует систему векторного прерывания с маскированием по любому разряду регистра состояния микросхемы K1809ВГ4. Возможность загрузки 16-разрядных данных дает существенную экономию времени ЦП, так как большая часть параметров команд графического сопроцессора имеет формат два байта.

На основе контроллера и серийных мипи- и микроЭВМ можно построить цветную графическую систему. Программное обеспечение такой системы можно условно разделить на аппаратно-зависимое и аппаратно-независимое

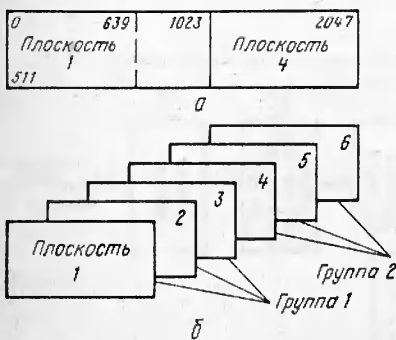


Рис. 2. Организация памяти: а — память отображения ГСП; б — логическая организация памяти

(графические стандарты, графические расширения языков высокого уровня, специализированные графические пакеты).

Назначение программного обеспечения первого уровня — создать универсальный и удобный интерфейс для программиста, использующего средства отображения графической информации. Аппаратный интерфейс контроллера представляет собой довольно сложную систему подготовки данных, обусловленную системой команд ГСП K1809ВГ4 и другими особенностями аппаратуры. Для того чтобы избавить прикладного программиста от необходимости изучения особенностей графического контроллера, разработан базовый набор графических подпрограмм, поддерживающий большинство возможностей аппаратуры и обеспечивающий удобный интерфейс с контроллером. Подпрограммы на ассемблере микроЭВМ «Электроника 60» собраны в графический пакет, который компонуется с прикладной программой в ОС РТ-11 или совместимой с ней. Для удобства использования пакета разработана библиотека макрообращений к подпрограммам.

Программы пакета реализуют пять групп графических команд: общего управления, управления поверхностями отображения, графических примитивов, атрибутов графических примитивов и переписи блоков.

Команды общего управления обеспечивают инициализацию графического режима, начальную установку атрибутов и параметров, очистку памяти отображения, увеличение изображения при выводе на экран, изменение стартового адреса (панорамирование), включение и гашение изображения.

Команды управления поверхностями отображения позволяют работать параллельно в трех плоскостях памяти отображения, образующих группу 1 или 2 (см. рис. 2). Эти команды осуществляют выбор группы плоскостей, установку типа записи в память, очистку группы или отдельных плоскостей, установку режимов двух 8-цветных поверхностей с заданием приоритетов или одной 64-цветной, управление видимостью поверхностей, включение аппаратной заливки в группах 1 и 2 памяти отображения.

Команды графических примитивов позволяют построить различные образы в заданной группе плоскостей памяти отображения. Ряд графических примитивов определяется заданием абсолютных координат X, Y или относительных координат DX, DY, определяющих смещение относительно текущей позиции (ТП). Две команды дают возможность перемещать ТП в точки (X, Y), или ТП+(DX, DY).

Остальные графические примитивы образуют две группы: примитивы типа линия и примитивы, связанные с ото-

бражением графического символа — массива 8×8 бит из внутренней памяти ГСП. Такое деление определяется раздельным заданием атрибутов для этих групп примитивов.

Графические примитивы типа линия позволяет построить точку (X, Y); точку ТП+(DX, DY); линию из ТП в точку (X, Y); линию из ТП в точку ТП+(DX, DY); прямоугольник с одним углом в ТП, другим — в точке ТП+(DX, DY); окружность радиуса R с центром в точке (X, Y); дуги окружности радиуса R с центром в точке (X, Y), с начальной и конечной точками, заданными относительно центра (DXB, DYB) и (DXE, DYE).

Графические примитивы типа графический символ предназначены для построения: прямоугольника, заполненного по образцу в памяти графического символа с одним углом в ТП, другим — в точке (X, Y);

заполненного прямоугольника с одним углом в ТП, другим — в точке ТП+(DX, DY);

графического символа (маркера) с центром в заданной точке (X, F), а также для выбора графического символа (маркера) по индексу и вывода алфавитно-цифрового символа (левый нижний угол матрицы символа в ТП).

Команды установки атрибутов графических примитивов также делятся на две группы. Для примитивов типа линия устанавливаются индексы цвета и фона линии, шаблон, определяющий тип линии (непрерывная штриховая и т. п.); для примитивов типа графический символ — индексы цвета и фона символа, образец заполнения прямоугольной области, коэффициент увеличения символа, ориентация алфавитно-цифрового символа (кратно 45°), смещение ТП после вывода алфавитно-цифрового символа.

Команды переписи блоков позволяют переместить заданную прямоугольную область из памяти отображения в память центральной ЭВМ и наоборот, заполнить прямоугольную область в памяти отображения данными из памяти центральной ЭВМ.

На основе описанного набора графических команд легко реализовать любые недостающие функции для конкретного приложения. Графические пакеты, работающие с конкретными терминалами, несложно настроить на работу с описанным графическим контроллером.

Телефон 291-67-41, Ленинград

Статья поступила 21.03.88

ФОТОЭЛЕКТРОННАЯ СИСТЕМА ПРЕОБРАЗОВАНИЯ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Система построена на основе ДВК2, оснащенного специализированными программно-аппаратными средствами для повышения скорости обработки изображений, и фоточувствительных микросхем с зарядовой связью (ФМЗС) [1, 2] для преобразования непрерывных и импульсных излучений в цифровой код (рис. 1).

Назначение и работа блоков. В качестве первичного ФЭП используется телевизионная камера [3]. БЗУ согласует скорости обмена информацией между ФЭП и ЭВМ. Емкость БЗУ, необходимая для записи одного или нескольких кадров изображения (110 Кбайт на кадр изображения, формируемого ФЭП на ФМЗС К1200ЦМ2, К1200ЦМ7), обеспечивается микросхемами ОЗУ динамического типа К565РУ5Б. Запись в БЗУ — последовательно-параллельная, т. е. несколько страниц памяти работают параллельно, а соответствующее число последовательно поступающих слов информации записывается во входные буферные регистры [4, 5]. Предусмотрена возможность подключения до 32 модулей БЗУ емкостью 256 Кбайт каждый.

Синхронная запись информации с ФЭП и БЗУ выполняется под управлением КВД по вырабатываемым ФИУ импульсам сопровождения элемента (ИСЭ), строки (ИСС) и кадра (ИСК) изображения [3]. БЗУ тактируется генератором ФИУ ФЭП. Регенерация ОЗУ обеспечивается последовательным обращением к столбцам матриц ЗУ микросхем.

Система находится в режиме отобра-

жения как при хранении записанной в БЗУ видеoinформации, так и при обмене данными между БЗУ и ЭВМ. Информация из микросхем памяти в выходные буферы модулей БЗУ считывается одновременно под управлением КВД по синхросигналам с ГТИ. Регенерация БЗУ при отображении происходит также, как при записи. К ШВД для передачи на УОВД подключаются выходные буферы модуля БЗУ, адрес которого задается КОД.

Обмен данными между системой и ЭВМ организуется через СИ, подключенный к системному каналу микроЭВМ (передача данных ведется асинхронно). Регенерация БЗУ будет более экономичной, если обмен выполняется на «фоне» отображения видеoinформации. Это означает, что на время считывания байта информации из БЗУ к адресным входам БЗУ подключается КОД вместо КВД и выбранная страница модуля БЗУ переводится в режим записи или считывания.

КОД обеспечивает произвольный доступ в БЗУ к элементам раstra изображения и последовательный доступ к элементам строк или столбцов с любыми начальным и конечным адресами и шагом приращения адреса по строке и столбцу; записывает в регистр слово состояния КОД и выдает на ШД информацию о достижении конечного адреса в строке и (или) столбце; выдает на ШД текущий адрес обмена и адрес ветвления программы для процессора микроЭВМ в зависимости от состояния КОД (рис. 2).

Каждая из двух одинаковых микро-

процессорных секций КОД включает 12-разрядный сумматор на основе 4-разрядных секций типа К1802ИМ1 для вычисления номера строки или столбца, буферный регистр для хранения конечного номера из микросхем К1804ИР1 и схему сравнения конечного адреса с текущим на компараторах К531СП1. Входы и выходы сумматора объединены и подключены к ШД через двунаправленный БА. Один из четырех входных буферов сумматора используется в качестве накопителя, два — для хранения начального номера и шага приращения. ФА состоит из двух 12-разрядных буферных регистров с тремя состояниями на выходе; МБД — из двух 8-разрядных регистров с тремя состояниями на выходе (служит для двунаправленного обмена данными между БЗУ и микроЭВМ по ШВД и ШД). УУВП содержит регистр состояний (РС) МПС с мультиплексором на входе и ОЗУ для хранения адресов ветвления программы.

Блоки КОД работают под управлением микрокоманд, поступающих из ПЗУ УУ через буферные регистры. Группа последовательно записанных в ПЗУ микрокоманд образует макрооперацию, которая выполняется перебором адресов ПЗУ с помощью счетчика по тактовым импульсам ГТИ. Вслед за последней микрокомандой макрооперации в ПЗУ записано нулевое слово остановки. Требуемая макрооперация выбирается записью в счетчик по команде микроЭВМ адреса первой микрокоманды макрооперации с ША.

Технические возможности и характеристики. Разработанная система позволяет оперативно управлять преобразованием непрерывных и импульсных излучений, записью изображений в реальном масштабе времени, обработкой и отображением видеoinформации. Команды управления системой могут

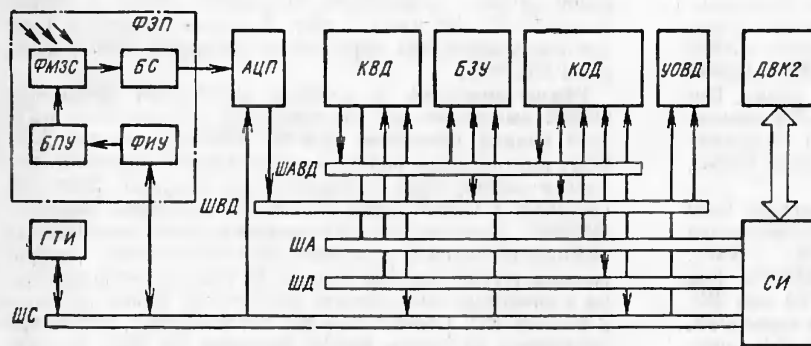


Рис. 1. Структурная схема фотоэлектронной системы преобразования и обработки изображений:

первичный фотоэлектронный преобразователь (ФЭП) на основе ФМЗС; формирователь импульсов управления (ФИУ); блок преобразователей уровней (БПУ); блок считывания выходного сигнала с ФМЗС (БС); блок аналого-цифрового преобразователя (АЦП); контроллер видеоданных (КВД); буферное запоминающее устройство (БЗУ); контроллер обмена данными (КОД); устройство отображения видеоданных (УОВД); генератор телевизионных импульсов (ГТИ); специализированный интерфейс (СИ); шина видеоданных (ШВД); шина адреса видеоданных (ШАВД); шина данных (ШД); шина синхронизации (ШС)

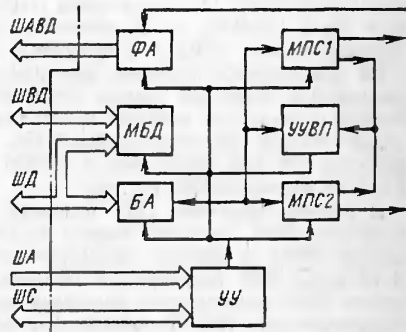


Рис. 2. Структурная схема контроллера обмена данными:

микропроцессорные секции строки и столбца (МПС1 и МПС2); формирователь адреса видеоданных (ФА); многорежимный буфер данных (МБД); устройство управления ветвлением программы (УУВП); буфер адреса (БА); устройство управления (УУ)

Выбор режима ФЭП	Включение автоматического или ждущего режима
Запись кадра	Отключение модуля БЗУ от шины видеоданных и подключение АЦП; подача синхросигналов с ФИУ ФЭП на БЗУ, КВД и УОВД; перевод модуля БЗУ в режим записи; перевод системы в режим отображения
Запуск преобразования ФЭП	Выработка импульса запуска с пульта оператора или микроЭВМ и выполнение команды «Запись кадра изображения»
Включение режима отображения видеoinформации с ФЭП (из БЗУ)	Подача синхросигналов с ФИУ ФЭП (ГТИ) на УОВД; подключение АЦП (БЗУ) и ШВД
Выбор модуля БЗУ*	Запись адреса модуля БЗУ в КОД
Вывод слова состояния КВД на ШД*	Вывод информации о режимах ФЭП (автоматический — ждущий) и модуля БЗУ (запись-считывание) на ШД
Выбор и выполнение макроопераций КОД*	Запись параметров алгоритма обмена; вычисление адреса растра изображения и обмен данными между БЗУ и микроЭВМ; выдача слова состояния КОД на ШД

Основные параметры системы

Число элементов разложения пространственного сигнала	360×288
Разрешающая способность, лин./мм	23
Спектральный диапазон регистрируемых излучений, мкм	0,4...1,1
Пороговая чувствительность на элемент разложения, фотонов	4000
Полная шкала амплитудной характеристики по каждому элементу, В	2
Разрядность по амплитуде в каждом элементе разложения	8
Погрешность квантования амплитуды, %	0,4
Время экспозиции, мс, не менее	1,15
Время кадра, мс, не менее	18,85
Допустимое время кадра (без охлаждения ФМЗС), мс	200
Частота дискретизации выходного сигнала, МГц, не более	7
Емкость модуля БЗУ, Кбайт	256
Время обращения к ОЗУ, мс, не менее	120
Время ввода кадра изображения в микроЭВМ, мс	450
Время ввода кадра в режиме скользящего окна размерами 3×3 элемента, с	4
Напряжение источников питания, В	5, -5,12, -12
Потребляемая мощность (без учета микроЭВМ), Вт	15

* Команда подается только от микроЭВМ

подаваться оператором и передаваться с микроЭВМ.

Благодаря использованию КОД удалось распараллелить процессы подготовки и обработки данных и органи-

зовать управление ветвлением программы процессора микроЭВМ.

115409, Москва, Каширское ш., 31, МИФИ; тел. 324-91-65

1. Матвеев В. И., Староверов Ю. Г. Полутоновая система технического зрения // Микропроцессорные средства и системы. — 1987. — № 12. — С. 68—69.
2. Суранов А. Я., Госьков П. И., Якунин А. Г. Микропроцессорная система измерения параметров би-нарных изображений на основе многоэлементного фотоприемника // Микропроцессорные средства и системы. — 1988. — № 1. — С. 54—56.
3. Дик П. А., Краснюк А. А., Степин В. Я. Телевизионная камера на фоточувствительной микросхеме с зарядовой связью с кадровым переносом // Электронная промышленность. — 1988. — № 4. — С. 85—87.
4. Дик П. А., Краснюк А. А., Степин В. Я., Яковлев А. В. Импульсный фотоэлектронный преобразователь с памятью на кадр // Координатно-чувствительные фотоприемники и оптико-электронные устройства на их основе. Тез докл. IV Всес. совещ. — Барнаул: АПИ. — 1987. — Ч. 1. — С. 15—16.
5. Верман В. Л. Модуль двухпортового ЗУ для систем автоматизированных измерений параметров телевизионных передающих приборов // Электронная промышленность. — 1987. — № 8. — С. 14—16.

Статья поступила 17.10.88

СРЕДСТВА ОТЛАДКИ

УДК 681.3.06

В. Л. Горбунов, М. А. Сулимина

КОМПЛЕКС ОТЛАДКИ ОЭВМ

Комплекс включает программное обеспечение и плату сопряжения (ПС) для стыковки управляющей ЭВМ (в данном случае ДВК) с отлаживаемым устройством. Управляющая ЭВМ (УЭВМ) и ПС обмениваются данными через селектор адреса (СА) (см. рисунок). При работе УЭВМ в ВУ используется адресное пространство 160000...160016 (входы АД селектора адреса установлены в нуль). Выходные сигналы СА выборки кристалла С0...С7 и записи чтения М1/М2 управляют работой ПС. Сигнал С0 формируется в СА при обращении к УЭВМ по адресу 160000, С1 — по адресу 160002 и т. д.

В режиме эмуляции ПЗУ основную роль играет блок в составе ОЗУ, счетчика адреса (СТА), мультиплексора адреса (МА) и шинных формирователей (BS1 и BS2). В качестве ОЗУ используется микросхема К537РУ9. Код адреса ОЗУ формируется выходным кодом СТА или ВУ, подключенным к ПС. Устройство, формирующее адрес ОЗУ, выбирается в соответствии с сигналом X (при X=0 к адресным входам ОЗУ подключается выход СТА, при X=1 — адресная часть разъема X3). Для записи начального адреса в СТА из УЭВМ используется обращение по адресу 160006 (сигналы С3, М1). Увеличение адреса СТА на единицу требует обращения УЭВМ к ПС по адресу 160010 (сигналы С4, М1 или С4, М2). Шинный формирователь BS1 подключает выход данных ОЗУ к разъему X3. Информацию с BS1 можно стробировать сигналом с отлаживаемого устройства CS2 (разъем X3).

Шину данных ОЗУ с общей шиной ДВК связывает BS2. Для записи данных в ОЗУ по адресу СТА из УЭВМ, т. е. из файла на рабочем устройстве, используется обращение по адресу 160010 (сигналы С4, М1); для считывания данных — обращение по адресам 160002 и 160010 (сигналы С1, М2 или С4, М2). В режиме эмуляции ПЗУ для считывания ОЗУ обращаются по адресу 160016 (сигналы С1, М2).

Режим эмуляции. В процессе выполнения программы ОЭВМ выставляет на внешнюю шину содержимое счетчика команд (контакты ARAM0...ARAM10 разъема X3). ОЗУ, имитирующее работу внешней памяти программ, выдает в соответствии с адресом код команды. Этот код поступает в ОЭВМ через разъем X3 (контакты DRAM0...DRAM7). Комплекс отладки предусматривает возможность анализа сигналов в произвольных восьми точках отлаживаемого устройства. Магистраль X5 подключается выходами к анализируемым точкам, обеспечивая запись сигналов в регистр RD. Содержимое RD опрашивается УЭВМ при обращении по адресу 160012 (сигналы С5, М2). Считанный из RD код редактируется и выводится на экран УЭВМ в форме, удобной для анализа. В режиме логического анализатора можно выводить на экран адрес команды (считывается в УЭВМ через RA по адресу 160004 (сигналы С2, М2)) и восемь сигналов логического анализатора, которые вводятся в УЭВМ по адресу 160012 (сигналы С5, М2) через RD. Кроме того, предусмотрена возможность выбора типа синхронизации устройства: внешняя (от ПС) или внутренняя. При внешней синхронизации

можно изменять характеристики синхрипульсов (число в пачке, скважность).

При отладке устройств с использованием процессоров КМ1813ВЕ1 и в некоторых других случаях возникает необходимость в аналого-цифровом и цифро-аналоговом преобразовании сигналов с возможностью наглядного отображения информации. Для аналого-цифрового преобразования предназначен блок АЦП, состоящий из собственно АЦП и регистра. При этом происходит обращение к УЭВМ по адресу 160014 (сигналы С6, М1 и С6, М2). Обратную задачу выполняет блок ЦАП, который включает в себя схему ЦАП, регистр и выходной усилитель ОУ. Обращаться к схеме ЦАП можно по адресу 160000 (сигналы С0, М1 и С0, М2).

Программное обеспечение (ПО) включает программу-отладчик ОТ18.SAV и набор кросс-средств для ОЭВМ К1816, К1820. Программа ОТ18 представляет собой фиксированный набор команд: пользователя, работы с памятью, анализатора, преобразования формата.

Команды пользователя

- help** — справочник программы ОТ18;
- date** — сообщение о текущей дате;
- type** — просмотр файла на экране;
- exit** — выход из программы ОТ18 в монитор; файл протокола, если его забыли закрыть командой nolisting, закрывается;
- show** — сообщение о состоянии файла-протокола (открыт или закрыт);
- graphics** — просмотр графика функции;
- listing** — запись последовательности действия при работе с ОТ18 в специальный файл-протокол (текущая дата, названия выполняемых команд и результаты их работы);
- nolisting** — закрывает файл протокола.

Команды работы с памятью

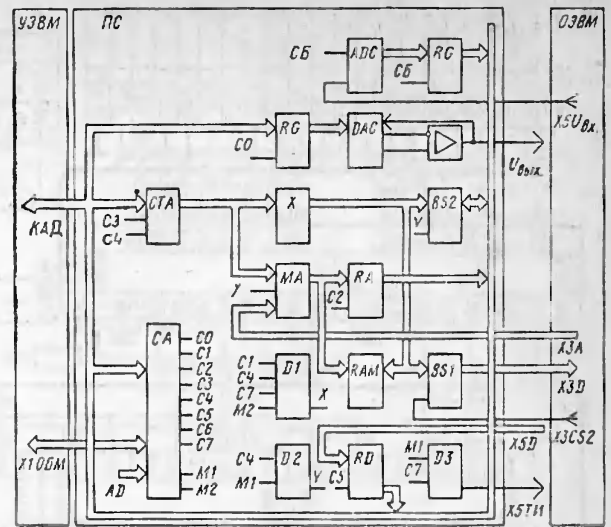
- test** — в каждый разряд каждой ячейки ОЗУ последовательно записываются, а затем считываются единица и ноль; после выполнения команды в каждой тестируемой ячейке записана единица; при наличии неисправных ячеек выдаются их адреса;
- load** — загрузка в ОЗУ программы в шестнадцатеричных кодах из файла;
- ram** — просмотр ОЗУ;
- rom** — просмотр ПЗУ;
- map** — составление карты памяти ПЗУ;
- comparing** — сравнение содержимого ОЗУ и файла.

Команды анализатора

- adc** — аналого-цифровое преобразование, режимы «шаг» и «цикл»; возможность синхронизации ВУ.
- dac** — цифро-аналоговое преобразование; входная информация записана в файле на рабочем устройстве;
- emulator** — эмуляция ПЗУ, режимы «шаг», «цикл» и «останов в контрольных точках»; аналогично команде adc — возможность синхронизации ВУ;

Команды преобразования формата

- ass16** — работа с программами ОЭВМ серии К1816 (входная информация — файл, полученный в результате работы программ КА48.SAV и L48.SAV, выходная — файл загрузки ОЗУ);
- ass20** — работа с программами ОЭВМ серии К1820; используется с программой АС420.SAV



Структурная схема отладочного комплекса

Разработка ПО с помощью отладочного комплекса

Если при первом опробовании программы в реальном времени появится несоответствие между ожидаемым и полученным результатами, необходимо перейти к режиму эмуляции с остановом в контрольных точках. Задавая адреса контрольных точек (произвольное число) и используя логический анализатор (регистр RD), можно отслеживать на экране одновременно восемь сигналов, соответствующих заданному адресу. В этом же режиме можно «проходить» программу от одной контрольной точки до другой либо по шагам, либо в цикле и, таким образом, детально исследовать «узкие места» своей программы. Пошаговый режим с подключением выводов логического анализатора к соответствующим контактам отлаживаемого устройства позволит убедиться в правильности тех или иных операторов программы.

Телефон 452-01-61, Москва

Статья поступила 30.10.88

УДК 681.3—181.1.06

А. Г. Горбунов, С. В. Миненко, А. Д. Мруга

ОТЛАДЧНЫЙ МОДУЛЬ НА ОСНОВЕ ГЕНЕРАТОРА ТЕСТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Общее развитие цифровой вычислительной техники и электроники позволило формализовать и упростить многие этапы разработки программного обеспечения и аппаратных

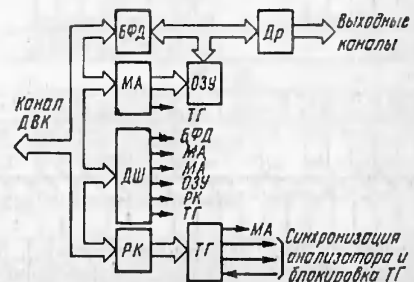
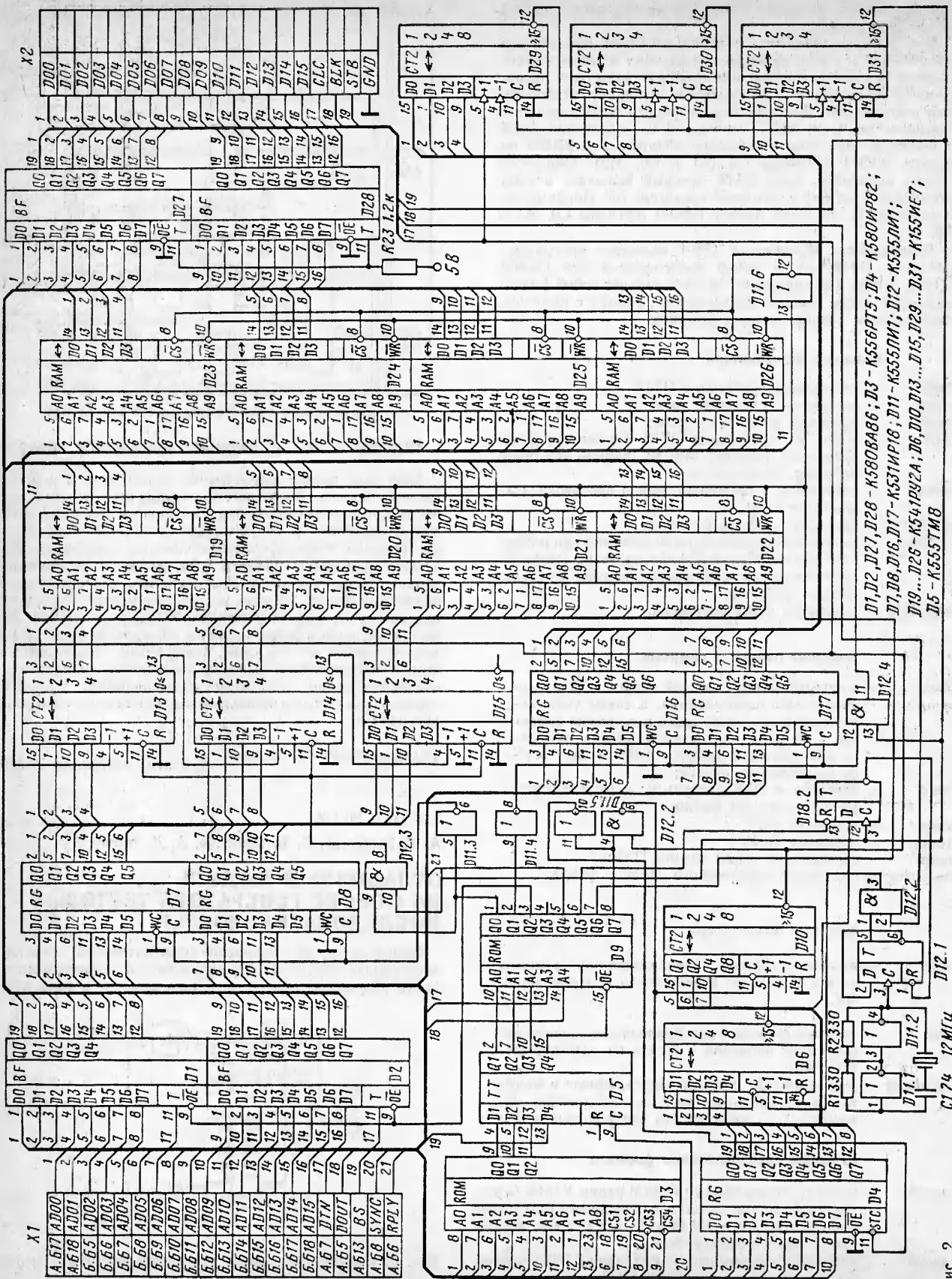


Рис. 1. Структурная схема генератора тестовых последовательностей



D1, D2, D27, D28 - К5808А86; D3 - К556РТ5; D4 - К5801Р82;
 D7, D8, D16, D17 - К531ПР18; D11 - К555ДМ1; D12 - К555ДМ1;
 D19... D26 - К541РУ2А; D6, D10, D13... D15, D29... D31 - К155ЕТ7;
 D5 - К555ТМ8

Рис. 2.

Карта прошивки ПЗУ D9(K155PE3)

- АДРЕС	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 X	FF	FF	FF	FF	FF	FF	FF	FF	FF	CF	AF	6F	EE	ED	FF	FF
1 X	CF	EB	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

средств. Однако отладка до сих пор сохранила черты своеобразного искусства, особенно когда дело касается больших схем и логически сложных программ. Предлагаемый простой комплект отладки основан на использовании генератора тестовых последовательностей, встроенного в микроЭВМ ДВК2М.

Генератор формирует и подает на проектируемую или диагностируемую систему входные воздействия, эмулирует отсутствующие дискретные устройства проектируемой системы. Обобщенная структурная схема генератора (рис. 1) включает дешифратор адреса (ДШ), входной буфер данных (БФД), драйверы (Др), управляемый тактовый генератор (ТГ), модификатор адреса (МА), ОЗУ, регистр команд (РК). Последовательность входных наборов, которую необходимо подать через драйверы Др на диагностируемую или проектируемую систему, записывается в ОЗУ. Частота тестирования и режим цикличности подачи воздействия устанавливаются по команде, занесенной в РК. Данные из памяти считываются последовательно от начального до конечного адреса.

Модификатор адреса содержит регистр адреса ОЗУ, в котором находится начальный адрес, и регистр хранения числа тактов в цикле. Выходной набор определяется запрограммированным алгоритмом тестирования (данные в ОЗУ), реакцией диагностируемой системы посредством приостановки ТГ и выдачи прерываний в ДВК.

Дешифратор адреса собран на элементах D3, D5, D9 (рис. 2), БФД — на D1, D2, Др — на D27, D28, ТГ и РК — на D4, D6, D10, D11.1, D11.2, D12.1, МА — на D7, D8, D12.4, D12.3, D13...D17, D18, D29...D31.

Генератор сопрягается с ДВК через интерфейс МПИ (ОСТ II 305.903-80). Адреса регистров генератора из области адресов внешних устройств ДВК (160000...177777), преобразованные микросхемой D3, сохраняются (по срезу сигнала SYNCX1) регистром D5 и дешифрируются ПЗУ. Содержимое ПЗУ D3 и D9 представлено в табл. 1, 2 соответственно. Генератор управляется посредством пяти программно доступных регистров: команд (176604_в), данных (176602_в), пуска (176600_в), адреса (176606_в) и счетчика тактов (176610_в). Разряды D0...D6 РК задают коэффициент деления частоты ТГ счетчиками D6, DA10; раз-

ряд D10 — режим цикличности (D10=0 — один цикл, D10=1 — непрерывный). На микросхемах D12.1, D18 собрана логика управления генератором с возможностью программного пуска (обращение к регистру пуска) и приостановкой (подачей сигнала на вход ВЛКХ2). Память генератора объемом 2Кх16 выполнена на микросхемах K541PY2A (время доступа 90 нс), что позволяет вести тестирование на частоте 10 МГц и обнаруживать ряд неисправностей, проявляющихся только на высоких частотах функционирования дискретной системы. Тактовый сигнал генератора CLK X2, сигнал канала нового цикла STB X2, блокировки ВЛК X2 и информационные сигналы D0...D15 выведены на высокочастотные розетки X2.

Программировать генератор желательно с применением адаптированных трансляторов, редакторов текстов, отладчиков ОС РАФОС. Удобен язык временных диаграмм. Настраивается комплекс в диалоговом режиме с помощью программ-утилит, каждая из которых предназначена для выполнения определенного вида работы. Нужную программу выбирают из меню команд. При использовании генератора в процессе отладки аппаратуры или комплексной отладки микропроцессорной системы требуется задавать последовательности входных наборов на языках проектирования конкретного МП*. В процессе отладки необходимо тестировать состояние большого числа шин. Для МП КР580ВМ80А это число равно 36 (16 адресных, 8 данных и 12 для управления и синхронизации), а для 8-разрядной микроЭВМ, построенной на основе комплекта серии К589, — не менее 70 (5 информационных шин по 8 разрядов, 9 разрядов микропрограммного адреса, 18 разрядов микрокоманды и не менее 3 — для управления и синхронизации). Для этого генератор целесообразно дополнить расширителем каналов (рис. 3). Информация с шин D0...D13 записывается в регистры P1...P4, адресуемые разрядами D14...D15, с помощью строблируемого ДШ. При обращении к P4 строб-импульс C4, задержанный элементом задержки (ЭЗ), фиксируется выходной набор в регистрах P5...P8, обеспечивая одновременность смены логи-

Таблица 1
Карта прошивки ПЗУ D3(K556PT5)

Адрес	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
01 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
02 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
03 X	F	F	F	F	F	F	F	F	F	3	F	2	F	1	F	0
04 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
05 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
06 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
07 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
08 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
09 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0A X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0B X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0C X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0D X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0E X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0F X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
10 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
11 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
12 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
13 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
14 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
15 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
16 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
17 X	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

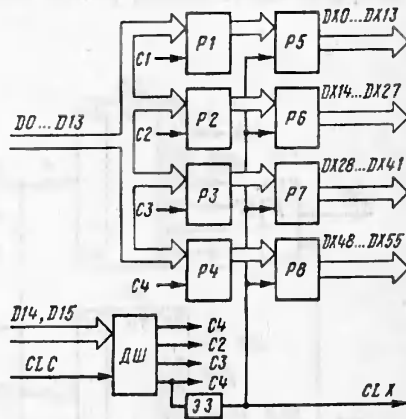


Рис. 3. Структурная схема расширителя каналов

* Микропроцессоры: В 3 кн.: Учеб. для ВТУЗов / Под ред. Л. И. Преснухина. — Кн. 3: Средства отладки, лабораторный практикум и задачник / Н. В. Воробьев, В. Л. Горбунов, А. В. Горячев и др. — М.: Высш. шк., 1986.

ческих комбинаций на выходах DX0...DX55. Такой способ увеличения каналов применим, когда можно допустить снижение быстродействия и уменьшение числа тактов в цикле. Предлагаемый генератор может служить основой для построения средств регистрации выходных реакций тестируемой системы в цифровой (логической) форме. Для этого вносятся некоторые изменения в принципиальную схему:

из МА исключаются регистр и счетчик числа тактов в цикле;

в ТГ вводится внешняя синхронизация;

изменяются адреса регистров (для одновременной работы двух и более устройств);

Др заменяются компараторами уровней для ввода выходных реакций в ОЗУ.

Анализ логических последовательностей на различных уровнях абстракции позволяет определить неисправность сначала в программно-временной области (конкретная команда программы системы или тестовой программы), а затем и в пространственной (конкретный разряд определенной шины). Для упрощения сравнения выходных реакций регистрируются только существенные для каждой конкретной задачи данные. Для этого в состав анализатора включены схемы запуска по кодовому слову и по несовпадению, реализованные на схемах ИСКЛЮЧАЮЩЕЕ — ИЛИ. Схема запуска по несовпадению решает и проблему борьбы с перемежающимися (случайными) неисправностями при ограниченных объемах ОЗУ.

Рассмотренные технические средства успешно применяются в практике отладки дискретных систем различного назначения (телевизионных адаптеров, волоконно-оптических линий связи в различных стандартах, контроллеров периферийных устройств ЭВМ).

Телефон 441-12-58, Киев, Мруга А. Д.

Статья поступила 24.04.87

УДК 681.3.06:326

И. В. Рябошапка

ОТЛАДОЧНЫЙ МОДУЛЬ ДЛЯ ОЭВМ КМ1816ВЕ48

Отладочный модуль (ОМ) совместно с управляющей микроЭВМ (УЭВМ) «Электроника К1-30» позволяет отлаживать программное обеспечение и проверять функционирование аппаратной части устройств на базе ОЭВМ КМ1816ВЕ48.

ОМ кроме блока ОЭВМ содержит промежуточный интерфейс, интерфейс адреса-данных, регистр команд (РК)

и представляет для УЭВМ набор портов ввода-вывода. При обращении к этим портам происходит обмен данными или вырабатываются управляющие сигналы.

Блок ОЭВМ включает ОЭВМ КМ1816ВЕ48 (D5), триггер D4.1 для пошагового выполнения команд, регистры D6, D7 для фиксации данных на младших разрядах порта P2 (рис. 1). Регистры используются в случае необходимости получить неизменяющиеся данные, выдаваемые портом P2 (младшие разряды порта P2 во время выполнения команды выдают данные и старшие разряды адреса следующей команды).

Подает на вход S триггера D4.1 высокого или низкого уровня задается соответственно пошаговый или автоматический режим работы ОЭВМ. При поступлении на вход С триггера D4.1 импульса CS3 ОЭВМ выполняет одну команду и делает останов. В данном ОМ используется пошаговый режим для того, чтобы во время останова УЭВМ записала в РК следующую команду для ОЭВМ.

РК содержит два буферных регистра D2 и D3 типа К589ИР12 и схему выбора. К этим регистрам ОЭВМ при чтении кода команды адресуются младшим разрядом адреса АО, который совместно с сигналом PМЕ выбирает необходимый регистр. Обращение к регистрам со стороны ОЭВМ циклическое, т. е. команды считываются последовательно из D2 и D3, а затем снова из D2 и т. д. Применение двух регистров и циклического режима считывания обусловлено тем, что команды могут быть одно- и двубайтовыми, и ОЭВМ может начать прием команды как с регистра D2, так и с регистра D3. После выполнения команды УЭВМ записывает в РК следующую команду и инициирует ее выполнение.

Шины блока ОЭВМ доступны для УЭВМ через интерфейс данных-адреса, состоящий из двух программируемых периферийных адаптеров (ППА) К580ВВ55 (рис. 2). Необходимые направления передачи каналов ППА устанавливаются УЭВМ записью соответствующего управляющего слова. Интерфейс адреса-данных позволяет считывать данные с портов P1 и P2, адрес следующей команды, записывать данные в порты P1 и P2.

Каналы С периферийных адаптеров не задействованы и могут быть использованы при расширении возможностей ОМ, например подключении внешнего ОЗУ ОЭВМ. В этом случае один канал С служит для обмена данными между ОЗУ и УЭВМ, а разряды другого канала С коммутируют необходимую структуру ОМ (включение внешнего или внутреннего ОЗУ, регистра команд, необходимых буферов и т. д.).

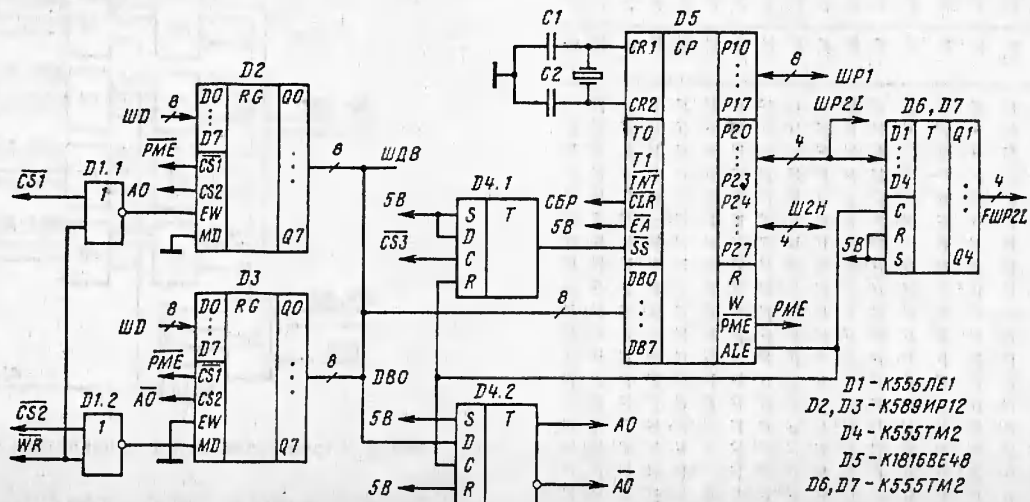


Рис. 1. Схема регистра команд и блока ОЭВМ

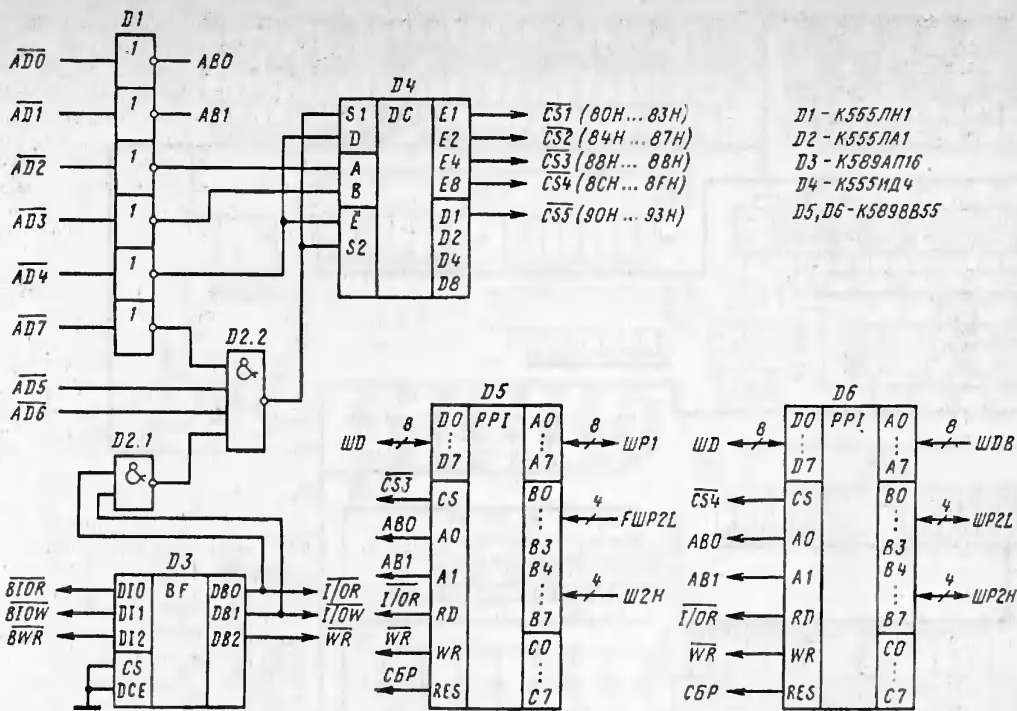


Рис. 2. Схема промежуточного интерфейса и интерфейса адреса-данных:
 WD — шина данных; AD0...AD7 — шина адреса; BIOR — сигнал чтения ВУ; BИOW — сигнал записи в ВУ; BWR — строб записи в ВУ; СБР — сброс

Промежуточный интерфейс содержит дешифратор адреса, формирующий сигналы CS1...CS5, буферные схемы сигналов чтения-записи I/OR, I/OW, WR и младших разрядов адреса ABO и AB1.

Для отладки программ в реальном масштабе времени необходимо к шине DB подключить внешнее ОЗУ программ. Выходы РК при обращении к ОЗУ переводятся в третье состояние. При работе с памятью программ (внешней или внутренней) с помощью РК выполнение отлаживаемой программы можно начать с любой точки. Для этого в РК записывается команда перехода на необходимый адрес и инициализации ее выполнения. После этого чтение команд переключается с РК на ОЗУ.

Программное обеспечение ОМ включает монитор и программу загрузки команд. Монитор позволяет читать и модифицировать содержимое регистров и внутреннего ОЗУ ОЭВМ. Обмен данными между ОЭВМ и УЭВМ происходит через порт P1 или P2.

Программа загрузки пересылает в РК команды отлаживаемой программы из ОЗУ управляющей микроЭВМ. В УЭВМ отлаживаемые программы хранятся с некоторым смещением по адресу. УЭВМ считывает адрес следующей команды с шин ШDB и ШP2L, прибавляет к этому адресу смещение, соответствующее начальному адресу отлаживаемой программы, и записывают адресуемый бит в один из регистров РК (в тот, который адресуете АО), а следующий байт — в другой регистр (на случай, если команда окажется двубайтовой); инициирует выполнение записанной команды; считывает адрес следующей команды и сравнивает с адресами контрольных точек останова. В случае совпадения адресов выполнение отлаживаемой программы прекращается и на логический пульт (экран дисплея) выводится адрес останова.

Достоинство ОМ в том, что он состоит всего из 13 микросхем (или 10, если не нужен обмен данными через шину P2) и имеет простое программное обеспечение.

270044, Одесса, пр. Шевченко, 1, Одесский политехнический ин-т; тел. 28-87-25

Статья поступила 3.03.88

УДК 681.326.3

О. Б. Малёжин, В. А. Верстаков

ПРОГРАММАТОР ПЛМ В СТАНДАРТЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

В работе [1] рассматриваются способы построения модуля программатора, обменивающегося с микроЭВМ по системной магистрали Multibus. Данная статья посвящена рассмотрению программатора, ориентированного на применение с микроЭВМ, имеющей систему команд микроЭВМ «Электроника 60».

Функциональная схема программатора ПЛМ КР556РТ1 и КР556РТ2 (рис. 1). Программатор обеспечивает информацией с микроЭВМ посредством интерфейса параллельного ввода-вывода, один из вариантов которого, реализованный на БИС серии К1801, приведен в [2]. При этом для передачи информации из микроЭВМ в программатор используется 16-разрядный регистр источника — РИ (ВД00В...ВД15В) и два разряда регистра состояний — РС (РС00В, РС01В), а для передачи информации из программатора — восемь разрядов регистра приемника — РП (ВВ00В...ВВ07В).

Возможны пять режимов работы программатора [3]: P1 — проверка матрицы И; P2 — проверка слоя программируемых инверторов и матрицы ИЛИ; P3, P4, P5 —

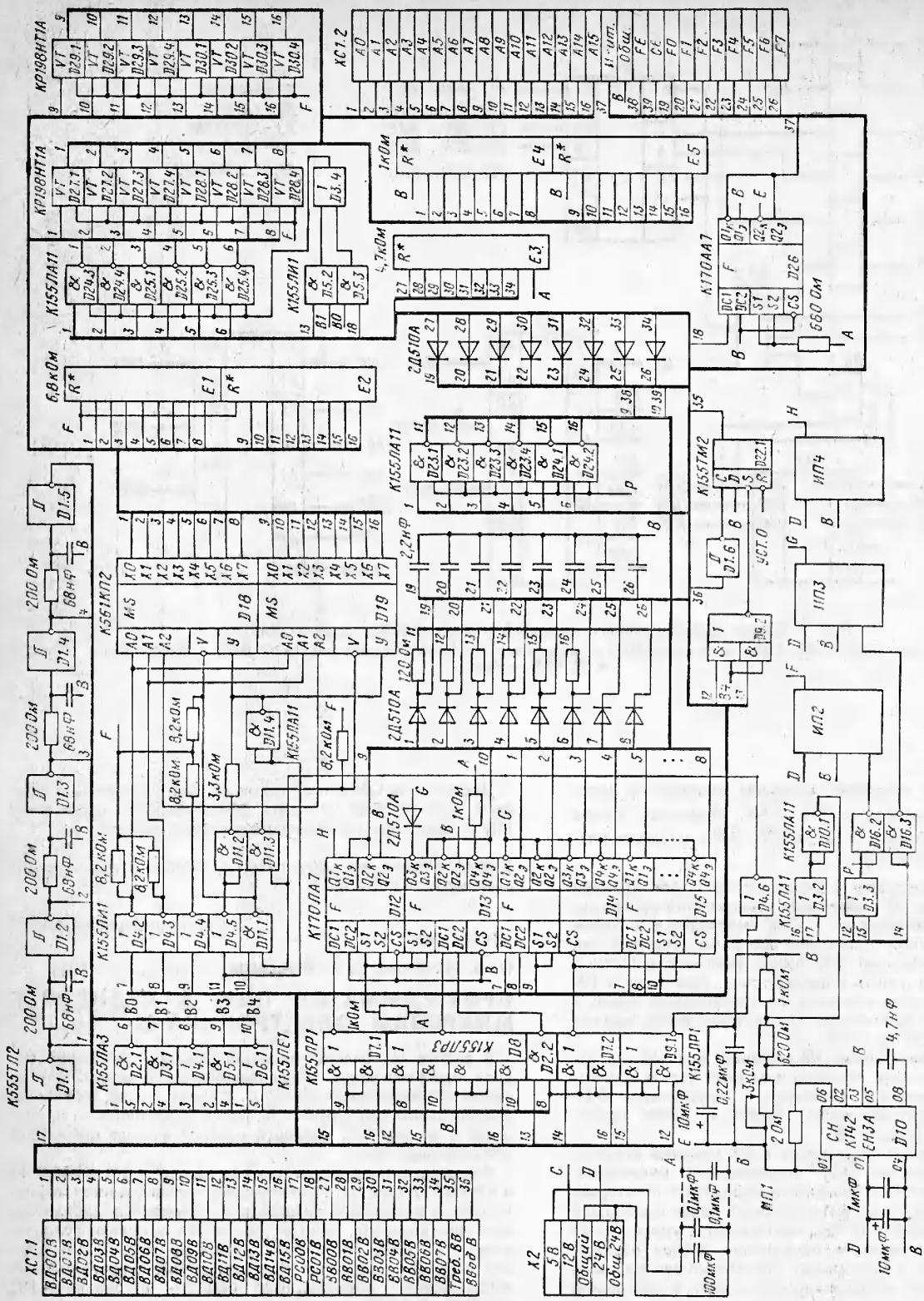


Рис. 1. Функциональная схема программатора П.Л.М

Режим	Назначение разрядов РИ															
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Режим 1	0	0	0	0	1	N	N	N	N	A	A	A	A	A	A	A
	Номер входа конъюнктора								Адрес конъюнктора							
Режим 2	0	0	0	1	0	0	0	0	0	0	N	N	N	N	N	N
	Номер входа конъюнктора								Адрес конъюнктора							
Режим 3	0	0	1	0	0	0	N	N	N	N	0	0	0	0	0	0
	Номер программируемого выхода								Адрес конъюнктора							
Режим 4	0	1	0	0	0	N	N	N	N	A	A	A	A	A	A	A
	Номер программируемого входа								Адрес конъюнктора							
Режим 5	1	0	0	0	0	0	A	A	A	A	N	N	N	N	N	N
	Адрес конъюнктора								Номер входа конъюнктора							

Рис. 2. Назначение разрядов РИ в различных режимах работы

программирование и контроль слоя инверторов, матрицы И и матрицы ИЛИ соответственно. При реализации этих режимов (рис. 2) номера их кодируются пятью разрядами РИ (ВД11В...ВД15В). Адреса (А) и номера (N) входов (выходов) представляются в инверсном виде.

Программатор содержит четыре стабилизированных источника питания (ИП1...ИП4), каждый из которых построен на основе стабилизатора напряжения К142ЕНЗ. При этом ИП1...ИП4 формируют напряжения Е, F, G, H так, что на входах программируемой ПЛМ в зависимости от внешних управляющих сигналов обеспечиваются напряжения:

$$\begin{aligned}
 U_{\text{пит}} &= [0...0,5 \text{ В}; 5 \text{ В} \pm 5\%; 8,75 \pm 0,25 \text{ В}]; \\
 A_i &= [0...0,8 \text{ В}; 3...5,5 \text{ В}; 10 \text{ В} \pm 5\%]; \quad (i=0...15); \\
 F_j &= [0...0,8 \text{ В}; 3...5,5 \text{ В}; 10 \text{ В} \pm 5\%; 17 \pm 1 \text{ В}]; \quad (j=0...7); \\
 FE &= [0...0,8 \text{ В}; 17 \pm 1 \text{ В}]; \\
 CE &= [0...0,8 \text{ В}; 3...5,5 \text{ В}; 10 \text{ В} \pm 5\%].
 \end{aligned}$$

Коммутатор D 26 служит для подачи на вход ПЛМ ($U_{\text{пит}}$) напряжения $B=0...0,5 \text{ В}$ или напряжения Е от ИП1 (см. рис. 1). Схема, построенная на основе мультимплексоров D18, D19, логических схем D24, D25.3, D25.4 и эмиттерных повторителей (ЭП) D27...D30, предназначена для формирования на входах A_i ($i=0,15$) напряжений логических уровней или напряжения F от ИП2.

Микросхемы D12...D15 служат для коммутации контролируемых и программируемых напряжений и подачи их

УКАЖИТЕ НОМЕР ДИЗЬЮНКТОРА (КОД: 0-7) --- 0
 УКАЖИТЕ ПРОГРАММИРУЕМОЕ СОСТОЯНИЕ КАЖДОГО ИЗ 48 ВХОДОВ ДИЗЬЮНКТОРА:
 А - ЕСЛИ КОНЪЮНКЦИЯ НЕ УДАЛЯЕТСЯ, Т.Е. ПЕРЕМЫЧКУ ОСТАВЛЯЮТ ЦЕЛЫЙ
 - - ЕСЛИ КОНЪЮНКЦИЯ УДАЛЯЕТСЯ, Т.Е. ПЕРЕМЫЧКУ ПРОЖИГАЮТ

ВХОД	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
01	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
17	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
18	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
19	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
20	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
21	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
22	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
23	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
24	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
25	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
26	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
27	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
28	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
29	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
30	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
31	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
32	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
33	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
34	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
35	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
36	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
37	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
38	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
39	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
40	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
41	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
42	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
43	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
44	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
45	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
46	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
47	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO
48	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO	-	NO

ДЛЯ ПРОДОЛЖЕНИЯ РЕЖИМА НАЖМИТЕ КЛАВИШУ "BK", ДЛЯ ВЫХОДА - ЛЮБУЮ КЛАВИШУ
 ДЛЯ ПЕРЕХОДА В ТРЕБУЕМЫЙ РЕЖИМ УКАЖИТЕ СООТВЕТСТВУЮЩИЙ НОМЕР (1...5) ---

Рис. 3. Реализация диалогового режима при программировании матрицы ИЛИ

КОДИРОВКА ПЛМ

СИСТЕМА ОБОЗНАЧЕНИЙ:																
A	-	ЦЕЛЫЙ ПЕРЕМЫЧКА	1	-	ПРЯМОЕ ЗНАЧЕНИЕ ВХОДНОЙ ПЕРЕМЕННОЙ	0	-	ИНВЕРСНОЕ ЗНАЧЕНИЕ ВХОДНОЙ ПЕРЕМЕННОЙ	N	-	КОНЪЮНКЦИЯ НЕ ЗАВИСИТ ОТ ВХОДНОЙ ПЕРЕМЕННОЙ	-	-	ПРОЖИГАЯ ПЕРЕМЫЧКА В СЛОЕ ИНВЕРТОРОВ ИЛИ В МАТРИЦЕ "ИЛИ"		
МАТРИЦА "И"																
ВХОДНАЯ ПЕРЕМЕННАЯ (A0...A15)																
1	01	11	21	31	41	51	61	71	01	91	101	111	121	131	141	151
18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02
1	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
3	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
МАТРИЦА "ИЛИ"																
ВЫХ. ФУНКЦИЯ (F0...F7)																
0	1	2	3	4	5	6	7									
18	17	16	15	14	13	12	11									
A	A	A	A	A	A	A	A									
-	1	-	1	-	1	-	1									
A	A	-	A	-	A	-	A									
-	1	-	1	-	1	-	1									
A	A	-	A	-	A	-	A									
47	1	0	N	1	0	N	1	0	N	1	0	N	1	0	N	1
48	1	1	A	A	0	0	N	N	1	1	0	0	N	N	A	A

Рис. 4. Таблица кодирования ПЛМ

в зависимости от управляющих сигналов, формируемых микросхемами D7, D8, D2.2, на входы F_j ($j=0,7$), Е, СЕ.

Часть устройства на основе микросхем D1, D2.1...D6.1 предназначена для получения последовательности сдвинутых относительно друг друга импульсов В0...В4, используемых для формирования временных диаграмм Р1...Р5. Запускается устройство по приходу импульса на вход РС00В.

Программа работы программатора в режимах Р1...Р5 написана на Макроассемблере и позволяет реализовать диалоговый режим (рис. 3) проверки и программирования ПЛМ. При этом используется алгоритм программирования, заключающийся в циклической подаче программируемых импульсов основной серии до тех пор, пока не будет зафиксирован прожиг соответствующей перемычки, после чего подается дополнительная серия программируемых импульсов. На рис. 3 представлен фрагмент информации, отображаемой на экране дисплея при программировании диэъюнктора 0 матрицы ИЛИ, на рис. 4 — таблица программирования для требуемого состояния ПЛМ.

Конструктивно программатор выполнен на плате размером 135×220 мм, предназначенной для установки в блоки сопряжения микроЭВМ «Электроника 60» или ДВК. Требуется один внешний источник питания +24 В.

Телефон 532-99-91, Москва

ЛИТЕРАТУРА

- Дьянов А. П., Щелкунов Н. Н. Технические средства программирования логических схем // Микропроцессорные средства и системы.—1986.— № 2.— С. 77—80.
- Малежин О. Б., Крылик Н. О., Преснухин Д. Л. Интерфейсы параллельного ввода-вывода микроЭВМ «Электроника 60» на основе БИС серии К1801, К588 // Микропроцессорные средства и системы.—1987.— № 5.— С. 80, 81.
- Щелкунов Н. Н., Дьянов А. П. Процедуры программирования логических матриц // Микропроцессорные средства и системы.—1986.— № 2.— С. 71—76.

Статья поступила 2.01.88.

пускается ЦЗ. ЭСКТ сохраняет попеременно в одном из двух контрольных массивов на ШНЦМД данные ФЗ, буферы ввода-вывода, относящиеся к ПП, адрес и состояние процессора на момент прерывания, содержимое аппаратного буфера ЦМД, буфер УОИ, регистры процессора и устройств, используемых системой. Кроме того, ЭСКТ вызывает подпрограмму из ПП, сохраняющую (перепиывающую в СОММО-область) регистры устройств, используемых ПП. При пуске инициализатор вызывает подпрограмму из ПП, восстанавливающую состояние этих регистров. Так как объем ШНЦМД равен 32 Кбайт, необходимо ограничение на объем контрольного массива: 16 Кбайт.

При установке ПП таблица векторов прерывания (его сопровождающая) перепиывается в область системных таблиц. В ядре ОС предусмотрено 20 входов по прерыванию от ВУ. Адреса этих входов записываются в соответствующие ячейки векторов прерывания, и при поступлении прерывания от ВУ управление получает не обработчик прерывания, а ядро ОС, которое выделяет пространство ОЗУ и передает затем управлению ЗОПВУ. Предоставление работы программе-обработчику прерывания происходит через 100 мкс — это минимальное время реакции на внешнее событие.

Интерпретатор командной строки обрабатывает входную строку символов и выполняет указанные действия. Кроме того, нажатие на любую из девяти функциональных клавиш вызовет выполнение подпрограммы, адреса которых находятся в таблице функциональных клавиш ПП.

В функции интерпретатора входят: установка ПП (с ЦМД или ПЗУ), прерывание и снятие ПП, установка флага завершения ПП, добавления ПП; ввод (коррекция) астрономического времени, вызов диагностического текста, перенаправление ввода-вывода на ФЗ, перенаправление ввода-вывода на ЦЗ, установка одноразовых задач по нажатию клавиш Ф1...Ф3. Добавляться может только ПП, состоящий из ЦЗ и ЗОПВУ.

Драйверы устройств. В ОС включены драйверы следующих устройств: клавиатуры, УОИ и накопителей на ЦМД (НЦМД). Драйвер клавиатуры работает по опросу и вводит код нажатой клавиши. Драйвер УОИ обеспечивает генерацию букв русского и латинского алфавитов, цифр и других специальных символов, а также графических примитивов (точка, отрезок, ломаная, дуга, окружность). Он имеет следующий набор команд: установить курсор в заданную позицию, зажечь или погасить точку, начертить отрезок с заданными координатами конца, начертить ломаную по точкам, начертить окружность, начертить дугу, вывести символ (КОИ-7) или специсимвол, забить (стереть, последний символ), вернуть каретку, перевести строку, стереть текущую строку, стереть экран, конец сообщения. Система команд может быть расширена до 32.

Драйвер накопителей НЦМД управляет штатным и съемным накопителями. Он выполняет следующие команды: идентификацию текущего накопителя, подгон заданной страницы, считывание страниц в буфер, запись из буфера в накопитель, разметку накопителя. При выполнении команд контролируется нумерация страниц.

Файловая система. ОС имеет оригинальную файловую систему (ФС) для НЦМД. На штатном накопителе ФС поддерживает два равновеликих файла, сохраняющих контрольную точку ФЗ на случай сбоя, отключения питания и т. п. На съемном накопителе ФС обеспечивает автоматическую каталогизацию рабочих файлов (до 255 файлов суммарного объема 31744 байт) и в принципе обходится без операции устранения пустот между файлами (SQUEEZE).

ФС выполняет следующие операции: запись, чтение и ударение файлов, чтение каталога файлов, форматирование и инициализацию накопителей, прямой доступ к файлам. После выполнения операции устанавливается признак завершения (успешно-неуспешно). Две смежные страницы 1024 и 0 отводятся под системные нужды; следующие 31 страница (1...31 — всего $31 \times 32 = 992$

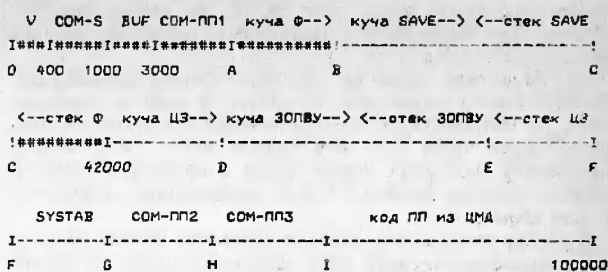


Рис. 2. Фактические значения адресов А...I в фиксированных ячейках COM-S.

- V — область векторов прерываний;
- COM — общий блок ядра системы, в который входят информация о текущем состоянии системы, распределение ОЗУ, активные задачи, адреса буферов ввода-вывода, их размеры и т. д.;
- BUF — буферы ввода с клавиатуры СПО-2 и вывода на УОИ, буфер сохранения регистров процессора и устройств;
- COM-ПП1 — общий блок ПП, установленного первым, COM-ПП2 содержит буферы ввода-вывода, используемые ПП, и все статические данные ФЗ, ЦЗ и ЗОПВУ; куча Ф, стек Ф — куча и стек фоновой задачи;
- SYSTAB — системные таблицы, в которых хранится информация об адресах вызова ЗОПВУ, ЦЗ и точек входа в ФЗ по нажатию функциональных клавиш;
- ПП с ЦМД — код всех программных составляющих ПП, загруженного с ЦМД

байт) — под каталог файлов, оставшихся 992 — под файлы. Файлы могут располагаться в несмежных страницах.

Библиотека системных подпрограмм и математических функций включает в себя модули арифметической и символической поддержки, подобные модулям из библиотек FT4LIB и PASLIB, а также некоторые сервисные подпрограммы, такие как контекстный поиск, перемещение массивов, выдача текущего времени и сохранение-восстановление регистров в стеке. В арифметическую и символическую поддержку входят целочисленная арифметика и арифметика с плавающей точкой, а также подпрограммы преобразований типа FLOAT TO INTEGER, FLOAT TO STRING и др. Возможно включение тригонометрических функций.

Стратегия распределения ресурсов

Операционная система предоставляет каждой задаче ресурс ОЗУ в виде трех массивов ячеек, каждый из которых имеет свои особенности программного доступа. Первый массив — фиксированной длины, адрес его начала указывается в фиксированной ячейке блока системных флагов и состояний (COM-S). Второй массив — переменной длины, расширяемый вверх, с организацией типа «куча». Третий массив — также переменной длины, это область, выделяемая под стек задачи.

Общее распределение ОЗУ с расчетом на наиболее сложную комбинацию задач представлено в виде диаграммы на рис. 2. Штриховкой показана область ОЗУ, сохраняемая в контрольном массиве.

Прикладные пользовательские пакеты

Состав пакета: машинный код фоновой задачи в формате переместимого загрузочного модуля; коды циклических задач и обработчиков прерываний; таблицы смещений входов в каждую задачу пакета относительно начала пакета, векторов прерывания и циклических задач; значения — объемы требуемой COMMON-области и всего пакета, а также размеры вышеуказанных таблиц.

Если пакет прошивается в ПЗУ на фиксированных адресах, то снимается требование о переместимости, но добавляется требование исполнения в ПЗУ. При указанной структуре ПП можно производить замену ПП просто путем замены микросхем ПЗУ, а также загружать ПП со съемной кассеты на ЦМД.

Пакет содержит девять специальных одноразовых задач, отнесенных к классу ЦЗ. В системе они пользуются ресурсами ЦЗ, однако вызываются только один раз на каждое нажатие клавиш Ф1...Ф9 на клавиатуре СПО-2. При-

ритет этих задач выше, чем ЭСКТ, но ниже, чем ЦЗ, поэтому они выполняются лишь тогда, когда нет запроса на ЦЗ.

В ПП может быть до 20 обработчиков прерывания. ЗОПВУ имеют наивысший приоритет, и сами не прерываются. На них действуют все ограничения ЦЗ, кроме аварийного снятия через 1 с. Все задачи могут использовать библиотеку системных подпрограмм и драйверов, регистр данных таймера, ячейки СОМ-S, содержащие астрономическое время.

В распоряжении пользователя находятся четыре колодки для подключения своих ПЗУ общим объемом 32 Кбайт и съемный накопитель ЦМД емкостью 32 Кбайт.

Телефон 42—15-40, Фрунзе

ЛИТЕРАТУРА

1. Устройства числового программного управления «Электроника МС 2101». Техническое описание и инструкция по эксплуатации.
2. Устройство ВЧС. Техническое описание.
3. Сингер М. Мини-ЭВМ PDP-11: Программирование на языке ассемблера и организация машины. Пер. с англ. — М.: Мир, 1984.
4. МикроЭВМ: В 8 кн.: Практик. пособие / Под ред. Л. Н. Преснухина. Кн. 4: Управляющие системы «Электроника НЦ» / Ю. Е. Чичерин. — М.: Высшая школа, 1988.

Статья поступила 15.04.88

Предлагаемые ниже статьи содержат описание средств отладки программ для ЭВМ, архитектурно совместимых с СМ ЭВМ, и для микропроцессорных систем на базе микросхем серий К1801, К1806. Одновременно статьи являются развернутым описанием базовых возможностей ПО комплекса «Электроника НЦ-803» (последняя статья — одна из конкретных реализаций комплекса). Нужда в средствах отладки программ очень велика, и, безусловно, многие программистские коллективы разработали для себя отладочный инструментарий². Ценность этих средств, как представляется, заключается в целостном подходе к процессу отладки, т. е. в наличии единой аппаратной, программной и языковой базы для проведения всех этапов отладки и тестирования программ и для комплексной отладки (интеграции) аппаратно-программных средств систем.

На базе символьного отладчика разработаны: модели систем на МП серий К1801, К1806, обеспечивающие возможность отладки под SI- или TS-монитором ОС РАФОС любых программ для указанных систем. Модели нестандартных устройств и абонентов разрабатываются пользователем и подключаются при генерации модели систем. Возможности отладчика дополнены средствами сбора разнообразной статистики об отлаживаемой программе;

ПО эмуляторов МП серий К580, К1801, К1806. Возможности отладчика расширены за счет дополнительных аппаратных средств в эмуляторах для поддержки отладки и тестирования в режиме реального времени.

УДК 681.3

А. В. Александров, М. И. Кушнир, А. В. Харламов

СИМВОЛЬНЫЙ ОТЛАДЧИК ДЛЯ ЯЗЫКА MACRO-11

В процессе разработки программ проектировщики, естественно, допускают много ошибок, причем в сложных системах на устранение этих ошибок (отладку) уходит основная часть времени и труда (до 60...70 %). Этим определяется значимость создания мощных и удобных средств отладки программ. Минимальный набор таких средств должен обеспечивать (пультом управления ЭВМ или пультовым отладчиком) возможность отображения и изменения значений программно-доступных объектов, а также запуска программы в автоматическом и пошаговом режимах и ее останова при заданных значениях счетчика команд.

Отладка программ настолько утомительна и трудоемка, что в состав операционных систем ОС РВ, РАФОС, ФОДОС, ОС ДВК входят отладчики (ODT, TODT) [1], включающие несколько более широкий набор функций, а главное, реализующие их в более удобной форме. Основным недостатком штатных отладчиков — ориентация на машинные коды, а не на язык ассемблера. В [6] описан отладчик ODT/X с расширенным набором отладочных функций. Недостаток такого отладчика — невозможность использования символьных меток. Необходимость работы с адресами вынуждает пользователя постоянно обращаться к листингу программы для определения адреса нужной переменной или команды, на что уходит значительная часть времени при отладке.

Предлагаемый символьный отладчик применяется для отладки программ в терминах языка ассемблера. Он функционирует под управлением ОС РАФОС (в том числе под TS-монитором), ФОДОС, ОС ДВК. Ведется работа по

адаптации к среде ОС РВ. На базе этого отладчика разработаны средства отладки для языка Фортран-IV [2] и ПО комплексов отладки МП-систем «Электроника НЦ-803» [3] и программ «Электроника МСО404» [5], начата разработка средств отладки для языка Си.

Подготовка программ к отладке

Большинство отладочных функций требует задания адресов для идентификации элементов данных или программ, на которые производится ссылка. Достоинство описываемого отладчика — возможность использования как адресов, так и символьных имен и меток (идентификаторов), определенных в исходном тексте программы на языке ассемблера. Для этого в технологическую цепочку подготовки программы к исполнению включена процедура формирования таблицы имен, которая по листингам и карте загрузки программы создает три файла: оверлейных сегментов, имен объектных модулей и имен переменных. Для неоверлейных программ файл оверлейных сегментов не формируется.

Собственно отладчик поставляется либо в виде отдельной программы (для TS- или SI-монитора при объеме памяти не менее 96 Кбайт), либо в виде объективной библиотеки, компонуемой с программой пользователя. Командный файл для компоновки формируется специальной програм-

¹ Горовой В. Р., Васильев Н. П. Комплекс средств отладки микропроцессорных систем «Электроника НЦ-803» // Микропроцессорные средства и системы. — 1987. — № 1.

² Белоусов Г. В., Болмазов И. В. и др. Инструментальная система отладки программ // Микропроцессорные средства и системы. — 1989. — № 1.

мой генерации отладчика с включением в него только указанных пользователем функций. Это позволяет найти компромисс между возможностями отладчика и объемом программы.

Описание отладочных функций

Отладочные функции запускаются одно- и двухсимвольными командами, чаще — односимвольными. Параметры команд (числа, имена) могут редактироваться символами (ЗБ) (стереть последний символ) и СУ/У (стереть весь параметр). Имя файла таблицы имен (имена файлов различаются только расширением) указывается отладчику с помощью команды NL (имя файла), после которой допустимы обращения ко всем именам, включенным в файл имен переменных. Ввод имени переменной должен начинаться с команды L (например, L \$END, LSTART). Если в различных программах используются одинаковые имена переменных, то для доступа к нужной переменной применяется команда L (имя программы), которая настраивает отладчик на работу с таблицей имен указанной программы.

Основные команды отладчика приведены в таблице. В угловых скобках указаны параметры команд. Параметры, заключенные в квадратные скобки, являются необязательными или имеют значение по умолчанию. Символом * помечены команды, которые могут быть включены или не включены в отладчик на этапе компоновки отлаживаемой программы с библиотекой. Во всех командах в качестве параметров (адрес) или (значение) может использоваться имя.

Команды отладчика

Синтаксис команды	Отладочная функция
[<адрес>;]G	Запустить программу с (адреса)
[<адрес>;]P	Выполнить одну команду с (адреса)
[<адрес>;](<N> B	Задать точку останова по (адресу) с номером (N)
:[<N>]B	Удалить точку останова с номером (N). Если N не указан, то удалить все точки останова
TB	Просмотреть таблицу точек останова по счетчику команд
*[<адрес>;<N>]W	Установить точку останова с номером (N) по изменению байта с (адресом)
*TW	Удалить точку останова с номером (N). Если (N) не указан, то удалить все точки останова
[<адрес>]/	Просмотреть таблицу точек останова по изменению байта
[<адрес>]0	Открыть слово с (адресом)
[<значение>] (ПС)	Открыть байт с (адресом)
[<значение>]a	Изменить содержимое открытого адреса и открыть следующий адрес
[<значение>] (BK)	Изменить содержимое открытого адреса и закрыть адрес
[<значение>]@	Изменить содержимое открытого слова и открыть слово по абсолютному адресу
[<адрес>;] (K) ID	Дизассемблировать (K) команд, начиная с адреса. По умолчанию (K)=1
[<адрес>;] (K) A	Показать и изменить (K) байт памяти, начиная с (адреса), в формате ASCII
PX[<значение>]	Показать и изменить значение слова в формате RADIX=50
*FD[<значение>]	Показать и изменить значение слова в формате целого десятичного числа одинарной точности

Синтаксис команды	Отладочная функция
*FJ[<значение>]	Показать и изменить значение двух слов в формате целого десятичного числа двойной точности
*FE[<значение>]	Показать и изменить значение двух слов в формате десятичного числа с плавающей запятой одинарной точности
*FD[<значение>]	Показать и изменить значение четырех слов в формате десятичного числа с плавающей запятой двойной точности
*NC[<метка>;] (файл)	Запустить командный (файл), начиная с (метки), по умолчанию — с начала (файла)
*%	Признак приостановки командного файла
*!	Признак комментария в командном файле
*aP	Пошаговое исполнение командного файла
*aG	Автоматическое исполнение командного файла
*[<0>]NP (файл)	Включить режим протоколирования всей информации, отображаемой на экране, в (файл)
*[<1>]NP (файл)	Включить режим протоколирования информации, входимой в клавиатуры терминала, в (файл)
*;NP	Завершить протоколирование
*[<адрес>;]SA	Включить режим записи в память команд, вводимых в мнемоническом коде, начиная с (адреса)
*[<адр1>;<адр2>]MB	Установить границы области памяти
*; MA	Определить текстовый аргумент поиска
*MA	Поиск текста в установленных границах
(аргумент);(маска) MD	Поиск битового (аргумента) по маске в установленных границах
*[<адрес>;]MP	Поиск обращений к (адресу) в установленных границах
*[<блок>]MS (файл)	Переписать области памяти в установленных границах в (файл), начиная с указанного блока. Если (блок) не указан, то в новый файл
*[<блок>]ML (файл)	Перепись (файла), начиная с (блока) в память, в установленных границах. По умолчанию (блок)=0
*[<K>]MF	Отформатировать область памяти в установленных границах (K) первыми байтами этой области. По умолчанию (K)=2
[<адрес>;]TA	Определить положение (адреса) в программе (показать метку, ближайшую к (адресу), имя программы, где определена метка, и номер сегмента)
*NA (файл)	Включить режим доступа на диск
*NO (файл)	Включить режим отладки оверлейных программ

Следует дополнительно сказать о двух последних командах отладчика.

Команда **NA** обеспечивает доступ к любой программе, расположенной на внешнем носителе (таким образом, отладчик заменяет собой системную программу PATCH), и ко всему внешнему носителю, что позволило наряду с описанными в [4] возможностями восстановить информацию на подпорченных носителях.

Команда **NO** дает возможность вести полноценную отладку оверлейных программ с доступом и установкой точек останова в нерезидентных оверлейных сегментах, а также с доступом к любой оверлейной программе, расположенной на внешнем носителе.

Из вышесказанного видно, что описанный отладчик имеет более богатый набор отладочных функций по сравнению с отладчиками ODT, TODT или ODT/X. По опыту эксплуатации можно сделать вывод, что отладчик является сокращающим время разработки программ. Кроме приведенных в [4] возможностей отладчик поддерживает диалоговый режим отладки и автоматический режим тестирования.

```

>ВКЛЮЧИТЬ РЕЖИМ ДОСТУПА НА ДИСК К ОВЕРЛЕЙНОЙ ПРОГРАММЕ ВТО
>НО
>ВВЕДИТЕ ИМЯ ФАЙЛА :ВТО
>ДОСТУП НА ДИСК? (Y/N)NY
>ОВЕРЛЕЙНЫЙ ДОСТУП НА ДИСК
>ВКЛЮЧИТЬ РЕЖИМ ПРОТОКОЛИРОВАНИЯ В ФАЙЛ DK:PROT.TXT
>ИМЯ ФАЙЛА ПРОТОКОЛА :PROT.TXT
>НАСТРОИТЬСЯ НА 1-Й ОВЕРЛЕЙНЫЙ СЕГМЕНТ ПО ИМЕНИ
>ИОВ
>ИЗМЕНИТЬ СОДЕРЖИМОЕ БАЙТА С АДРЕСОМ .ВЕСС1+15
%L.ВЕСС1:110074 +15.000 =0 01
>ПЕРЕКЛЮЧИТЬСЯ В РЕЖИМ ОТЛАДКИ ОВЕРЛЕЙНОЙ ПРОГРАММЫ
>ОО
>ОВЕРЛЕЙНЫЙ ДОСТУП
>ДИАГНОСТИЦИРОВАТЬ 2 КОМАНДЫ И ЗАДАТЬ ОСТАНОВ В 1-М ОВЕРЛЕЙНОМ СЕГМЕНТЕ
110532 120
110532 012700 000010 ST01 : MOV 000010 R0
110534 012701 000011 MOV 000011 R1
>НАСТРОИТЬСЯ НА 2-Й ОВЕРЛЕЙНЫЙ СЕГМЕНТ ПО ИМЕНИ ПРОГРАММЫ
11082
>ИЗМЕНИТЬ КОМАНДЫ С МЕТКОЙ ВТО2
110824 110820
110824 MOV #20,R#Z
>УСТАНОВИТЬ ТОЧКИ ОСТАНОВА ПО АДРЕСУ 110620
110621:110620:B
>НАСТРОИТЬСЯ НА КОРНЕВОЙ СЕГМЕНТ ПО ИМЕНИ ПРОГРАММЫ
11088
>ИЗМЕНИТЬ СОДЕРЖИМОЕ РЕГИСТРА R6
R#6 000000 1000
>Исполнить 1 команду с метки ST0B
110800:110444 JF
PC R0 R1 R2 R3 R4 R5 RP N Z UC
110400 000000 000000 000000 000000 000000 000000 001000 0 0 0 0
110450 012701 000001 MOV 000001 R1
>Проск программы с текущего адреса
>O>
R:000 001
PC R0 R1 R2 R3 R4 R5 RP N Z UC
110536 000010 000001 000002 000003 000004 000005 001000 0 0 0 0
110534 012701 000011 MOV 000011 R1
>ОТРАБОТАТЬ 0-Я ТОЧКА ОСТАНОВА. ОСТАНОВ В 1-М ОВЕРЛЕЙНОМ СЕГМЕНТЕ
>ОПРЕДЕЛИТЬ ПОДЛОЖИЕ АДРЕСА ОСТАНОВА В ПРОГРАММЕ
>ПОКАЗАТЬ ВЛИЯЮЩИЕ МЕТКИ, СМЕЩЕНИЕ ОТ ЭТОЙ МЕТКИ К АДРЕСУ,
>ИМЯ ПРОГРАММЫ, В КОТОРОЙ ОПРЕДЕЛЕНА МЕТКА, И НОМЕР СЕГМЕНТА
110534:ТА-ВТО1 +000004 001 001
>ПРОДОЛЖИТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ
>O>
R:001 002
PC R0 R1 R2 R3 R4 R5 RP N Z UC
110624 000020 000011 000012 000013 000014 000015 001000 0 0 0 0
110624 012701 000021 MOV 000021 R1
>ОТРАБОТАТЬ 1-Я ТОЧКА ОСТАНОВА. ОСТАНОВ ВО 2-М ОВЕРЛЕЙНОМ СЕГМЕНТЕ
>ПОКАЗАТЬ ИНФОРМАЦИЮ ОБ ОВЕРЛЕЙНЫХ СЕГМЕНТАХ
>О
>СЕГМ. АДРЕС В. АДР. РАМЕР БЛОК
000 000000 110526
001 110530 110614 000044 000111 ; РЕЗИДЕНТ
002 110616 110700 000064 000112 ; РЕЗИДЕНТ ; НАСТРОЕН
- 110700
>ВКЛЮЧИТЬ РЕЖИМ ОТЛАДКИ ОВЕРЛЕЙНОЙ ПРОГРАММЫ И ЗАВЕРШИТЬ ПРОТОКОЛИРОВАНИЕ
>ИО
>ИП
>

```

Пример использования отладчика, в котором изменяются оверлейная программа и данные на диске и отлаживается программа. Символом выделены комментарии. Адреса, соответствующие именам, печатаются отладчиком.

Телефон 532-82-62, Москва

ЛИТЕРАТУРА

1. Валикова Л. И., Вигдорчик Г. В., Воробьев А. Ю., Лукин А. А. Операционная система СМ ЭВМ РАФОС.— М.: Финансы и статистика, 1984.
2. См. статью на с. 62 настоящего выпуска.

3. Горовой В. Р., Васильев Н. П. Комплекс средств отладки микропроцессорных систем «Электроника НЦ-803» // Микропроцессорные средства и системы.— 1987.— № 1.— С. 44—46.
4. См. статью на с. 63 настоящего выпуска.
5. См. статью на с. 67 настоящего выпуска.
6. Белоусов Г. В., Болмазов И. В. и др. Инструментальная система отладки программ // Микропроцессорные средства и системы.— 1989.— № 1.— С. 47—48.

Статья поступила 12.07.88

УДК 681.3

А. В. Харламов, М. И. Кушнир СРЕДСТВА УПРАВЛЕНИЯ ТЕСТИРОВАНИЕМ И ОТЛАДКОЙ ПРОГРАММ

В процессе исполнения программа взаимодействует с внешней средой (внешними устройствами и другими программами), которую обычно приходится моделировать. На этапе автономной отладки программы моделирование внешней среды обеспечивает человек, формируя в диалоговом режиме желательные воздействия абонентов. Символьный отладчик [1] — средство автономной отладки отдельных программ. Однако за автономной отладкой следует комплексная отладка нескольких программ, а также, как правило, неоднократные модификации автономно отлаженной программы, инициированные обнаружением ошибок и изменением требований к программе. Диалоговый режим тестирования программы достигает 30...40 % общих затрат на разработку программ и значительно определяют качество созданного программного продукта [3].

Для автономно отлаженной программы должны быть разработаны и отлажены тесты, позволяющие в автоматическом режиме проверить работоспособность программы; при исполнении тестов моделирование внешней среды также должно выполняться в автоматическом режиме. При комплексной отладке программ возникает задача отслеживания более сложных условий для останова, чем значения счетчика команд или изменения ячейки. Алгоритмы тестирования и условия останова могут быть сколько угодно сложными, поэтому при использовании отладчика для тестирования и комплексной отладки программ требуются средства анализа значений переменных отлаживаемой программы и изменения хода процесса тестирования или отладки по результатам анализа. Поэтому необходимо дополнить отладчик [1] алгоритмическим языком управления тестированием и отладкой (ЯУТО), который содержал бы в качестве доступных объектов переменные отлаживаемой программы. В настоящее время существуют различные системы, позволяющие автоматизировать процесс отладки [4, 5]. Однако с их помощью нельзя реализовать сложные алгоритмы тестирования. Кроме того, авторам не известны подобные системы, которые функционировали бы под управлением ОС ФОДОС, РАФОС, ОС ДВК.

ЯУТО разработан для использования в командных файлах символьного отладчика [1]. Он позволяет автоматическим (без вмешательства оператора) управлять исполнением командного файла, т. е. анализировать переменные отлаживаемой программы и результаты выполнения директив отладчика, осуществлять переход на любую метку текущего или другого командного файла, вызывать вложенные командные файлы и др. ЯУТО состоит из объединенного набора директив отладчика [1] и операторов языка.

Константа ЯУТО — целое 16-битовое восьмеричное число, которому может предшествовать знак. Отрицательные числа задаются в дополнительном коде. Левые значащие нули или лишние разряды (если они есть) игнорируются.

В операторах ЯУТО различают переменные отлаживаемой программы и переменные командного файла. Переменные отлаживаемой программы задаются в формате

⟨Указатель типа адресации⟩: ⟨Адрес переменной⟩.

Адрес переменной может задаваться восьмеричным числом, именем, определенным в исходном тексте отлаживаемой программы, регистром процессора (R0, ..., R7, RS), выражением, заключенным в скобки (см. ниже). Задание адреса переменной именем возможно только после выполнения директивы отладчика NL (см. [1]). Все переменные программы глобальные, т. е. доступны для всех подпрограмм и вложенных командных файлов.

Переменные командного файла задаются в формате ⟨Указатель типа адресации⟩ ⟨Имя переменной⟩.

Имя переменной командного файла — строка не более чем из трех символов. Все переменные командного файла, начинающиеся с символа $\$$, глобальные. Это означает, что они доступны всем подпрограммам и вложенным командным файлам. Остальные переменные доступны только тем подпрограммам и вложенным командным файлам, в которых они определяются. Допускается одновременное определение до 128 различных переменных командного файла.

Указатель типа адресации определяет способ доступа к переменной (табл. 1).

Выражение может быть константой, переменной либо комбинацией из этих компонент и одного или более знаков операций. Перечень всех операций, разрешенных в выражении, и их приоритет приведены в табл. 2.

Бинарные и упорные операции с одинаковым приори-

тетом выполняются слева направо и справа налево соответственно. Для изменения обычного порядка вычислений в выражении используются скобки. Выражение, заключенное в скобки, рассматривается как отдельная компонента, т. е. вначале вычисляется ее значение, которое затем используется в вычислении значения выражения.

Значение выражения или его части, заключенной в скобки, можно использовать как адрес переменной отлаживаемой программы. Для этого нужно поставить перед открывающейся скобкой знак «:» и необходимый указатель типа адресации. Например, в выражении : (1000+AREA) значение переменной с именем AREA увеличивается на 1000 и используется в качестве указателя адреса необходимого значения.

Метка ставится перед любой директивой или оператором ЯУТО для повторного обращения к фрагменту командного файла. Перед меткой ставится точка, а за ней — двоеточие.

Интерпретатор ЯУТО может реализовать два способа поиска метки: в файле и каталоге. В первом случае интерпретатор прежде всего проверяет таблицу меток прямого доступа. Если метка в таблице определена, то осуществляется непосредственный переход на нее без поиска в файле. Если метка в таблице не определена, интерпретатор начинает поиск метки от текущей позиции до конца командного файла. Если метка не найдена, то поиск ведется от начала файла до текущей позиции. В случае обнаружения метки ее имя и адрес заносятся в таблицу меток прямого доступа. В заполненной таблице, рассчитанной на 64 метки, вновь определенные метки заменяют ранее определенные, начиная с первой метки. Длина метки — не более шести символов.

Во втором случае командный файл должен быть предварительно обработан программой LIBS.SAV, которая заносит имена и адреса всех меток в каталог, размещаемый в начале файла. Собственно командный файл начинается с блока, следующего за каталогом. Имена меток могут быть занесены в каталог как в коде RADIX-50, так и в коде КОИ-7 (ключи программы LIBS.SAV/R и /A соответственно). Длина метки — не более шести символов для кода RADIX-50 и не более четырех символов для кода КОИ-7.

Как правило, ссылаются на метку в операторах ЯУТО так: ⟨Метка⟩; ⟨Имя командного файла⟩.

Если имя командного файла, в котором определена метка, не указано, то управление передается на метку в текущем командном файле. Указывать имя файла при ссылке на метку запрещено в операторах .DO и PRINT.

Операторы ЯУТО:

.GO ⟨Метка⟩ обеспечивает безусловный переход на указанную метку.

.IFA ⟨Выражение⟩ ⟨Метка 1⟩, ⟨Метка 2⟩, ⟨Метка 3⟩ передает управление на первую, вторую или третью из указанных меток, если значение выражения меньше нуля, равно нулю или больше нуля соответственно. Любые две метки или одна могут быть пропущены, однако запятые, разделяющие их, должны быть сохранены. После последней указанной метки запятые сохранять не обязательно. Пропуск метки означает, что управление будет передано на следующую за оператором .IFA строку.

.IFL ⟨Выражение 1⟩ ⟨Операция⟩ ⟨Выражение 2⟩ ⟨Метка 1⟩, ⟨Метка 2⟩ передает управление на первую метку, если результат логического сравнения двух выражений — истина. В противном случае управление передается на вторую метку. Любая из меток может быть пропущена. Если пропущена первая метка, то запятая должна быть сохранена. Пропуск метки означает, что управление в данном случае будет передано на следующую за оператором строку. Операции, допустимые в данном операторе, перечислены в табл. 3.

.CALL ⟨Метка⟩ ⟨PC1, ..., PCN⟩; ⟨P31, ..., P3N⟩ вызывает подпрограмму с указанной меткой. ⟨PC1, ..., PCN⟩ — список параметров, передаваемых подпрограмме по ссылке. В этом списке могут быть указаны только переменные

Таблица 1

Адресация	Указатель	Значение
Прямая	Пусто	Значение переменной с указанным адресом или именем
Непосредственная	#	Адрес переменной (запрещен для переменных командного файла и регистров)
Косвенная	@	Содержимое ячейки с адресом, равным значению переменной

Таблица 2

Тип операции	Знак операции и ее описание	Приоритет	
Арифметическая	Унарные +, —	1	
	.ABS.	абсолютная величина	1
	**	возведение в степень	2
	*, /	умножение, деление	3
	+, —	сложение, вычитание	4
Логическая	.NOT.	.NOT.A истинно тогда и только тогда, когда A ложно	1
	.AND.	A.AND.B истинно тогда и только тогда, когда A и B истинны	5
	.OR.	A.OR.B истинно тогда и только тогда, когда либо A, либо B, либо A и B истинны	6
	.EQV.	A.EQV.B истинно тогда и только тогда, когда A и B истинны или A и B ложны	7
	.XOR.	A.XOR.B истинно тогда и только тогда, когда A истинно, а B ложно или когда B истинно, а A ложно	7

Таблица 3

Обозначение	Операция
>	Больше
<	Меньше
=	Равно
><	Не равно
>=	Больше или равно
<=	Меньше или равно

командного файла, так как переменные программы всегда доступны любой подпрограмме. (PЗ1, ..., PЗN) — список параметров, передаваемых подпрограмме по значению. Этот список может включать в себя любые переменные и выражения. Значения передаваемых параметров присваиваются переменным командного файла с именами .NN (NN — порядковый номер фактического параметра в списке). Параметры, передаваемые по значению, доступны только вызванной подпрограмме и при выходе из нее теряются. Любой из списков параметров может быть опущен. Если отсутствует список параметров, передаваемых подпрограмме по ссылке, то «;», разделяющая списки, должна быть сохранена.

.RETURN предназначен для возврата из подпрограммы и завершения вложенного командного файла. Вместо этого оператора допустимо использование управляющего символа $\langle \text{CY/R} \rangle$.

.SET (Переменная) - (Выражение) присваивает переменной значение выражения. Если после знака « \Rightarrow » отсутствует выражение, определяющее значение указанной переменной, то выполнение командного файла приостанавливается и ожидается ввод значения переменной с терминала. После того как будет набрано значение и нажата клавиша (BK), выполнение командного файла будет продолжено.

.TYPE (Выражение) печатает на экране терминала значение выражения.

.DO (Метка) (Выражение) применяется для циклической обработки фрагмента командного файла. Метка, указанная в операторе, определяет область его действия. Все директивы и операторы ЯУТО, расположенные между оператором и этой меткой, включаются в его область действия. Значение выражения определяет число циклов. Операторы **.DO** и **.IF**, а также директива **NC** прекращают действие оператора **DO**, если передают управление за пределы его области действия.

Поскольку в состав ЯУТО входят как операторы, так и директивы отладчика [1], результаты исполнения директив, отображаемые на экране, должны быть доступны для анализа, проводимого с помощью операторов ЯУТО. Не приводя полного описания формируемой директивами информации, укажем для примера: номер точки останова после директивы **G**, адрес найденного кода после директивы **MD**, значение контрольной суммы после директивы **ML**. Информация о результате исполнения директив сохраняется в специальной области отладчика. Оператор **.FUNCTION** (Имя переменной командного файла) позволяет присвоить переменной командного файла адрес этой области и сделать ее, таким образом, доступной для анализа. Полезно, чтобы определенная здесь переменная была глобальной (начиналась с символа $\langle \circ \rangle$). Это обеспечит доступ к ней всех подпрограмм и вложенных командных файлов и снимет необходимость определения в них других переменных новым оператором **.FUNCTION**.

.ERROR/NOERROR разрешает (запрещает) приостановление выполнения командного файла при обнаружении синтаксической ошибки.

.QUIET/NOQUIET запрещает (разрешает) печать на экране терминала.

.PRINT (Метка) печатает на экране терминала строку командного файла с указанной меткой. Необходимо, чтобы

строка начиналась и заканчивалась одним и тем же ограничителем, которым может быть любой символ (например, «/», «>», «!» и т. п.).

.EXIT прекращает выполнение командного файла и передает управление отладчику. Вместо этого оператора допустимо использование управляющего символа $\langle \text{CY/E} \rangle$.

Директива **C** позволяет подключить к точке останова по счетчику команд определенную метку из текущего командного файла. Это означает, что после выхода отлаживаемой программы на точку останова управление автоматически передается на заданную метку командного файла. Необходимо, чтобы связываемые с точками останова метки имели числовые имена. Синтаксис:

$\langle \text{МЕТКА} \rangle$; $\langle \text{НТО} \rangle$ **C** — подключить метку к точке останова с номером $\langle \text{НТО} \rangle$;

$\langle \text{НТО} \rangle$ **C** — отключить метку от точки останова с номером $\langle \text{НТО} \rangle$;

C — отключить метки от всех точек останова.

Рассмотренный язык позволил сделать шаг вперед на пути автоматизации процесса тестирования и отладки ПО и показал, что предоставляемые им средства достаточны для решения данных задач. ЯУТО используется в качестве исполнительной подсистемы в системе автоматического построения тестов на программы [2] и для отслеживания сложных условий останова и сбора информации о работе программы. Разработан пакет стандартных программ на ЯУТО, обеспечивающих: останов по значению переменной, по выходу значения переменной за указанные границы и после указанного числа прохождений данной точки программы; вывод в файл последовательности значений переменной, значений указанных переменных в заданных точках программы и последовательности прохождения программы через заданные точки. ЯУТО применяется также для поиска контекста в числовом файле.

Телефон 532-82-62, Москва

ЛИТЕРАТУРА

1. См. статью на с. 58 настоящего выпуска.
2. См. статью на с. 63 настоящего выпуска.
3. Липаев В. В. Тестирование программ.— М.: Радио и связь, 1986.
4. Тюрин В. Ф. Операционная система ДИСПАК.— М.: Наука, 1985.
5. Гришман Р. Критерии языка отладки // Сб. статей: Средства отладки больших систем.— М.: Статистика, 1977.— С. 57—72.

Статья поступила 12.07.88

УДК 681.3

Н. А. Бучнева, Н. П. Васильев, М. И. Кушнир СРЕДСТВА ОТЛАДКИ ДЛЯ ЯЗЫКА ФОРТРАН-IV

Для отладки программ, написанных на языке высокого уровня, необходимо решить те же задачи, что и для отладки программ, написанных на ассемблере, а именно: обеспечить доступ к программно-доступным объектам, пооператорное и автоматическое исполнение программ и останов исполнения программ по заданным условиям. Доступ на уровне исходного языка и пооператорное исполнение программ существенно зависят от используемого языка, исполнение программ в автоматическом режиме и останов по условиям обеспечиваются с помощью процедур доступа и непосредственно от используемого языка не зависят.

Целесообразно выделить два уровня доступа для целей отладки. На нижнем уровне осуществляется доступ к содержимому ОЗУ по адресам и к регистрам; содержимое отображается и меняется в виде кодов в выбранной системе счисления. На верхнем уровне определяются адреса, соответствующие объекту исходного языка, занимающему

указанный адрес; содержимое памяти отображается и меняется в виде объектов исходного языка (текст оператора, значение переменной в соответствии с ее типом). Верхний уровень доступа обеспечивает привязку средств отладки к языку программирования.

Возможность разработки верхнего уровня доступа существенно зависит от компилятора используемого языка: при отсутствии информации о распределении памяти, выполненном компилятором, разработка верхнего уровня невозможна; при использовании динамического распределения памяти алгоритмы привязки исходного текста к адресам существуют, но могут оказаться слишком сложными для выполнения в реальном масштабе времени; статическое распределение памяти при наличии информации о распределении на выходе компилятора обеспечивает возможность реализации достаточно простых алгоритмов верхнего уровня доступа.

Рассмотрим отладчик, выполненный на базе символьного отладчика [2, 3], для языка Фортран-IV ОС РАФОС. Поставляемый в составе РАФОС компилятор языка Фортран-IV распределяет память статически и выдает в листинге полную информацию о распределении виртуальной памяти программы между подпрограммами и данными. Отображение виртуальной памяти на физическую, выполняемое компоновщиком ОС РАФОС, отражается в карте загрузки. Таким образом, в процессе подготовки программы формируется вся информация, необходимая для реализации в отладчике верхнего уровня доступа. В этой связи основной недостаток отладчика для Фортрана EDT [1], входящего в состав ОС РАФОС, — отсутствие автоматической привязки имен переменных к адресам и средств отображения текста программы на терминале. Поэтому при работе с FDT верхний уровень доступа выполняется с помощью листингов и карты загрузки.

Определение соответствия между адресами и объектами языка. Привязка операторов и переменных Фортрана к адресам могла бы выполняться по листингам и карте загрузки всякий раз, когда это необходимо для реализации отладочных функций, однако достаточно трудоемкая обработка текстовых файлов привела бы к большим задержкам реакции отладчика на директиву. Поэтому в процесс подготовки программы к исполнению включена вспомогательная программа, формирующая по листингам и карте загрузки набор табличных файлов, содержащих следующую информацию: адреса расположения операторов в ОЗУ и текстов операторов в листингах; таблицы имен переменных и меток программы, имен подпрограмм, оверлейных сегментов (для оверлейных программ).

Описание отладочных функций. Отладка фортрановских программ — факультативная возможность отладчика [2], определяемая при его генерации. Таким образом, все описанные в [2] возможности отладки на уровне языка ассемблера (отображение и изменение программы и данных в различных форматах; пошаговое и автоматическое исполнение программы; точки останова по счетчику команд, по времени, по изменению ячейки; поиски и форматирование памяти; отладка оверлейных программ, протоколирование и аппарат командных файлов [3]) реализуются в отладчике для Фортрана. Ограничим описанием дополнительных директив (возможностей):

F(NOP) трактуется как адрес начала фортрановского оператора с номером (NOP); директива **L(метка)** — как адрес объекта (переменной или оператора) с именем (метка). Поскольку Фортран допускает только числовые метки операторов, а, по мнению авторов, применение символьных меток удобнее, комментарии, начинающиеся с последовательности символов «/./», считаются метками, и их текст может использоваться для доступа к тексту программы.

L(имя) настраивает отладчик на работу с подпрограммой с именем (имя). Директивы **F(NOP)** и **L(метка)** относятся к тексту и переменным подпрограммы, на которую настроен отладчик.

FD отображает фрагмент листинга, содержащий указанный оператор.

FR выполняет один оператор Фортрана.

Директивам **FD** и **FR** может предшествовать адрес, отделенный от директивы символом «;». Адрес указывается директивами **F(NOP)**, **L(метка)** или числом. При указании адреса числом директива **FD** отображает оператор, занимающий данный адрес, а директива **FR** запускает фрагмент программы от указанного адреса до конца занимающего этот адрес оператора.

FR выполняет оператор в режиме пошаговой прокрутки (обращения к исполнителю в системе Фортрана реализуются в автоматическом режиме), отслеживая выходы счетчика команд за пределы исполняемого оператора.

После выполнения директивы **FR** и выхода на точку останова на экране терминала отображается текст оператора, перед исполнением которого произошел останов, и осуществляется настройка на подпрограмму, содержащую этот оператор. При отлаживании фортрановской программы как ассемблерной (такая возможность желательна для программ взаимодействия с аппаратурой) директива **XA** позволяет отображать в точке останова мнемонику команды, перед исполнением которой произошел останов, а директива **XF** — вернуться к отображению текста оператора Фортрана.

TF отображает содержимое стека вызовов текущей фортрановской подпрограммы. Такая возможность обеспечивает идентификацию причин заикливания, прерывания по зависанию, несуществующему коду команды и др.

В заключение особо следует подчеркнуть, что в отличие от отладчика FDT [1] обеспечивается отладка процедур, написанных на ассемблере, и фортрановских программ, скомпилированных в прямом коде и с ключом /NOLINENUMBERS.

Телефон 532-82-62, Москва

ЛИТЕРАТУРА

1. Валиков Л. М., Вигдорчик Г. В., Воробьев А. Ю., Луккин А. А. Операционная система СМ ЭВМ РАФОС.— М.: Финансы и статистика, 1984.
2. С. статью на с настоящего выпуска.
3. С. статью на настоящего выпуска.
4. Горовой В. Р., Васильев Н. П. Комплекс средств отладки микропроцессорных систем «Электроника НЦ 803» // Микропроцессорные средства и системы.— 1987.— № 1.— С. 44—46.

Статья поступила 12.07.88

УДК 681.3

А. А. Станоткин

АВТОМАТИЗАЦИЯ ПОСТРОЕНИЯ ТЕСТОВ ДЛЯ ПРОГРАММ ПО ВНЕШНЕЙ СПЕЦИФИКАЦИИ

Выбор стратегии для системы автоматического построения тестов

Известны два подхода к тестированию программ: с использованием внутренней структуры программы (стратегия «белого ящика») и внешней спецификации (стратегия «черного ящика»). В соответствии с этими подходами развивалась автоматизация тестирования программ, однако стратегии «белого ящика» отдавалось предпочтение. Проводился анализ различных систем автоматизации тестирования программ [1, 2], основанных на стратегии «белого ящика», хотя эта стратегия обладает следующими недостатками: полностью зависит от структуры программы (при изменении программы, как правило, внешняя спецификация остается неизменной, тогда как внутренняя структура меняется и как следствие этого — повторная генерация тестов); не позволяет обнаружить

несоответствие между внешней спецификацией на программу и самой программой.

Стратегии «белого ящика» и «черного ящика» имеют один общий недостаток: исчерпывающее тестирование невозможно [3]. Так почему же при выборе стратегии, лежащей в основе какой-либо системы автоматизации тестирования программ, предпочтение отдается «белому ящику»? Ответ на этот вопрос может быть следующим. Тестирование программ развивалось из тестирования аппаратных средств. Чтобы проверить работоспособность аппаратуры, необходимо убедиться в работоспособности ее отдельных элементов, следовательно, применить стратегию «белого ящика». Судить о правильности программы можно исходя из внешней спецификации на программу. Кроме того, при приемосдаточных испытаниях интересней и понятней проверять программу на тестах, построенных по методам «черного ящика». В настоящее время системы автоматизации тестирования на основе стратегии «черного ящика» автору неизвестны.

Майерс рекомендует [3] тестирование программ начинать с применения стратегии «черного ящика» и лишь потом достраивать тесты в зависимости от внутренней структуры программ.

Рассмотрим систему автоматического построения тестов, основанную на стратегии «черного ящика». Система содержит следующие части, поддерживающие определенные этапы тестирования: язык спецификаций (позволяет составить внешнюю спецификацию на программу); генератор тестов (по спецификации, применяя методы «черного ящика», строит тестовые наборы). Составной частью системы может быть подсистема, позволяющая исполнять тестируемую программу на заданных тестовых наборах и контролировать результаты работы программы. Если такая подсистема присутствует, то система автоматического построения тестов превращается в систему автоматического тестирования программ.

Язык спецификаций

Язык спецификаций (ЯС) — формальный язык, на который переводится спецификация, написанная на естественном языке. Основная задача, поставленная перед ЯС: описание входных и выходных данных объекта и функциональной связи между ними. Объект и его входные и выходные данные определяются в зависимости от требуемого уровня детализации спецификации. Например, в качестве объекта можно рассматривать один оператор языка программирования при составлении спецификации на компилятор, а при составлении спецификации на программный комплекс — весь программный комплекс или его отдельные подсистемы. Уровень детализации спецификации оказывает непосредственное влияние на объем спецификации и тестов, составленных по спецификации. (Следует помнить, что язык спецификаций отличается от традиционных языков программирования.)

Способ описания и основные конструкции языка. В соответствии с традиционными формами Бэкуса — Наура синтаксические конструкции обозначаются словами, заключенными в угловые скобки «*<*и*>*». Эти слова описывают также природу или смысл данных конструкций и используются в последующем описании семантики. Символ «*|*» — разделитель при перечислении возможных значений описываемой конструкции. Основная лексика состоит из символов, подразделяющихся на буквы, цифры, специальные символы, и служебных слов.

Для выделения начала комментария используется символ «*\$*»; конец комментария — символ «*;*». Между любыми конструкциями в языке спецификаций допускается сколько угодно пробелов и табуляций. Поэтому оформляется спецификация так, что ее структура становится наглядней. Конструкции языка отделяются друг от друга символами «*,*» или «*;*».

Спецификация на языке — это блок. Он содержит раздел описаний, в котором определяются все объекты, ис-

пользуемые в нем, и раздел операторов, задающий функциональные связи между описанными объектами:

```
(БЛОК)::=BEGIN(РАЗДЕЛ ОПИСАНИЙ)(РАЗДЕЛ ОПЕРАТОРОВ)
END
```

Раздел описаний — константы, типы и переменные. Описание констант позволяет поставить в соответствие идентификатору постоянное значение (тестовое, вещественное, целое, восьмеричное, символьное). Константу в спецификации можно употреблять и без предварительного описания. При этом ее нужно каждый раз указывать в стандартном виде. Использование идентификаторов для констант дает возможность сгруппировать в начале спецификации описания характерных величин.

Для чисел, являющихся константами целого или вещественного типа, используется обычная десятичная система записи. ЯС допускает константы, записанные в восьмеричном виде. Если за целой константой следует буква «*B*», то число будет интерпретироваться как записанное по основанию 8. Текст (текстовая константа) — это последовательность символов, заключенная в апострофы.

В начале определения констант стоит слово CONST и этот раздел имеет следующий вид:

```
(ОПИСАНИЕ КОНСТАНТЫ)::=CONST(ОПРЕДЕЛЕНИЕ КОНСТАНТЫ);
(ОПРЕДЕЛЕНИЕ КОНСТАНТЫ)::=(ИДЕНТИФИКАТОР)=(КОНСТАНТА)!
(ОПРЕДЕЛЕНИЕ КОНСТАНТЫ),(ИДЕНТИФИКАТОР)=
=(КОНСТАНТА)
```

Тип данных определяет множество значений, принимаемых переменной. Каждая встречающаяся в спецификации переменная должна быть сопоставлена с одним и только одним типом.

ЯС имеет пять предопределенных типов: десятичный целый (INTEGER), вещественный (REAL), восьмеричный (OCTAL), табличный (TABL) и символьный (CHAR). Кроме того, можно определять типы данных, соответствующие структуре данных тестируемой программы: типы, задаваемые перечислением, отрезки типов и образцы.

Описания типов начинаются словом TYPE:

```
(ОПИСАНИЕ ТИПА)::=TYPE(ОПРЕДЕЛЕНИЕ ТИПА);
(ОПРЕДЕЛЕНИЕ ТИПА)::=(ИДЕНТИФИКАТОР)=(ТИП)!
(ОПРЕДЕЛЕНИЕ ТИПА),(ИДЕНТИФИКАТОР)=(ТИП)
```

Описание *переменной* связывает идентификатор и некоторый тип данных. Оно начинается с зарезервированного слова VAR:

```
(ОПИСАНИЕ ПЕРЕМЕННОЙ)::=VAR(ОПРЕДЕЛЕНИЕ ПЕРЕМЕННОЙ);
(ОПРЕДЕЛЕНИЕ ПЕРЕМЕННОЙ)::=(ПОСЛЕДОВАТЕЛЬНОСТЬ ИДЕНТИФИКАТОРОВ):(ТИП)!
(ОПРЕДЕЛЕНИЕ ПЕРЕМЕННОЙ),(ПОСЛЕДОВАТЕЛЬНОСТЬ ИДЕНТИФИКАТОРОВ):(ТИП)
(ТИП)::=OCTAL|INTEGER|CHAR|REAL|TABL|(ТИП)
(ТИП)::=(ТИП ПЕРЕЧИСЛЕНИЯ)! (ОТРЕЗОК ТИПА)! (ТИП ОБРАЗЦА)
```

Если это удобно, вместо идентификатора переменной в спецификации можно использовать имя типа данных.

В определении типа, задаваемого перечислением, должны быть те значения, которые может принимать переменная данного типа. Значения записываются в виде текстовых констант:

```
(ТИП, ЗАДАВАЕМЫЙ ПЕРЕЧИСЛЕНИЕМ)::=(ПОСЛЕДОВАТЕЛЬНОСТЬ ТЕКСТОВЫХ КОНСТАНТ)
((ПОСЛЕДОВАТЕЛЬНОСТЬ ТЕКСТОВЫХ КОНСТАНТ))::=(ТЕКСТОВАЯ КОНСТАНТА)!
(ПОСЛЕДОВАТЕЛЬНОСТЬ ТЕКСТОВЫХ КОНСТАНТ),(ТЕКСТОВАЯ КОНСТАНТА)
```

Отрезок типа получается введением ограничения на предопределенные типы (за исключением типа «*TABL*»). Ограничение определяется заданием диапазона минимального и максимального значений констант.

```
(ОТРЕЗОК ТИПА)::=(КОНСТАНТА)<(КОНСТАНТА)
```

Кроме простых типов в ЯС существует структурный тип «образец», который строится из простых. Обра-

зец предназначается для описания форматов данных в более наглядной и лаконичной форме. Список полей можно рассматривать как шаблон. Он может накладывать ограничения на некоторые части строки, другие же фрагменты могут быть переменными разных типов, в том числе и типа «образец». Возможно рекурсивное описание. Синтаксис определения типа «образец» следующий:

```

<ТИП ОБРАЗЦА> ::= РАТ <СПИСОК ПОЛЕЙ>
<СПИСОК ПОЛЕЙ> ::= <ПОЛЕ> | <СПИСОК ПОЛЕЙ> <РАЗДЕЛИТЕЛЬ> <ПОЛЕ>
<ПОЛЕ> ::= <КОНСТАНТА> | <ПЕРЕМЕННАЯ>
<разделитель> ::= ' ' | ','

```

где ' ' — выбор; ',' — конкатенация.

Выражения — это конструкции, состоящие из операндов и знаков логических операций и описывающие функциональную связь операндов. Существуют три логические операции AND — конъюнкция (и); OR — дизъюнкция (или) и NOT — отрицание (не).

Результат вычисления выражения — таблица истинности, построенная по определенным правилам [3]. Таблица используется генератором тестов для формирования тестовых наборов.

Раздел операторов. Операторы языка спецификаций позволяют: указать функциональную связь между описанными переменными спецификации; поставить в соответствие переменную спецификации переменной тестируемой программы (это свойство используется при работе с исполнительной подсистемой); поставить в соответствие функциональную связь между описанными переменными спецификации какой-либо переменной спецификации.

```

<ОПЕРАТОР> ::= <ОПЕРАТОР СЛЕДОВАНИЯ> | <ОПЕРАТОР ПРИСВАИВАНИЯ> | <ОПЕРАТОР ПРОЦЕДУРЫ> | <БЛОК>

```

Оператор *следования* позволяет поставить в соответствие функциональные связи между переменными спецификации (записанной в виде выражения) и переменной другой спецификации (следствие из функциональной связи). Синтаксис оператора следования:

```

<ОПЕРАТОР СЛЕДОВАНИЯ> ::= FROM (ВЫРАЖЕНИЕ) FOLLOW (ИДЕНТИФИКАТОР)

```

Оператор *присваивания* предназначен для замены функциональной связи между переменными спецификации, задаваемой некоторым выражением, одной переменной. В различных операторах следования могут присутствовать одни и те же комбинации причин. Для сокращения времени обработки спецификации генератором тестов повторяющиеся комбинации причин лучше объединять в отдельные выражения, результат обработки которых будет присвоен переменной. В дальнейшем можно использовать эту переменную в других выражениях. Синтаксис оператора присваивания:

```

<ОПЕРАТОР ПРИСВАИВАНИЯ> ::= <ПЕРЕМЕННАЯ> ::= <ВЫРАЖЕНИЕ>

```

После того, как генератор тестов закончил свою работу и создал файл тестовых наборов, возникает вопрос: Как использовать полученные тесты? Есть два пути: подавать тестовые значения на вход тестируемой программы и проверять результаты вручную; поручить выполнение и проверку исполнительной подсистеме. Понятно, что второй путь предпочтительней, но он требует от пользователя дополнительных затрат труда при написании спецификации. Чтобы обеспечить возможность подготовки тестов к выполнению их исполнительной подсистемой, в ЯС введен оператор процедуры. Основная функция, возложенная на этот оператор, — указание соответствия между переменной тестируемой программы и переменной спецификации. Процедуры могут быть с параметрами (процедуры доступа) и без параметров.

Описание процедуры состоит из заголовка, в котором указываются идентификатор процедуры и формальные параметры, и тела. В качестве формальных параметров процедуры можно использовать любые идентификаторы спецификации. Тело процедуры — набор директив и операторов

исполнительной подсистемы с вставленными в него параметрами. В ЯС имеются четыре предопределенные процедуры доступа: к десятичным целым, восьмеричным, вещественным и символьным переменным тестируемой программы.

Обращение к процедуре — идентификатор описанной ранее процедуры с указанными фактическими параметрами. В качестве переменной тестируемой программы можно указать любую текстовую константу, а в качестве переменной спецификации — соответствующий идентификатор, используемый в спецификации.

Процедуры без параметров могут использоваться для выполнения подсистемой определенных действий, которые не зависят от спецификации программы. Например, запуск тестируемой программы с определенным адресом и останов ее в указанном месте, начальная инициация программы и т. п. Синтаксис оператора процедуры:

```

<ОПЕРАТОР ПРОЦЕДУРЫ> ::= <ЗАГОЛОВОК ПРОЦЕДУРЫ> <ТЕЛО ПРОЦЕДУРЫ> ENDPROC
<ЗАГОЛОВОК ПРОЦЕДУРЫ> ::= PROC <ИДЕНТИФИКАТОР ПРОЦЕДУРЫ> ((СПИСОК ПАРАМЕТРОВ)) | PROC <ИДЕНТИФИКАТОР ПРОЦЕДУРЫ>
<СПИСОК ПАРАМЕТРОВ> ::= <ИДЕНТИФИКАТОР> | <ИДЕНТИФИКАТОР>
<ТЕЛО ПРОЦЕДУРЫ> ::= <ДИРЕКТИВЫ ИСПОЛНИТЕЛЬНОЙ ПОДСИСТЕМЫ>
<ОБРАЩЕНИЕ К ПРОЦЕДУРЕ> ::= <ИДЕНТИФИКАТОР ПРОЦЕДУРЫ> (<ИДЕНТИФИКАТОР, ИСПОЛЪЗУЕМЫЙ В СПЕЦИФИКАЦИИ>, <ТЕКСТОВАЯ КОНСТАНТА>)

```

Генератор тестов

Генератор тестов осуществляет синтаксический контроль описания программы на ЯС и по ней строит тестовые наборы. Результатом работы генератора тестов являются два файла: листинга, содержащий описание программы на ЯС с помеченными в ней синтаксическими ошибками, и тестовых наборов. Исполнительная подсистема использует для своей работы файл тестовых наборов.

В процессе синтаксического анализа генератор создает таблицу встреченных в описании идентификаторов и таблицу операторов следования. Если описание правильное, т. е. отсутствуют синтаксические ошибки, генератор приступает к обработке таблиц идентификаторов и операторов следования. Генератор просматривает таблицу идентификаторов и для каждого элемента в зависимости от его типа и в соответствии с принципами построения тестов по методам граничных значений и эквивалентных разбиений [3] строит множество значений.

Правила, используемые генератором при обработке таблиц идентификаторов. Если идентификатору соответствует константа, то берется ее значение. Правила генерации входных наборов по описаниям типов: если тип стандартный, то в качестве значений берутся два случайных типа из множества значений и два условия перехода через нуль; отрезок — границы отрезка, граничные условия и условия перехода через нуль; перечисление — все значения типа; образец — все значения этого образца, т. е. в образец подставляются конкретные значения полей (они формируются в соответствии с принципами, описанными выше).

Правила, используемые генератором при обработке таблиц операторов следования. Элемент таблицы операторов следования содержит ссылки на выражение и следствие (элемент в таблице идентификаторов). Выражение преобразуется в таблицу истинности. Каждая строка таблицы соответствует тестовому набору. Если элемент строки равен единице, то значения соответствующей переменной должны присутствовать в тестовом наборе. Следует помнить, что значения переменных спецификации записываются в выходной файл в порядке их появления в выражении.

Так как переменная может принимать несколько значений, то одна строка таблицы истинности превращается в N тестовых наборов (N — максимальное число значений переменных, входящих в данное выражение). Каждому тестовому набору соответствует определенное значение следствия (выходного набора).

Правила генерации выходных наборов по описаниям типов: если тип стандартный, то в качестве выходного значения выдается имя типа и пользователю предоставляется возможность проверять результаты самому; отрезок — границы отрезка и тип границ; перечисления — все значения этого типа; образец — параметризованное значение образца с указанием типов переменных полей.

В процессе генерации тестовых наборов, перед тем как записывать значение переменной спецификации в выходной файл, генератор анализирует, было ли обращение к процедуре для данной переменной, подставляет в параметризованное тело конкретные значения и тело записывает в выходной файл. Правила генерации входных и выходных наборов для переменных и соответствующих типов совпадают.

Методика использования ЯС

При обработке описания программы генератор тестов использует метод функциональных диаграмм [3], поэтому методики описания программ на ЯС и построения тестов методом функциональных диаграмм частично совпадают. Описание программы на ЯС и ее тестирование осуществляются в несколько этапов:

1) описание программы на естественном языке разбирается на рабочие участки — семантически законченные части описания программы. Это может быть описание команды, директивы, модуля и т. п.;

2) входная и выходная области программы разбираются на конечное число классов эквивалентности так, чтобы можно было предположить, что все значения, являющиеся представителями некоторого класса, программа обрабатывает одинаково. При разбиении на классы эквивалентности следует помнить, что различные два типа классов: правильные (правильные входные данные программы) и неправильные (ошибочные входные значения). Описания констант, типов, переменных ЯС позволяют описать выделенные классы эквивалентности;

3) классам эквивалентности при написании спецификации ставятся в соответствие идентификаторы;

4) среди классов эквивалентности определяются причины и следствия. Причина — отдельное входное условие или класс эквивалентности входных условий. Следствие есть выходное условие или преобразование системы (остаточное действие, которое входное условие оказывает на состояние программы);

5) идентификаторы причин объединяются логическими операциями OR, AND, NOT в соответствии с семантическим содержанием спецификации в выражение;

6) с помощью оператора следования ЯС выражению (комбинации причин) ставится в соответствие следствие;

7) полученное описание программ на ЯС вводится с помощью любого текстового редактора в ЭВМ;

8) введенный файл обрабатывается генератором тестов;

9) программа тестируется на полученных тестовых наборах.

Предположим, Вам необходимо проверить правильность процедуры установки даты. Процедура в качестве входных данных получает символьную строку с датой следующего формата:

<день>—<месяц>—<год>,

где день может принимать значения от 1 до 31; месяц — 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'; год — от 1989 до 2000. Результат работы процедуры — установление системной даты и выдача сообщения «дата установлена».

Рассмотрим некоторые из указанных этапов подробнее. 1-й этап. В качестве рабочего участка выбираем описание процедуры установления даты.

2-й и 3-й этапы. Для упрощения спецификации будем рассматривать только правильные классы эквивалентности,

в качестве которых при анализе спецификации можно выделить следующие входные и выходные условия:

день изменяется от 1 до 31;
 месяц может принимать значения 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec';
 год изменяется от 1989 до 2000;
 строка с датой должна иметь формат: <день>—<месяц>—<год>;
 сообщение — «дата установлена».

4-й этап. Анализируем спецификацию процедуры установки даты и определяем причины и следствия. Причина для данной процедуры — строка с датой, а следствие — сообщение, выдаваемое процедурой.

5-й и 6-й этапы. С помощью оператора следования причине (строке с датой) ставим в соответствие следствие (сообщение, выдаваемое процедурой).

7-й этап. Получаем полное описание процедуры установки даты на ЯС:

```
BEGIN
$ установка даты;
CONST
TRUE — 'дата установлена',
TYPE
MONTH=('Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov,
Dec),
DAY=1:31,
YEAR=1989:2000,
DATE=PAT DAY, '-', MONTH, '-', YEAR :
FROM DATE FOLLOW TRUE:
END.
```

8-й этап. После обработки спецификации процедуры установки даты генератором тестов будут сформированы следующие тестовые значения даты:

1—Dec—1989	16—Aug—1998	30—Apr—1999
31—Nov—2000	1—Jul—1989	16—Mar—1998
2—Oct—1990	31—Jun—2000	1—Feb—1989
30—Sep—1999	2—May—1990	31—Jan—2000

9-й этап. Для тестирования процедуры установки даты необходимо подавать входные и анализировать выходные условия. Способ подачи и анализа условий зависит от конкретной реализации процедуры.

Реализация. Экспериментальная система автоматического построения тестов была разработана в 1986—1987 гг. и использовалась для тестирования генератора тестов, а также программ, написанных на языке MACRO-11. В состав системы входят описанные выше язык спецификаций и генератор тестов. В качестве исполнительной подсистемы применяется символьный отладчик для программ, имеющих пакетный режим выполнения директив [4, 5]. Генератор и отладчик функционируют под управлением ОС РАФОС или других аналогичных ОС. В исполнительной подсистеме используется файл тестовых наборов.

Телефон 532-82-62, Москва

ЛИТЕРАТУРА

1. Бичевский Я. Я., Борзов Ю. В. Тестирование программ ЭВМ.— Рига: ЛГУ им. П. Стучки, 1985.— С. 41—56.
2. Борзов Ю. В. Тестирование программ с использованием символического выполнения // Программирование.— 1980.— № 1.— С. 51—58.
3. Майерс Г. Искусство тестирования программы: Пер. с англ.— М.: Финансы и статистика, 1982.— 176 с.
4. См. статью на с. 58 настоящего выпуска.
5. См. статью на с. 62 настоящего выпуска.

Статья поступила 8.08.88

А. Н. Бучнев, Н. П. Васильев, В. Р. Горовой, Ю. П. Крылатых, М. И. Кушнир

КОМПЛЕКС ОТЛАДКИ ПРОГРАММ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ

Рассмотрим комплекс «Электроника МС0404», предназначенный для отладки программ микропроцессорных систем на базе микросхем К1801ВМ2, К1806ВМ2.

Аппаратурные средства комплекса

Структурная схема комплекса приведена на рисунке. Функции ЦУС выполняет ДВК «Электроника НЦ-80-20» [1], совместимый по архитектуре и системе команд с СМ ЭВМ, имеющей в своем составе память объемом 56 Кбайт, накопители на ГМД и печатающее устройство. Функции ЛСО выполняет блок логических устройств (БЛУ) «Электроника НЦ-803» [2]. В состав БЛУ входят блоки сопряжения (БС) и обмена (БО), логический анализатор (ЛА) и компаратор (ЛК), эмулятор памяти (ЭП), генератор слов (ГС).

Устройство БС, подключенное к магистрали процессор — память ВСЭ, обеспечивает доступ к адресуемому пространству системы; БО — связь БЛУ с ДВК (все устройства БЛУ доступны по адресам на системной магистрали ДВК, реализация интерфейса обмена по магистрали ДВК возложена на БО), замеры времени, подсчет и обработку радиальных запросов от устройств. Радиальные запросы выдают ЛА, ЛК, ГС, ЭП, а также СОВ и ВСЭ.

БЛУ может находиться в двух режимах: «опрос» и «контроль». В режиме «контроль» БЛУ выполняет функции по слежению за работой отлаживаемой программы. В режиме «опрос» все ресурсы БЛУ доступны ЦУС, в этом режиме пользователь описывает те события в системе, при которых он хотел бы прекратить слежение и перейти к анализу результатов ее работы, т. е. перевести БЛУ из режима «контроль» в режим «опрос». В качестве событий могут указываться: интервал времени, обращение к эмулируемому внешнему устройству, переход процессора в пультовый режим, последовательность адресов и данных на магистрали МП-системы.

Отсчет интервалов времени обеспечивается СОВ; эмуляция внешних устройств — ЭП и ГС (с помощью ЭП

можно читать и записывать по адресам эмулируемых устройств, а с помощью ГС — выдавать векторы прерывания). Абоненты моделируются программно. Переход процессора в пультовый режим распознается ВСЭ, а последовательность адресов и данных на магистрали МП-системы — ЛК.

Устройством ЛА запоминается информация о трассе программы. Объем памяти ЛА 1024 32-разрядных слова. Информация в память устройства ЛА записывается в момент появления сигнала «достоверность адреса (данных)», вырабатываемого ВСЭ, или сигнала от ЛК. В состав БЛУ входят два устройства ЛА. Одно из них подключено к магистрали системы и используется для запоминания адресов и данных. 16 разрядов каждого слова памяти этого устройства отведено под значения управляющих линий магистрали (эти значения позволяют распознать вид информации и операцию на магистрали: адрес, данные, чтение, запись, прерывание, прямой доступ в память и пр.), другие 16 разрядов хранят значения информационных линий. Второе устройство ЛА частью своих входов подключено к таймеру СОВ, что позволяет зафиксировать с точностью до 1 мкс время появления события на магистрали системы; часть входов не задействована в комплексе и может использоваться для сбора информации о поведении аппаратуры, подключенной к комплексу (при отладке системы на аппаратном уровне ЛА может использоваться в асинхронном режиме для запоминания с частотой до 20 МГц состояний подключенных линий).

Устройство ВСЭ обеспечивает: эмуляцию памяти системы (56 Кбайт); хранение программ пультового режима процессора; электрофизическое сопряжение с системой; эмуляцию МП К1801ВМ2 и К1806ВМ2; формирование сигнала «достоверность адреса (данных)»; управление эмуляцией памяти с адреса (данных); управление эмуляцией памяти с адресами 0...157777 и МП.

Поскольку объем ЭП БЛУ составляет 32 Кбайт, что недостаточно для эмуляции всей памяти МП К1801ВМ2 и К1806ВМ2, в состав ВСЭ включена дополнительная память объемом 64 Кбайт, разбитая на восемь блоков по 8 Кбайт. Блоки с номерами 1...7 предназначены для эмуляции памяти системы, восьмой блок — для хранения пультовой программы (пультовый режим МП К1801ВМ2, К1806ВМ2 задействован в комплексе для реализации доступа к регистрам МП и пуска).

Комплекс может работать совместно с реальной системой и автономно. При автономной работе комплекса внешние устройства эмулируются с помощью ЭП и ГС, абоненты системы модели-

руются программно и комплекс целесообразно использовать в режиме управления, в котором перевод БЛУ в режим «опрос» всегда вызывает приостанов выполнения программ системы и прерывания счета времени. При работе с реальной системой и особенно при подключении реальных абонентов может сказаться необходимым режим подслушивания, в котором комплекс лишь следит за работой системы, не оказывая на нее никаких воздействий. Основное средство отладки в режиме подслушивания — сбор и последующий анализ трассы программы.

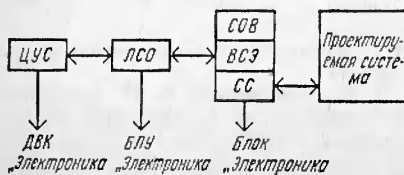
Программное обеспечение комплекса

Программное обеспечение (ПО) комплекса построено на базе операционной системы ОС ДВК и обеспечивает управление работой комплекса в процессе тестирования и отладки программ системы и программ моделей. Создание программных моделей абонентов возлагается на пользователей комплекса (разработка программ и аппаратуры проводится, как правило, одновременно). В состав ПО комплекса входят модели терминала, дисков и некоторых специализированных внешних устройств, а также интерфейс взаимодействия моделей внешних устройств с моделями абонентов.

При ориентации на программное моделирование внешней среды возникают дополнительные требования к функционированию комплекса: он должен обеспечивать совместную отладку программ системы и моделей.

В организации ПО комплекса фундаментальным является понятие адресного пространства (АП). АП — совокупность процедур: доступа к памяти и регистрам, пуска в пошаговом и автоматическом режимах, обработки основа программы. Число адресных пространств (1...256) задается при генерации ПО. Основная часть ПО не зависит от вида и числа АП: специальная директива обеспечивает переключение на нужное пользователю АП. В комплексе «Электроника МС0404» ПО работает с двумя АП — пространством программ моделей и системы. Директивы языка отладки одинаковы в обоих АП, за исключением дополнительных директив, которые поддерживаются аппаратурными средствами комплекса.

Программирование устройства ЛК обеспечивает селекцию обменов по магистрали системы, описанных адресом (набором или диапазоном адресов), видом обмена (чтение, запись, чтение или запись) и, если необходимо, значением (набором или диапазоном значений) данных. Распознаваемое ЛК событие может использоваться в качестве условия останова программ системы для выборочного сбора трассы программы, подсчитываться устройством БО для обеспечения останова после указанного числа событий или отображения информации о работе программы.



Трасса программы собирается устройством ЛА. В качестве трассы может быть последовательность команд и их операндов или последовательность работы с данными. В первом случае трасса собирается по сигналам «достоверность адреса (данных)» и с помощью ЛК описывается область адресов программы, для которой собирается трасса (это позволяет избавиться, например, от информации о работе ОС). Во втором случае трасса собирается по сигналам от устройства ЛК и ПО комплекса позволяет зафиксировать последовательность значений некоторой ячейки памяти или последовательность обращений (адреса и данных) к набору или диапазону адресов. Трасса программы, собранная по сигналам «достоверность адреса (данных)», отображается в виде дизассемблированных строк

с именами и значениями операндов и языка Фортран, если указана отладка фортрановских программ. Трасса, собранная по сигналам от устройства ЛК, отображается в виде имен и значений данных. Для любой записи в трассе можно запросить время возникновения на магистрали записанного события.

Для отладки системы на аппаратурном уровне обеспечивается приостановка ее или (в режиме подслушивания) перевод БЛУ в режим «опрос» по последовательности сигналов на линиях, подключенных к аппаратуре пользователя, и отображение собранной логической информацией в виде временных диаграмм.

Телефон 532-82-62, Москва

ЛИТЕРАТУРА

1. Заморин А. П., Мячев А. А., Селиванов Ю. П. Вычислительные машины, системы, комплексы: Справочник.— М.: Энергоавтоматиздат, 1985.
2. Горовой В. Р., Васильев Н. П. Комплекс средств отладки микропроцессорных систем «Электроника НИ-803» // Микропроцессорные средства и системы.— 1987.— № — С: 44—46.
3. См. статью на с. 58 настоящего выпуска.
4. См. статью на с. 62 настоящего выпуска.
5. См. статью на с. 63 настоящего выпуска.

Статья поступила 12.07.88

КРАТКОЕ СООБЩЕНИЕ

УДК 681.326+681.3.06

В. В. Андреев, С. И. Щербаков

КОМПЛЕКС ДЛЯ ПРОГРАММИРОВАНИЯ МИКРОСХЕМ

Разработан программно-аппаратный комплекс (ПАК) для записи информации в БИС УФРПЗУ К573РФ1... К573РФ7, однократно программируемые ПЗУ КР556РТ4... КР556РТ18, К155РЕ3, К541РТ1, РТ2 БИС ПЛМ КР556РТ1, РТ2, ОЭВМ КМ1816ВЕ48, КМ1816ВЕ51, КМ1813ВЕ. Он состоит из управляющей микроЭВМ типа ДВК, программатора «Электроника МС9401» и пакета управляющих программ. При работе с данным комплексом пользователь получает в свое распоряжение систему программирования для подготовки, редактирования, хранения и документирования информации.

В процессе эксплуатации потребители могут встречать микросхему одного и того же типа с разными значениями напряжения программирования, поэтому максимальная амплитуда импульса программирования устанавливается внешним источником, от которого запитываются формирователи высоковольтных сигналов.

Внешний источник питания, включаемый независимо от микроЭВМ ДВК, создает опасность попадания напряжения высокого уровня на установленную в переходное устройство микросхему при отключенном питании ДВК. Для исключения таких ситуаций в схему введена блокировка.

Управляющая программа хранится на ГМД и поддерживается ОС типа RT-11, ОСДВК; она обменивается информацией через специальные регистры программатора: 177150 — данных, 177152 — адреса, 177154 — команд, 177156 — состояния. Временные диаграммы режимов чтения и записи информации формируются установкой и сбором соответствующих битов в регистре команд.

Регистр состояния используется для создания импульсов длительностью 50 мкс, 1 мс, 50 мс в режиме программирования и для автоматического выбора управляющей подпрограммы.

При подготовке данных для программирования используется буфер, расположенный внутри управляющей программы. Он заполняется тремя способами: чтением файла кодов или информации из микросхемы либо вводом с клавиатуры ДВК. Режим работы ПАК выбирается оператором из меню, выводимого на экран монитора ДВК, и состоит из следующих основных операций:

чтения файла с ГМД в буфер, записи буфера на файл, входного контроля микросхем, чтения информации из микросхемы в буфер, записи информации из буфера в микросхему, сравнения информации буфера и микросхемы, распечатки буфера на экране и его редактирования, получения листинга буфера.

В режиме меню задается адресация буфера и смещение ПЗУ, поэтому чтение-запись информации возможно с произвольного адреса ПЗУ (решается проблема работы с массивами данных, значительно превышающими емкость ПЗУ). Для более точного отображения содержимого ПЗУ в буфере при его смещении, отличным от нуля, формируется поле исходного состояния ПЗУ на размер смещения.

В режиме записи информации после программирования текущего байта контролируется правильность записи. При этом, время работы с неисправными образцами экономится, так как после нахождения первого же дефектного байта управляющая программа выдает сообщение об ошибке и выходит из данного режима.

Принцип модульности позволяет оперативно наращивать программные средства для расширения функциональных возможностей ПАК при соответствующих технических условиях на выбранный тип программируемой микросхемы.

Телефон 460-38-61, Москва

ЛИТЕРАТУРА

1. Домнин С. Б., Иванов Е. А., Кушир В. Д., Муренко Л. Л. Вычислительный отладочный комплекс «Электроника МС 0701» // Электронная промышленность.— 1986.— № 2.— С. 51—53
2. Кобылинский А. В., Липовецкий Г. П. Однокристальные микроЭВМ серии К1816 // Микропроцессорные средства и системы.— 1986.— № 1.— С. 10—19.

Статья поступила 19.2.88

УДК 681.327.0

С. Л. Разенков, Е. Ю. Тимофеев

МЕТОД УВЕЛИЧЕНИЯ ПРОТЯЖЕННОСТИ МАГИСТРАЛИ ИНТЕРФЕЙСА КОП

Интерфейс КОП (канал общего пользования по ГОСТ 26.003-80) предназначен для соединения функционально и конструктивно законченной аппаратуры и широко используется в автоматизированных измерительных системах (АИС) различного назначения. Заданные показатели скорости и достоверности обмена информацией в электрических схемах возбуждителей и приемников достигаются при длине магистрали не более 20 м.

Для увеличения длины интерфейсных шин КОП предлагается использовать волоконно-оптическую линию связи (ВОЛС), способную передавать цифровые сигналы без ретрансляции со скоростью более 10 Мбит/с на расстояние в несколько километров*.

Линии интерфейса КОП двунаправленные, поэтому цифровые сигналы передаются по одной ВОЛС, а по другой принимаются от удаленной аппаратуры (УА).

ВОЛС приема и передачи мультиплексированы на одну магистраль КОП устройством управления направлением передачи (УУНП), а подключаются к КОП через два устройства сопряжения (УС), одно из которых входит в состав УУНП (рис. 1).

В результате анализа команд контроллера КОП УУНП подключает или блокирует соответствующие линии передатчика и приемника ВОЛС в зависи-

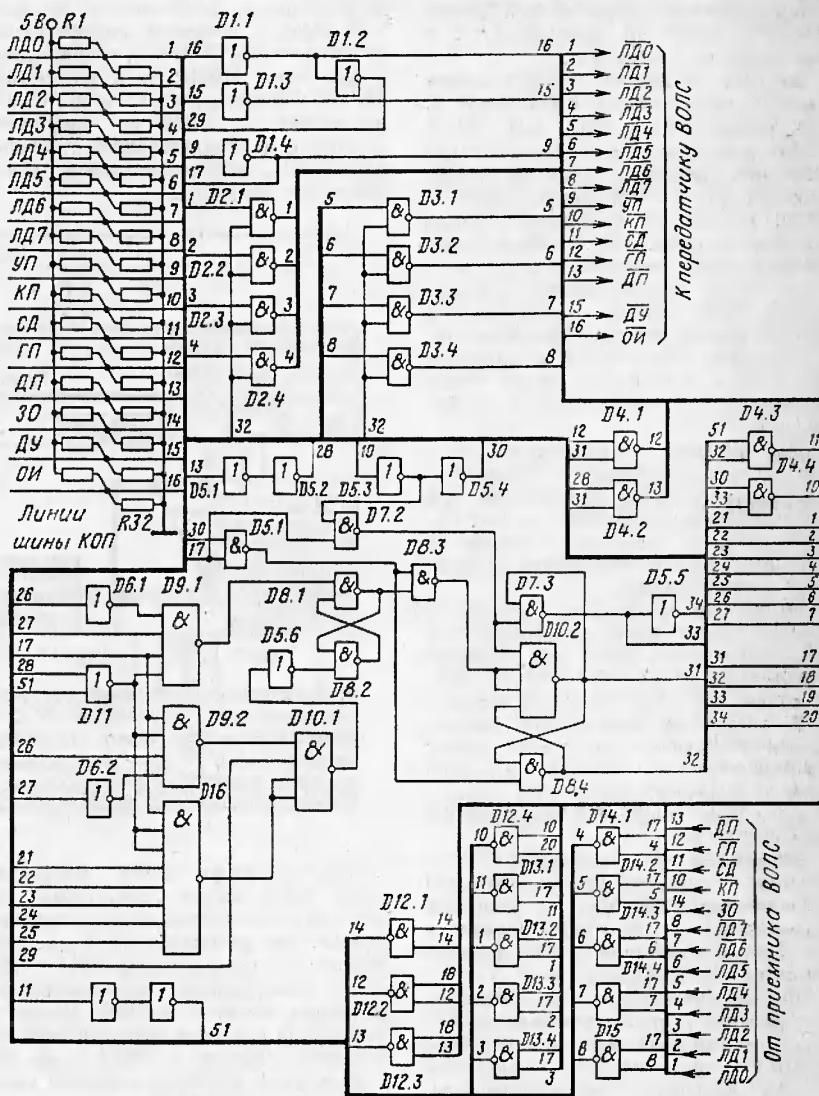


Рис. 2. Принципиальная схема УУНП

мости от одного из трех возможных состояний интерфейса (рис. 2):

передача команд и сообщений устройств от ЭВМ к УА (ЭВМ управляет УП, ДУ, ОИ, КП, СД, ЛД0...ЛД7; УА—ГП, ДП и ЗО);

передача сообщений устройств от УА к ЭВМ (ЭВМ управляет УП, ДУ, ОИ, ГП и ДП; УА—ЛД0...ЛД7, ЗО, КП, СД);

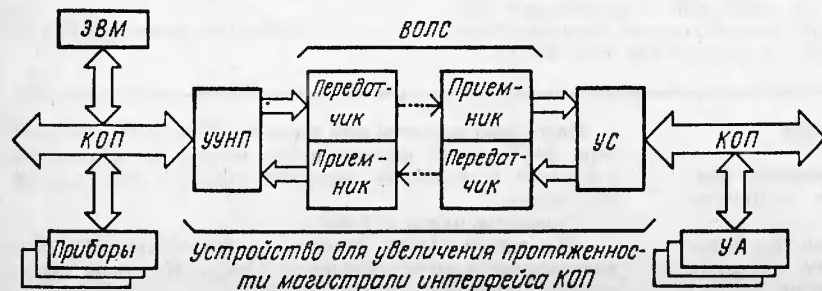


Рис. 1. Структура распределенной АИС на основе ВОЛС

* Раденко М. Е., Сеньков В. К. Реализация волоконно-оптического интерфейса для микропроцессорных систем // Микропроцессорные средства и системы.— 1986.— № 2.

состояние параллельного опроса (ЭВМ управляет УП, ДУ, ОИ, КП; УА — ЛД0...ЛД7, ЗО; СД, ГП и ДП не используются).

Для идентификации этих состояний необходимо и достаточно построить работу УУНП по следующим правилам:

в состоянии параллельного опроса линии подключать по команде ИДТ — УПАКП;

при передаче сообщений от УА переключать линии по команде ГАИ в пассивном состоянии УП;

во всех остальных случаях линии должны удерживаться в состоянии 1.

Команды ГАИ, ГАП, ОИ, НПД, НПМ, необходимые для идентификации состояний, дешифрируются на элементах D9, D10.1, D16, команда ИДТ — D7.2. Триггер D8 фиксирует сигналы от дешифратора. D10.2 (D8.4) переключает линии интерфейса в состояние 1 на время выдачи команд контроллером КОП (после снятия режима выдачи команд повторяет функции D8).

Триггеры D10.2, D8.4 и элементы D7.3, D5.5 подключают линии интерфейса КОП в соответствии с состоянием 3.

Для правильной работы УУНП в прикладных управляющих программах необходимо адреса приемников или источников, находящихся в составе УА, передавать в последнюю очередь (в большинстве АИС требование реализуется по умолчанию).

После незначительных доработок УУНП может быть использовано в локальных сетях АИС при передаче управления от основной ЭВМ к ЭВМ, находящейся в составе УА, и обратно.

В устройстве сопряжения каждая линия интерфейса нагружена одним входом логического элемента, а в качестве возбудителей (D12...D15) использованы ИМС К155ЛА13 ($I_{\max} = 48$ мА при $U \leq 0,5$ В).

Передачик и приемник ВОЛС — серийно выпускаемое устройство «Электроника МС4101», с помощью которого можно передавать состояния 19 параллельных линий в ТТЛ-уровнях со скоростью 8 МБод.

Применение ВОЛС с УУНП позволяет создавать распределенные АИС, сохраняя все возможности интерфейса КОП, и за счет электрической развязки с УА увеличить число устройств, объединяемых одной магистралью, до 30.

Телефон 588-37-09, Москва

Статья поступила 3.12.87

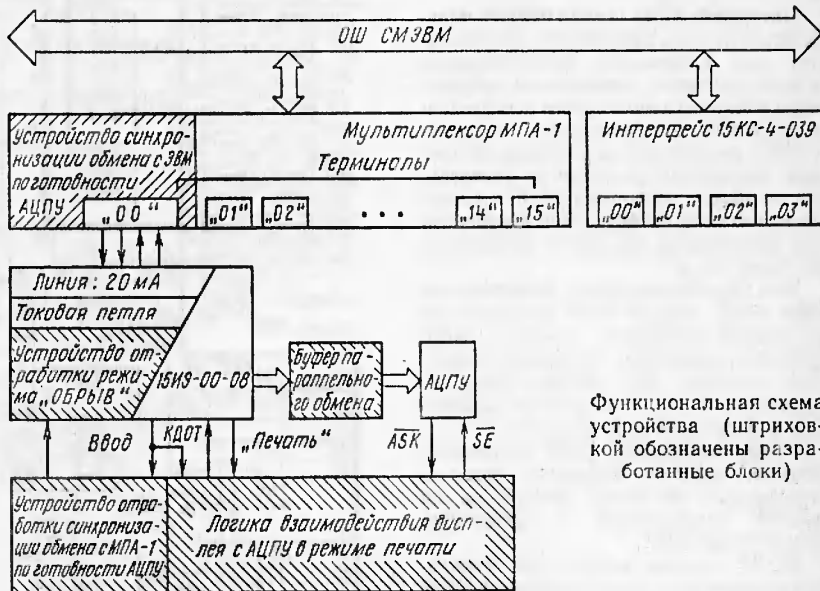
УДК 681.3

Д. Б. Линин, В. В. Корчагин

УСТРОЙСТВО ПОДКЛЮЧЕНИЯ АЦПУ ПРИ МИНИМАЛЬНОЙ ЛИНИИ СВЯЗИ

Устройство предназначено для подключения АЦПУ параллельного действия (ПРД-АЦПУ) DZM-180, DARO-1156 к ЭВМ-системе на базе СМ ЭВМ с помощью периферийных терминалов 15ИЭ-00-013, сопряженных с ней через ИРПС типа контроллер 15КС-4-039 или мультиплексор МПА-1, удаленных от ЭВМ на расстояние до 1000 м, с использованием типичных для ИРПС четырехпроводных линий связи (см. рисунок).

Использование типичной для ИРПС связи не требует дополнительных линий для обеспечения надежности обмена ЭВМ-АЦПУ (синхронизация по готовности АЦПУ) и позволяет организовать полноценное рабочее место потребителя (не зависимое от других) на любом удаленном терминале. Число таких мест ограничивается числом терминалов, поддерживаемых многопользовательской системой. Мультиплексор МПА-1 с 16 ветвлениями заменяет четыре



Функциональная схема устройства (штриховкой обозначены разрабатываемые блоки)

Оно содержит шесть микросхем серии К155 четыре транзистора, два оптрона и реализуется на трех печатных платах: две встраиваются в дисплей, третья — в мультиплексор МПА-1. Дисплей оборудован дополнительным разъемом связи с АЦПУ. Элементы устройства питаются от внутренних источников дисплея и МПА-1 (5 В).

В режиме «Печать» скорость комплекта «Дисплей-АЦПУ» равна скорости примененного АЦПУ; в частности, для DZM-180 средняя скорость обмена — 45...55 строк/мин, максимальная скорость ограничивается возможностями МПА-1 и дисплея (до 9600 бит/с).

контроллера 15КС-И-039 без снижения скорости обмена информацией.

Устройство соответствует требованиям надежности стандартных средств ЭВМ-системы. Подключение одного дополнительного АЦПУ повышает производительность ЭВМ в 1,5 раза. Годовой экономический эффект от внедрения одного устройства — не менее 3 тыс. руб.

601100, Петушки Владимирской обл., ОКБМ при заводе «Токамак», лаборатория САПР; тел. 2-17-25

Сообщение поступило 24.2.88

КОНТАКТ С НАМИ НАДЕЖЕН И ПЕРСПЕКТИВЕН

Предприятиям и организациям, занимающимся разработкой радиоэлектронной аппаратуры и эксплуатацией ЭВМ!

Значительно расширить возможности и придать большую универсальность работе ваших систем позволит применение печатной платы для макетирования, совместимой с персональными компьютерами типа IBM PC.

70 «Микропроцессорные средства и системы» № 2, 1990

Плата дает возможность устанавливать до 90 корпусов типа ДИП, имеет двустороннюю печать, металлизацию отверстий и покрытие контактов разъема под данный тип машин.

Стоимость платы 175 руб.

Наш адрес: 111112, Москва, ш. Энтузиастов, 6. Государственный научно-технический Центр «Контакт», отдел лазерной техники.

Телефон 128-90-06

ИНТЕРФЕЙСНЫЕ МОДУЛИ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Интерфейсные модули ИРПР-16/8 представлены двумя группами устройств: ИРПР-16/8-1 и ИРПР-16/8-2, выполненных на одной полуплате в стандарте микроЭВМ «Электроника 60» и имеющих сопряжение в соответствии с НМ МПК по ВТ 29—80 или ОСТ 11 305.917-84. Сигналы данных управления и состояния передаются и принимаются активными низкими уровнями, однако на разъеме платы предусмотрены резервные входы-выходы с возможностью инвертирования отдельных сигналов периферийных устройств, работающих с сигналами высокого уровня.

Модули ИРПР-16/8 имеют перестраиваемую разрядность. Обмен 16-разрядными словами организован по двунаправленным линиям связи, что существенно сокращает аппаратные затраты. В качестве сигнала состояния от источника предварительно выдается сигнал ВЫВОД, инициируемый микроЭВМ и однозначно определяющий состояние двунаправленных линий данных для предстоящего режима вывода. Для режима ввода сигнал ВЫВОД остается пассивным.

Обмен 8-разрядными словами осуществляется также по 16-разрядному каналу. Однако линии данных при этом становятся однонаправленными: линии младшего байта работают только на вывод, старшего — только на ввод. Принятые данные мультиплексируются в режиме ввода с линиями старшего на линии младшего байта данных в микроЭВМ. Благодаря этому можно использовать стандартное программное обеспечение для управления серийно выпускаемыми периферийными устройствами.

Устройства ввода и вывода функционально не зависят друг от друга, представлено в ИРПР-16/8 каждое с помощью регистра команд и состояний и 16-разрядного (8-разрядного) регистра данных. Однако адреса и векторы прерываний для такой пары устройств являются смежными на магистрали микроЭВМ.

Формат регистра команд и состояний источника (РКС-И): 15РКС-И — ошибка, 7РКС-И — готовность данных источников (вырабатывает также запрос на прерывание), 6РКС-И — разрешение прерывания, 5РКС-И — состояние источника.

Формат регистра команд и состояний приемника (РКС-П): 15РКС-П — ошибка, 14РКС-П — сброс ВУ, 7РКС-П — готовность к приему данных, 6РКС-П — разрешение прерывания, 5РКС-П — состояние приемника (завершено). Запрос на прерывание вырабатывается 7РКС-П или 5РКС-П (определяется распайкой ответной части разъема кабеля интерфейса, т. е.

типом управляемого периферийного устройства). Такая пара устройств ввода-вывода соединяется с модулем ИРПР-16/8 общим интерфейсным кабелем и общим 40-контактным разъемом типа СНО52-40.

Модуль ИРПР-16/8-1 — однопортовый, управляет одной парой периферийных устройств ввода-вывода в режимах программного обмена и прямого доступа к памяти (ПДП).

Модуль ИРПР-16/8-2 — двухпортовый, управляет двумя парами периферийных устройств ввода-вывода только в режиме программного обмена и содержит два 40-контактных разъема А и Б для подключения двух кабелей связи.

Основу модуля ИРПР-16/8-1 составляют пять БИС обмена информацией (ОИ) К1802ВВ1 и одна БИС И К1802ВВ2 (рис. 1).

Каналы А1...А4 четырех БИС ОИ и каналы А5 и В5 пятой БИС ОИ подключены к магистрали микроЭВМ через приемопередатчики К589АП16; канал Х5 пятой БИС ОИ — непосредственно к магистрали для вывода разрядов адреса А16...А19, а два разряда канала С5 этой БИС — через передатчики для вывода разрядов адреса А20, А21.

Каналы Х1...Х4 первых четырех БИС ОИ обеспечивают обмен 16(8)-разрядными данными с ВУ ввода и вывода, а каналы В1...В4 этих БИС используются для мультиплексирования старшего байта данных на линии младшего байта данных при вводе в микроЭВМ в однобайтовом режиме. Наборное поле адресов векторов прерывания подключено к каналам С1,

С2 БИС ОИ и позволяет задавать два смежных адреса для приемника и источника данных.

Однобайтовый режим обмена задается с помощью перемычки. Сигнал ВЫВОД определяет режим работы для двунаправленных линий данных. Схема прерывания выполнена на ИМС М555ВН1, а схема управления ПДП — на БИС К1802ВВ2. Адресный селектор позволяет настраивать модуль на любые четыре смежных адреса из верхних 4К слов адресного пространства микроЭВМ.

Модуль ИРПР-16/8-1 работает в режимах программного обмена с ВУ, ПДП и прямого доступа к РД выбранного ВУ. Распределение регистров первых четырех БИС ОИ: RG0 — 16-разрядный регистр начального адреса памяти (младшая часть адреса); RG3 — 16-разрядный регистр конечного адреса памяти (младшая часть адреса); RG2 — 16-разрядный РД-П, RG1 — 16-разрядный РД-И.

Распределение регистров для пятой БИС ОИ: RG0, RG2 являются частью разрядов РКС-П; RG0 содержит разряды А16...А19 старшей части изменяемого начального адреса памяти; два разряда RG2 соответствуют разрядам А20, А21 старшей, не изменяемой части этого же начального адреса памяти, а два других разряда — это разряды признаков ПДП и ПУСК для задания режима работы. RG3 составляет часть разрядов РКС-И и содержит разряды А16...А19 старшей части конечного адреса памяти; RG1 не используется. Выход переноса СО из пятой БИС ОИ служит для аварийного завершения

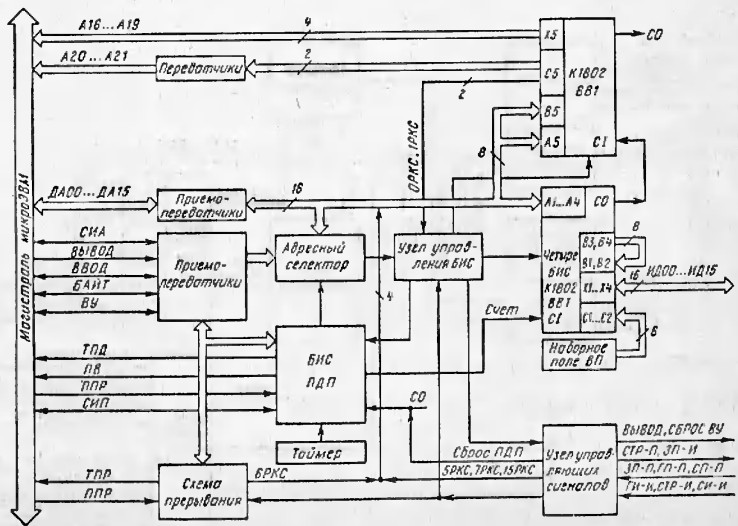


Рис. 1. Функциональная схема модуля ИРПР-16/8-1

режима ПДП при неправильном задании начальных параметров обмена.

Таким образом, модуль имеет изменяемый 20-разрядный адрес А00...А19, охватывающий пространство в 1 Мбайт, и еще два разряда А20, А21 для выбора сегмента памяти, в пределах которого организуется режим ПДП.

Формат РКС-П: 15РКС-П — ошибка, 14РКС-П — сброс ВУ, 13РКС-П...08 РКС-П — разряды А21...А16 начального адреса памяти, 7РКС-П — готовность к приему данных, 6РКС-П — разрешение прерывания, 5РКС-П — состояние приемника (завершено), 1РКС-П — режим ПДП, 0РКС-П — пуск ПДП.

При программном обмене запрос на прерывание вырабатывается 7РКС-П или 5РКС-П (определяется распайкой ответной части разъема кабеля интерфейса), т. е. типом ВУ. В режиме ПДП запрос вырабатывается всегда при окончании ПДП или при появлении ошибки. Появление ошибки может быть вызвано не только пропаданием сигналов ГИ-И, ГП-П от ВУ, но также и появлением сигнала таймаут от схемы контроля времени задержки ответа от ВУ, если оно больше допустимого при приеме очередного слова или байта данных. При этом режим ПДП сбрасывается и микроЭВМ освобождается.

Формат РКС-И: 15РКС-И — ошибка, 11РКС-П...08 РКС-И — разряды А19...А16 конечного адреса памяти, 7РКС-И — готовность данных источника (вырабатывает также запрос на прерывание), 6РКС-И — разрешение прерывания, 5РКС-И — состояние источника, 1РКС-И — вывод по прямому доступу в РД ВУ, 0РКС-И — ввод по прямому доступу из РД ВУ.

Введение двух дополнительных разрядов при прямом доступе к РД ВУ (но не к памяти), например при обмене с квазидиском, связано с тем,

что начальный адрес, выставляемый на магистраль микроЭВМ, в этом случае изменяться не должен, поскольку обмен идет с РД ВУ, имеющим фиксированный адрес. Однако адрес надо постоянно наращивать до совпадения с конечным адресом для выработки сигнала окончания обмена по прямому доступу, поэтому он загружается дважды: второй раз — в свободный РД-И RG1 во время вывода данных (именно с выходов этого регистра адрес выставляется на магистраль микроЭВМ). Во время ввода данных начальный адрес дополнительно загружается в свободный регистр данных приемника RG2 и с его выходов выставляется на магистраль микроЭВМ, не изменяясь до конца обмена. Следовательно, разряды 1РКС-И, 0РКС-И свидетельствуют о том, с какого регистра БИС ОИ будет выставляться неизменяемый начальный адрес на магистраль при обмене с РД ВУ. Режим ПДП может быть снят в любой момент сигналом состояния источника СИ-И, приходящим от ВУ по интерфейвному кабелю ИРПР.

Модуль ИРПР-16/8-1 кроме шести БИС содержит 30 микросхем малой степени интеграции и потребляет ток 2,5 А от источника 5 В.

Этот модуль функционально заменяет модули: И2 (устройство параллельного обмена), И3 (устройство ПДП), И7 (ИРПР), выпускаемые Воронежским ПО «Электроника».

Основу схемы модуля ИРПР-16/8-2 составляют четыре БИС ОИК1802ВВ1, каналы А которых подключены через приемопередатчики К589АП16 к магистрали микроЭВМ (рис. 2).

Каналы Х и С этих БИС обеспечивают обмен (16(8)-разрядными данными ВУ ввода и вывода портов А и Б соответственно. Каналы В1...В4 БИС ОИ мультипликсируют старший байт данных на линии младшего

байта при вводе в микроЭВМ в однобайтовом режиме, который задается двумя переключками отдельно для портов А и Б.

Сигнал ВЫВОД, устанавливающий режим работы для двунаправленных линий данных, формируется предварительно в адресной фазе команды вывода от микроЭВМ; обмен данными для портов А и Б может совмещаться во времени. МикроЭВМ после выдачи слова данных в порт А не задерживается до завершения обмена, а сразу начинает обслуживание устройств порта Б.

Схема прерывания выполнена на двух ИМС М559ВН1; адреса векторов прерывания для портов А и Б устанавливаются независимо друг от друга, что дает определенные преимущества при компоновке системы. Адресный селектор позволяет настраивать модуль на любые восемь смежных адресов из верхних 4К слов адресного пространства микроЭВМ.

С помощью одного модуля ИРПР-16/8-2 можно организовать обмен 16-разрядными данными с другой микроЭВМ имеющей такой же модуль. При этом один порт всегда должен работать только на вывод, другой — только на ввод. Это связано с тем, что в этом случае оба устройства являются активными и могут одновременно обращаться друг к другу, а арбитраж на малом интерфейсе с двунаправленными линиями данных отсутствует. Возможен также и однобайтовый обмен между двумя микроЭВМ. При этом задействуется только один порт модуля, а второй остается свободным для других периферийных устройств.

Модуль ИРПР-16/8-2 кроме четырех БИС ОИ К1802ВВ1 содержит еще 34 микросхемы малой степени интеграции и потребляет ток 2 А от источника 5 В.

Модули устойчиво работают с ленточным кабелем марки ЛВ-30-020 длиной 10 м, подключаемым к обоим портам обмена А и Б. В качестве ВУ использовались микроЭВМ «Электроника МС 1212», печатающее устройство «Электроника УВВПЧ-30-004», а также несколько нестандартных устройств. Скорость обмена модули ИРПР-16-8 не ограничивают.

Телефон 220-46-11, Киев

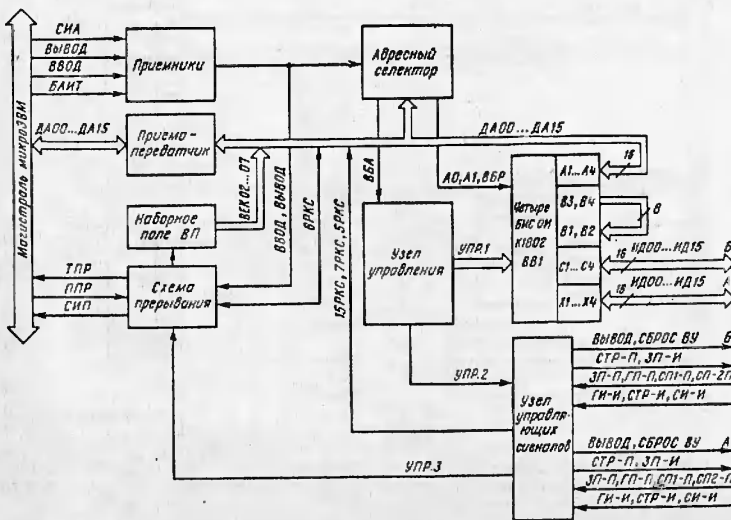


Рис. 2. Функциональная схема модуля ИРПР-16/8-2

Статья поступила 29.07.88

ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА ОБЩЕЙ МАГИСТРАЛИ В МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМАХ

При проектировании мультипроцессорных управляющих вычислительных систем (МУВС) возникает необходимость в организации взаимодействия между отдельными процессорными модулями (ПМ), доступа их через общую магистраль (ОМ) к модулям разделенного ресурса (МРР). Особые трудности возникают для мультиплексных ОМ, например МПИ (ОСТ 11.305.903-80), в связи с тем, что формирование запроса ОМ происходит при обращении ПМ к адресам, соответствующим МРР [1]. Доступ к магистрали обеспечивается поступлением сигнала разрешения (PM) от арбитра, причем к этому времени адрес уже может быть снят с мультиплексной магистрали (рис. 1). Запоминание адреса, восстановление временной диаграммы, соответствующей мультиплексной магистрали и необходимой для обращения ПМ и МРР, требуют значительных временных и аппаратных затрат. Их можно свести к минимуму, если выполнить МУВС на основе МПК БИС K588 [2]. Процессор [3] имеет внутренние сигналы синхронизации передачи информации В, В1 и П. Сигнал В сопровождается информацией из арифметического устройства (АУ), а сигнал В1, поступающий в

АУ с БИС системного контроллера (СК), обеспечивает ввод информации. Сигнал П является ответным сигналу В и подтверждает прием информации.

Простоты обращения ПМ через ОМ к МРР при минимальных затратах времени можно достичь, задерживая поступление сигнала В с АУ на СК, а следовательно формирование фронта 1/0 сигнала СИА до получения результатов анализа попадания выдаваемого АУ адреса в зону, соответствующую обращению к ОМ. При обращении процессора к блокам самого ПМ сигнала В с АУ должен сразу поступать на СК, а при обращении к ОМ — задерживаться до поступления с арбитра сигнала разрешения доступа к магистрали. При этом

$$\begin{aligned} Z_{\text{пр}} &= V_{\text{АУ}} + \overline{\text{СИА}} + \overline{\text{ДА15}}, \\ V1_{\text{СК}} &= V_{\text{АУ}} + Z_{\text{пр}} \cdot \text{PM}, \\ \text{BK} &= \text{PM}, \\ V_{\text{АУ}} &= V1_{\text{СК}} + \text{Ввод}. \end{aligned}$$

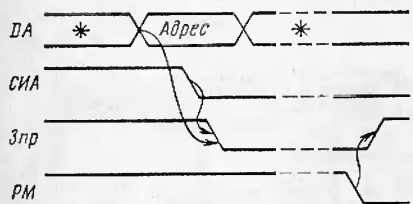


Рис. 1. Сигналы процессора при обращении к общей магистрали

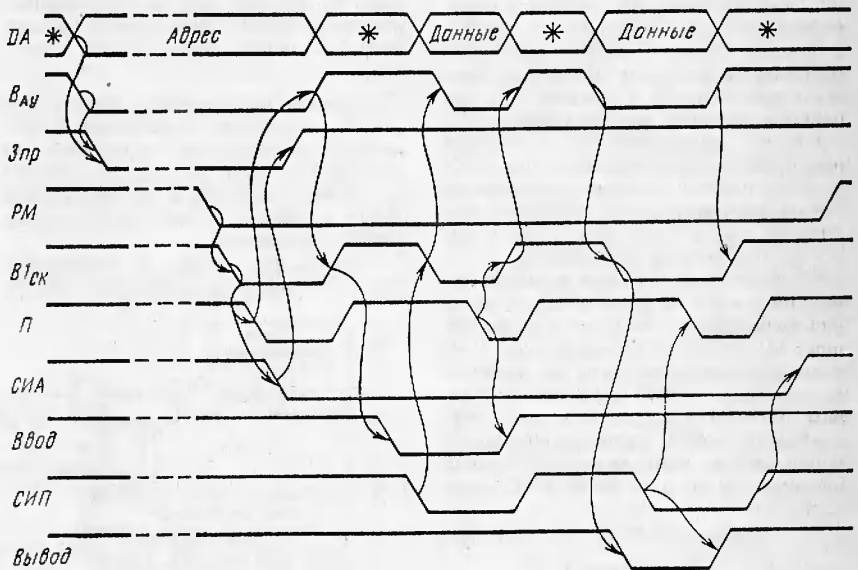


Рис. 2. Временная диаграмма цикла ввод-модификация-вывод

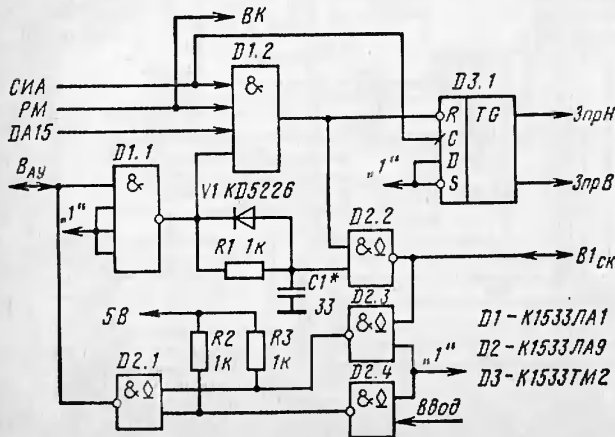


Рис. 3. Интерфейс процессорного модуля

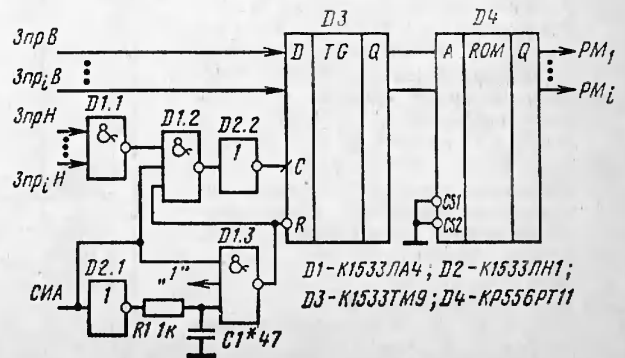


Рис. 4. Арбитр общей магистрали

АУ сохраняет выдаваемый адрес на внутренней магистрали ПМ, так как ответный сигнал П будет отсутствовать, т. е. не потребуются специальные регистры для запоминания адреса. Сигнал В с приходом на СК формирует необходимую временную диаграмму, соответствующую магистрали МПИ, при работе с внутренними блоками ПМ и обращении через ОМ к МРР (рис. 2). Выдачу адреса можно идентифицировать по пассивному уровню сигнала СИА с СК (при выдаче данных уровень сигнала СИА активный).

Допустим, что зона адресов ПМ, соответствующая МРР, составляет 16... 32К (А15=1). Тогда необходимый алгоритм интерфейса каждого ПМ можно отразить выражениями:

где Зпр — запрос ОМ; РМ — разрешение доступа к ОМ; ВК — выборка буферов, соединяющих ПМ и ОМ.

Последнее выражение отражает отсутствие обратной связи между сигналами V_{AU} и $V_{1СК}$ и поступление сигнала V_1 с СК на АУ при вводе информации в процессор. Возможная практическая реализация предложенного алгоритма приведена на рис. 3. Цепочка $R_1 C_1$ и триггер предотвращают «гонки» в схеме интерфейса и арбитра магистральной, обусловленные временными задержками элемента $D_{1.2}$ и буферов ОМ.

Для организации взаимодействия между ПМ, обращения ПМ к МРР и увеличения производительности системы целесообразно использовать параллельный арбитраж запросов ПМ на доступ к ОМ [4]. Схема простого арбитра с высоким быстродействием приведена на рис. 4.

Окончание обращения ПМ к ОМ фиксируется по срезу сигнала СИА на ОМ. По нему же короткий импульс с одновибратора $D_{1.3}$, $D_{2.1}$ обнуляет триггер и снимает сигнал РМ. При появлении запросов на доступ к ОМ код приоритетов записывается в триггер. ПЗУ выполняет функции дешифратора, выдавая сигнал разрешения ПМ со старшим приоритетом. Приоритетность при необходимости легко изменить введением регистра, соединенного со старшими разрядами адреса ПЗУ $A_{n-1} \dots A_n$, и записью в него кода приоритетов.

Предлагаемые в статье решения сокращают в 1,5–2 раза время обращения процессорных модулей к разделенному ресурсу системы, уменьшая в 2–3 раза аппаратные затраты на реализацию интерфейса ОМ и обеспечивая таким образом возможность высокоэффективной работы мультипроцессорных управляющих вычислительных систем, построенных на базе МПК БИС серии К588.

Телефон 23-78-36, 23-53-33, Уфа

ЛИТЕРАТУРА

1. Рыбаков В. Г. Стандартные магистрали для мультипроцессорных вычислительных систем. // Зарубежная радиоэлектроника. — 1986. — № 4. — С. 3. 23.
2. Бобков В. А., Чернуха Б. Н., Свиридович В. С., Ключников В. Н. Расширенный микропроцессорный комплект К588 // Микропроцессорные средства и системы. — 1987. — № 1. — С. 6–7.
3. Черняховский Д. Н., Шиллер В. А., Юровский А. А. Процессор с системой команд и интерфейсом микроЭВМ «Электроника 60» на БИС серии КР588 // Электронная промышленность. — 1983. — № 9. — С. 11–13.
4. Пангишвили И. В., Стецюра Г. Г. Микропроцессорные системы. — М.: Наука, 1980. — 136 с.

Статья поступила 4.01.88

УДК 681.327

Т. В. Бабик, А. Ф. Теслюк

СОПРЯЖЕНИЕ ИНТЕРФЕЙСА МПИ С ВНУТРЕННИМ ИНТЕРФЕЙСОМ ПЭВМ «ЭЛЕКТРОНИКА МС0585»

Системный интерфейс (Э85) серийно выпускаемой ПЭВМ «Электроника МС 0585» имеет ряд отличий от стандартного интерфейса МПИ [1]: в него введены дополнительные сигналы (МАЦВ, МВАД, МД, МДЗПМБ, МДЗПСБ), обусловленные особенностями схемы центрального процессора [2, 3].

Узел сопряжения интерфейсов обеспечивает обмен данными в двух режимах: в режиме программного обмена и ПДП. Он реализован на двух модулях, которые устанавливаются соответственно на позициях 7, 8 кроссплаты системного модуля Э85 (рис. 1). Первый модуль предназначен для сопряжения линий адреса данных, второй — линий управляющих сигналов.

В режиме программного обмена (активное устройство — центральный процессор) информация передается из интерфейса Э85 в циклах записи по сигналу КДЧТН и из интерфейса МПИ в циклах чтения (для режима ПДП — наоборот).

Сигналы управления из интерфейса Э85 поступают в интерфейс МПИ,

причем сигнал МОБМЛ через схемы усилителей на линию сигнала КОБМН, МВУЛ — на КВУН, МДЧТЛ — на КДЧТН, МОТВЛ — на КОТВН, а сигналы КПЗПН и КДЗПН формируются схемой ФВ из МВАД, МДЗПМБ, МДЗПСБ.

В режиме ПДП (активное устройство — ПДП интерфейса МПИ) сигналы управления в интерфейсе Э85 вырабатываются схемой управления, реализованной в виде автомата на ПЗУ (рис. 2). В ПЗУ хранится микропрограмма работы автомата.

В связи с существенными отличиями в реализации схемы обработки прерываний в Э85 от стандартной схемотехники МПИ [1] выполнение процедуры прерывания в интерфейсе МПИ отличается от аналогичной процедуры во внутреннем интерфейсе Э85.

В качестве сигнала, инициирующего процедуру прерывания Э85, может быть использован сигнал КТПРН, формирующий сигнал запроса обмена в интерфейсе Э85 (МЗОАЛ или МЗОБЛ в зависимости от положения переключки во втором модуле), по которому в Э85 генерируется вектор прерывания с кодом 340_6 или 344_6 соответственно.

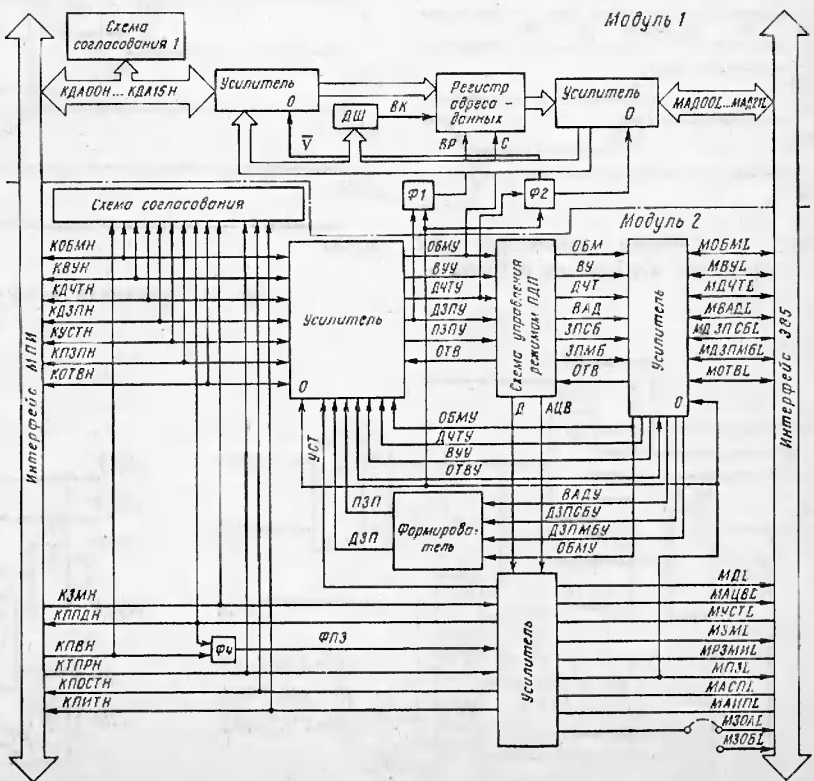


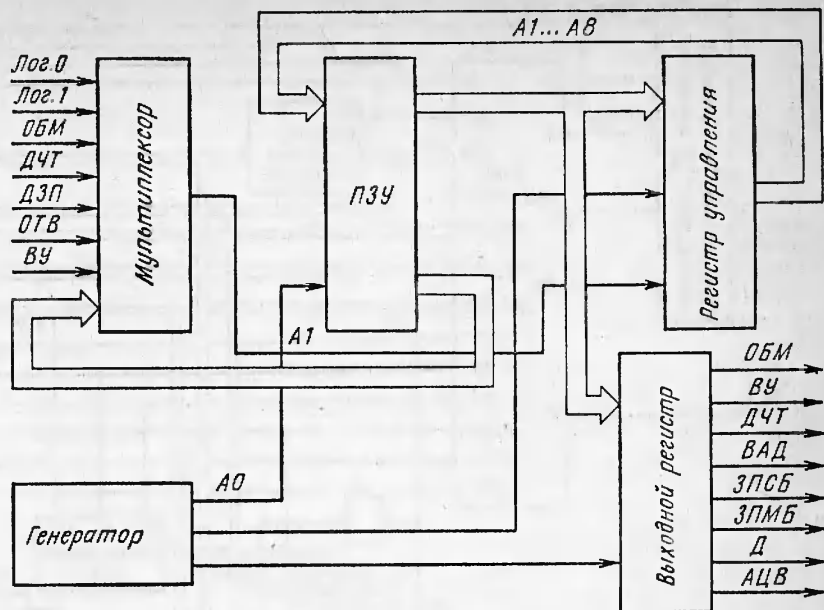
Рис. 1. Функциональная схема узла сопряжения

Рис. 2. Структурная схема автомата ПДП

Дальнейшее обслуживание процедуры прерывания (поиск источника прерывания, приоритет устройств МПИ) осуществляется программно через регистры внешних устройств МПИ (сигнал КТПРН от устройства МПИ необходимо снять). 290060, Львов, Научная, 7 а, НИИ БРЭА; тел. 72-85-23

ЛИТЕРАТУРА

1. ГОСТ 2675.51-86. Межмодульный магистральный параллельный интерфейс МПИ.
2. Хацкевич Л. Д., Проценко И. Г. Профессиональная персональная ЭВМ «Электроника МС 0585» // Микропроцессорные средства и системы. — 1988. — № 2. — С. 3.
3. Комплекс вычислительный персональный «Электроника МС0585». Техническое описание 2.791.026 ТО. Статья поступила 17.03.89



УДК 681.325.—181.48

А. В. Бедарев, В. С. Гравов, В. А. Дмитриев, В. О. Патрикеев

УНИВЕРСАЛЬНЫЙ ОДНОПЛАТНЫЙ КОНТРОЛЛЕР НА ОДНОКРИСТАЛЬНОЙ ЭВМ

Микроконтроллер (МК) предназначен для работы в составе АСУТП децентрализованного типа в качестве узла управления локальным технологическим процессом (ТП). Он может быть использован для управления прибором, сбора данных, связи измерительных и исполнительных устройств с ЭВМ.

МК состоит из следующих узлов (см. рисунок): однокристальной ЭВМ КР1816ВЕ35 со схемой внешнего тактового генератора и схемой формирования сигнала «сброс»;

регистра-засылки младшего байта адреса внешнего запоминающего устройства КР580ИР82;

памяти программ объемом 4 Кбайт на двух микросхемах К573РФ2;

памяти данных объемом 1 Кбайт на двух микросхемах КР541РУ2 со страничной адресацией 256 байт на страницу и схемой выбора ОЗУ;

схемы управления записью-чтением внешних устройств; адаптера параллельного интерфейса КР580ВВ55;

со схемами приемника и передатчика по стандарту ИРПС; трехканального таймера КР580ВИ53;

контроллера клавиатуры и индикации КР580ВВ79; схемы прерываний.

Скорость обмена данными по стандартному интерфейсу ИРПС составляет 75...960 Бод; длина линии связи — до 3 км, число проводов в линии — 4. Интерфейс обладает высокой помехозащищенностью и гальванической развязкой с ЭВМ. МК насчитывает три канала по восемь линий ввода-вывода без стробирования или два канала по восемь линий со стробированием. Предусмотрено пять линий запросов прерывания. Все линии TTL-совместимые. К МК можно подключить до 16 8-сегментных знаковых индикаторов и клавиатуру до 64 кнопок. Напряжение питания снимается с двух источников 5 В, 1А; 12 В, 0,04 А. Построен МК на 18 микросхемах.

ЭВМ исполняет программу, записанную в памяти программ и формирует все необходимые сигналы управления

обменом данными с внешним ОЗУ и периферийными БИС. Порты периферийных БИС адресуются как ячейки внешней памяти данных. Для разделения операций с памятью данных и портами используется разряд А10 адреса (вывод Р22 КР1816ВЕ35). Внешние схемы тактового генератора и формирования сигнала «сброс» обеспечивают правильную последовательность сброса периферийных БИС и ЭВМ. Регистр-засылка фиксирует байт адреса внешнего ЗУ, передаваемый по шине данных. Подключение БИС памяти программ и данных особенностей не имеет. ОЗУ активизируется при А10=0 и наличии \overline{RD} или \overline{WR} . Номер страницы ОЗУ образуют разряды А8, А9 шины адреса МК, т. е. состоящие выводов Р20, Р21 КР1816ВЕ35 в цикле обращения к ОЗУ. Схема управления внешними устройствами анализирует разряд адреса А10 и устанавливает сигналы \overline{IORD} и \overline{IOWT} с временными соотношениями, соответствующими ТУ на БИС серии КР580.

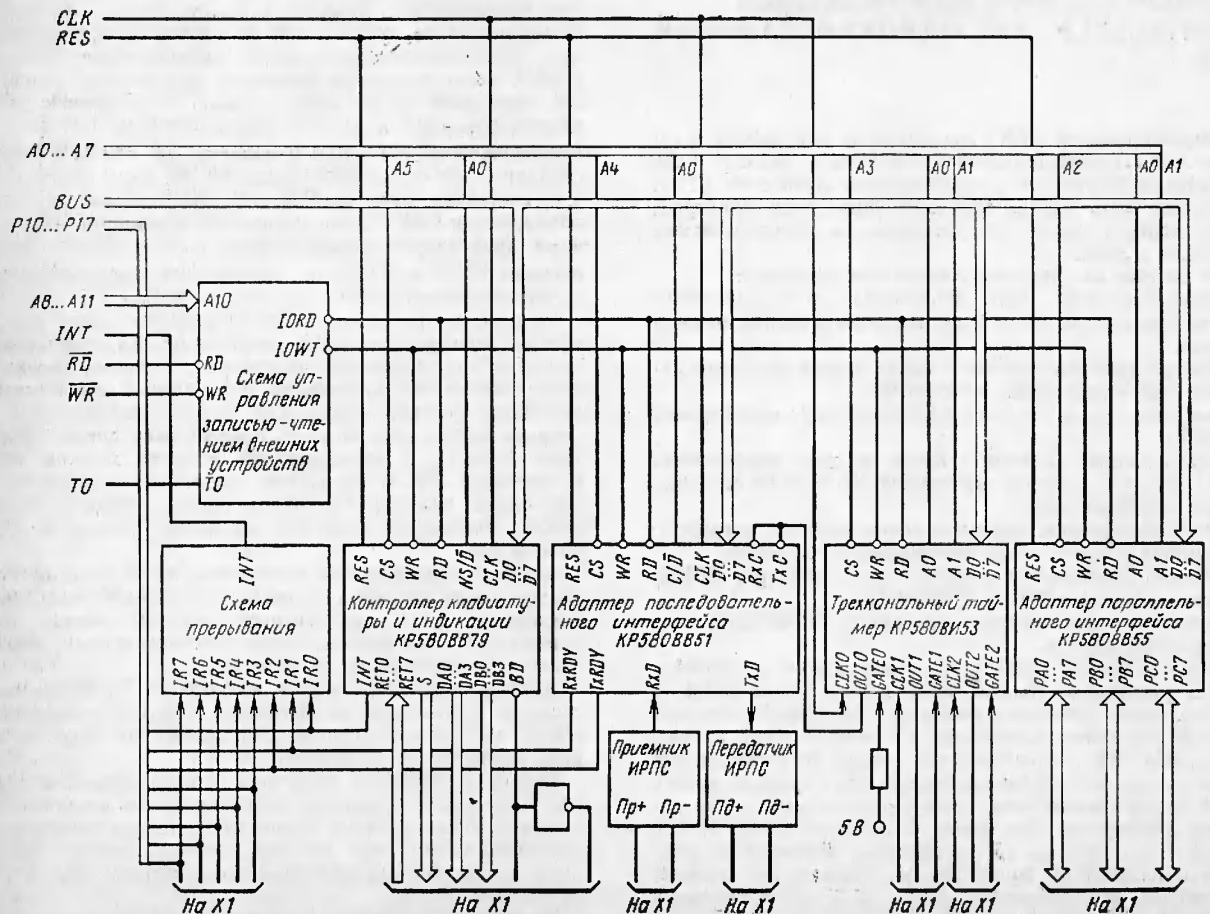
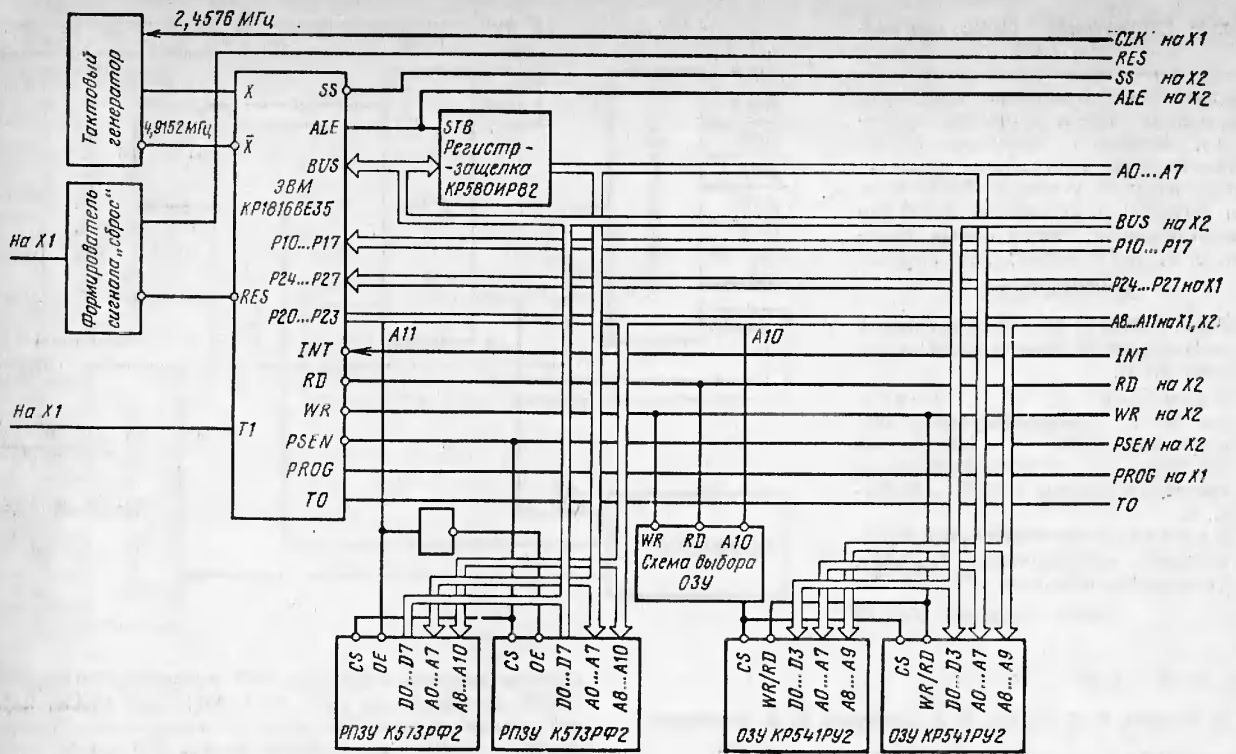
Разряды шины адреса А0 и А1 выбирают порт внутри каждой периферийной БИС, разряды А2...А5 — кристаллы адаптера параллельного интерфейса, таймера, адаптера последовательного интерфейса, контроллера клавиатуры и индикации соответственно. Для этого в соответствующем разряде адреса должен быть сформирован сигнал Лог. 0 (при А10=1), а остальные три разряда должны быть в состоянии Лог. 1. При А10=1 и появлении 0 на более чем одном разряде А2...А5 во время операции чтения может возникнуть конфликт на шине данных и сбой работы МК.

Адаптер параллельного интерфейса КР580ВВ55 обеспечивает стробированный и нестробированный ввод-вывод информации по параллельным каналам связи, сбор данных с внешних измерительных устройств и (или) управление исполнительными устройствами.

Адаптер последовательного интерфейса КР580ВВ51 со схемами передатчика и приемника реализует интерфейс ИРПС для связи с внешней ЭВМ или устройством, имеющим аналогичный интерфейс.

Таймер КР580ВИ53 необходим для измерения временных интервалов (частоты, периода, длительности) или подсчета числа событий. Один из счетчиков микросхемы (нулевой) служит для задания тактовой частоты приемника и передатчика адаптера последовательного интерфейса КР580ВВ51.

Контроллер клавиатуры и индикации обслуживает индикаторное табло, опрашивает органы управления или состояния датчиков.



Функциональная схема контроллера

Схема прерываний формирует из восьми входных сигналов запросов прерываний IR0...IR7 один сигнал запроса прерывания INT для КР1816ВЕ35, причем назначения сигналов IR0...IR3 определены внутренними потребностями контроллера. Использование остальных входов схемы оставлено на усмотрение пользователя. Входы, оставшиеся неиспользованными, необходимо соединить с общим проводом схемы. Для анализа запросов прерываний и возможности работы по опросу при запрещенных прерываниях входы IR0...IR7 соединены также с выводами P10...P17 порта КР1816ВЕ35. Разработанное программное обеспечение позволяет обслуживать запросы прерываний в соответствии с назначенными в программе приоритетами, а также маскировать нежелательные в данном фрагменте программы запросы.

Плата МК снабжена двумя внешними разъемами X1, X2: на X1 выведены сигналы, предназначенные для подключения МК к управляемым устройствам, ЭВМ, клавиатуре, индикаторам; X2 служит для целей отладки и диагностики устройства с помощью специального пульта.

МК эксплуатируется в составе устройства автоматизированного ввода информации о потребителе нефтепродуктов при отпуске их на нефтебазах, а также в автономном измерителе веса.

Телефон 930-90-09, Москва

Статья поступила 27.10.88

УДК 681.33

А. Л. Лилени

КОМПЛЕКСНЫЙ УМНОЖИТЕЛЬ НА МИКРОПРОЦЕССОРАХ К1815ВФ3

Микропроцессоры К1815ВФ3 (рис. 1) с программируемой структурой позволяют одновременно выполнять до трех арифметических операций над пятью операндами. В работе* показана реализация умножителя комплексных чисел на четырех микросхемах К1815ВФ3, а также примеры выполнения других операций обработки сигналов: цифровой фильтрации, дискретного преобразования Фурье. Однако возможности микропроцессоров К1815ВФ3 значительнее выше.

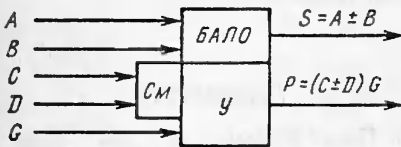


Рис. 1. Условное обозначение и основные функции МП К1815ВФ3

При умножении комплексных чисел $W=U \cdot V$ можно затрачивать не четыре, а три действительных умножения, если воспользоваться алгоритмом**

* Бобков В. А., Сузпаров А. И., Гузик В. Ф. и др. Микропроцессор К1815ВФ3 для цифровой обработки сигналов // Микропроцессорные средства и системы. — 1986. — № 2. — С. 21—23, 85.

** Лилени А. Л. Устройство для умножения комплексного числа на комплексную константу. — А. С. (СССР № 1120316) кл. G06F 7/52. — Опубликовано. 1984, Бюл. № 39.

$$\begin{aligned} \text{Re}W &= \text{Re}U \cdot (\text{Re}V - \text{Im}V) + \\ &+ (\text{Re}U - \text{Im}U) \cdot \text{Im}V, \\ \text{Im}W &= (\text{Re}U - \text{Im}U) \cdot \text{Im}V + \\ &+ \text{Im}U \cdot (\text{Re}V + \text{Im}V). \end{aligned}$$

который легко реализуется на трех микросхемах К1815ВФ3 (рис. 2). Один сум-

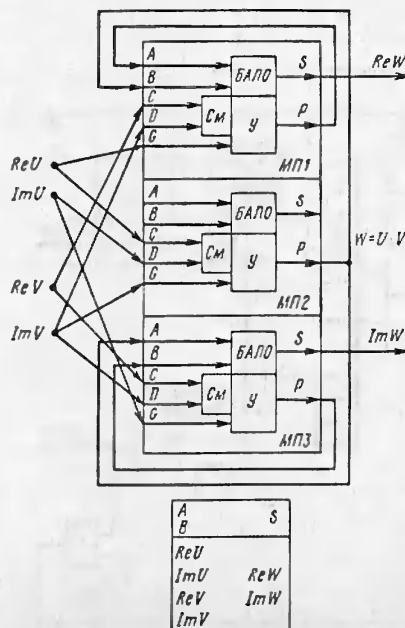


Рис. 2. Умножитель комплексных чисел

«ОРГТЕХНИКА-90» Международная специализированная выставка в Свердловске с 14 по 21 июня 1990 г.

Роботрон Экспорт-Импорт из ГДР выставляет: 32-разрядные ЭВМ с многоместной системой; проблемно-ориентированный комплекс подготовки и обработки производственных данных (ПОК ОПД.Х) со сбором производственных данных (рекомендации КамАЗа);

цифровую обработку изображений для медицины, биологии и материаловедения; кассовый терминал для рационализации продажи билетов;

технику направленной радиосвязи; 16-разрядные ЭВМ с решениями для разработки программ, вычислительных сетей и системы САПР/АСУТП. Наши специалисты ждут Вас!

Роботрон Экспорт-Импорт

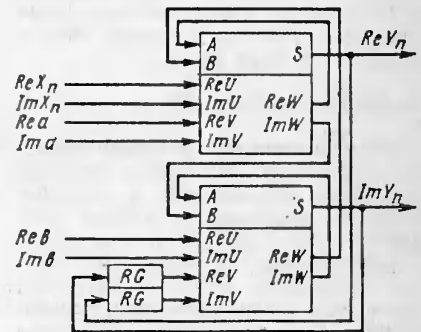


Рис. 3. Рекурсивный комплексный фильтр

матор при этом остается свободным. Он может использоваться при дальнейшем наращивании устройств обработки сигналов.

На рис. 3 приведен пример схемы рекурсивного цифрового фильтра, описываемого уравнением

$$Y_n = aX_n + bY_{n-1},$$

где X_n, Y_n — входная и выходная последовательность отсчетов комплексного сигнала, соответственно ($n = -0, 2, \dots$); a и b — комплексные коэффициенты. При построении фильтра использованы (см. рис. 2) комплексные умножители. Если один из сомножителей, например V , является константой, то сумма $\text{Re}V + \text{Im}V$ и разность $\text{Re}V - \text{Im}V$ можно вычислить предварительно и часть сумматоров высвободить для выполнения других операций.

Телефон 408-61-22, Москва

Сообщение поступило 4.01.88

МИКРОКОНТРОЛЛЕР ДЛЯ ОКЕАНОЛОГИЧЕСКОЙ ИЗМЕРИТЕЛЬНОЙ АППАРАТУРЫ С ПОНИЖЕННЫМ ЭНЕРГОПОТРЕБЛЕНИЕМ

Океанологические измерительные приборы имеют характерные особенности, определяемые спецификой применения: минимально возможное энергопотребление, универсальность по отношению к типу объекта управления и унификация по вводу-выводу.

Как правило, работа океанологических измерительных приборов носит дискретный характер. Например, преобразователи в большинстве случаев опрашиваются с частотой не более 50 раз/с [1] (скважность времени ожидания к времени опроса составляет 10...1000 и более раз). Многопотребляющие схемы можно отключать на время ожидания (энергопотребление прибора уменьшится практически на эту величину). Принцип отключения на время ожидания положен в основу работы микроконтроллера (МК).

Особенности работы МК (см. рисунок)

МК работает в двух режимах: ожидания (устройство управления питанием МК, ОЗУ, входной порт и выходные регистры включены) и работы (МК включен полностью). Устройство управления питанием МК коммутирует питание процессора, РПЗУ и TTL-схем, выявляет источник включения, вырабатывает программируемые временные интервалы и тактовые импульсы синхро-

низации, а также сигналы блокировки и начального импульса сброса.

В устройстве определения источника включения входят: порт С, регистрирующий запросы на включение МК (сигналы с выхода С поступают на входной порт МК, а также на схему ИЛИ, управляющую ключом К1, коммутирующим U_{cc}); мультиплексор (МХ) для сброса соответствующего триггера порта СW после отработки программы.

Устройство выработки временных интервалов содержит кварцевый генератор тактовых импульсов (ГТИ-1), задающий тактовую частоту для интервального таймера (ПИТ), включающего МК через заданный временной интервал, программируемый процессором перед очередным периодом ожидания.

Микропотребляющий кварцевый генератор ГТИ-2 вырабатывает тактовые импульсы, которые через ключ К2 поступают в процессор. Устройство блокировки формирует защитный импульс, предотвращающий прохождение помехи на ОЗУ, и выходные регистры МК (на период коммутации U_{cc} и время режима ожидания) и разрешает прохождение сигнала WRCW (запись в порт СW) через схему ИЛИ на мультиплексор МХ. Схема сброса вырабатывает импульс RES, необходимый для начального обнуления порта СW и ПИТ при первом включении МК.

Программное обеспечение контроллера написано на языке ассемблера и «зашито» в РПЗУ. Оно состоит из программы-монитора, дополненной подпрограммами импульсного ввода-вывода, программы идентификации источника включения и рабочих программ.

Для целей океанологического приборостроения разработано несколько образцов МК на базе МП К580ИК80А, ОЭВМ серии К1816 и комплекта СМ600 — производства НРБ [2], полностью совместимого с микропроцессорной системой МС6800 фирмы Motorola. Все эти варианты практически идентичны по своим характеристикам и строго придерживаются общей структурной схемы, показанной на рисунке.

Технические характеристики МК

	K580	K1816	СМ600
Тактовая частота, МГц	2	2	1
Число входов включения	4	4	1
Объем, Кбайт:			
ОЗУ	8	4	8
РПЗУ	8	4	8
Порты:			
последовательные	1	5	1
параллельные	4×8	2×8	2×8+4 дополнительные линии
Напряжение питания, В	5,0	5,0	18
Потребляемый ток, мА:			
дежурный	0,5	0,4	0,070
рабочий	400	200	650
Габаритные размеры, мм	60×320	60×320	60×320

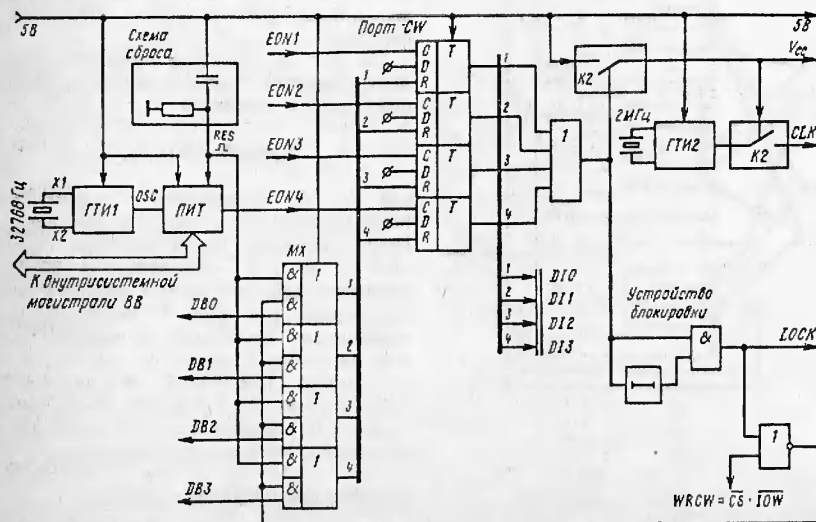
Данные контроллеры разработаны для исследований, выполняемых в Институтах океанологии АН СССР и Болгарии. Опытные образцы устройств успешно прошли испытания на судах.

117218, Москва, В-218, ул. Красикова, 23, Институт океанологии им. П. П. Ширикова АН СССР; тел. 129-19-90

ЛИТЕРАТУРА

1. Парамонов А. Н., Кушнир В. М., Зайкин В. М. Автоматизация гидрофизического эксперимента. — Л.: Гидрометеоздат, 1982. — 223 с.
2. Донов В. С. Микропроцессорный комплект БИС СМ600 // Микропроцессорные средства и системы. — 1988. — № 5. — С. 3—5.

Статья поступила 2.01.89



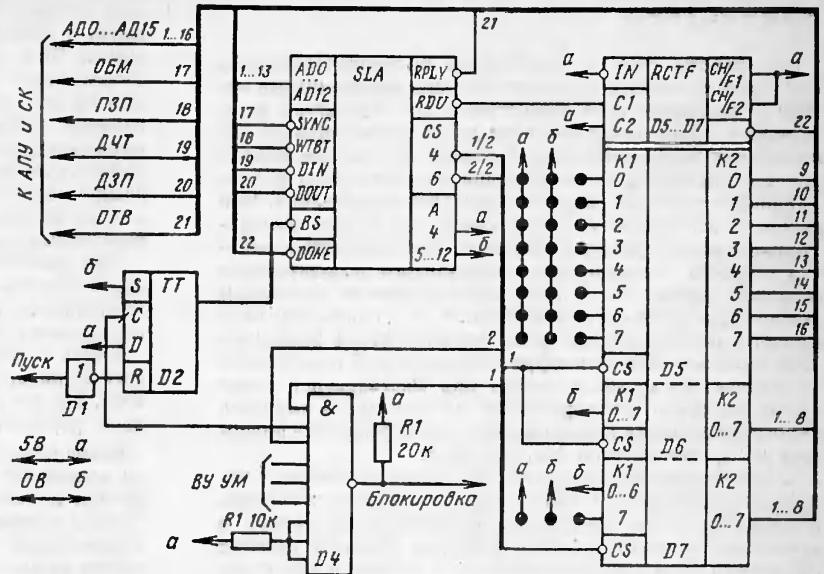
Устройство управления питанием МК

УСТРОЙСТВО НАЧАЛЬНОГО ПУСКА ДЛЯ ВСТРАИВАЕМЫХ МИКРОЭВМ НА БАЗЕ БИС СЕРИИ K588

Устройство обеспечивает автоматический выход процессора на стартовый адрес программы, хранящейся в ПЗУ встраиваемой системы, при включении питания и перезапуске микроЭВМ. Функционально оно представляет собой две ячейки памяти с адресами 24_а и 26_а, программно доступные для чтения только во время выполнения режима начального пуска (при этом предполагается, что на процессоре выбран режим начального пуска с адреса 24_а). Для подключения к ЭВМ* вход Пуск соединяется с одноименным входом микроЭВМ (см. рисунок). Свободные входы D4 подключаются ко входам выборки регистров умножителя SEL1...SEL3 D11, а выход D4 — ко входам выборки шинных формирователей D3, D6, D9 (элемент D12.2 из схемы микроЭВМ исключается). Все остальные цепи-сигналы интерфейса МПИ связаны с соответствующими выходами БИС АЛУ K588BC2 и системного контроллера K588BG1.

При включении питания или повторном запуске микроЭВМ на вход Пуск поступает импульс обнуления триггера D2, разрешающий работу селектора адреса D3. Далее при последовательном обращении процессора к ячейкам 24_а и 26_а устройство выдает стартовый адрес программы и слово состояния процессора (ССП) соответственно. На время обращения к этим двум ячейкам выходные шинные формирователи микроЭВМ переводятся в третье состояние

* Бобков В. А., Горовой В. В., Лаврусенко В. А., Черняковский Д. Н. Одноплатная микроЭВМ на основе БИС серии K588 // Микропроцессорные средства и системы. — 1988. — № 1. — С. 8.



D1 — K564ЛН2; D2 — K1564ТМ2; D3 — K560ВТ1; D4 — K533ЛА2; D5...D7 — K5088А1
Принципиальная схема устройства начального пуска

сигналом с выхода D4, что исключает из обмена все внешние модули системы. Поэтому блокировка нулевого банка ОЗУ на время выполнения начального пуска в данном случае не требуется.

Значения стартового адреса и ССП задаются переключками (для стартового адреса — старший байт D5, для ССП — старший бит младшего байта D7; остальные биты — нулевые). По завершении цикла обращения к ячейке с адресом 26_а на выходе CS6 селектора

D3 формируется положительный перепад, которым в триггер D2 записывается Лог. 1, запрещающая работу селектора адреса; устройство отключается и процессору становится доступным все

подле памяти (в том числе ячейки ОЗУ с адресами 24_а и 26_а). Это позволяет ввести в состав программного обеспечения встраиваемой системы подпрограмму обработки прерывания по аварии источника питания с адрес-вектором 24_а. Следующее включение устройства в работу произойдет только после прихода импульса на вход Пуск.

Кишинев-14, ул. Пушкина, 6, НПО «Квант», отделение ВНИИТ; тел. 24-11-13

Статья поступила 19.10.88

КРАТКИЕ СООБЩЕНИЯ

АВТОМАТИЗИРОВАННАЯ ОБУЧАЮЩАЯ СИСТЕМА

Автоматизированная обучающая система, разработанная в Бакинском научно-учебном центре производственного объединения вычислительной техники и информатики, предназначена для подготовки специалистов по техническому обслуживанию и ремонту средств вычислительной техники. Позволяет обучать основам тестирования устройств и приобретать навыки диагностирования и устранения неисправностей.

Главное достоинство системы — возможность моделирования любой из встречающихся на практике технических неисправностей, воспроизведения ее в изучаемом электронном блоке и программного управления в диалоговом режиме процессом поиска.

Техническая база: диалоговый вычислительный комплекс ДВК2М с подключенным к магистральному каналу дополнительным программно-управляемым устройством имитации технических неисправностей.

Основные функции: диалоговый режим работы, режим «подсказки», распечатка протокола работы на печатающем устройстве, самообучение, обучение, контроль знаний, имитация технических неисправностей практически в любой электронной аппаратуре.

Программы разработаны на языках Паскаль, Макро-ассемблер в рамках ОС ФОДОС 2.

В комплект поставки основного варианта системы входят: устройство имитации технических неисправностей (модификация принципиальной электрической схемы согласуются с заказчиком);

контроллер устройства имитации технических неисправностей;

диск 1 — специализированный резидент, содержащий грузозачные модули программ обслуживания системы;

диск 2 — грузозачные модули обучающих и контролируемых программ;

диск 3 — документация по эксплуатации и обслуживанию комплект технической документации.

370132, Баку, ул. Буниатова, 3, БАКНУЦ АзербПОВТИ; тел. 25-41-82. Алиев Рамиз Тофик оглы

Сообщение поступило 1.09.88

А. А. Жучков, А. И. Мягих, А. Ю. Титков

ПРОГРАММИРУЕМЫЙ ПРИЕМОПЕРЕДАТЧИК КОДА МАНЧЕСТЕР-2

Для работы ЭВМ с удаленными приемопередающими устройствами широко применяются высокоэффективные методы кодирования типа Манчестер-2 [1]. Структура передатчика и приемника этого кода легко реализуется на интегральных микросхемах малой степени интеграции, однако рекомендованные схемные решения подразумевают работу с априорно заданной скоростью передачи информации. При работе с реальными линиями связи, имеющими индивидуальные характеристики, возникает необходимость изменения скорости передачи для оптимального использования возможностей линии. В этом случае схемные параметры передатчика остаются без изменений, а в приемнике надо изменить времязадающие цепи формирователя дополняющего импульса. Эта процедура с последующей подстройкой формирователя продолжительна при многократной смене частот передачи. Формирователь дополняющего импульса с программируемым таймером упрощает перестройку приемника на другие частоты (см. рисунок).

Приемопередатчик собран на БИС последовательного интерфейса КР580ВВ51А (D1) и программируемого таймера КР580ВИ53 (D2). Устройство рассчитано на работу в дуплексном и полудуплексном режимах. Канал 1 таймера D2 должен быть запрограммирован в режиме 3 (делитель частоты) [2] с выходной частотой, вдвое превышающей частоту передачи. Тактовый сигнал поступает на синхровходы триггеров D3.1 и D3.2 через буферный элемент D5.1. Триггер D3.1 работает в режиме счетного триггера и подает тактовую частоту для передатчика D1. Выходной и тактовый сигналы передатчика суммируются по модулю 2 на элементе D5.2 — на выходе сумматора появляется код Манчестер-2. Буферный триггер D3.2 устраняет переходные процессы, возникающие при несовпадении фронтов выходного сигнала передатчика и тактовой частоты, на его выходе — код Манчестер-2 без помех. Этот триггер сдвигает выходной код на полпериода тактовой частоты по отношению к выходу сумматора.

Входной код Манчестер-2 поступает на схемы выделения исходной тактовой частоты и информационного сигнала. Сумматор по модулю 2 (D5.3) с инвертором D5.4 и вы-

ходной RC-цепочкой выделяет фронты входного сигнала, образуя короткий импульс отрицательной полярности при любом фронте входного сигнала. Длительность импульса определяется параметрами RC-цепочки. Канал 0 таймера D2 программируется в режиме 5 с периодом времени, равным 2/3 номинального периода тактовой частоты входного сигнала. При отсутствии смены фазы входного сигнала канал 0 таймера постоянно перезапускается импульсами с выхода D5.3; триггер D4.1 в режиме счетного триггера восстанавливает частоту входного сигнала. При смене фазы триггер D4.1 переключается импульсом с выхода канала 0 таймера (таймер не перезапускается, так как отсутствует импульс с выхода D5.3). Восстановленный тактовый сигнал подается на синхровход триггера D4.2, на выходе которого воспроизводится исходный информационный сигнал. С помощью инверсий тактовой частоты устраняются влияния переходных процессов.

При включении приемника фаза восстановленной тактовой частоты и выходной информационный сигнал устанавливаются произвольным образом. Однако по первому переданному биту происходит самонастройка приемника.

Канал 2 таймера D2 используется для индикации входного сигнала. Он программируется в режиме 1 (одновibrator). На его тактовый вход подается опорная частота, на вход управления — восстановленная тактовая частота, а выход подключается к входу DSR БИС D1. В этом случае он выполняет функцию аналогичную сигналу «индикатор уровня линейного сигнала» (линия 109 интерфейса С2). Период одновибатора выбирается равным 2/3 периода восстановленной тактовой частоты. Уровень Лог. 0 поддерживается на выходе одновибатора входным сигналом.

Максимальная скорость передачи зависит от линии связи и способа сопряжения. В описываемой схеме скорость передачи ограничивается последовательным интерфейсом КР580ВВ51А (максимальная скорость составляет 62 КБод).

123182, Москва, ИМГ им. Курчатова АН СССР; тел. 196-02-00

ЛИТЕРАТУРА

1. Шевкопляс Б. В. Микропроцессорные структуры. Инженерные решения. — М.: Радио и связь, 1986.
2. Торгоев О. И. Программируемый таймер КР580ВИ53 и его применения // Микропроцессорные средства и системы. — 1984. — № 1. — С. 77—84.

Статья поступила 28.12.87

УДК 681.327

А. Е. Краснопеов

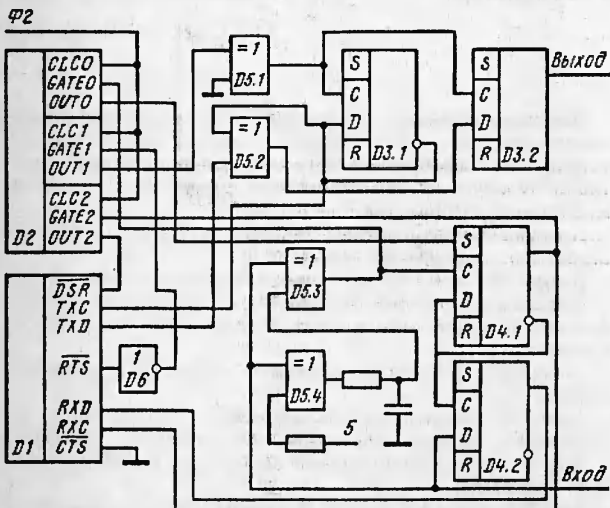
УСТРОЙСТВО СОПРЯЖЕНИЯ ИНТЕРФЕЙСОВ ЕС 9004 И «ЭЛЕКТРОНИКА 60»

Предлагаемое устройство решает часто встречающуюся практическую задачу — регистрацию и хранение данных, получаемых ПЭВМ, входящей в систему физического эксперимента, для последующей обработки с помощью устройства подготовки данных на магнитной ленте ЕС 9004.01 [1]. Модуль сопряжения интерфейсов выполнен на базе серийного И5 [2], выпускаемого в виде полной платы микроЭВМ «Электроника 60». Устройство (см. рис) собрано на свободной половине платы И5 (ред. 3). Разрядность регистров — 8 бит, адрес РС—177014, РД—177016, дешифратор адреса — элемент D1.

Формат РС

РПРД	РПРМ	ППБ	ЗАПРОС
7	6	5	4
3	2	1	0

Разряды 4 и 5 устанавливаются ЭВМ для переключения интерфейсной шины данных соответственно на прием и передачу, разряды 1 и 2 — ЭВМ и устройством подготовки (УПДМЛ) для синхронизации обмена данными. Конструк-

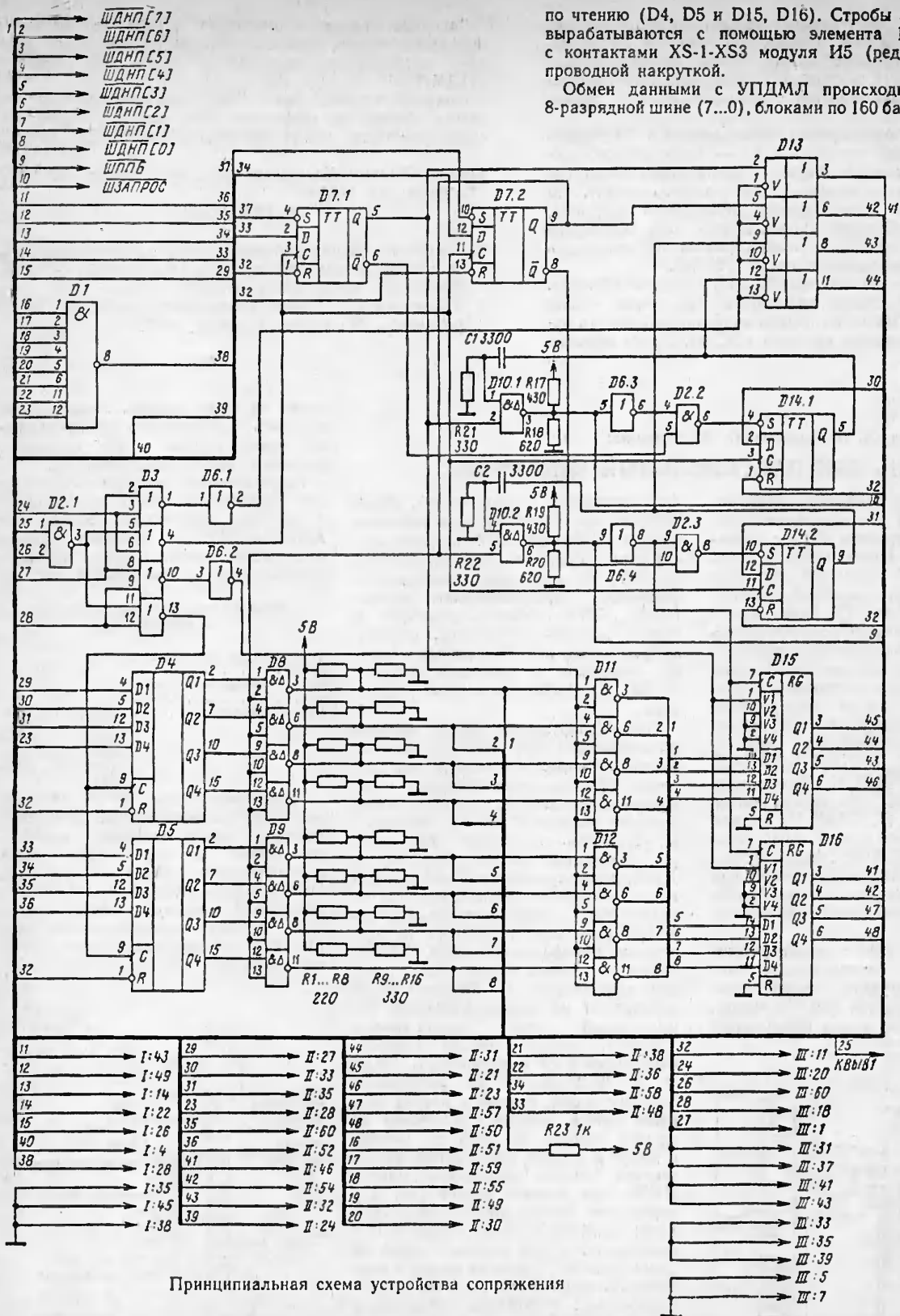


D1 — КР580ВВ51А; D2 — КР580ВИ53; D3, D4 — К155ТМ2; D5 — К155ЛП5; D6 — К155ЛН1

Схема программируемого приемопередатчика кода Манчестер-2

по чтению (D4, D5 и D15, D16). Стробы записи и чтения вырабатываются с помощью элемента D3. Соединения с контактами XS-1-XS3 модуля И5 (ред. 3) выполнены проводной накруткой.

Обмен данными с УПДМЛ происходит побайтно по 8-разрядной шине (7...0), блоками по 160 байтов, с помощью



Принципиальная схема устройства сопряжения

тивно PC представляет собой четыре триггера (D7.1, D7.2 и D14.1, D14.2). Формат РД совпадает с младшим байтом слова данных ЭВМ. Физически это два регистра, один из которых со стороны ЭВМ доступен по записи, а другой —

синхронизирующих сигналов ШПГБ (прием-передача байта) и ШЗАПРОС. По фронту сигнала ШПГБ байт данных поступает в буфер УПДМЛ и устанавливается сигнал ШЗАПРОС на следующий байт. После приема

160-го байта УПДМЛ записывает блок на ленту и посылает сигнал ШЗАПРОС (требование передачи следующего блока). В процессе записи байты, сопровождаемые сигналом ШППБ, поступают на шину данных. Правильность передачи проверяется с помощью данных, принимаемых ЭВМ.

При входе в подпрограмму цикла записи в R0 должно содержаться число блоков, а в R1 — адрес первого элемента блока. Элементы блока — двухбайтные слова (на ленту записывается младший байт каждого слова). По окончании цикла записи блока формируется задержка, так как УПДМЛ контролирует запись (при безаварийной работе на выходе из подпрограммы R0 очищен, в противном случае содержит число 177777_е).

Данные с МЛ считываются УПДМЛ в режиме «Выход». Циклы чтения и записи аналогичны. До начала чтения лента устанавливается на первом записанном блоке (в режиме «Выход» нажата клавиша «ОСВ»). Далее обменом управляет ЭВМ.

УДК 681.324

В. В. Елистратов, П. И. Пальгин, О. В. Трещалин

УНИВЕРСАЛЬНЫЙ ПАРАЛЛЕЛЬНЫЙ ИНТЕРФЕЙС

Универсальный параллельный интерфейс разработан для подключения любых внешних устройств (ВУ) с параллельной организацией шины к микроЭВМ ДВК. К таким ВУ относятся графопостроитель и цветной графический терминал (рис. 1). Связь между устройствами в канале осуществляется по принципу «активный-пассивный». В любой момент времени только одно устройство является активным и управляет циклами обмена информацией в канале. Данные передаются по каналу с помощью специальных сигналов синхронизации К ВВОД Н, К ВЫВОД Н, К СИА Н, К СИП Н (Н — низкий активный уровень). На сигнал активного устройства (К СИАН Н) должен поступить ответный сигнал от пассивного устройства (К СИПН). Адрес пассивного устройства назначается синхронно кодом адреса под управлением фронта установки в активное состояние сигнала К СИА Н [1].

16-разрядный код с мультиплексированной шины адреса-данных микроЭВМ ДВК поступает на магистральный приемопередатчик (МПП), исходно работающий на вывод информации

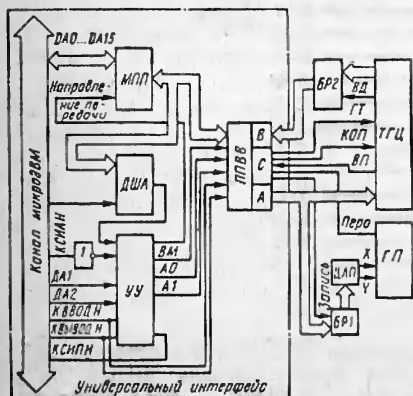


Рис. 1. Блок-схема интерфейса

Задержка передачи принятых байтов на проверку определяется быстродействием УПДМЛ. Ошибка при передаче индицируется сообщением «Сбой» на мониторе УПДМЛ.

При работе устройства с ДВКЗ время записи и чтения одного блока не превышает 300 мс. Без изменения схемы устройство может работать с УПДМЛ ЕС 9002.01.

249020, Обнинск Калужской обл., пр. Ленина, 82, НПО «Тайфун»; тел. 2-55-40

ЛИТЕРАТУРА

1. Устройство подготовки данных на магнитной ленте с интерфейсом для устройств ввода-вывода ЕС 9004. Документы по эксплуатации. БНР, «Изот».
2. Толстых Б. Л. Мини- и микроЭВМ семейства «Электроника». — М.: «Радио и связь», 1987.

Статья поступила 12.07.88

слова на шину данных. Можно организовать стробированный обмен данными между портом и ВУ, используя регистр С как управляющий [2].

Разработанный интерфейс осуществляет duplexный обмен информацией с цветным графическим терминалом А543-11 (ТГЦ) и передачу данных на графопостроитель Н-306 (ГП) с преобразованием в аналоговый сигнал.

Основные технические характеристики терминала

Разрешающая способность, точек	320×270
Число индицируемых цветов	7
Обмен информацией	16-разрядная шина с двумя контрольными битами (КОП — код операции, ГТ — готовность)

Графопостроитель (ГП) регистрирует две взаимосвязанные логические величины, представленные в аналоговой форме. Для организации связи с графическими средствами регистры ППВВ программируются следующим образом: А — на вывод адреса ячейки ОЗУ, данных видеoinформации, для ГП; В — на ввод содержимого ячейки видеоОЗУ; С1 (младший байт)

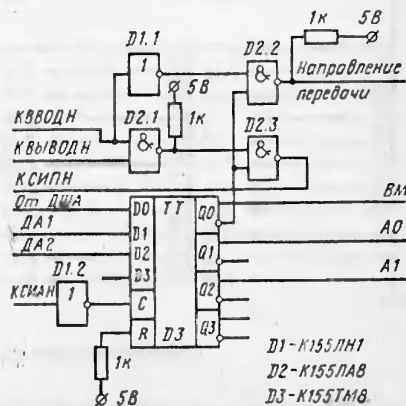


Рис. 2. Принципиальная схема устройства управления

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
КОП				Адрес ОЗУ терминала											

Рис. 3. Формат команды цветного терминала

— на вывод управляющих сигналов КОП и ГТ; С2 (старший байт) — на ввод сигналов состояния. Лог. 1 на шине КОП сопровождает команду и адрес ячейки ОЗУ ТГЦ из А в формате, указанном на рис. 3, а Лог. 0 — поступление данных видеoinформации или цвета (в зависимости от прошедшего ранее адреса ОЗУ цветного терминала). Сигнал ГТ инициирует запись либо команды с адресом, либо информации в ТГЦ. Данные из ОЗУ терминала отображаются на экране цветного монитора. При поступлении команды «Чтение данных» из ячейки памяти терминала на его выходной шине появляется информация о содержимом указанной ячейки. Сигнал «Выдача» (ВД) стробирует информацию в буферном регистре (БР), которую можно считать из регистра В ППВВ. Сигнал «Выполнено» (ВП), фиксирует завершение цикла запись-чтение и окончание передачи данных.

Универсальный параллельный интерфейс позволяет управлять ГП с помощью ППВВ, используемого для связи с ТГЦ. Управляющие сигналы

регистра С1 разделяют функции. Через регистр А на ГП выдаются два байта, определяющих координаты пера. Весь интерфейс выполнен на 20 микросхемах и занимает половину платы ДВК. ПО для интерфейса может быть написано на любом языке, имеющем команды обращения к ячейкам памяти (для ассемблера это команды пересылки данных по адресам, назначенным для данного ВУ, для языка Паскаль — сопоставление переменных и содержимых ячеек памяти с помощью процедуры ORIGIN). Разработано ПО для графических средств. Любое графическое изображение может быть представлено совокупностью элементарных объектов — точек и отрезков прямой.

Для вывода примитива на экран ТГЦ или ГЦ разработан драйвер на языке MACRO-11. В его систему команд входят две процедуры отображения — точки и отрезки заданного цвета. Для ГП вместо цвета передается команда поднять или опустить перо. По команде на экран или поле графопостроителя выводится примитив. Драйвер включается в состав ОС и загружается вместе с ней. Пользователь, зная систему команд и размеры полей цветного терминала и графопостроителя, на любом языке высокого уровня (БЭЙСИК, Фортран, Паскаль,

Си) может создать необходимый графический пакет, используя примитивы, определенные драйвером (можно использовать стандартные графические пакеты программ, переписав базовые процедуры). В качестве примера был создан небольшой пакет графических программ, включающий в себя построение образов, фигур, кривых, масштабирование этих образов и преобразование их координат, построение цифр и текста из русских и латинских символов. Основное назначение драйвера — использование устройств, подключаемых к интерфейсу, как стандартных (АЦПУ, системный терминал, накопитель).

115409, Москва, Каширское ш., д. 31, МИФИ, каф. 27; тел. 324-31-34 (Елестратов Вячеслав Васильевич)

ЛИТЕРАТУРА

1. МикроЭВМ «Электроника НМС 11100.1» (МС 1201).— Техническое описание. 0.305.019. Т01. 1985.
2. Хвош С. Т., Варлинский Н. Н., Попов Е. А. Микропроцессоры и МикроЭВМ в системах автоматического управления: Справочник.— Л.: Машиностроение, 1987.— С. 85—90.

Статья поступила 2.08.88

УДК 681.327

С. П. Медведев, В. Б. Абрамов, Р. М. Печерская, В. В. Коршунов

МНОГОКАНАЛЬНЫЙ ИНТЕРФЕЙС ПАРАЛЛЕЛЬНОГО ОБМЕНА

Основу интерфейса для подключения ВУ к магистральному параллельному интерфейсу (МПИ) составляют БИС серии КР580 (см. рисунок). Интерфейс позволяет принимать по восьми каналам 16-разрядные слова, информацию о готовности восьми ВУ, организовать прерывания и передачу сигналов запуска для пяти ВУ, передать и принять одно 16-разрядное слово. Согласование с двуправленной шиной МПИ осуществляется шинными формирователями D1...D4. Они все время включены на прием информации от ЭВМ и только в момент активного сигнала К ВВОД переключаются на передачу к ЭВМ. Таким образом, прием адреса возможен в любой момент времени. Дешифрация адреса реализуется элементами D5...D8.1 и буферным регистром D10. В адресном пространстве ЭВМ интерфейс занимает область 160000...160010. Первые четыре четных адреса связаны с БИС параллельного обмена, адрес 160010 предусмотрен для организации прерываний.

Применение двух БИС КР580ИК55 дает возможность получить три 16-разрядных канала А, В и С. Половина канала С (восемь разрядов) используется для приема сигналов готовности от ВУ. Из режима чтения в режим записи БИС переключается сигналами К ВВОД и К ВЫВОД.

Канал А рассчитан только на прием информации и соединен с 16 мультиплексорами К155КП5 (D14...D29), позволяющими увеличить число принимаемых каналов до восьми. Выходные сигналы многих измерительных приборов (например, цифровых измерителей емкости, вольтметров, АЦП) не согласованы с TTL-логикой и имеют уровень Лог.1, равный 6 В, и Лог.0, равный — 6 В, поэтому на

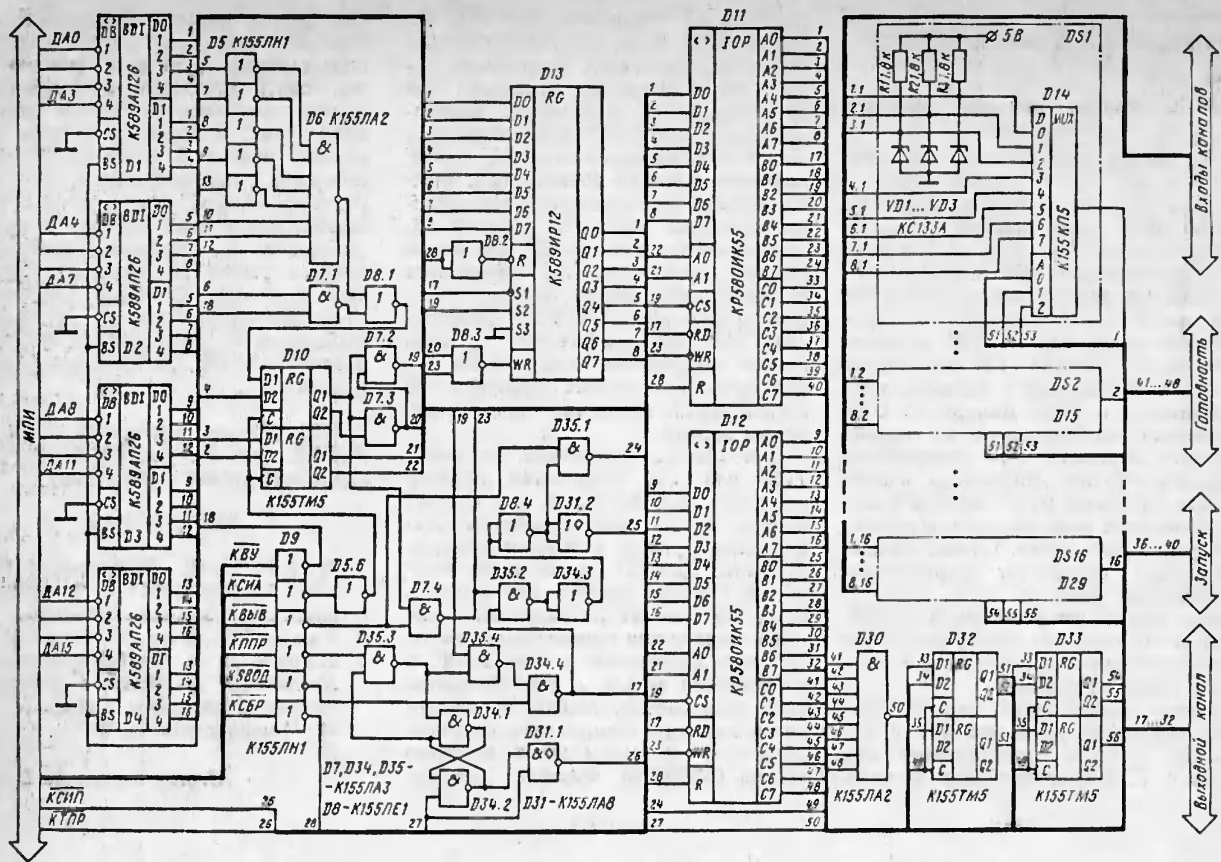
входе трех каналов установлены преобразователи уровней R, VD1. Мультиплексоры (номер принимаемого канала) управляются сигналами второй половины канала С, которые запоминаются в буферных параллельных регистрах D32, D33. Регистры стробируются сигналом К ВЫВОД и работают каждый с восемью мультиплексорами.

Протокол обмена с МПИ предполагает ответный сигнал К СИП от пассивного устройства, формируемый из К ВВОД или К ВВОД с помощью D7.4, D9.4 и D31.2.

Прерывания организованы следующим образом. Информация о готовности ВУ к обмену поступает на канал С и элемент D30 (эти сигналы обычно передаются логическим нулем). При появлении любого сигнала готовности срабатывает RS-триггер (D34.1, D34.2) и с помощью D31.1 формируется сигнал требования прерываний К ТПР, который поступает в канал ЭВМ. ЭВМ передает сигнал К ППР, активизирующий многофункциональный буферный регистр D13 для передачи вектора прерываний, сбрасывает RS-триггер обеспечения прием следующего сигнала готовности. Передача вектора прерываний стробируется сигналом К ВВОД. Далее ЭВМ начинает обработку прерываний.

Все восемь сигналов готовности имеют одинаковый приоритет и один вектор прерываний, поэтому для получения информации о номере устройства необходимо считать данные канала С БИС параллельного обмена. Применение микросхемы К589ИР12 для хранения вектора прерываний позволяет легко перепрограммировать вектор в процессе работы программы. Для этого нужно значение записывается по адресу 160010. Для запуска ВУ задействованы пять разрядов канала С, но если этого недостаточно, можно использовать также канал В.

Таким образом, перепрограммируемый интерфейс обеспечивает подключение к ЭВМ восьми 16-разрядных каналов, управление запуском ВУ, прием информации об их готовности. Три из восьми каналов имеют преобразователи уровней. Дополнительный канал (канал В) можно



Принципиальная схема многоканального интерфейса параллельного обмена для ДВК

использовать для приема и передачи информации, а также в качестве двух независимых 8-разрядных каналов. Если микросхему K155КП1 заменить на K155КП5, то число принимаемых каналов можно увеличить до 16.

440017, Пенза, ул. Красная, 40. Пензенский политехнический институт, каф. микроэлектроники; тел. 69-84-29

Статья поступила 14.07.88

УДК 681.325

Б. Г. Борисов

ВНУТРИСХЕМНЫЙ ПРОТОТИП-ЭМУЛЯТОР ПЭ-48

Универсальный одноплатный прототип-эмулятор предназначен для создания аппаратно-программного отладочного комплекса разработки микроконтроллеров на основе ОЭВМ КР1816ВЕ35, КР1816ВЕ39 и КМ1816ВЕ48 с использованием стандартных кросс-средств. Созданный на основе ПЭ-48 отладочный комплекс позволяет изучить особенности функционирования ОЭВМ данного семейства: использовать все ресурсы ОЭВМ без каких-либо ограничений, отлаживать программные и аппаратные средства контроллера. Процесс отладки программных средств ОЭВМ включает assembling исходного текста программы; загрузку полученного объектного кода для ОЭВМ в ПЭ-48; аппаратное (через соответствующий разъем) и (или) программное (из программной модели) введение в прототип-процессор ПЭ-48 внешних (по отношению к разрабатываемому контроллеру) сигналов и информации;

устранение выявленных ошибок путем непосредственной коррекции объектного кода или изменения исходного текста программы.

В процессе отладки разработчик имеет возможность создавать и анализировать до 15 полных копий ОЭВМ в различных контрольных точках (число которых не ограничено) и синхронизировать работу отладчика с обрамляющей моделью. После окончания отладки можно записать полученный объектный код программы в резидентное УФРПЗУ ПЭ-48. Разработка прототипа-эмулятора для конкретного типа ЭВМ (СМ3, СМ4, СМ1420, ДВК2, ДВК3, ДВК4, СМ1800, ЕС1840, ЕС1841 и др.), а также его конфигурация (одноплатный вариант или с выносным прототип-процессором) осуществляются по договору с заказчиком.

Типовой вариант ПЭ-48 выполнен на ОЭВМ КР1816ВЕ35, трех БИС КР580ВВ55А, четырех БИС КР537РУ10, БИС КР537РУ3А, К573РФ2 и 37 ИМС серий К555 и К561.

Телефон 273-88-02, Москва

Сообщение поступило 9.12.88

УДК 681.326.34

В. И. Джиган, С. В. Щербаков

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ С УНИФИЦИРОВАННЫМ ИНТЕРФЕЙСОМ МПИ

Ширина спектра частот аналоговых сигналов определяется временем обработки одного отсчета T и ограничивается значением $\Delta F = 1/2T$. Интервал T зависит от быстродействия микроЭВМ, алгоритма обработки, способа сопряжения микроЭВМ с АЦП, в частности от алгоритма передачи данных (АПД) из АЦП в микроЭВМ.

В устройствах обработки сигналов используются синхронный и асинхронный АПД: если параметры устройства

(алгоритма) обработки не изменяются во времени и интервал T не меньше времени преобразования АЦП, то синхронный, если изменяются — асинхронный. Выполнение синхронного АПД сводится к исполнению команды запуска АЦП на предыдущей итерации алгоритма обработки сигналов, а преобразованные данные используются на следующей итерации. При выполнении асинхронного АПД АЦП сопрягается с микроЭВМ посредством регистра состояния (РС), устанавливающего готовность АЦП к преобразованию, запуск АЦП на преобразование и готовность последовательности обращений РС даже для быстродействующих АЦП оказывается существенным, что приводит к увеличению интервала времени T , а следовательно, уменьшению допустимого значения ΔF .

Однако при использовании быстродействующих АЦП, у которых время преобразования аналоговых сигналов не превышает длительности цикла обмена информацией

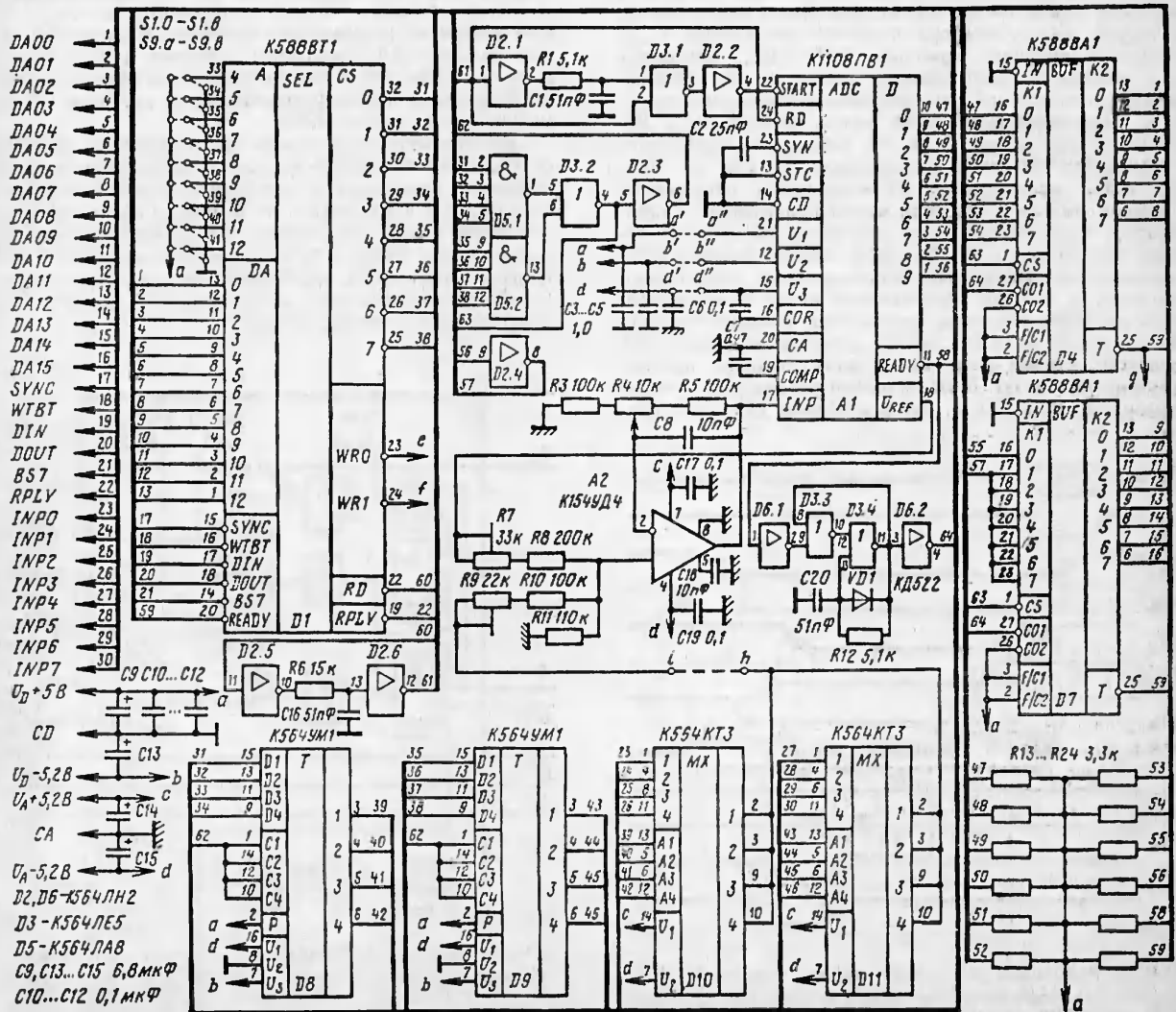


Рис. 1. Принципиальная схема модуля АЦП

ЦП микроЭВМ, можно сократить срок выполнения АЦП. Для этого необходимо обеспечить выборку аналогового сигнала, запуск АЦП, преобразование и считывание данных с АЦП в течение цикла ввода информации.

Принципиальная схема соответствующего модуля АЦП приведена на рис. 1, а временная диаграмма его работы — на рис. 2. Модуль разработан на основе ИМС серии К564 и БИС серии К588 для малопотребляющей микроЭВМ — на основе МПК КМДП БИС серии К588 [1]. БИС АЦП К1108ПВ1 (А1) работает в 10-разрядном режиме с временем преобразования аналогового сигнала 0,9 мкс. БИС имеет внутренний тактовый генератор и источник опорного напряжения. Вход STC задания режима работы (10-разрядный или 8-разрядный) БИС АЦП соединяется с общей шиной CD источников питания UD цифровых цепей модуля; вход компенсации COMP внутреннего источника опорного напряжения через конденсатор C7 — с общей шиной источников питания UA аналоговых цепей модуля [2].

Работает модуль АЦП следующим образом. При обращении в цикле ввода по одному из восьми адресов, задаваемых на шине адреса А селектора (СА) D1, на выходе СА формируется один из сигналов CS0...CS7, по которому с помощью аналогового мультиплексора D10, D11 выбирается канал входного сигнала. Аналоговый сигнал с выхода мультиплексора поступает на один из входов сумматора, реализованного на операционном усилителе А2. На второй вход сумматора подается напряжение UREF внутреннего опорного источника БИС АЦП. Благодаря этому уровень входного сигнала смещается на половину диапазона напряжений преобразуемых униполярных сигналов и достигается биполярный режим работы АЦП. Регулировкой резисторов R4 и R9 добиваются требуемого коэффициента передачи аналогового сигнала на вход БИС АЦП; регулировкой R7 — смещения напряжения на выходе А2 таким образом, чтобы при нулевом напряжении на аналоговых входах устройства на выходах регистра данных АЦП D9...D0 после преобразования устанавливался код 1000000000. Для получения дополнительного кода, в котором представлены числа в микроЭВМ семейства «Электроника», достаточно проинвертировать старший разряд кода D9...D0. На шину адреса-данных DA микроЭВМ данные поступают с магистральных приемопередатчиков D4, D7. В D7 знаковый разряд расширяется на разряды DA10...DA15, что необходимо для согласо-

вания форматов данных АЦП и 16-разрядной шины DA микроЭВМ.

Цикл ввода данных сопровождается сигналом DIN, по которому на выходе СА формируется сигнал RD. С помощью элементов D2.5, D2.6 и цепочки R6, C16 сигнал RD СА задерживается на время t_1 . Длительность этого интервала определяется задержкой аналогового сигнала выбранного канала на выходе мультиплексора относительно сигналов CS0...CS7 СА и временем установления сигнала на выходе входного операционного усилителя. С помощью элементов D2.1, D2.2, D3.1 и цепочки R1, C1 по фронту задержанного сигнала RD (RDDL) формируется короткий импульс запуска АЦП START.

Минимальная длительность сигнала START равна периоду импульсной последовательности внутреннего тактового генератора, а максимальная не должна превышать время преобразования АЦП. Длительность сигнала START задается цепочкой R1, C1. Ограничение на максимальную длительность этого сигнала накладывается из-за того, что при низком уровне на входе запуска по истечении времени преобразования t_4 происходит автоматический перезапуск АЦП. По окончании интервала времени t_4 АЦП вырабатывает сигнал готовности данных READY, который через одновибратор на элементах D3.3, D3.4, D6.1, D6.2 поступает на вход управления направлением передачи информации CO1 магистральных приемопередатчиков. Длительность этого сигнала определяется цепочкой R12, C20 исходя из минимальной длительности передачи данных по шине DA. Появление истинных данных на этой шине сопровождается низким уровнем сигнала на выходе T магистральных приемопередатчиков, по которому СА формирует сигнал ответа RPLY.

Согласно протоколу обмена информацией в микроЭВМ семейства «Электроника» максимально допустимое время t_5 не должно превышать 10 мкс. При использовании данного модуля АЦП в микроЭВМ на основе микропроцессорного комплекта КМДП БИС серии К588 длительность сигнала цикла обмена SYNC, в течение которого происходит выбор канала, запуск АЦП, преобразование аналогового сигнала и ввод полученного кода, не превышает 5 мкс при выполнении команд с вводом данных.

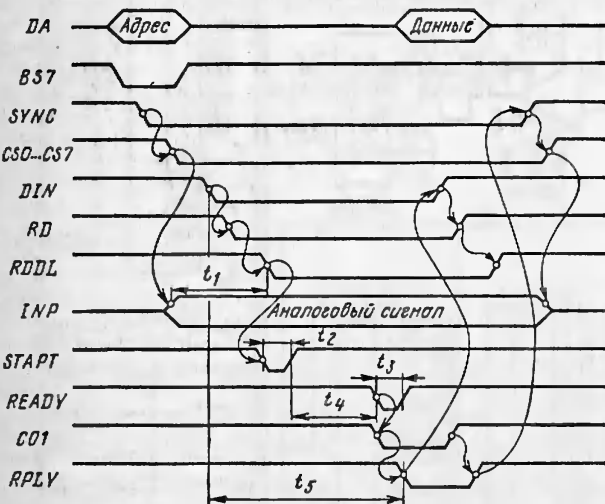


Рис. 2. Временная диаграмма работы модуля АЦП:

t_1 — время установления аналогового сигнала на входе; $t_2 = 1/f_T$ — минимальная длительность сигнала запуска; $t_3 = t_2$ — длительность сигнала готовности данных; t_4 — время преобразования; $t_5 \leq 10$ мкс — максимально допустимое время задержки сигнала RPLY относительно сигнала DIN; f_T — частота внутреннего тактового генератора

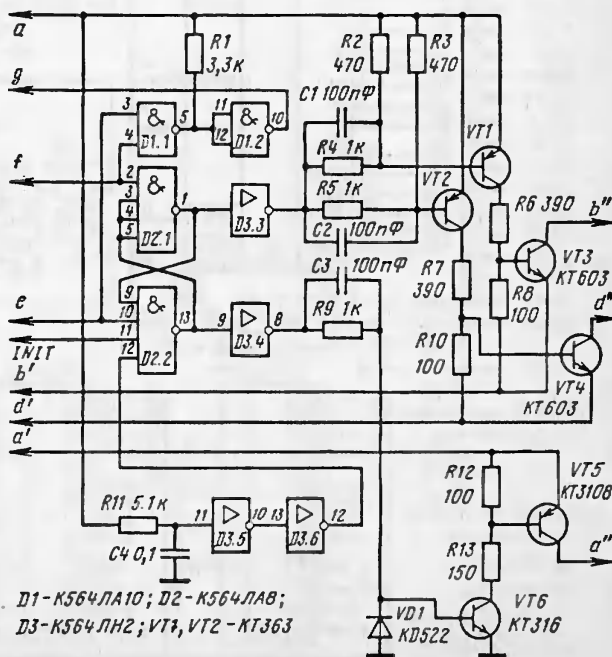


Рис. 3. Устройство ключевого питания БИС АЦП

Важным и интересным является способ снижения потребляемой модулем мощности, которая определяется в основном БИС АЦП и равна 0,9 Вт [2], что в несколько раз превышает мощность потребления микроЭВМ на основе МПК КМДП БИС серии К588 в динамическом режиме [1]. Эффективное средство снижения мощности, потребляемой модулем, — ключевое питание БИС АЦП. Время выхода БИС К1108ПВ1 на рабочий режим после подачи питающих напряжений не превышает 1 мс во всем диапазоне рабочих температур. Эта величина обусловлена зарядом конденсаторов С6 и С7. Емкость этих конденсаторов обеспечивает правильное функционирование БИС К1108ПВ1 [2]. Если аналоговые сигналы выбираются с частотой меньше 1 кГц или между сериями преобразований, проводимых с высокой частотой, есть длительные паузы, то модуль целесообразно дополнить устройством ключевого питания БИС АЦП (рис. 3), которое подключается между точками «а-а», «b-b», «d-d». Устройство управляется сигналами байтовой записи данных WR0 и WR1, вырабатываемыми СА при выполнении команд пересылки байтов произвольных чисел по любому из допустимых адресов. Оно содержит RS-триггер (элемент D2), два ключа отрицательного (VT1, VT3 и VT2, VT4) и ключ положительного напряжения (VT5 и VT6). Особенность ключей — низкое (несколько десятков микроватт) потребление мощности в выключенном состоянии.

При подаче питающих напряжений на модуль АЦП устройство устанавливается в выключенное состояние. В начале программного формирования интервала времени подачи питания на БИС АЦП (>1 мс) сигнал с выхода WR1 СА (рис. 1) поступает на вход RS-триггера (точка «f») (рис. 1, рис. 3) и переводит устройство ключевого питания во включенное состояние. По истечении указанного интервала времени сигналом с выхода WR0 СА (точка «e») RS-триггер переводится в исходное состояние и питание БИС АЦП снимается. Сигнал RPLY при выводе байтовой информации формируется логическими элементами D1.1, D1.2 (рис. 3). Выход логического элемента D1.2 (рис. 3) соединяется со входом готовности данных READY СА (точка «g») (рис. 1, рис. 3).

Телефон 535-96-42, Москва, Джиган В. И.

ЛИТЕРАТУРА

1. Джиган В. И., Лужецкий С. Г. МикроЭВМ управления фазированной антенной решеткой // Архитектура, схемотехника и математическое обеспечение микропроцессорных систем: Межвузовский сб. — М.: МИЭТ. — 1986. — С. 131—136.
2. Игнатов Б. И., Рябов Е. А., Сотский Д. В. и др. Быстродействующий однокристалльный АЦП К1108ПВ1 // Электронная промышленность. — 1986. — № 2. — С. 16—19.

Статья поступила 4.08.88

УДК 681.326

А. Е. Злачевский

ОБ ОДНОМ НЕПРИЯТНОМ «СЮРПРИЗЕ» БАЙТОВОГО ПАРАЛЛЕЛЬНОГО ИНТЕРФЕЙСА, ВЫПОЛНЕННОГО НА БИС СЕРИИ К1801

Устройство байтового параллельного интерфейса (УБПИ) для микроЭВМ с каналом МПИ легко выполнить на двух БИС серии К1801: К1801ВП1-033 в режиме контроллера байтового параллельного интерфейса (КБПИ) и К1801ВП1-034 в режиме устройства передачи информации (УПИ) [1]. С помощью такого УБПИ выводятся данные с микроЭВМ «Электроника МС 1201» на печатающее устройство [2]. Двухлетняя эксплуатация наших восьми комплексов ДВК1М и ДВК2М подтверждает надежную и безошибочную работу таких УБПИ.

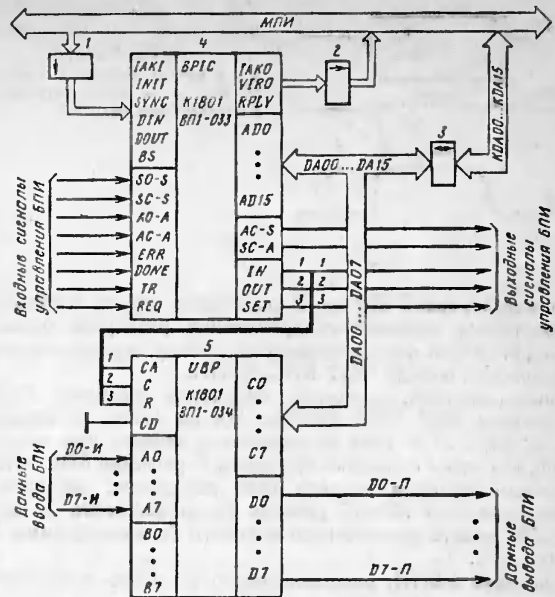


Рис. 1. Упрощенная схема модуля, содержащего УБПИ: 1, 2, 3 — каналные приемники; передатчики, препередатчики соответственно; 4 — БИС КБПИ; 5 — БИС УПИ

Однако при разработке периферийных модулей на основе УБПИ пришлось столкнуться с довольно неприятным «сюрпризом»: модули безошибочно работали на адаптер-удлинителе, а в составе кроссплаты ДВК через несколько часов работы начинали давать сбои при выводе информации. Справедливости ради необходимо отметить, что не все модули вели себя подобным образом: были среди них и такие, которые работали без сбоев неограниченное время.

Для определения причины неисправности в память ДВК записывалась следующая простая программа:

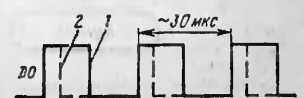
```
START: MOV#377, @#RD WYW
MOV#0, @#RD WYW
BR START
```

Программа поочередно выводит в регистр данных вывода УБПИ нулевой байт, а также байт, в котором все биты равны единице. Затем цикл повторяется. Пуск программы с адреса, определенного меткой START, осуществляется переводом клавиши ПРОГРАММА — ПУЛЬТ ДВК в положение ПУЛЬТ.

Упрощенная схема модуля, содержащего УБПИ, приведена на рис. 1. При буферировании выводимых данных (а именно такой режим УБПИ нам требовался) вход CD БИС УПИ необходимо заземлить. Модуль на адаптере-удлинителе подключался к кроссплате ДВК, программа запускалась на выполнение. На выводах D0...D7 БИС УПИ наблюдались импульсы прямоугольной формы со скважностью, близкой к двум (рис. 2). Спустя некоторое время (не более 30 мин), длительность положительных импульсов на некоторых из этих выводов заметно уменьшилась. На выходе OUT БИС КБПИ отрицательный импульс за это же время удлинился на 20...40 нс, и при «прогретых» БИС его длительность составила 900 нс (рис. 3). Длительность импульсов адреса-данных DA00...DA07 на соответствующих входах БИС УПИ и БИС КБПИ не изменялась как при «холодных», так и при «прогретых» БИС,

Рис. 2. Осциллограммы импульсов на выходе D0 БИС УПИ:

1 — в начале работы программы; 2 — через 30 мин



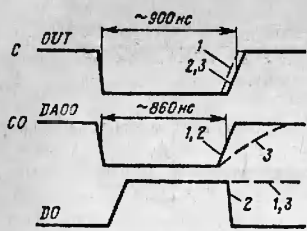


Рис. 3. Осциллограммы импульсов на выходах БИС УПИ:

1 — в начале работы; 2 — через 30 мин; 3 — после установки интегрирующих емкостей

оставалась равной 860 нс. В последнем случае использовался режим внешней синхронизации развертки осциллографа, запуск осуществлялся по началу отрицательного импульса на выходе OUT БИС КБПИ.

Таким образом, в момент окончания импульса OUT на С-входе БИС УПИ импульс DA на одном из входов С0...С7 БИС УПИ (или на нескольких входах) уже закончился, поэтому в выходном буферном D-регистре БИС УПИ в соответствующем разряде (или нескольких разрядах) устанавливается низкий уровень (осциллограмма 2, сигнал D0) вместо положенного высокого (осциллограмма 1, сигнал D0).

Проверка УБПИ, расположенного на плате микроЭВМ «Электроника МС 1201», показала, что длительность импульса OUT не превышала 800 нс для любой платы микроЭВМ при сколь угодно долгом времени работы программы, а длительность импульсов адреса-данных DA была такой же, как и ранее, т. е. 860 нс. Секрет безошибочной работы этого УБПИ состоит в том, что на платах микроЭВМ установлены более быстродействующие

БИС КБПИ, чем те, которыми завод-изготовитель одаривает потребителей, руководствуясь, по-видимому, пресловутым «остаточным принципом».

Потребовалась доработка печатных плат модулей: к выходам С0...С7 БИС УПИ подключены восемь интегрирующих конденсаторов емкостью 180...240 пФ, вторые обкладки которых заземлены. Импульсы DA00...DA07 удлиняются на 40...50 нс, и информация заносится в выходной D-регистр БИС УПИ без искажений и сбоев (осциллограммы 3 на рис. 3). Конденсаторы указанных емкостей не оказывают влияния на ввод данных от УБПИ в ЭВМ.

Модули, содержащие УБПИ и доработанные указанным образом, успешно функционируют в составе микропроцессорной системы обработки телемеханической информации, которая является основой автоматизированной системы контроля и управления движением городского электротранспорта и эксплуатируется на троллейбусных маршрутах г. Донецка.

340086, Донецк, ул. Донецкая, 39, ПО «Донецкгорэлектротранс», отдел АСУ; тел. 93-52-31.

ЛИТЕРАТУРА

1. Глушкова Г. Г. Рекомендации по применению микросхем серии К1801 // Микропроцессорные средства и системы.— 1989.— № 2.— С. 91—94.
2. 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.

Статья поступила 10.02.88

УДК 681.3.06

Ю. Э. Ионан

УПРАВЛЕНИЕ РАБОТОЙ ПРОГРАММНЫХ МОДУЛЕЙ

Управляющие программные средства для объединения различных модулей в единую систему строятся в основном на языке ассемблера, который довольно сложен в освоении программистом-непрофессионалом.

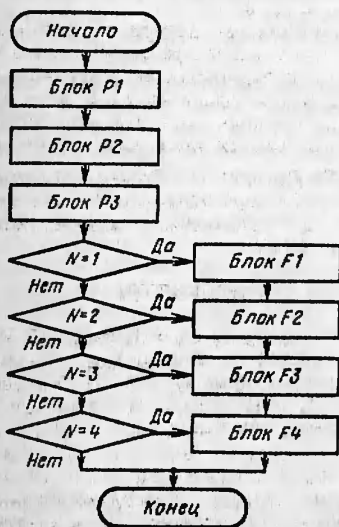


Рис. 1. Блок-схема меню

Система меню, организованная с помощью языка Паскаль и директив косвенного командного файла (ККФ), — один из путей решения задачи создания управляющих программ без привлечения языка ассемблера. Выбор языка Паскаль связан с тем, что он в отличие от языка Фортран имеет команды управления курсором дисплея, особенно эффективные при разработке наглядных диалоговых режимов работы. Основу меню составляют Паскаль-программы и ККФ. Паскаль-программы организуют формирование файла исходных данных в диалоговом режиме работы с использованием команд управления курсором, а ККФ обеспечивает подключение соответствующих программных модулей (рис. 1).

Блок P1 создает файл выбранной ра-

боты, в соответствии с которым подключится программный модуль с помощью директивы ККФ. В данном случае предполагается обработка Фортран-программой файла исходных данных, сформированного в блоке P2. Файлы выбранной работы и исходных данных организуются на внешних носителях. Блок P3 введен для просмотра данных. Язык программирования для этих блоков — Паскаль (рис. 2).

Программы блоков с F1 по F4, написанные на языке Фортран, обрабатывают файл исходных данных (рис. 3). Причем в блоке F1 эти данные обрабатываются как целые числа, в блоке F2 — как действительные, в блоке F3 — как целые и действительные, а в блоке F4 — как символы. Межблочные связи организованы с помощью директив

```

VAR F:TEXT; NAME:PACKED ARRAY[1..8] OF CHAR; N:CHAR; I:INTEGER;
BEGIN
  NAME:='MENU.OAT';
  REWRITE(F,NAME);
  WRITE(CHR(27),CHR(69));
  WRITELN(' ');
  WRITELN('      ЕСТЬ 4 ВИДА РАБОТ С ФОРТРАН-ПРОГРАММОЙ:');
  WRITELN(' ');
  WRITELN('  1. ОБРАБОТКА ДАННЫХ В ФОРМАТЕ ЦЕЛЫХ ЧИСЕЛ:');
  WRITELN('  2. ОБРАБОТКА ДАННЫХ В ФОРМАТЕ ДЕЙСТВИТЕЛЬНЫХ ЧИСЕЛ:');
  WRITELN('  3. ОБРАБОТКА ДАННЫХ В ФОРМАТЕ СМЕШАННЫХ ЧИСЕЛ:');
  WRITELN('  4. ОБРАБОТКА ДАННЫХ В ФОРМАТЕ ЛИТЕРНЫХ СИМВОЛОВ:');
  FOR I:=1 TO 4 DO WRITELN(' ');
  WRITE('      СООБЩИТЕ НОМЕР ВЫБРАННОЙ РАБОТЫ -');
  READLN(N);
  WRITELN(F,N);
  CLOSE(F);
END.

```

Рис. 2, а

```

CONST SIZE=3;
TYPE
  IND=1..SIZE;
  KAR=RECORD
    FIO:PACKED ARRAY[IND] OF CHAR;
    DIO:PACKED ARRAY[IND] OF CHAR;
    MIO:PACKED ARRAY[IND] OF CHAR;
  END;
VAR DM:KAR; NAME:PACKED ARRAY[1..9] OF CHAR;
    FZ:TEXT;
BEGIN
  WRITE('УКАЖИТЕ ИМЯ ФАЙЛА ДЛЯ ПЕРЕДАЧИ ИНФОРМАЦИИ');
  WRITE(' В ФОРТРАН-ПРОГРАММУ -');
  READLN(NAME);
  WRITE(CHR(27),CHR(69));
  WRITE(CHR(27),CHR(89),CHR(35),CHR(44));
  WRITE('ВВОДИМ ПЕРВОЕ ПСЕВДОЧИСЛО:');
  READLN(DM.FIO);
  WRITE(CHR(27),CHR(89),CHR(40),CHR(55));
  WRITE('ВВОДИМ ВТОРОЕ ПСЕВДОЧИСЛО:');
  READLN(DM.DIO);
  WRITE(CHR(27),CHR(89),CHR(50),CHR(44));
  WRITE('ВВОДИМ ТРЕТЬЕ ПСЕВДОЧИСЛО:');
  READLN(DM.MIO);
  WRITE(CHR(27),CHR(69));
  REWRITE(FZ,NAME);
  WRITELN(FZ,DM.FIO);
  WRITELN(FZ,DM.DIO);
  WRITELN(FZ,DM.MIO);
  CLOSE(FZ);
END.

```

Рис. 2, б

```

CONST SIZE=3;
TYPE
  IND=1..SIZE;
  K=RECORD
    F:PACKED ARRAY[IND] OF CHAR;
    D:PACKED ARRAY[IND] OF CHAR;
    M:PACKED ARRAY[IND] OF CHAR;
  END;
VAR FC:TEXT; D:K;
    NAME:PACKED ARRAY[1..9] OF CHAR;
BEGIN
  WRITE('УКАЖИТЕ ФАЙЛ ДЛЯ РАСПЕЧАТ
        КИ');
  WRITE(' ДАННЫХ ИЗ ПАСКАЛЬ-ПРОГРАМ
        МЫ >');
  READLN(NAME);
  RESET(FC,NAME);
  READLN(FC,D.F);
  READLN(FC,D.D);
  READLN(FC,D.M);
  WRITELN(D.F,D.D,D.M);
  CLOSE(FC);
END.

```

Рис. 2, в

Рис. 2. Программы блоков P1 (а), P2 (б), P3 (в)

```

LOGICAL *1 FIL(15)
WRITE(5,500)
READ(5,501,ERR=100) NFD,FIL
CALL ASSIGN(1,FIL,NFD)
500  FORMAT('ИМЯЗОВИТЕ ИМЯ ФАЙЛА
* ДАННЫХ,СОЗДАННОГО В ПАСКАЛЬ
* ПРОГРАММЕ:');
501  FORMAT(Q,15A1)
    READ(1,*)I,J,K
    D=I+J/K
    WRITE(5,*)I,J,K,D
    CALL CLOSE(1)
100  STOP
    END

```

Рис. 3, а

Рис. 3. Программы блоков F1 (а), F2 (б), F3 (в), F4 (г)

```

LOGICAL *1 FIL(15)
WRITE(5,500)
READ(5,501,ERR=100) NFD,FIL
CALL ASSIGN(1,FIL,NFD)
500  FORMAT('ИМЯЗОВИТЕ ИМЯ ФАЙЛА
* ДАННЫХ,СОЗДАННОГО В ПАСКАЛЬ
* ПРОГРАММЕ:');
501  FORMAT(Q,15A1)
    READ(1,4)C,A,R
    D='GG'
    WRITE(5,5)C,A,R,D
    4  FORMAT(A4)
    5  FORMAT(1X,A4)
    CALL CLOSE(1)
100  STOP
    END

```

Рис. 3, б

```

LOGICAL *1 FIL(15)
WRITE(5,500)
READ(5,501,ERR=100) NFD,FIL
CALL ASSIGN(1,FIL,NFD)
500  FORMAT('ИМЯЗОВИТЕ ИМЯ ФАЙЛА
* ДАННЫХ,СОЗДАННОГО В ПАСКАЛЬ
* ПРОГРАММЕ:');
501  FORMAT(Q,15A1)
    READ(1,*)C,A,I
    D=C+A/I
    WRITE(5,*)C,A,I,D
    CALL CLOSE(1)
100  STOP
    END

```

Рис. 3, в

```

LOGICAL *1 FIL(15)
WRITE(5,500)
READ(5,501,ERR=100) NFD,FIL
CALL ASSIGN(1,FIL,NFD)
500  FORMAT('ИМЯЗОВИТЕ ИМЯ ФАЙЛА
* ДАННЫХ,СОЗДАННОГО В ПАСКАЛЬ
* ПРОГРАММЕ:');
501  FORMAT(Q,15A1)
    READ(1,*)C,A,S
    D=C+A/S
    WRITE(5,*)C,A,S,D
    CALL CLOSE(1)
100  STOP
    END

```

Рис. 3, г

ККФ (1, 2) (рис. 4) Вся система программ отлажена в ОСРВ на УБК

СМ1420.

Особенность передачи данных из Паскаль-программы в Фортран-программу — файл данных в Паскаль-программе должен быть описан как TEXT (см. рис. 2 б). Фортран-программа работает с данными из этого файла в соответствии с их описанием (см. рис. 3). Сформированные файлы в Паскаль-программе можно просмотреть программой редактирования текстов типа TED или EDI. Данные для передачи ККФ должны быть также описаны в Паскаль-программе как TEXT (см. рис. 2, а). Это связано с тем, что процессор ККФ обрабатывает данные из файла как символьные строки (см. рис. 4).

```

.ENABLE SUBSTITUTION
.ENABLE GLOBAL
.ENABLE QUIET
RUN P1
RUN P2
RUN P3

```

```

.DISABLE QUIET
.OPENR MENU,DAT
.READ N
.IF N EQ "1" .GOTO L10
.IF N EQ "2" .GOTO L20
.IF N EQ "3" .GOTO L30
.IF N EQ "4" .GOTO L40
.CLOSE
.L50: .EXIT
.L10: RUN F1
     .GOTO L50
.L20: RUN F2
     .GOTO L50
.L30: RUN F3
     .GOTO L50
.L40: RUN F4
     .GOTO L50

```

Рис. 4. Программа косвенного командного файла

Наличие этой особенности необходимо учитывать при формировании разветвлений в ККФ.

Рассмотренный подход к организации управляющих программ показывает одну из возможностей создания меню на основе директив ККФ и языка высокого уровня типа Паскаль, которые осваиваются значительно быстрее,

241037, Брянск, пр. Станке Димитрова, 3, БрТИ, каф. «Вычислительная техника и прикладная математика»; Тел. 21-29-89

ЛИТЕРАТУРА

1. Операционная система ОС РВ СМ ЭВМ: Справочное издание / Г. А. Егоров, В. Л. Кароль, И. С. Мостов и др. — М.: Финансы и статистика, 1987. — 292 с.
2. Бирюков В. В., Рыбаков А. В., Шакула Ю. П. Введение в систему программирования ОС РВ. — М.: Финансы и статистика, 1986. — 192 с.

Сообщение поступило 20.06.88

С. И. Ситников

УСТРОЙСТВО КОНТРОЛЯ РАБОТЫ АВТОМАТИЗИРОВАННОГО ЭКСПЕРИМЕНТАЛЬНОГО КОМПЛЕКСА

При эксплуатации систем автоматизации научных исследований особое внимание уделяется достоверности получаемой информации. Основные причины отказов — случайные сбои микроЭВМ, не связанные с выходом из строя технических средств. Необходимо определить сам факт сбоя, перезагрузить ОС, автоматически запустить экспериментальную программу заново. МикроЭВМ «Электроника 60» автоматически перезапускается при сбоях питания, поэтому достаточно симитировать сбой питания на ее магистрали.

Модуль «регенератор функционирования программ» для микроЭВМ «Электроника 60», определяющий факт сбоя и вырабатывающий сигнал перезапуска микроЭВМ [1], имеет ряд недостатков, не позволяющих эксплуатировать автоматизированные экспериментальные комплексы без постоянного контроля со стороны обслуживающего персонала. Во-первых, при выходе из строя каких-либо частей автоматизированного экспериментального комплекса модулем постоянно регистрируется факт сбоя и перезапускается микроЭВМ. Во-вторых, если в результате сбоя процессор микроЭВМ перейдет к выполнению микропрограммы связи с пультовым терминалом, то перезапуск микроЭВМ не будет произведен. В-третьих, не предусмотрена возможность отключения модуля (необходимо при разработке и отладке программного обеспечения или обработке новых результатов измерений).

В описываемом устройстве контроля работы автоматизированного экспериментального комплекса на базе микроЭВМ «Электроника 60» нет этих недостатков (см. рисунок). На вход 12-разрядного счетчика подается сигнал с шины К ПРТ Н магистрали микроЭВМ; при его переполнении запускается схема имитатора сбоя питания, которая подает потенциалы низкого уровня на шины К ПИТН В и К ПОСТН В; микроЭВМ перезапускается. Применение

внешнего источника счетных импульсов позволяет легко отключать устройство контроля. При обращении к регистру устройства по адресу 177400_в или появлении потенциала низкого уровня на шине К СБРОС Н содержимое 12-разрядного счетчика сбрасывается (период между обращениями не должен превышать 81, 92 с; при нарушении периодичности выполняется процедура начального запуска микроЭВМ).

Устройство контроля также отслеживает информацию на экране дисплея. При появлении символа «@» (код ASCII 100_в) или «.» (код ASCII 056_в) устанавливается соответствующий триггер и сбрасывается содержимое 12-разрядного счетчика. При этом считается, что если к моменту переполнения 12-разрядного счетчика на выходе триггера «@» Лог. 1, то процессор микроЭВМ выполняет микропрограмму связи с пультовым терминалом, а если установлен триггер «.», то микроЭВМ прервала выполнение текущей программы и передала управление монитору.

Последовательность действий устройства контроля при переполнении 12-разрядного счетчика:

схема сравнения анализирует причины происшедшего и предыдущего сбоя по содержимому регистров текущего контроля, коду причины предыдущего сбоя и состоянию триггеров «@» и «.»

Счетчик числа сбоя очищается при несовпадении причин сбоя или сбоя по неконтролируемой причине, в противном случае его содержимое увеличивается на единицу.

Содержимое регистра текущего контроля и триггеров «@» и «.» переписывается в регистр кода причины предыдущего сбоя;

триггеры «@» и «.» сбрасываются и очищается регистр текущего контроля;

запускается имитатор сбоя питания.

Причины сбоя можно разделить на пять групп: сбоя питания; микроЭВМ прервала выполнение текущей программы и передала управление монитору; сбоя при загрузке ОС; переход процессора микроЭВМ к выполнению микропрограммы связи с пультовым терминалом или сбоя на неконтролируемом участке программы; сбоя на контролируемом участке программы (всего может быть до 15 контролируемых участков).

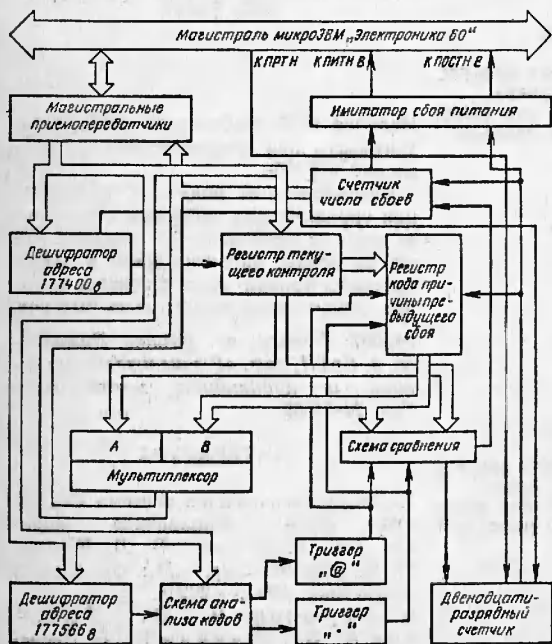
Если девять раз подряд происходят сбои по одной и той же контролируемой причине, блокируется запуск имитатора сбоя питания. Из этого состояния устройство контроля можно вывести, только выключив и вновь включив питание микроЭВМ.

Формат программно-доступного регистра с адресом 177400_в

- 0...3 — номер контролируемого участка программы
- 4 — процессор микроЭВМ перешел к выполнению микропрограммы связи с пультовым терминалом
- 5 — микроЭВМ прервала выполнение текущей программы и передала управление монитору
- 6 — неконтролируемый участок программы
- 7 — сбой питания
- 8 — чтение регистра кода причины предыдущего сбоя
- 9...12 — содержимое счетчика числа сбоя

Если бит 8 равен 0, то биты 0...3, 6 отражают содержимое регистра текущего контроля (биты 4,5 равны 0), в противном случае биты 0...6 отражают содержимое регистра кода причины предыдущего сбоя.

Случаи отказа экспериментальной аппаратуры можно отделить от случайных сбоях в работе микроЭВМ, номеру участка программы (внося в регистр текущего контроля номер контролируемого участка, сбрасывая бит 6 этого регистра в начале контролируемого участка и устанавливая его в конце), на которых проверяются условия (готовность внешнего устройства, наличие LAM-запросов от модулей КАМАК). При случайных сбоях устройство инициирует перезагрузку операционной системы в микроЭВМ, восстанавливая работоспособность автоматизированного комплекса. Достаточно большое время (81, 92 с) между сбросами счетчика позволяет предварительно обрабатывать



Структурная схема устройства контроля работы автоматизированного экспериментального комплекса

накопленную информацию и записывать ее на внешний носитель.

Простой из-за сбоев микроЭВМ при работе с устройством контроля в составе автоматизированного экспериментального комплекса одноплечового времяпролетного спектрометра осколков деления [2] не превышают 10 мин в сутки при круглосуточной работе спектрометра без наблюдения за его работой со стороны экспериментаторов. При эксплуатации комплекса без устройства контроля в сутки терялось 6...8 ч из-за сбоев в ночное время.

Устройство контроля работы автоматизированного экспериментального комплекса на базе микроЭВМ «Электроника 60» выполнено на микросхемах серий К155, К531, К559.

115409, Москва, Каширское ш., 31, МИФИ; тел. 324-28-35

ЛИТЕРАТУРА

1. Корнин Е. Г. Повышение отказоустойчивости систем автоматизации на основе микроЭВМ «Электроника 60» // Микропроцессорные средства и системы. — 1986. — № 3. С. 40—42
2. Пятков Ю. В., Слюсаренко А. И., Чураков А. К. и др. Система накопления и обработки данных автоматизированного времяпролетного спектрометра незамедленных продуктов деления. — Препринт. — Москва, 1986. (МИФИ 009—86).

Статья поступила 14.10.87

УДК 681.32

Ю. Ю. Сорокин, С. П. Максимак, В. В. Субач

ПОСЛЕДОВАТЕЛЬНЫЙ ИНТЕРФЕЙС ЧЕТЫРЕХ ВНЕШНИХ УСТРОЙСТВ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Предлагаемое устройство (У) с четырьмя независимыми каналами стандартного ИРПС выполнено на полулпате Е2 и содержит 25 интегральных схем и 16 оптронов. В нем использованы универсальные синхронно-асинхронные приемопередатчики (УСАПП) КР580ВВ51; управление преры-

ваниями осуществляется программируемым контроллером прерываний (ПКП) КР580ВН59 [1, 2] (см. рисунок). Программируемый делитель частоты допускает использование доступных кварцевых резонаторов на ряд частот $f = 39321/n \pm 5\% \text{ [кГц]}$, где $n = 5...64$.

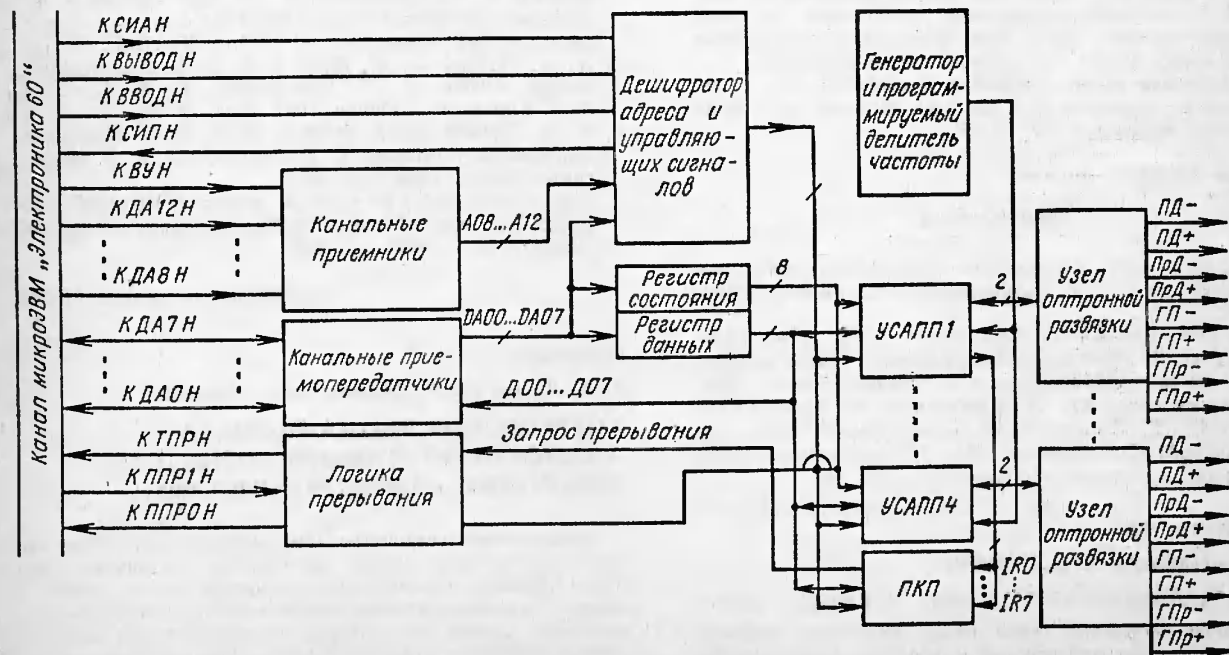
В адресном пространстве ЭВМ У предствлено всего двумя последовательными адресами регистров состояния (РС) и данных (РД). Значения этих адресов выбираются из диапазона 177000...177776 при установке перемычек на плату. Адрес единственного вектора прерывания платы в пределах 0...376 аппаратно программируется пользователем независимо от выбранных адресов РС и РД.

По каждому из четырех каналов У могут быть независимо выбраны скорости для асинхронного обмена из стандартного ряда 19,2; 9,6... 0,15 кБод и до 38 кБод — для синхронного. Остальные характеристики обмена по каналу устанавливаются программно при начальной инициализации У и при работе программы могут быть изменены.

Управление обработкой запросов на прерывание обеспечивает обслуживание каналов с фиксированными приоритетами, с циклическим понижением приоритета последнего обслуженного канала, с маскированием отдельных приемников и передатчиков. Выбор одного из перечисленных режимов производится программно.

Драйвер У идентифицирует источник прерывания (приемник или передатчик канала, имеющий максимальный текущий приоритет) по номеру после выполнения трех команд MOVВ с РС или РД в качестве одного из операндов. Завершение обработки текущего прерывания требует еще 5 байт ориентированных команд пересылки. В любое время двумя командами может быть запрещено прерывание, сброшены группы каналов или все У.

«Накладные расходы» в восемь команд на обработку одного прерывания незначительны при побайтном обмене с низкоскоростными устройствами и при организации межмашинной связи с эмуляцией терминальной линии одной из ЭВМ. Понизить затраты на непосредственное обслуживание У при высокоскоростном обмене пакетами данных можно, например, работая по прерываниям лишь с заголовком пакета (1...4 байт). Остальная часть пакета (несколько сотен байтов) обрабатывается с программным опросом готовности приемника или передатчика канала (стоимость



Структурная схема 4-х канального УПО

управления У снижается в среднем до 2...3 команд на байт).

С помощью программной установки готовности линии к обмену (управление сигналом ГПр) процессорное время при работе с У используется наиболее рационально. Изменения этого сигнала сопровождаются прерываниями, которые могут быть применены для программного исключения попытки обмена данными с каналом, не готовым к этому обмену.

Устройство предоставляет пользователю ряд дополнительных сервисных возможностей. Опрос битовых флагов готовности приемников или передатчиков возможен в специальном режиме «чтение РС опроса» ПКП. Кроме того, программой может быть прочитан байт состояния каждого УСАПП [2]. Это позволяет наряду с анализом готовности приемника и передатчика данного канала определить наличие ошибок в работе канала и их тип. Диагностируются ошибки по паритету (если контроль на четность/нечетность был заказан ранее) и «потери» стоповых битов при приеме, переполнение внутреннего буфера приемника УСАПП размером в 1 байт, разрывы токовых петель приемника или передатчика. Указанные возможности диагностики ошибок облегчают их коррекцию на уровне протоколов обмена.

Большое число программно-управляемых режимов работы в сочетании с высокой скоростью обмена (до 56 КБод) и развитой диагностикой ошибок обеспечивают эффективное использование У при объединении микроЭВМ в ЛС. Разработанное ПО позволяет выбором из исходного пакета на MACRO 11 получать набор сегевых драйверов и утилит, учитывающих конкретную конфигурацию аппаратных средств и требования пользователей. Оно ориентировано на работу У в составе микроЭВМ «Электроника 60» или ДВК под управлением ОС, совместимой с RT-11 (версии 4 и 5). Поддерживаются протоколы блок-пакетной передачи файлов с элементами маршрутизации пакетов.

При включении в сеть мини-ЭВМ «Электроника 100/25», «Электроника 79», СМ4 в качестве главной ЭВМ сетевое ПО дополнительно обеспечивает поддержку протокола виртуального терминала многопользовательской ОС разделения времени (типа TSX или NTS ВЦ ИОНХ АН СССР) главной ЭВМ. Разработана программа телезагрузки ОС микроЭВМ по одному из каналов У с внешних носителей главной ЭВМ, размещаемая в ПЗУ и стартующая при запуске микроЭВМ. При этом можно работать на микроЭВМ без собственных дисковых накопителей (снабженной «квасидиском» [3]). Для архивизации результатов работы используются удаленные внешние носители.

В настоящее время завершена разработка ПО, позволяющего использовать У в качестве средства мультитерминальной поддержки ОС RT-11.

Телефон 246-06-80, Москва

ЛИТЕРАТУРА

1. Аналоговые и цифровые интегральные микросхемы / Под ред. С. В. Якубовского — М.: Радио и связь, 1985.
2. Алексенко А. Г., Галицын А. А., Иванникова Л. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.
3. Сорокин Ю. Ю., Лаврентьев В. В., Максимиак С. П., Субач В. В. «Электронный диск» для микроЭВМ «Электроника 60М» // Микропроцессорные средства и системы.— 1986.— № 5.— С. 92.

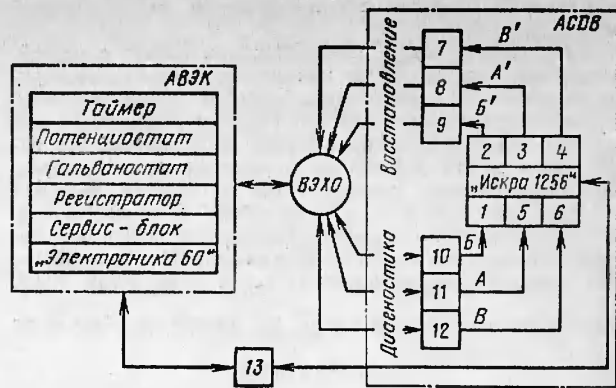
Статья поступила 27.01.87

УДК 681.3—541.135.3

С. Л. Гольдштейн, А. Б. Корчагин

АВТОМАТИЗИРОВАННЫЙ СТЕНД ДВ-1

Автоматизированный стенд (АС) ДВ-1 (см. рисунок) предназначен для диагностики и восстановления работоспособности высокотемпературных электрохимических объектов (ВЭХО), работающих в составе замкнутых систем



Структура автоматизированного стенда диагностики и восстановления ДВ-1:

1 — АЦП (БИФ 015-11); 2 — ЦАП (БИФ 015-10); 3..6 — БИФ 015-55; 7 — система загрузки исходного материала; 8 — устройство добавления и отбора электролита; 9 — источник тока либо вакуумный насос; 10 — импульсный источник тока; 11 — датчик уровня; 12 — контрольно-измерительные устройства; 13 — интерфейс

автоматического регулирования типа потенциостат-объект [1].

АС ДВ-1 основан на новых способах [2, 3] и функционирует совместно с автоматизированным комплексом АВЭК [4], удостоенным Большой Золотой медали и Диплома международного ярмарки в Лейпциге (ГДР).

Канал А диагностирует уровень электролита в капсуле электрода сравнения газового типа (А' — его восстановление); Б — сопротивление диафрагмы ВЭХО (Б' — восстановление сопротивления); В — состав жидких фаз (В' — восстановление рабочего состава).

Телефон 44-86-06, Свердловск

ЛИТЕРАТУРА

1. Гольдштейн М. Л., Гольдштейн С. Л., Распопин С. П. Потенциостат для идентификации высокотемпературных электрохимических систем // Приборы и системы управления.— 1978.— № 11. С. 28—29.
2. А. с. 1178801 СССР, МКИ С 25 С7/2. Электрохимическая ячейка. С. Л. Гольдштейн, Г. Б. Смирнов, А. Б. Корчагин.— Оpubл. 1985. Бюл. № 34.
3. А. с. 1376032 СССР, МКИ С 02 № 27/48. Потенциостатическая установка. С. Л. Гольдштейн, А. Б. Корчагин.— Оpubл. 1988. Бюл. № 7.
4. Высокотемпературный электрохимический комплекс: Тез. докл. 6-й Всесоюзной конференции по электрохимии.— М.— 1982.— С. 285

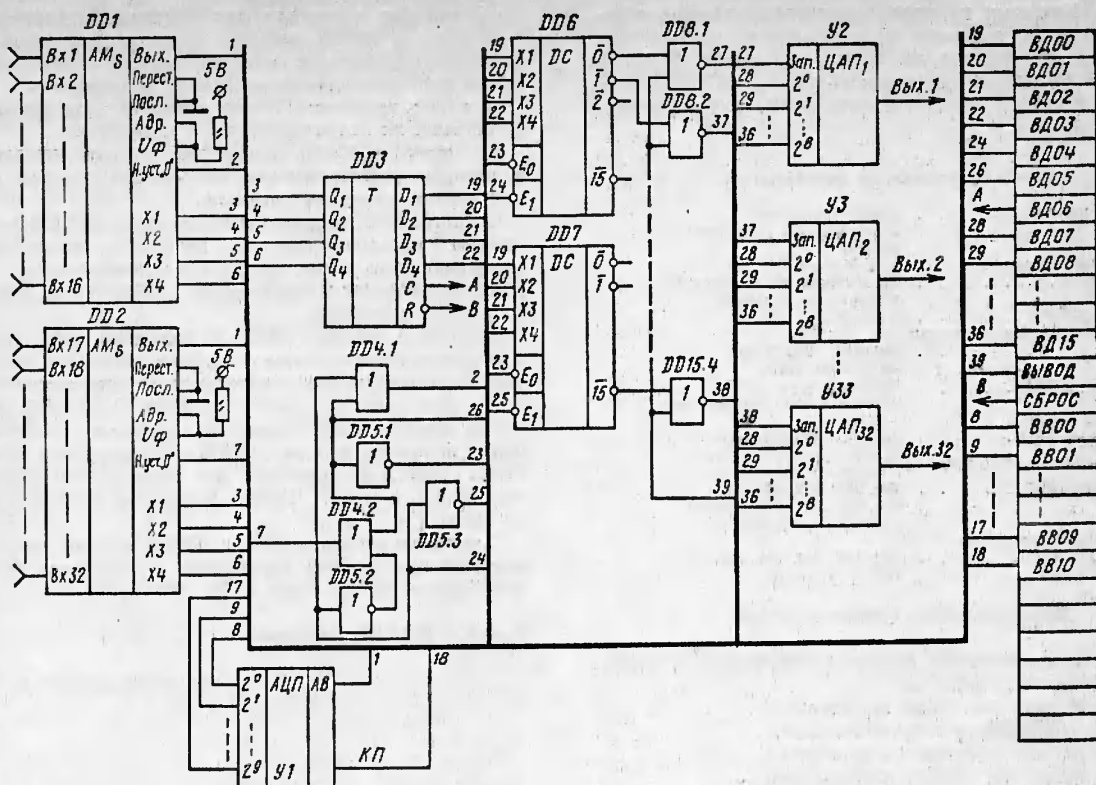
Сообщение поступило 1.02.88

УДК 681.327.2

А. А. Лапин, Ю. В. Литовка, А. М. Шаникеев

УСТРОЙСТВО ВВОДА-ВЫВОДА АНАЛОГОВОЙ ИНФОРМАЦИИ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Предлагаемое устройство (см. рисунок) выполнено на базе перспективных узлов: коммутатора аналоговых сигналов К543КН2, аналогоцифрового преобразователя Ф4243, цифро-аналогового преобразователя Ф4241. Оно предназначено для систем управления технологическими процессами, в частности, опытный образец устройства эксплуатируется в системе управления роботизированным технологическим комплексом гальванопокрытий. Для подключения



Принципиальная схема устройства ввода-вывода:

DD1, DD2 — К543КН2; DD3 — К155ТМ8; DD4 — К155ЛЛ1; DD5 — К155ЛН1; DD6, DD7 — К155ИД3; DD8, DD15 — К155ЛАЗ; Y1 — Ф4243; Y2...Y33 — Ф4241; КП — конец преобразования

внешних устройств используется устройство параллельного обмена И2 15КС-180-032 [1].

Аналоговая информация вводится в следующей последовательности. Адрес входного канала устанавливается на шинах ВД00...ВД03 порта вывода платы И2, зашелкивается триггером DD3 и поступает на адресные шины X1...X4 коммутаторов аналоговых сигналов DD1, DD2. Работа коммутаторов разрешается подачей Лог. 0 на вход *Начальная установка* (при подаче Лог. 1 на этот вход коммутатор переходит в высокоимпедансное состояние по каналу аналоговый вход-выход). Для работы коммутаторов в адресном режиме на входы *Адресный опрос* подается Лог. 1, на вход *Последовательный опрос* — Лог. 0. Кроме того, для работы коммутатора в 16-канальном варианте, при котором канал, указанный параллельным кодом на шинах X1...X4, подключается к выходу, 3, подается Лог. 0 на вход *Перестройка*.

Аналоговый сигнал со входа, номер которого пришел по адресным шинам ВД00...ВД03, через коммутатор поступает на АЦП Y1 [2], где преобразуется в цифровой код, запоминаемый в выходном регистре АЦП. Завершение преобразования фиксируется Лог. 1 на выходе КП АЦП. Командой ввода через шины ВВ00...ВВ10 платы И2 информация из выходного регистра АЦП вводится в процессор.

При выводе аналоговой информации адрес канала устанавливается на шинах ВД00...ВД03 и поступает на вход дешифраторов адреса DD6, DD7, вырабатывающих сигнал *Запуск* на один из 32 ЦАП Y2...Y33 [3]. На цифровые входы всех ЦАП поступает информация с шины ВД07...ВД15 порта вывода платы И2, при этом работает только тот ЦАП, на который пришел сигнал *Запуск*. Импульс *Запуск* формируется объединением сигналов с выхода дешифратора и сигнала *Вывод* платы И2.

Выбор одного из двух аналоговых коммутаторов или одного из двух дешифраторов осуществляется сигналом

ВД04; выбор режима вывода (выбор канала ввода или канала вывода) — сигналом ВД05 и простой логикой DD5, DD5.

Телефон 2-26-01, Тамбов

ЛИТЕРАТУРА

1. Устройство параллельного обмена И2 15КС-180-032: Техническое описание и инструкция по эксплуатации 3.858.383, 1980.
2. Преобразователь аналого-цифровой Ф4223: Техническое описание и инструкция по эксплуатации 2П4.949.152.
3. Преобразователь цифро-аналоговый Ф4241: Техническое описание и инструкция по эксплуатации 2П4.949.146.

Сообщение поступило 26.07.88

УДК 681.327

В. П. Уфимцев, В. А. Убогов, В. Г. Линник, А. Н. Буданов ОДНОПЛАТНЫЙ КОНТРОЛЛЕР ВОСЬМИ ПОСЛЕДОВАТЕЛЬНЫХ ИНТЕРФЕЙСОВ

Одноплатный контроллер обеспечивает обмен данными между общей шиной мини-ЭВМ типа СМ4 и восемью внешними устройствами, имеющими стандартный последовательный интерфейс «стык С2». Конструктивно контроллер выполнен на одной плате размерами 250×280 мм, вставляется в свободное место блока расширения системы и дает не более одной условной единицы нагрузки на общую шину мини-ЭВМ.

Контроллер имеет 32 программно-доступных регистра, занимающих непрерывную область памяти в странице вво-

да-вывода. Каждому из каналов передачи данных соответствуют четыре регистра и два вектора прерывания. Для преобразования данных из параллельного формата в последовательный и обратно используется БИС КР581ВА1, являющаяся универсальным асинхронным приемопередатчиком.

Основные функциональные характеристики

Способ передачи данных	Последовательный, асинхронный в дуплексном режиме
Число разрядов в знаке	5, 6, 7 или 8
Контроль данных	По четности, по нечетности и отсутствию контроля
Число битов STOP	1 или 2
Скорость приема-передачи данных, бит/с	300, 600, 1200, 2400, 4800, 9600, 19200
Взаимодействие с ЭВМ	Программное и (или) с использованием прерывания
Базовый адрес регистров контроллера	776400, 776500, 776600, 776700 (устанавливается с помощью перемычек)
вектора прерывания	100, 200, 300, 400, 500, 600, 700 (устанавливается с помощью перемычек)
Сигнальные линии интерфейса «Стык С2»	100, 102, 103, 104, 105, 106, NOT BUSY

Программно-доступные регистры

Каждый канал передачи данных имеет четыре программно-доступных регистра:

- RCSP — регистр состояния приемника;
- RBUF — буферный регистр приемника;
- CSR — регистр состояния передатчика;
- XBUF — буферный регистр передатчика.

Разработаны варианты контроллера для ЭВМ СМ4, СМ1420 и СМ1600.

454080, Челябинск, пр. Ленина, 76, ЧПИ, каф. АУ; тел. 39-90-47, 39-93-42, 42-04-34

Сообщение поступило 23.05.88

УДК 681.3.06

О. А. Яворский, Н. Ф. Чердиченко

ПАКЕТ ПРОГРАММ ОБРАБОТКИ ТЕКСТОВ ДЛЯ ПЭВМ «ЭЛЕКТРОНИКА МС0585»

Для автоматизации подготовки текстовых документов, в том числе и сложной структуры, в учреждениях при использовании ПЭВМ «Электроника МС0585» разработан пакет программ ТЕКСТ-85.

Пакет программ ТЕКСТ-85 предоставляет пользователю следующие возможности:

- подготовку и редактирование схемы (структуры документа), состоящей из элементов, каждый из которых соответствует некоторым фрагментам документа. При этом структура документа может иметь иерархический характер, т. е. представлять собой дерево фрагментов;

- редактирование текстовых фрагментов, в том числе операции вставки, замены, удаления некоторых частей текста от символа до фрагмента в целом, поиска и замены заданных частей текста, оформления шрифтов, задания правых и левых границ текста, отступов, центрирования и выравнивания частей текста, переноса слов по правилам русского языка;

- объектный подход к операциям работы с текстом, при котором выделяется объект текста (символ, слово, строка, предложение, абзац, кадр, фрагмент в целом), а затем с ним выполняются операции копирования, перемещения, форматирования и редактирования;

оформление документа для печати с заданием длины и ширины страниц, размера межстрочных промежутков, размеров верхнего и нижнего полей и отступов, указания числа выводимых на печать экземпляров.

В пакет включены также средства подготовки форм (бланков), их редактирования и заполнения.

Во время работы создается каталог пользователя, в котором регистрируются все его файлы (на жестком или гибком магнитных дисках).

Средства пакета ориентированы на взаимодействие с конечным пользователем и включают многоуровневые «ниспадающие» меню, функциональные клавиши, выводы с диагностических и поясняющих сообщений на любом этапе работы с пакетом.

Наличие в пакете ТЕКСТ-85 большого набора функций и объектно-ориентированного «дружественного» интерфейса дает возможность пользователю достаточно быстро освоить его и применить при автоматизации подготовки документов в учреждениях совместно с другими пакетами общего назначения для ПЭВМ «Электроника МС0585». Пакет ТЕКСТ-85 работает под управлением ИНМОС-85 версии 1.2 и выше. Тексты программ пакета написаны на языке Си.

С третьего квартала 1988 г. ТЕКСТ-85 поставляется ведомственным фондом алгоритмов и программ (ВСФАП) при Бердянском филиале ИПИ АН СССР.

Телефон 3-63-83, Бердлинск

Сообщение поступило 26.08.88

РЕКЛАМА

АДАПТЕР СОПРЯЖЕНИЯ ИНТЕРФЕЙСОВ

Разработан комплект адаптеров для сопряжения между собой в любых комбинациях следующих интерфейсов:

- последовательный интерфейс С2 (V24, RS-232C),
- последовательный интерфейс ИРПС (IFSS),
- параллельный интерфейс ИРПП (IFSP),
- параллельный интерфейс ИРПП-М (CENTRONICS).

Адаптеры могут быть использованы для подключения к ЭВМ и персональным компьютерам принтеров и других периферийных устройств с несовместимыми интерфейсами, для сопряжения ЭВМ и персональных компьютеров между собой (например, персональных компьютеров Искра 1030, ЕС 1840, IBM PC XT/AT, ЭВМ серии ЕС и СМ), а также для сопряжения персональных компьютеров с аналого-цифровыми преобразователями (АЦП) при аналоговом вводе информации в персональный компьютер.

Адаптеры и АЦП с интерфейсом С2 (RS-232C) имеют средние габариты 100×100×20 мм и автономное питание.

Телефоны для справок: 216-22-01, 216-02-57, Москва (Кузьминов А. Ю., Мацея А. В.)

УДК 34.002.6

Захаров В. В. Авторские права разработчиков программных средств // Микропроцессорные средства и системы.— 1990.— № 2.— С. 6.

Охрана авторских прав на программные средства вычислительной техники (ПС ВТ) требует совершенствования действующего законодательства, учета особенностей новых объектов права и совокупности отношений, возникающих в связи с их созданием и использованием. Предлагаемый вариант правового режима ПС ВТ позволяет сделать правовое регулирование в данной сфере адекватным объективно складывающимся хозяйственным отношениям.

УДК 621.3.049.771.14

Белоус А. И., Ковалевский В. К., Кособрюхов В. А., Пархомчук А. А., Чернуха Б. Н. Универсальный временной преобразователь на основе БИС K512PS11 // Микропроцессорные средства и системы.— 1990.— № 2.— С. 15.

Показаны возможности использования однокристалльного преобразователя частота-код K512PS11 в качестве универсального временного преобразователя.

УДК 681.323

Петух А. М., Ободник Д. Т., Верховой В. П. Генератор векторов на основе матричной БИС // Микропроцессорные средства и системы.— 1990.— № 2.— С. 17.

Рассмотрен генератор векторов для устройств отображения графической информации, выполненный на одном кристалле матричной БИС.

УДК 681.325.5:681.386

Злотник Е. М., Анищенко В. В., Бокуть В. В. Микропрограммируемый графический контроллер // Микропроцессорные средства и системы.— 1990.— № 2.— С. 19.

Рассмотрен графический контроллер на основе микропрограммируемого микропроцессорного комплекта БИС серии KP1804. Подробно описаны процессы взаимодействия контроллера с видеопамятью и видеопроцессором и обмена данными с центральным процессором KP1810BM86 через системную шину EC1840/1841.

УДК 681.3.06

Рябов А. Н., Ануфриев С. В. Подпрограммы с переменным числом параметров на Фортране системы РАФОС // Микропроцессорные средства и системы.— 1990.— № 2.— С. 30.

Рассматривается подход, позволяющий реализовать на Фортране системы РАФОС (ФОДОС) программные единицы с переменным числом формальных параметров.

УДК 681.3.022

Курлянд В. М., Лагута В. В. Графическая система для микроЭВМ ДВКЗ // Микропроцессорные средства и системы.— 1990.— № 2.— С. 40.

Графическая система предназначена для использования в научных расчетах и системах автоматизации научного эксперимента. Ее резидентная часть занимает 1,7 Келов памяти и функционирует как часть ОС РАФОС. Система построена по модульному принципу и допускает расширение языка графических команд пользователем. Может быть легко настроена для работы с различными устройствами вывода графической информации.

УДК 681.3.022

Нифонтов Ю. В., Калита В. В., Бобков Г. М., Яфаев А. А. Модули отображения УВК семейства CM1800 // Микропроцессорные средства и системы.— 1990.— № 1.— С. 42.

Рассмотрены серийные модули отображения УВК CM1800, предназначенные для преобразования информации в видеосигнал и построения символьных и графических изображений на экранах черно-белых и цветных модулей индикации. Приведен состав модулей, основные характеристики, особенность сопряжения с УВК.

UDC 34.002.6

Zakharov V. V. The lowly status of software products. // Microprocessor devices and systems.— 1990.— N 2.— P. 6.

The reliable copyright protection of the software products shows the need to improve existing law system, taking into account the peculiarities of these new objects of right and relations, which appear during software development and use. The proposed variants of lowly support of software products enables to create adequate regulations for current industrial relations.

UDC 621.3.049.771.14

Belous A. I., Kovalevsky V. K., Kosobryukhov V. A., Parhomchuk A. A., Chernukha B. N. Universal time converter using K512PS11 LSI. // Microprocessor devices and systems.— 1990.— N 2.— P. 15.

The main features of frequency-to-code converter type K512PS11 as universal time converter are shown.

UDC 681.323

Petuh A. M., Obodnik D. T., Verhovoy V. P. Vector generator using logic array LSI. // Microprocessor devices and systems.— 1990.— N 2.— P. 17.

The single-chip ULA vector generator for graphic displays is described.

UDC 681.325.5:681.386

Zlotnik E. M., Anishenko V. V., Bokut V. V. Microprogrammable graphic controller. // Microprocessor devices and systems.— 1990.— N 2.— P. 19.

The graphic controller Built using bit-slice microprocessor set KP1804 (AMD29xx) is described. The processes of controller interaction with videoprocessor and videoRAM and data I/O transfer with i8086 CPU via EC1840/1841 system bus are explained in detail.

UDC 681.3.06

Ryabov A. N., Anufriev S. V. Subroutines with variable parameter count in FORTRAN. // Microprocessor devices and systems.— 1990.— N 2.— P. 30.

The approach which enables creation of program units with variable parameter count in RT11/RAFOS Fortran-IV is explained.

UDC 681.3.022

Kurlyand V. M., Laguta V. V. Graphic system for DVK3 microcomputer. // Microprocessor devices and systems.— 1990.— N 2.— P. 40.

Graphic system for scientific calculation tasks and experiment automation is proposed. Its resident part occupies 1.7K words memory and works as a component of RAFOS operating system. The system has modular architecture and permits user-defined extensions of graphic command set. It can be easily reconfigured for various graphic output devices.

UDC 681.3.022

Nifontov Yu. V., Kalita V. V., Bobkov G. M., Yafaev A. A. CM1800 video display adapters. // Microprocessor devices and systems.— 1990.— N. 1.— P. 42.

The standard display adapters for CM1800 microcomputer provided for symbolic and graphic display on black-and-white and color monitors are described. The adapter structure, technical specifications and interfacing to CM1800 are explained.

УДК 681.3-181.1.06

Горбунов А. Г., Миненко С. В., Мруга А. Д. Простой отладочный модуль на основе генератора тестовых последовательностей // Микропроцессорные средства и системы.— 1990.— № 2.— С. 49.

Рассмотрен простой комплект отладки с помощью генератора тестовых последовательностей, встроенных в ДВК2М. Генератор тестовых последовательностей работает в режимах генерации входных воздействий и регистрации выходных реакций. Приведены структурные и принципиальная схемы.

УДК 681.327

Арсенин В. П., Охрименко В. Р. Интерфейсные модули для микроЭВМ «Электроника 60» // Микропроцессорные средства и системы.— 1990.— № 2.— С. 71.

Модули ИРПР-16/8 с перестраиваемой разрядностью данных для микроЭВМ типа «Электроника 60», выполненные на одной полуплате, предоставляют широкие возможности для работы различных периферийных устройств с параллельным обменом информацией.

УДК 681.326.34

Джиган В. И., Щербakov С. В. Аналого-цифровой преобразователь с унифицированным интерфейсом МПИ // Микропроцессорные средства и системы.— 1990.— № 1.— С. 85.

Представлена схема сопряжения восьмиканального биполярного АЦП на основе микропроцессорного комплекта КМДП БИС серии К588. Рассматривается возможность динамического питания БИС АЦП.

UDC 681.3-181.1.06

Gorbunov A. G., Minenko S. V., Mruga A. L. A simple debug unit using test sequence generator. // Microprocessor devices and systems.— 1990.— N 2.— P. 49.

A simple microprocessor debug tool with test sequence generator built into DVK-2 microcomputer is described. The test generator outputs logic sequences to the device under test and records output sequences. The block and circuit diagrams of the device are included.

UDC 681.327

Arsenin V. P., Ohrimenko V. R. Parallel port interface cards for "Electronika-60". // Microprocessor devices and systems.— 1990.— N 2.— P. 71.

The IRPR-16/8 multi-purpose parallel port cards with variable word size on half-size board for "Electronika 60" Q-bus simplify connection of multiple devices with parallel interface to computer.

UDC 681.326.34

Dzhigan V. I., Sherbakov S. V. ADC with a Q-bus interface. // Microprocessor devices and systems.— 1990.— N 1.— P. 85.

The circuit diagram of the 8-channel bipolar ADC built with K588 family LSIs and Q-bus interface. The possibility of the pulsed powering of controller is discussed.

Оформление рукописи

1. Объем статей (включая аннотацию, рисунки, выводы, список литературы) не должен превышать:

25 страниц машинописного текста для обзорной статьи;

15 страниц для статей о конкретных аппаратно-программных комплексах, устройствах и системах;

10 страниц для статей концептуального характера.

2. Рукопись представляется в двух экземплярах, напечатанных через два интервала на одной стороне листа стандартного размера.

3. Используемую литературу необходимо давать общим списком в конце статьи в последовательности, соответствующей упоминанию источников в тексте.

4. Рисунки должны быть выполнены тушью с соблюдением ГОСТов.

5. В качестве иллюстраций (предпочтительно на широкой пленке), которые могли бы быть опубликованы на вкладке или обложке журнала.

6. К статье должно быть приложено направление учреждения, в котором выполнена работа, со всеми необходимыми сопроводительными документами (акты, справки и т. д.), а также сведения об авторах, содержащие их адреса и номера телефонов; число авторов статьи должно быть не более четырех.

7. Перед названием статьи следует указывать индекс согласно универсальной десятичной классификации (УДК).

Заместитель главного редактора С. М. Пеленос

Номер подготовили:

Е. И. Бабич, Г. Г. Глушкова, В. М. Ларионова, Е. П. Граве

Корректор Е. М. Кучерявенко

Технический редактор Г. И. Колосова

Адрес редакции журнала:

103051, Москва, Малый Сухареvский пер., д. 9А

Телефоны: 208-73-23, 208-19-94

Сдано в набор 10.01.90. Подписано к печати 24.04.90.

Формат 84×108¹/₁₆ Офсетная печать Т-08932

Усл. печ. л. 10,08 Уч.-изд. л. 14,2. Тираж 87.000.

Заказ 6577 Цена 1 р. 10 к.

Орган Государственного комитета СССР

по вычислительной технике и информатике

Набрано на ордена Трудового Красного Знамени

Чеховском полиграфическом комбинате

Государственного комитета СССР по печати

142300, г. Чехов Московской области

Отпечатано в Московской типографии № 13 ПО «Периодика»

Государственного комитета СССР по печати

107005, г. Москва, Денисовский пер., 30. з. 169

РОБОТРОН

Роботрон — один из крупнейших комбинатов в ГДР. Здесь разрабатываются и изготавливаются изделия вычислительной и канцелярской техники, электронной измерительной и радиорелейной техники, а также бытовой электроники. Деятельность многочисленных специалистов обеспечивает наивысшие результаты работы и внедрение новых продуктов, методов и прикладных решений. Пользователи роботронских изделий в стране и за рубежом стали хорошими партнерами, относящиеся с доверием к качеству и экономичности наших изделий.

Сотрудники фирмы Роботрон знают о проблемах и требованиях пользователей различных областей применения, формах организации предприятий. Поэтому они могут найти и реализовать необходимые для решения задач пути. Все это, — начиная с разработки продуктов, предоставления технических и программных

средств вплоть до передачи, — осуществляется в тесном сотрудничестве с пользователями.

Программа поставок охватывает электронные пишущие машинки, терминалы и комплексные информационно-поисковые системы. Мы предоставляем технические средства и программное обеспечение для экономической работы ЭВМ на рабочем месте, для бюро и персональных ЭВМ, систем автоматизированного проектирования и управления производством (САПР/ГАП) вплоть до управления с помощью ЭВМ сложных технологических процессов.

Особое место занимает разработка и предоставление системного и отраслевого программного обеспечения для пользователей в стране и за рубежом.

Через двух- и многосторонние соглашения с другими социалистическими странами согласовано четкое разделение

труда. При этом обязательно обеспечивается совместимость техники между собой. Основой для этого является Единая система электронной вычислительной техники социалистических стран — ЕС ЭВМ — и система малых ЭВМ — СМ ЭВМ.

За последние годы скачкообразно увеличилось производство 8- и 16-разрядных персональных ЭВМ. Одновременно начато производство 32-разрядных ЭВМ.

Огромные возможности комбината Роботрон при проектировании и реализации «малолюдного» производства характеризуются настройкой технических и программных средств на условия пользователя.

Сюда относятся:

8-разрядная ЭВМ роботрон СМ1904, предназначенная для решения задач планирования производства и производственных мощностей, обработки заказов вплоть до отчетности;

16-разрядные профессиональные ЭВМ роботрон СМ1910 и ЕС1834, предназначенные для решения задач НИИОКР, обеспечения качества и надежности;

32-разрядная мини-ЭВМ роботрон СМ1710 и производительные рабочие станции для проектирования схем и топологии печатных плат, конструирования с помощью двух- и трехмерных изображений и использования в качестве управляющей ЭВМ при автоматизации производственных процессов.



ИАНА

версия 1.1

ИА логовая программная система по **НА**дежности
(Вероятностные расчеты, статистическая оценка и оптимизация
надежности сложных технических систем)

ИАНА предоставляет средства для решения большинства
надежностных задач, возникающих при проектировании, испытаниях и
эксплуатации сложных технических систем:

РАСЧЕТ всех гостированных показателей
надежности при всех принятых в теории надежности
распределениях продолжительности безотказной
работы и восстановления элементов, а также на основе
статистической информации о результатах испытаний
элементов и систем.

ОПТИМИЗАЦИЯ затрат на систему при
выполнении требований по надежности, также и
оптимизация надежности при ограничениях на
затраты.

ОЦЕНКА эффективности сложных систем с
учетом надежности.

СТАТИСТИЧЕСКАЯ обработка резуль-
татов эксплуатации и испытаний, позволяющая по
выборке (в т. ч. и по цензурированной) построить
гистограмму и получить оценку параметров любого
априори заданного распределения.

ИАНА позволяет внедрить современные методы теории
надежности в практику, сократив при этом затраты на анализ
надежности до минимума.

ИАНОЙ могут пользоваться не только специалисты по
надежности, но и инженеры, непосредственно занимающиеся
проектированием и эксплуатацией, студенты и аспиранты технических
вузов.

ИАНА--это удобный и гибкий диалог, графическое представление
результатов, развитая система подсказок и подробная документация.

ИАНА предназначена для использования на ПЭВМ типа РС XT/AT
(и совместимых моделях) с жестким магнитным диском.

Цена программной системы **ИАНА**--3700 р. (для вузов и
академических институтов--3000 р.). Для оснащения дисплейных классов
вузов поставляется учебный комплект на льготных условиях.

Пользователям версии 1.0 версия 1.1 поставляется бесплатно.

Подробная информация высылается бесплатно.

Руководители разработки: Ушаков
И. А., Рогожин В. С., Козлов М. В.

Наш адрес: Москва, 119899,
Ленинские горы, центр сп ДИАЛОГ в
МГУ

Телефон: 939-17-96

Телефакс: 939-38-93

ЦЕНТР
ВДиалог^{сп}