



МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

1 | 1986

ISSN 0233-4844

Однокристалльный 16-разрядный микропроцессор КМ1810ВМ86 аппаратно и программно совместимый «снизу — вверх» с МПК БИС серии К580. Модуль телевизионного адаптера персональной ЭВМ «Ириша»: принципиальная схема, топология печатной платы, описание режимов работы. Однокристалльные 4- и 8-разрядные микроЭВМ серий КР1820, КМ1816: архитектура, технические характеристики, система команд, режимы работы, примеры применения.

Персональная ЭВМ ПК 8001 на базе микропроцессора КР5801К80 с развитыми средствами передачи, обработки отображения алфавитно-цифровой и графической информации в системе автоматизации физического эксперимента.

ПЗУ — богатое разнообразие применений в микропроцессорных и радиоэлектронных устройствах. **Правовое регулирование** общественных отношений в области производства и обмена машинной информации: проблемы, противоречия, предложения.





KR1820BE1 KR1820BE2

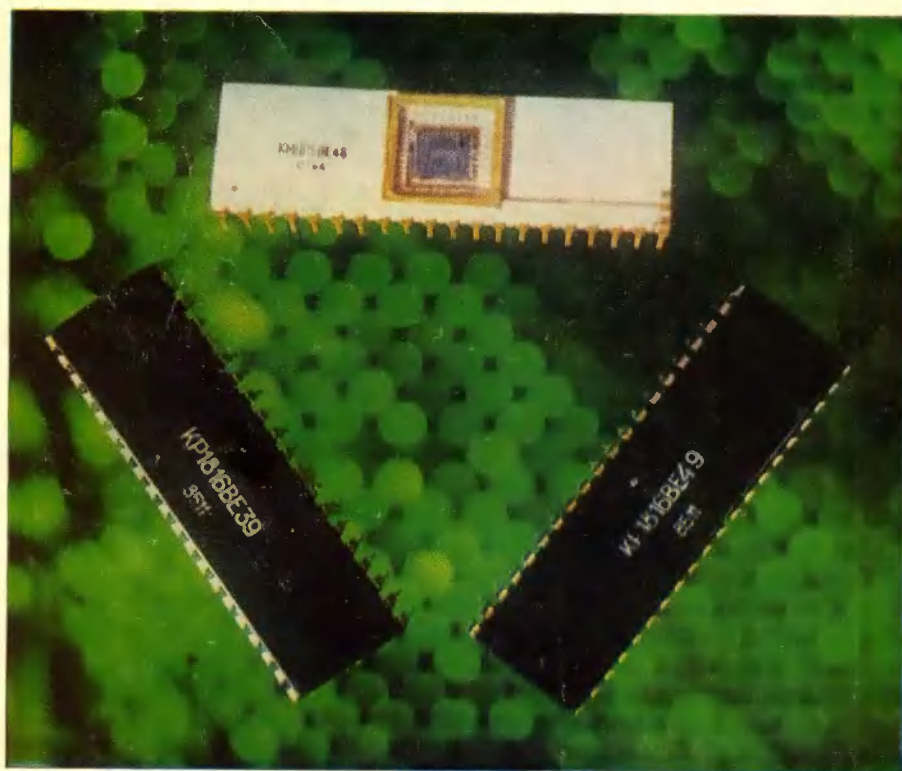
4-разрядные

Функционально - законченные универсальные устройства, содержащие на одном кристалле ОЗУ емкостью 64К слов, ПЗУ емкостью 1К байт (только KR1820BE2), порты ввода—вывода, таймер-счетчик, тактовый генератор, синхронизатор, схему прерываний.

Изготавливаются по *n*-канальной МОП-технологии, обеспечивающей высокое быстродействие и возможность использования одного источника питания.

(См. статью Бобкова В. А., Бурмистрова Ю. Н. и др. на с. 55.)

ОДНОКРИСТАЛЬНЫЕ МИКРОЭВМ



Данные микросхемы имеют идентичную структуру и отличаются быстродействием, типом и объемом внутренней памяти программ, объемом внутреннего ОЗУ (см. статью Кобылинского А. В., Липовецкого Г. П. на с. 10.)

8-разрядные

КМ1816В48

КМ1816В39

КМ1816В49

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО НАУКЕ И ТЕХНИКЕ

Издается с 1984 года



МП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 1 / 1986 МОСКВА

СОДЕРЖАНИЕ	Ершов А. П. — Колонка редактора	2
	Карась И. З. — Вопросы правового обеспечения информатики	3
МИКРОПРОЦЕССОРНАЯ ТЕХНИКА	Кобылинский А. В., Липовецкий Г. П. — Однокристалльные микроЭВМ серии К1816	10
	Бобков В. А., Бурмистров Ю. Н., Кособрюхов В. А., Уткин Ю. В., Чернуха Б. Н. — Однокристалльные 4-разрядные микроЭВМ серии КР1820	19
	Кобылинский А. В., Москалевский А. И., Темченко В. А. — Однокристалльный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86	28
ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ	Велихов Е. П., Персианцев И. Г., Рахимов А. Т., Рой Н. Н., Скурихин А. В., Щербаков О. А. — Персональный компьютер в системе автоматизации физического эксперимента	34
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	Алексенко А. Г., Гапоненко А. В., Иванников А. Д., Курилов И. Д. — Разработка и отладка микропрограммного обеспечения цифровых систем на основе секционированных микропроцессоров	37
	Лебедев Г. В. — Разработка интерактивных программ на основе принципа непосредственного редактирования информации	44
ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ	Мирский Г. Я. — Встроенная микропроцессорная система — основа современного средства измерения	47
	Сасов А. Ю. — Микротомография и цифровая обработка изображений на микроЭВМ «Искра 226»	53
	Гусев В. Н., Купцов В. В., Пантелеев С. В., Роговцев А. А. — Интеллектуальный терминал на базе ДВК-1	59
УЧЕБНЫЙ ЦЕНТР	Романов В. Ю., Барышников В. Н., Воронов М. А., Паначев Ф. И. — Графические возможности персональной ЭВМ «Ириша»	61
	Гойхман П. А., Добровинский М. Е., Удовченко А. Б., Фланчик Б. С. — Устройство приоритетного прерывания для микроЭВМ «Электроника 60»	73
	ПЗУ — РАЗНООБРАЗИЕ ПРИМЕНЕНИИ	
	Лукьянов Д. А. — ПЗУ — универсальный элемент цифровой техники	75
	Щелкунов Н. Н., Дианов А. П. — ПЗУ вместо произвольной логки	83
	СПРАВОЧНАЯ ИНФОРМАЦИЯ	
	Микросхема динамического ОЗУ КР565РУ6	88
	«КРУГЛЫЙ СТОЛ» ЖУРНАЛА	
	Ершов А. П. — Как учить программированию	91
	Рефераты статей	94

Редакционная коллегия:

А. Г. Алексенко
В. В. Бойко
В. М. Брябри
К. А. Валиев
Г. Р. Громов
(ответственный секретарь)
В. И. Иванов
М. Б. Игнатьев
А. В. Каляев
С. С. Лавров
В. В. Липаев
Б. Н. Наумов
(зам. главного редактора)
С. М. Пеленов
(зам. главного редактора)
А. К. Платонов
Ю. А. Чернышев
В. А. Чиганов
И. И. Шагурин

Редакционный совет:

Ю. Е. Антипов
Р. Л. Ашастин
Е. П. Велихов
Н. Н. Говорун
Г. И. Кавалеров
И. И. Малашинин
В. А. Мясников
Ю. Е. Нестерихин
И. В. Прангишвили
Л. Н. Преснухин
В. В. Симаков
В. И. Скурихин
В. Б. Смолов
Ю. М. Соломешев
Н. Н. Шереметьевский

Номер подготовили: Г. Г. Глушкова,
В. М. Ларионова, С. С. Матвеев

Корректор Г. Г. Казакова
Технический редактор
Л. А. Горшкова
Художник А. П. Бачурин

Адрес редакции: 101820,
проезд Серова, 5, редакция журнала
«Микропроцессорные средства
и системы»
Телефоны 228-18-88; 221-99-26

Сдано в набор 2.1.86
Подписано к печати 05.02.86 Т-03947
Формат 84×108¹/₁₆. Бумага № 1.
Высокая печать. Усл. печ. л. 10,08
Уч.-изд. л. 13,72. Тираж 51 000.
Заказ № 440. Цена 1 руб. 10 коп.
Орган Государственного комитета СССР
по науке и технике
Московская типография № 13
ПО «Периодика» ВО «Союзполиграфпром»
Государственного комитета СССР
по делам издательства, полиграфии
и книжной торговли.
107005, Москва, Б-5, Денисовский пер., д. 30

На первой странице обложки — Общий вид персональной ЭВМ ПК 8001

К ЧИТАТЕЛЯМ

Ровно два года назад академик Е. П. Велихов обратился к начальному 5-тысячному контингенту читателей, представляя им первый номер журнала «Микропроцессорные средства и системы».

Сейчас, вступая в 1986-й год, первый год пятилетки ускорения, мы рады приветствовать более чем 50-тысячный коллектив подписчиков журнала и в их лице — всех специалистов народного хозяйства, связанных с разработкой, применением и использованием микропроцессорных средств и систем.

Успешно завершив предыдущую пятилетку, в преддверии XXVII съезда нашей партии, весь советский народ переживает благотворный период политического и нравственного подъема, вызванного усилением борьбы за развитие лучших сторон жизни советского общества: осуществление социальной справедливости, поддержка народной инициативы, гласность, укрепление сознательной трудовой дисциплины, повышение качества труда и жизни, ускорение научно-технического и социального прогресса, усиление демократического начала в решении общественных проблем.

Перед инженерно-техническими силами страны стоит историческая задача: привести в действие огромный социально-экономический потенциал современной науки, на главных направлениях технического развития пройти своим путем от научного достижения до его воплощения в передовой технологии, выйти на высший уровень производительности труда, внести достойный вклад в развитие мировой науки и техники, в международное разделение труда.

В первых рядах тех, кому суждено осуществить эти предначертания, стоит многотысячная армия специалистов по электронике, связи, приборостроению, вычислительной технике и ее применению. На них лежит особая ответственность, поскольку указанные разделы техники сейчас становятся основой почти каждого продвижения к повышению производительности труда и эффективности управления.

Опережающее развитие всей индустрии информатики, включение информационного обеспечения наряду со связью в производственную инфраструктуру провозглашены программными положениями нашей партии. В недавно принятой комплексной программе научно-технического прогресса стран — членов СЭВ электронизация народного хозяйства была названа первой среди пяти наиболее приоритетных направлений.

Редакционные органы и авторский коллектив нашего журнала будут настойчиво стремиться к тому, чтобы «Микропроцессорные средства и системы» стали бы верным и полезным помощником своим читателям в их многотрудной повседневной работе.

Ведущий отдел журнала, посвященный микропроцессорной технике, будет стараться своевременно и полно знакомить читателей с новинками промышленной микропроцессорной техники, с типовыми техническими решениями, опробованными в передовых и успешных разработках.

Заботясь о расширении круга специалистов, повышении их квалификации и желая ускорить распространение положительного опыта, мы будем совершенствовать работу нашего «Учебного центра».

Существенное развитие получит отдел журнала, посвященный персональным ЭВМ, — прежде всего для того, чтобы подготовить специалистов народного хозяйства к эффективному применению этого важнейшего вида микропроцессорной техники, который в предстоящей пятилетке в нарастающих количествах начнет приходить на рабочие места трудящихся.

Мы усилим практическую направленность отдела программного обеспечения, содействуя распространению информации о наиболее удачных и массовых программных разработках, в частности базовых алгоритмов и прикладных программ для персональных ЭВМ.

Опираясь на корреспондентский актив, мы будем продолжать знакомить читателей с наиболее интересными и поучительными применениями микропроцессорных средств, обращая особое внимание на связь технических решений с реальным улучшением характеристик изделия или системы управления.

Укрепляя сложившиеся отделы журнала, мы будем в то же время, опираясь на пожелания читателей, расширять тематику журнала и разнообразить способы подачи материала, используя дополнительные возможности перехода журнала на двухмесячную периодичность. Особое значение мы придаем дальнейшему совершенствованию переписки с читателями и обеспечению всех форм обратной связи.

Мы благодарим ответивших на анкету журнала в 1985 г. и желаем всем читателям больших успехов в их работе на благо нашей отчизны.

А. П. Ершов

ВОПРОСЫ ПРАВОВОГО ОБЕСПЕЧЕНИЯ ИНФОРМАТИКИ

Катализатором современного научно-технического прогресса, как известно, являются микроэлектроника, вычислительная техника и приборостроение, вся индустрия информатики. Они оказывают решающее влияние на эффективность средств труда, технологических систем во всех отраслях... Это действительно новая техника, влекущая за собой революционные перемены в производстве.

М. С. Горбачев.

Коренной вопрос экономической политики партии

1. Постановка проблемы

В проекте «Основных направлений экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года» выдвинута задача «Ускоренно развивать выпуск средств автоматизации инженерного труда, малых электронно-вычислительных машин высокой производительности, персональных электронно-вычислительных машин. Увеличить производство программных средств для вычислительной техники и автоматизированных систем управления»*.

Перестройка права обычно начинается с запаздыванием по отношению к перестройке общественных отношений, которые должны им регулироваться. Если переход к энергетическому производству** занял сотни лет, то переход к освоению информационного производства*** совершается на порядок быстрее [3, с. 3, 4]. Время требует ускорения темпа развития правового обеспечения информатики, так как стихийно формирующиеся отношения могут замедлить и деформировать желательные для общества изменения. В статье сделана попытка проанализировать происходящие и предстоящие в ближайшее время изменения общественных отношений и сформулировать предложения по правовому регулированию и защите некоторых из них.

Введение в общественное производство автоматизированных систем и роботов приводит к замещению

орудием труда функций, которые традиционно включались в понятие «труд». Процесс общественного производства, сложившийся на основе трех элементов: предмет труда, орудия труда и непосредственно труд, претерпевает существенные изменения из-за качественной перестройки орудий труда и самого труда. Вслед за процессом производства изменяются и общественные отношения по присвоению и обмену продуктами производства.

Использование вычислительных систем (ВС) изменяет правовое положение человека в общественном производстве — усложняются возможности оценки последствий поведения человека, так как ВС способны принимать «самостоятельные» решения. ВС совместно с людьми становятся компонентами социальной формы движения [4]. Следовательно, общественные отношения в социальной сфере также будут заметно эволюционировать.

2. Исходные посылки

Под машинной информацией (МИ) будем понимать информацию, зафиксированную в виде, «непосредственно доступном обработке на ЭВМ, включая ее передачу с электронными скоростями без пространственного перемещения носителя» [3, с. 6]. При этом к категории МИ следует отнести информационные фонды в виде баз данных (БД) и программных средств (ПС)*.

* Между ПС и БД нет четкой границы: и то и другое концентрирует определенные знания (например, специфицирующее программирование в основном сводится к созданию наборов данных). Отнесение ПС к продукции производственно-технического назначения, регламентированное постановлением ГКНТ, Госнабза СССР... № 641 от 13 февраля 1984 г., в перспективе должно охватывать и БД как самостоятельный объект поставки.

Понятием ВС будем охватывать организационно и технически обособленную совокупность вычислительных средств, реализующих функции ввода, хранения, обработки и вывода МИ. Действующая правовая регламентация процессов информационного обмена между людьми неприменима в полном объеме к обмену МИ [5]. С развитием ЭВМ процессы создания и обмена информацией в области производства и управления, ранее совпадавшие с деятельностью людей либо носившие вспомогательный характер, обособляются. Именно в обмене «ВС — ВС» из-за количественного роста объемов и скорости обмена информация приобретает свойство товара. Во все возрастающих масштабах МИ создается и отчуждается в рамках вновь формирующихся информационных общественных отношений [6], порождая противоречия в сферах производства, потребления и социального общения.

Противоречия между формой и содержанием управления производством МИ

Рассмотрим противоречия, возникающие в вопросах планирования и учета общественного производства МИ, а также в вопросах организации труда.

Сложившаяся система управления производством вещей* предполагает возможность сопоставления денежной оценки производственной продукции и ее общественной ценности. Такое сопоставление в полном объеме

* Принятый в настоящее время термин материальное производство по мере развития информатики должен включать не только производство вещей, но и производство МИ, поэтому в дальнейшем будем пользоваться термином производство вещей. Производство вещей может осуществляться как при энергетическом, так и при информационном производстве; производство МИ — функция только информационного производства.

* «Правда», 9 ноября, 1985 г.

** Под энергетическим будем понимать производство, основанное на использовании небологических источников энергии (электрической, тепловой, ядерной).

*** Определим информационное производство как основанное на использовании вычислительных систем для обработки информации и управления.

ме неприменимо к производству МИ, так как производство вещей не ориентировано на использование МИ в качестве средства производства и предмета труда: не учитываются тиражируемость и отсутствие физического износа МИ. Это приводит к планированию и оценке деятельности в области производства МИ по затратам (загрузка ЭВМ, заработной плате на выполнение работ и т. п.), а не по информационной ценности*. Блок МИ в отличие от вещей может тиражироваться, поэтому использование бухгалтерского баланса и оперативного учета, ориентированных на свойства вещественных объектов, не обеспечивает отражения всех процессов движения и использования МИ.

Действующая система общественного производства базируется на посылках об экономической эффективности крупного машинного производства. Эта же посылка первоначально была справедлива и для производства МИ, так как только крупные ВЦ могли быть оснащены мощными ВС. Однако развитие терминальных сетей, организация дистанционного доступа к базам данных позволяют приблизить средства производства к участникам процесса, порождают новые элементы в организации общественного труда и сужают диапазон применения посылки о целесообразности территориальной и организационной концентрации труда в области производства МИ.

Учет МИ как важнейшего компонента средств производства, особенно для предприятий непосредственно производящих МИ, а также затраты на ее создание по-разному отражаются (либо вообще не получают отражения) в системе учета деятельности предприятия. Это приводит к невозможности экономически сопоставить деятельность предприятий в рамках существующей системы управления [7], замедляет «...разделение труда, ступень которого зависит от развития производительной силы, достигнутой в данный момент» [1], что, в свою очередь, тормозит рост производительности труда в производстве МИ.

Противоречия между формой и содержанием потребления МИ

Машинная информация обладает двойственным характером: как продукт материального производства она может непосредственно ис-

пользоваться в производстве, как продукт интеллектуальной деятельности — тиражироваться с ничтожно малыми затратами по отношению к стоимости первоначального создания. Сложившаяся правовая система проводит четкую грань между порядком присвоения продуктов интеллектуального и производственного труда. Продукты интеллектуального труда составляют интеллектуальную собственность, охраняются авторским или изобретательским правом и распространяются посредством печатных источников. Их присвоение реализуется через интеллект человека, оплата осуществляется вне распределительных отношений (например, авторский гонорар за статью не связан с процессом применения изложенных идей в производстве). Продукты производственной деятельности присваиваются посредством перехода права собственности (для государственных предприятий — права оперативного управления)* за установленную цену, которая определяется как «денежное название овеществленного в нем (в товаре.— И. К.) количества общественного труда» [2, с. 117]. Но производство МИ не сводится к овеществлению труда! Труд материализуется не в вещной, а в информационной форме.

Сохранение формы присвоения МИ, характерной для продукции интеллектуальной деятельности, вступает в противоречие с содержанием присвоения МИ как продукции производственного назначения. Так как МИ может присваиваться без оплаты, соответствующей ее ценности, то слабо стимулируется доведение МИ до уровня высококачественной продукции. Кроме того, возможно индивидуальное присвоение блоков МИ (особенно ПС) путем копирования с последующей поставкой под видом новой разработки для извлечения нетрудовых доходов. Последняя тенденция может получить существенное развитие по мере расширения сферы применения ПЭВМ.

Противоречия между формой правового регулирования и содержанием общественных отношений

Включение ВС, наделенных интеллектом, в процессы межличностного общения изменяет критерии оценки возможностей субъективного восприятия людьми причинно-следст-

венных связей между событием и его результатом. Традиционный взгляд на ВС как на бездумное орудие исполнения воли человека порождает противоречие между априорно предполагаемой обязанностью и фактической возможностью человека сознавать и предвидеть последствия своих деяний при взаимодействии с ВС. Недооценка «интеллектуальных» особенностей ВС может привести к ошибкам в анализе субъективной стороны деяний из-за проблемы выявления вины человека и ослабить эффективность действующих правоохранительных норм. Возрастание роли ВС в социальной среде в сочетании с неразработанностью ГОСТов, регламентирующих их «поведение» без изменения действующих норм, эквивалентно исключению части людей из сферы правового регулирования. По мере проникновения ВС в различные области деятельности указанное положение может привести к трудно предсказуемым последствиям.

3. Регулирование процессов производства МИ

Цель производства МИ состоит в создании продукта, обладающего потребительной стоимостью. В свою очередь, потребительная стоимость связана с объемом МИ. При одинаковых свойствах МИ может занимать различный объем, поэтому для оценки потребительной стоимости необходимо использовать понятие, характеризующее общественно необходимый объем МИ для обеспечения заданных свойств.

Соотношение фактического объема и информационной ценности позволяет судить об эффективности программы, плотности заполнения базы данных и т. п. Другими словами, информационная ценность играет ту же роль, что и цена вещественных объектов. Одновременно введем понятие информационного баланса как средства учета наличия и движения МИ в процессе производства. По аналогии с традиционным информационный баланс должен отражать поступление МИ на предприятия и процесс ее обработки, учитывать возможность создания копий, изменения объема и информационной ценности в этом процессе. Далее будем предполагать, что информационная ценность и информационный баланс используются в процессе обмена и производства МИ (подробное рассмотрение указанных вопросов, в том числе плана счетов, правил корреспонденции и т. п., является предметом самостоятельного исследования).

Выделим два класса действий с МИ: создание новой МИ и распространение уже полученной МИ без изменений. В первом классе можно

* Под информационной ценностью будем понимать средний общественно необходимый объем машинной памяти для выполнения заданной функции (например, объем базы данных, длина программы в символах и т. п.).

* В дальнейшем изложении отличие права оперативного управления от права собственности, заключающееся в ограничении возможности государственного предприятия распоряжаться закрепленным за ним имуществом, будет предполагаться без специальных оговорок.

выделить производство МИ человеком с помощью ВС и создание МИ в процессе работы ВС, во-втором — тиражирование МИ и передачу ее по каналам связи.

Создание новой МИ с помощью ВС как орудия труда (пошаговое программирование, подготовка массива научно-технической информации и т. п.) приводит к появлению новых информационных ценностей. Этот процесс по характеру труда близок к научной работе, так как в нем отсутствует предмет труда. МИ как продукт трудовой деятельности создается путем отображения идеальных объектов (мыслей, в том числе и зафиксированных ранее, но предварительно прошедших через сознание человека). Вместе с тем процесс создания новой МИ имеет сходство и с производством вещей, так как МИ обладает потребительскими свойствами и ее создание невозможно без использования орудия труда — ВС*.

Создание новой МИ в процессе работы ВС связано с использованием МИ в качестве предмета труда. Например, формирование сводной отчетности из первичной информации. Цель указанного процесса состоит в переработке и производстве МИ. По характеру он близок к автоматизированному производству вещей — участие в нем человека необязательно или носит вспомогательный характер. На практике при создании МИ оба описанных процесса могут использоваться совместно.

Тиражирование МИ (как процесс физического репродуцирования носителей информации) и передача ее по каналам связи (как обеспечение дистанционного доступа к МИ) зависят от технической оснащенности и наличия сетей ВС. Оба процесса могут характеризоваться объемом воспроизведенной МИ и ее информационной ценностью.

Рассмотрим возможные основания ответственности предприятий по производству МИ. В настоящее время производство новой МИ осуществляется по договорам на НИР и

* Вопрос о том, каким объектом считать МИ — материальным или идеальным, требует самостоятельного изучения. Являясь формой представления информации, МИ может рассматриваться как идеальный объект [8]. Однако МИ существует вне сознания человека, составляет часть орудия труда, может поставляться и использоваться, не проходя через сознание человека. Это позволяет отнести ее к материальным объектам. В статье МИ условно рассматривается как материальный невещественный объект.

ОКР [9]. При этих формах практически исключается гражданско-правовая ответственность предприятий за качество продукции, в том числе из-за неопределенности понятия «качество научной разработки». При необходимости созданная МИ тиражируется. Технические условия на поставляемый блок МИ не включают функциональных свойств продукта — следовательно, предприятие, осуществляющее тиражирование, отвечает только за идентичное воспроизведение первоначального блока. В результате этого риск, связанный с ошибками в МИ, полностью ложится на заказчика. Указанный факт экономически отображается через систему ценообразования, исключающую стоимость труда по созданию первоначального блока МИ из стоимости копии.

Рассмотрим вариант установления ответственности изготовителя МИ, ориентированный на потребности заказчика. МИ является изделием повышенной сложности, поэтому интересы заказчика диктуют организацию поставок ВС «под ключ» либо гарантированное качество работы ВС при включении в нее новых блоков МИ [10]. Возможно также введение риска как основания гражданской ответственности предприятия-изготовителя, если виновник не выявлен либо его нет. При этом ответственность за потери из-за сбоев в работе ВС ложится на рискующую сторону, если она не доказала вину противоположной стороны. Причинами сбоев могут быть несовершенство технических норм, сложность ВС и т. п. Существует и вариант освобождения от ответственности рискующей стороны при действии непреодолимой силы. Предприятие-изготовитель в течение срока, обусловленного договором, должно нести гражданско-правовую ответственность на основании риска за результат функционирования созданного им блока МИ и обязано доказывать, что сбой в работе произошли по вине заказчика.

Фактически, риск как основание гражданско-правовой ответственности сводится к правилу распределения убытков [11], поэтому введение принципа риска требует решения вопроса об образовании государственных резервных и ведомственных страховых фондов. Восстановительная функция этих фондов должна состоять в компенсации части ущерба, вызванного вычислительными авариями (повреждение ЭВМ или МИ), стоимость восстановления потерь от которых может быть весьма значительной. Страховые фонды должны образовываться из взносов предприятий-производителей и предприятий-заказчиков. Тарификация страховых взносов должна зависеть от запрашиваемых гарантий и мер защиты, принятых пользователем.

В рамках предлагаемой структуры производства правовое положение участников процесса выпуска МИ может строиться, например, следующим образом:

предприятие производит МИ единичного потребления и поставляет ее с ответственностью на основании риска;

предприятие создает новую МИ с использованием блоков МИ чужого производства. После сборки и отработки поставка пользователю также осуществляется с ответственностью на основании риска. Данная схема является рекурсивной. В договорах оговаривается степень ответственности и круг факторов риска, возлагаемых на поставщика. Заказчик может потребовать возмещения потерь в случае, если поставщик не может доказать вину заказчика или действие непреодолимой силы;

предприятие по тиражированию приобретает блок МИ в собственность (оперативное управление) как средство производства за полную цену из оборотных средств (в настоящее время такая передача осуществляется практически бесплатно) и включает в цену поставки стоимость блока, деленную на предполагаемый тираж. Предприятие по тиражированию несет имущественную ответственность на основании риска за ошибки в поставляемой МИ по принципу «ответственность за результат» и имеет право регрессивного иска к предприятию-изготовителю первоначального блока МИ. Передача копий регламентируется договором поставки;

предприятие по тиражированию осуществляет только посреднические услуги по распространению МИ в интересах создателя исходного блока МИ. При этом оно, подобно коммиссионеру, наделяется правом тиражирования (распоряжения) блоком МИ, действует по поручению собственника блока МИ (либо планового задания) и получает вознаграждение, соответствующее затратам. Передача копий регламентируется договором поставки (подряда), в особые условия которого входит ответственность на основании риска предприятия-изготовителя за качество МИ. Ответственность изготовителя наступает, если предприятие по тиражированию докажет, что оно не исказило МИ при копировании.

Следует отметить, что хранение страховой копии МИ, являющееся в настоящее время обязанностью фондов алгоритмов и программ, может выделиться в самостоятельную функцию и, следовательно, в договорное отношение, предусматривающее возмещение затрат на хранение и компенсацию заказчику ущерба от потери МИ. Можно предвидеть также развитие договорных отношений по передаче МИ, аренде ВС и т. п.

Установление цены, отражающей

общественно необходимые затраты на создание МИ, соотносении затрат с информационной ценностью и отражение на балансе предприятия, позволят решить вопрос о финансовом управлении соответствующей деятельностью. Возможные расхождения в стоимости приобретения МИ и, следовательно, различная стоимостная оценка фондов предприятия при равной информационной ценности могут компенсироваться рентными (фиксированными) платежами как средством выравнивания финансового положения предприятий, находящихся в различных условиях и получающих вследствие этого различную прибыль [12].

Наряду с проблемами управления общественным производством путем правовой регламентации деятельности предприятий возникают проблемы управления трудом в рамках самих предприятий. Одним из основных условий трудового договора, отличающим его от гражданского-правового договора подряда (ст. 64 ОГЗ)*, является подчинение трудящегося правилам внутреннего распорядка (ст. 8 ОЗоТ)**.

Трудовое законодательство в отдельных случаях предусматривает надомную форму работы [15]. Обычно эта форма применяется для изготовления товаров народного потребления, оказания услуг, изготовления изделий народных художественных промыслов. Учитывая перспективы развития ПЭВМ, можно предположить, что надомная форма создания программных средств и подготовки данных может получить широкое развитие.

Появление ПЭВМ в личной собственности приведет к объединению групп лиц с целью создания блоков МИ (в первую очередь ПС) подобно возникновению авторских коллективов. Проблема объединения групп лиц для достижения общей цели по-разному решается для творческой и производственной деятельности. В первом случае право признает и разрешает такой союз (ст. 99, 116 ОГЗ), во втором — запрещает (ч. 2, ст. 434 ГК РСФСР). Целесообразно узаконить коллективную деятельность по созданию МИ вне предприятий, так как по характеру труда она близка к творческой, хотя результатом может являться законченный комплект средств производства в форме МИ.

Таким образом, в характере труда по производству МИ можно ожидать сближения некоторых видов деятельности с традиционно производственными, а также развития специфических форм индивидуальной и

рассредоточенной организации работ. Предстоит решить вопрос и о материальном вознаграждении индивидуальных и коллективных создателей МИ (особенно ПС) на основе коэффициентов трудового участия с учетом степени вклада в работу, качества МИ и спроса на нее. Одновременно необходимо усилить ответственность работников за конечные результаты деятельности.

4. Регулирование процессов обмена и потребления МИ

Рассмотрим возможные пути защиты права на МИ и методы договорного регулирования товарообмена МИ (отчуждение и присвоение МИ). Сложившиеся правоотношения базируются на четком разделении прав на вещь и на продукт интеллектуальной деятельности. Права в отношении вещей и ценных бумаг устанавливаются в рамках права собственности (разд. II ОГЗ). Права в отношении продуктов интеллектуальной деятельности — в рамках авторского права (разд. IV ОГЗ) и права на открытие, изобретение, рационализаторское предложение, промышленный образец (разд. V, VI ОГЗ). Авторское право защищает преимущественно форму выражения мыслей, изобретательское — в основном существо новых идей. Возможна также специфическая защита новой информации как производственного, экономического или государственного секрета. Такая защита может осуществляться независимо или в сочетании с авторской или изобретательской.

В развитых капиталистических странах складывается система защиты ПС (и баз данных) в рамках авторского и патентного права [16]. Защита ПС в рамках авторского права допускает возможность извлечения из листинга алгоритма и воспроизведения его в форме новой программы. При этом на создателей ПС распространяются положения Женевской конвенции об авторском праве от 6 сентября 1952 г.: автор получает право ограничивать распространение ПС, оговаривать размер вознаграждения за его использование и т.п. (ст. 98 ОГЗ). Но ПС и БД создаются, как правило, коллективным трудом, поэтому речь идет в основном не об авторском праве отдельных лиц, а о праве организаций ограничивать и определять возможности использования созданных ими ПС.

Защита ПС в рамках изобретательского права затрудняет использование одного и того же алгоритма для различных реализаций ПС. Разнообразие возможных реализаций программ по заданному алгоритму значительно превос-

ходит число возможных реализаций устройства по описанию изобретения, поэтому такая форма защиты вряд ли целесообразна. Фактически в обоих вариантах защиты ПС речь идет о закреплении за крупными капиталистическими фирмами в сфере вычислительной техники монопольных прав на использование идей. В СССР в настоящее время МИ не защищена правом от произвольного присвоения, в том числе и безвозмездного [17], что резко ослабляет экономические рычаги управления развитием информатики.

Полезность МИ далеко не однозначно связана с новизной или оригинальностью идей, заложенных в ее создание, поэтому представляется целесообразным отделить творческую новизну МИ от ее конкретных потребительских свойств, выявляющихся и проявляющихся в распределительных отношениях. Творческая новизна МИ может фиксироваться в рамках уже сложившихся форм (например, в виде публикации) либо развита с учетом специфики МИ. МИ может являться объектом охраны права собственности, как и другие виды продукции производственно-технического или народного потребления. Отметим, что интегральные микросхемы, которые можно рассматривать как одну из форм представления ПС, защищаются именно таким образом (топология микросхем относится к категории идей, физическая реализация — к категории вещей).

Рассмотрим применительно к МИ содержание института права собственности (ст. 9 ОГЗ), включающего в пределах, установленных законом, следующие полномочия: владеть (физически обладать), пользоваться (извлекать из вещи полезные свойства) и распоряжаться (определять судьбу вещи). Владение МИ в отличие от владения вещью позволяет скопировать ее и использовать без износа и следов, поэтому основной способ защиты права собственности путем истребования имущества из чужого незаконного владения (ст. 28 ОГЗ) для МИ является неконструктивным.

Защита прав на МИ может осуществляться в рамках обязательств, возникающих вследствие приобретения или сбережения имущества без достаточных оснований (ст. 4 ОГЗ) и порождающих обязанность возместить стоимость имущества и неосновательно полученные доходы (ст. 437 ГК РСФСР). Одновременно может быть поставлен вопрос об уголовно-правовой защите МИ от повторения и незаконного присвоения, для чего необходимо выделить МИ как самостоятельный предмет уголовно-правовой охраны. В настоящее время присвоение МИ путем несанкционированного копирования не может квалифицироваться как

* Здесь и далее — Основы гражданского законодательства [13].

** Здесь и далее — Основы законодательства о труде [14].

хищение, так как в разъяснении Пленума Верховного Суда СССР № 4 от 11 июля 1972 г. с изменениями, внесенными постановлением Пленума Верховного Суда СССР № 13 от 21 сентября 1977 г. и № 6 от 27 ноября 1981 г. [18], подчеркивается, что хищение сопряжено с изъятием ценностей из фондов организации. При неправомерном обращении в собственность МИ может не изыматься из фондов, а копироваться. Это нисколько не влияет на существо процесса незаконного присвоения, однако качественно меняет квалификацию преступления. Наряду с уголовно-правовыми могут применяться и меры административно-правовой защиты МИ.

Так же, как для вещных объектов, пользование МИ протекает по-разному для потребляемой и непотребляемой МИ со следующими особенностями. Потребляемые вещи (горючее, продукты питания и т. п.) утрачивают свою первоначальную форму в процессе однократного использования. Потребляемая МИ теряет свою ценность (например, оперативная сводка, разовая справка и т. п.) в течение короткого отрезка времени. Непотребляемые вещи в процессе использования подвергаются физическому и моральному износу, МИ — только моральному.

Право распоряжаться МИ можно считать аналогичным праву на распоряжение вещью, включая в это понятие и специфическую функцию — копирование и передачу по каналам связи. Так как права собственника ограничены законами и связаны с его правоспособностью (ст. 8 и 12 ОГЗ), то введение нового объекта права — МИ* — требует расширения содержания правоспособности путем введения понятия информационной правоспособности, определяющего возможность иметь МИ в собственности или в оперативном управлении.

Принятая посылка о включении МИ в круг объектов права собственности и вызванная этим необходимость расширения правоспособности лиц требует рассмотрения вопроса о способе реализации информационной правоспособности, т. е. информационной дееспособности, заключающейся в возможности присвоения МИ. Очевидно, что информационная дееспособность реализуется только посредством ВС. Учитывая перспективу развития ПЭВМ, будем считать, что ВС может принадлежать или находиться в пользовании не только организаций, но и граждан.

* Наряду с общественными отношениями к объектам права относят и предметы, по поводу которых возникают общественные отношения [19].

Для удобства построения правовых конструкций, учитывающих возможности ВС, может быть введена юридическая фикция*, наделяющая ВС дееспособностью и признающая их субъектами права. Отметим, что в период первой промышленной революции с помощью фикции «юридическое лицо» была отобрана необходимость экономического и правового обособления предприятий, акционерных обществ и т. п. [20]. Вопрос о правовых последствиях действия ВС и о регулировании «поведения» ВС рассмотрен в п. 5 настоящей статьи.

Охрана прав на МИ в рамках института права собственности требует индивидуализации всех копий конкретного блока МИ. В противном случае одна копия блока МИ не обособляется и не может охраняться. Решение этой проблемы требует паспортизации каждой копии и, следовательно, регламентации порядка тиражирования и порядка учета МИ у всех собственников, владельцев и пользователей МИ. Указанное требование является обязательным для упорядочения процессов присвоения и производства. Оно может решаться в рамках единого для всех участников информационных правоотношений информационного баланса, отражающего получение, обработку и отчуждение МИ. Попытка общегосударственной регламентации учета МИ сделана в Великобритании, где согласно законопроекту о надзоре за использованием информации [21] предусмотрена процедура учета и отчетности, обязательная для всех организаций и лиц, создающих и поддерживающих базы данных. Закон вводит единые журналы регистрации банков данных (ежегодную отчетность фирм по обработке данных перед Палатой Общин), регламентирует порядок исправления данных и внесения новых.

Рассмотрим особенности субъектов права собственности на МИ. Так как одним из компонентов ВС, ис-

* Заведомо неправильное утверждение, условно принимаемое правом для удобства использования правовых норм. Необходимо подчеркнуть, что наделение ВС дееспособностью — юридический прием, опирающийся на то, что «воля» ВС — это набор программных средств, созданных людьми и в определенном смысле моделирующий человеческое мышление. Учет специфики ВС возможен и непосредственно через развитые нормы, регламентирующие правовое положение ВС как объекта права, обладающего особой сложностью. Однако автору представляется, что предлагаемый в статье путь, не влияя на существо проблемы, упрощает построение юридических норм.

пользуемых при создании МИ, являются технические средства, то в настоящее время право собственности (оперативного управления) МИ может принадлежать только социалистическим предприятиям (как государственным, так и кооперативным, общественным и т. п.). По мере распространения ПЭВМ правом собственности на МИ должны быть наделены и отдельные граждане. МИ может поступать в их собственность путем приобретения (например, игровых и учебных программ), а также создаваться как продукт индивидуальной деятельности на собственности ПЭВМ. Принципиально новыми моментами являются возможность создания таким путем МИ, относящейся к средствам производства (например, специализированных ПС), и вытекающая из этого необходимость расширения круга объектов личной собственности (ст. 25 ОГЗ). Кроме того, возможно создание и приобретение МИ группой лиц. Такая МИ должна принадлежать коллективу на правах общей собственности (ст. 26 ОГЗ).

Вариант построения договорных обязательств по поставкам МИ с ответственностью на основании вины может быть дополнен предложенной в п. 3 ответственностью на основании риска. В ряде применений ВС могут рассматриваться как источник повышенной опасности, так как вырабатываемая ими МИ служит для управления предприятиями, средствами передвижения и т. п. Признание этого факта распространяет на собственника (пользователя ВС) внедоговорную ответственность, возникающую вследствие причиненного вреда источником повышенной опасности, но не обязательно по вине владельца источника повышенной опасности (ст. 90 ОГЗ). При этом фактически рассматривается ответственность за риск ошибки в МИ, что смыкается с общим представлением об ответственности за поставки МИ.

Таким образом, в сфере потребления и обмена МИ предлагается: урегулировать общественные отношения в рамках права собственности, защитив его обязательством, связанным с неосновательным приобретением и использованием имущества, а также уголовно и административно-правовыми средствами;

наделить ВС информационной дееспособностью;

расширить сферу применения принципа возложения ответственности на основании риска на договорные отношения с тем, чтобы применить соответствующий подход как к использованию ВС, так и к процессу их производства и поставки.

5. Регулирование социальных отношений

Регулятивные правовые нормы эффективны только в том случае, если они доступны пониманию участников урегулированных правом общественных отношений. Их соблюдение обеспечивается совокупностью политических, философских, эстетических взглядов. Наряду с этим важнейшим инструментом обеспечения стабильности общественных отношений являются правоохранительные институты, предусматривающие неблагоприятные правовые последствия для нарушителей охраняемых отношений. Условием для наступления таких последствий служит вина нарушителя, определяемая как психическое отношение лица к своему противоправному поведению и его последствиям. Вина складывается из интеллектуального и волевого моментов.

В советском праве при наличии вины предусмотрена гражданско-правовая (ч. 2, ст. 88 ОЗГ*; ч. 1, ст. 49 ОЗоТ), уголовная (ст. 3 Основ уголовного законодательства [22]), дисциплинарная (п. 24 Типовых правил внутреннего трудового распорядка для рабочих и служащих предприятий, учреждений, организаций)** и административная (ст. 7 Основ законодательства об административных правонарушениях [24]) ответственность граждан. Во всех случаях выделяется умышленная и неосторожная вина.

Остановимся на проблеме интеллектуального момента вины, так как именно интеллектуальные возможности человека должны рассматриваться при включении ВС в социальную сферу. Интеллектуальный момент умысла предполагает осознание лицом общественной опасности своего поведения (деяния) и предвидения его общественно опасных последствий. Интеллектуальный момент неосторожности — предвидение либо обязанности и возможность предвидения общественной опасности своего поведения.

В настоящее время возможность вероятностной оценки причинно-следственных связей между деянием и наступившими вредными последствиями отрицается, так как приводит к «оценочной» теории вины, согласно которой вина в определенной степени производна от сознания судей и не является в полной мере

* В гражданском праве возможно наступление неблагоприятных последствий без вины, например, в случае причинения вреда источником повышенной опасности (ст. 90 ОЗГ).

** Утверждены постановлением Госкомтруда СССР № 213 от 20 июля 1984 г. [23].

объективным отражением психической деятельности субъекта правонарушения. Пленум Верховного Суда СССР в постановлении № 2 от 18 марта 1963 г. «О строгом соблюдении законом при рассмотрении судами уголовных дел» подчеркнул, что «вредные последствия», независимо от их тяжести, могут быть вменены лицу лишь в том случае, если оно действовало умышленно или допустило по неосторожности» [25]. Указанные требования в сочетании с низким уровнем правовой регламентации информатики, неразработанностью стандартов и специальных правовых норм практически исключают возможность конструкций правовых норм с материальным составом, при котором противоправное деяние связывается с наступлением конкретных материальных последствий. В этом случае выявление интеллектуального момента вины в отношении наступивших последствий потребовало бы анализа образовательного уровня, знания лицом принципов работы и надежности ВС.

Исходя из вышесказанного можно заключить, что для регулирования поведения людей при работе с ВС целесообразно строить правоохранительные нормы с формальным составом. В этом случае на основе (предварительного) анализа возможных последствий поведения участника правоотношения создатель норм предписывает возможные и запрещает вредные способы поведения. Участник правоотношений несет ответственность не за последствия своих действий или бездействия, а за нарушение инструкций (законов). Так, например, должностные инструкции оператору должны включать не требования обеспечить сохранность БД, а перечень операций по ее поддержанию, разработанный технологом. При этом возможна характерная для использования научно-технических достижений ситуация гражданской ответственности предприятия за ущерб на основании риска и отсутствие возможности выявить вину его сотрудников.

Отметим, что право капиталистических стран идет по пути создания формальных конструкций законодательных актов, охраняющих ВС от посягательства. Так, в США согласно Федеральному закону о преступлениях с помощью ЭВМ предусматривается уголовное наказание за умышленную попытку доступа к вычислительной системе в целях обмана или приобретения материальных благ, за умышленный незаконный доступ, видоизменение или попытку видоизменения ВС [26]. Конструкция закона формальная — наказание назначается за действие и за попытку к действию. Принятый в 1983 г. законопроект Великобритании «О предотвращении посягательства на секретность информации, обрабаты-

ваемой с помощью ЭВМ, вследствие ее неправильного использования» [21] также предусматривает уголовно-правовые санкции за нарушение порядка ведения баз данных, незаконное использование МИ и нарушение порядка работы с банком данных, а не за реальные последствия этих действий.

Стабильность общественных отношений требует урегулирования правом «поведения» ВС. При этом возникает проблема создания таких ГОСТов, которые учитывали бы технологические и социальные аспекты поведения (например, решение альтернативы «остановка производства со значительными материальными потерями или выброс вредных веществ в окружающую среду»). По мере развития автоматизированных и автоматических систем, а также организации ВС с недоступными для восприятия человека темпами изменения ситуации указанная проблема будет приобретать все большую актуальность.

Правовое регулирование отношений с использованием ВС предполагает ее информационную дееспособность, рассмотренную в п. 4 данной статьи. Принципиальным моментом для надления ВС дееспособностью является система юридической ответственности за последствия ее «поведения» и юридический состав (совокупность юридических фактов), наделяющий ВС дееспособностью. В противном случае лицам, работающим с ВС, неизбежно будут вменяться в вину последствия деятельности ВС независимо от наличия или отсутствия их вины. Результатом этого будет либо явный и совершенно обоснованный отказ от использования ВС, либо скрытый отказ, проявляющийся в полном ручном дублировании работы ВС. Так, в обзоре судебной практики «Использование в качестве доказательств документов и заключений экспертов, подготовленных средствами электронной вычислительной техники» [27] зафиксировано, что выходные документы ВС (табуляграммы) рассматриваются в качестве письменных доказательств только при наличии подписи уполномоченных лиц. Но это означает, что сама ВС рассматривается лишь как средство автоматизированной подготовки документов, не отличающееся от арифмометра и пишущей машинки.

Может быть предложена следующая схема правоотношений в социальной сфере:

ВС наделяется дееспособностью на основании официальных испытаний (аналогично системе профессиональных экзаменов);

предоставление МИ рассматривается как форма поставки либо услуг с обязательной фиксацией перехода МИ от одного собственника (владельца) к другому;

вина за последствия выявляется на основе процессуальных норм, учитывающих ВС как фиктивный субъект права; невиновность человека презюмируется.

Ответственность за «вину» ВС в течение гарантийного срока несут в дисциплинарном порядке члены комиссии по приемке системы и организация-поставщик, после истечения гарантийного срока — организация-собственник (пользователь) ВС.

Наделение ВС дееспособностью и включение их в социальную сферу требует процессуальной регламентации собирания доказательств при работе с ВС, а также регламентации процессов обыска ВС, выемки и наложения ареста на МИ. Наряду с этим необходимо ввести адекватные приемы экспертизы, в том числе входящие в сферу криминалистической техники. Эти приемы должны позволить выявить и зафиксировать следы обработки данных. Кроме того, необходимо создать соответствующие аппаратные и программные средства вычислительной экспертизы [5]* и обязательной машинной регистрации процессов обработки информации.

6. Заключение

В статье рассмотрены некоторые аспекты общественных отношений, связанные с развитием информатики, на основе предпосылки о способности ВС к обработке информации и выполнению трудовых функций. Выделены противоречия между формой и содержанием общественных отношений в сферах присвоения и производства МИ, а также в сфере социального управления, связанные с использованием ВС. Сформулированы предложения по правовому регулированию и охране информационно-правовых отношений с учетом усложнения причинно-следственных связей во вновь формирующейся социальной структуре и размывания четких границ между способностями человека к обработке информации и соответствующими возможностями ВС.

В статье намечены некоторые возможные пути развития правового обеспечения информатики как вновь формирующейся отрасли [28]**. По

* Соя-Серко Л. А., Корф Н. И. Некоторые вопросы защиты информации от несанкционированного доступа в АСУ, с. 17—19.

** Предполагается, что правовое обеспечение информатики должно включать нормы, сгруппированные на основании единства предмета регулирования, традиционно относящиеся к различным отраслям права: гражданскому, уголовному, финансовому, трудовому, административному, процессуальному.

мере развития вычислительной техники, особенно ПЭВМ и сетей связи, можно ожидать перестройки общественных отношений на базе новых производительных сил. Ряд высказанных в работе предположений носит, очевидно, дискуссионный характер, однако необходимость правового регулирования и охраны новых отношений, порождаемых развитием информатики, не вызывает сомнений.

ЛИТЕРАТУРА

1. Маркс К., Энгельс Ф. Фейербах. Противоположность материалистического и идеалистического воззрений. Избранные произведения.— М.: Политиздат, 1983, т. 1, с. 76.
2. Маркс К., Энгельс Ф. Соп., изд. 2-е, т. 23.
3. Ершов А. П. Комплексное развитие системного программного обеспечения. Препринт 469.— ВЦ СО АН СССР, Новосибирск, 1983,— 38 с.
4. Шалютин С. М. Искусственный интеллект (гносеологический аспект).— М.: Мысль, 1985, с. 187.
5. Всесоюзная научно-практическая конференция «Организационно-правовые вопросы создания документов средствами вычислительной техники и использования их в народном хозяйстве». Тез. докл. ГКНТ, Минюст СССР, ГАУ при Совете Министров СССР.— М.: ВЦ Статуправления, 1984,— 161 с.
6. Венгеров А. Б. Право и информатика в условиях автоматизации управления (теоретические вопросы).— М.: Юрид. лит., 1978, с. 64—66.
7. Карась И. З. Опыт функционирования промышленного предприятия по производству программных средств.— Микропроцессорные средства и системы, 1985, № 1, с. 36—42.
8. Лифшиц М. Об идеальном и реальном.— Вопросы философии, 1984, № 10, с. 120—145.
9. Типовое положение о порядке заключения хозяйственных договоров и выдачи внутриминистерских заказов на проведение научно-исследовательских, опытно-конструкторских и технологических работ и типовой договор. СП СССР, 1968, № 19, с. 133.
10. Sartre F. Les caracteristiques du contral. „cle en main” informatique — Minis et micros, 1985, N 227, p. 37—38.
11. Ойгензихт В. А. Проблема риска в гражданском праве.— Душанбе, Ирфон, 1972, с. 223.
12. Положение о фиксированных пла-

тежах в бюджет.— СП СССР, 1966, № 23, ст. 211.

13. Закон СССР от 8 декабря 1961 г. Ведомости Верховного Совета СССР, 1961, № 50, ст. 526.
14. Закон СССР от 15 июля 1970 г. Ведомости Верховного Совета СССР, 1970, № 29, ст. 265.
15. Бюллетень Госкомиздата СССР, 1982, № 1.
16. Авторские права на средства программного обеспечения ЭВМ.— Электроника, 1985, т. 58, № 5, с. 71—76.
17. Анисимов Г. Н., Бакланов В. Н., Волковицкий К. Е. О правовой охране алгоритмов и программ для ЭВМ.— Вопросы изобретательства, 1976, № 8, с. 9.
18. Бюллетень Верховного Суда СССР, 1972, № 4; 1977, № 6; 1982, № 1.
19. Юридический энциклопедический словарь.— М.: Советская энциклопедия, 1984, с. 208.
20. Черниловский З. М. Презумпции и фикции в истории права.— Советское государственное право, 1984, № 1, с. 98—105.
21. «A Bill to prevent the invasion of privacy through the misuse of computer information.— Advances in Computer system security», 1984, ch. 9.2, legal Requirements for Computer security, p. 477—483.
22. Закон от 25 декабря 1958 г.— Ведомости Верховного Совета СССР, 1959, № 1, ст. 6.
23. Бюллетень Госкомтруда СССР, 1984, № 11.
24. Постановление Верховного Совета СССР от 23 октября 1980 г.— Ведомости Верховного Совета СССР, 1980, № 44, ст. 310.
25. Сборник постановлений Пленума Верховного Суда СССР, 1924—1963.— М.: Изд-во Известий Советов депутатов трудящихся СССР, 1964, с. 171.
26. A Bill to amend title 18, United states code, to make a crime the use, for fraudulent or other illegal purposes of Computer owned or operated by the United states, certain financial institutions and entities affecting interstate commerce.— «Advances in Computer system security», 1984, ch. 9.2, legal Requirements for Computer security, p. 389—349.
27. Бюллетень Верховного Суда СССР, 1982, № 6, с. 20—26.
28. Венгеров А. Б., Барабашева Н. С. Нормативная система и эффективность общественно-го производства.— М.: Изд-во МГУ, 1985, с. 241—242.

Статья поступила 5 сентября 1985 г.

УДК 681.3.06

А. В. Кобылинский, Г. П. Липовецкий

ОДНОКРИСТАЛЬНЫЕ МИКРОЭВМ СЕРИИ К1816

Однокристалльные 8-разрядные микроЭВМ серии К1816—КМ1816ВЕ39, КМ1816ВЕ48, КМ1816ВЕ49 — представляют собой функционально законченные устройства, содержащие на кристалле центральный процессор, ОЗУ данных и ЗУ программ, многоканальный интерфейс ввода-вывода, 8-разрядный таймер-счетчик, векторную схему прерываний, тактовый генератор, устройство синхронизации. Все это обеспечивает универсальность, автономность и гибкость их применения в устройствах самого различного назначения (от локальных систем автоматики до устройств управления бытовыми приборами). Микросхемы серии К1816 имеют идентичную структуру и отличаются быстродействием, типом и объемом внутренней памяти программ, объемом внутреннего ОЗУ (табл. 1).

Таблица 1

Отличительные особенности микросхем серии К1816

Наименование микросхемы	Тип программной памяти	Объем памяти программ (ПЗУ), байт	Объем памяти данных (ОЗУ), байт	Максимальная тактовая частота, МГц
КМ1816ВЕ39	Нет	Нет	128	11,0
КМ1816ВЕ48	1К	1К	64	6,0
КМ1816ВЕ49	ПЗУ	2К	128	11,0

В каждой схеме предусмотрена возможность расширения памяти программ до 4К байт, памяти данных до 384 байт и увеличения числа линий ввода-вывода за счет подключения внешних кристаллов памяти программ (ПЗУ), ОЗУ и интерфейсов ввода-вывода серии КР580.

Условное графическое обозначение микросхемы приведено на рис. 1, назначение выводов — в табл. 2.

Структурная организация

Арифметическо-логическое устройство (рис. 2) — параллельное 8-разрядное устройство, позволяющее выполнять арифметические, логические операции и операции сдвига над данными, представленными в двоичном коде, а также обрабатывать данные, представленные в двоично-десятичном коде.

Блок программной памяти (БПП) состоит из ПЗУ, счетчика команд, дешифратора адреса и дешифратора команд.

Память, расположенная на кристалле, занимает адреса 0000H...03FFH (КМ1816ВЕ48) или 07FFH (КМ1816ВЕ49) (рис. 3). Внешняя память занимает адреса с 0400H (КМ1816ВЕ48) или с 0800H (КМ1816ВЕ49) по 0FFFH. Все поле адресов 0000H...0FFFH разбито

Таблица 2

Назначение выводов

Номер вывода	Обозначение вывода	Назначение вывода
1	T0	Тестируемый вход 0
2	CR1	Вход тактового генератора
3	CR2	Выход тактового генератора
4	$\overline{\text{CLR}}$	Сброс
5	$\overline{\text{SS}}$	Пошаговый режим
6	$\overline{\text{INT}}$	Прерывание
7	$\overline{\text{EMA}}$	Выборка из внешней памяти
8	R	Чтение
9	PME	Считывание из внешней памяти
10	W	Запись
11	ALE	Фиксация адреса
12—19	DB0...DB7	Шина данных (Порт P0)
20	GND	Общий
21—24, 35—38	P20...P27	Порт P2
25	PROG	Строб расширителя ввода-вывода
26	U_{DD}	Напряжение питания при программировании
27...34	P10...P17	Порт P1
39	$\overline{\text{TI}}$	Тестируемый вход 1
40	U_{CC}	Напряжение источника питания +5 В

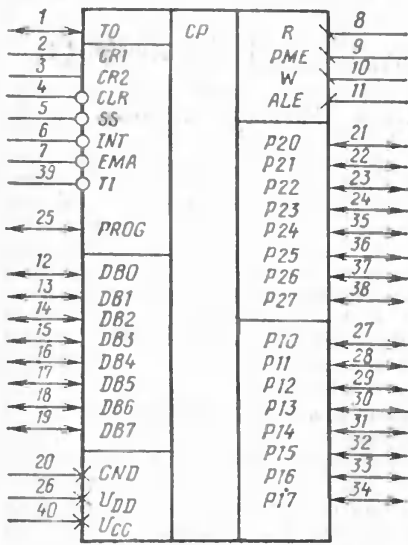


Рис. 1. Условное графическое обозначение микроЭВМ

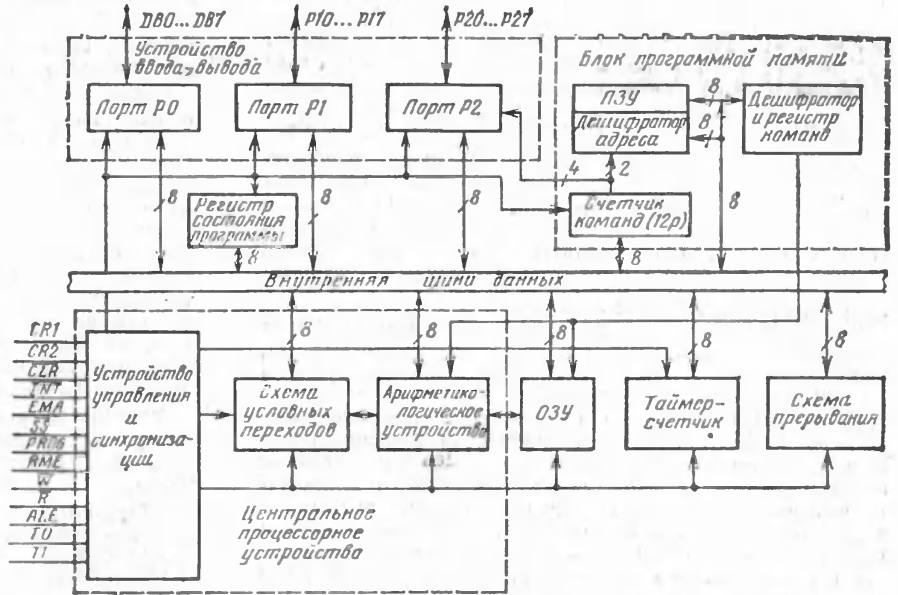


Рис. 2. Структурная схема микроЭВМ

на два банка — банк 0 с адресами 0000H..07FFH и банк 1 с адресами 0800H по 0FFFH. Переключение банков осуществляется программным путем.

12-разрядный счетчик команд (СК) организован таким образом, что его содержимое увеличивается после выбора каждого байта команды и может изменяться при выполнении команд условных, безусловных переходов и при прерываниях. Старший разряд СК изменяется только программно (команды SEL MB0, SEL MB1). СК разбит на две части: счетчики младших (биты 0..7) и старших (биты 8..11) разрядов. При использовании внешней программной памяти биты 0..7 СК поступают через порт P0 (DB0..DB7), а биты 8..11 через порт P2 (P20..P23) (см. рис. 1).

Ячейки ОЗУ (64—KM1816BE48 и 128—KM1816BE49)

разбиты на два банка регистров общего назначения (РОН) с адресами 00H..07H — банк РОН0 и с адресами 08H..0FH — банк РОН1 (рис. 4). Переключение банков осуществляется программным путем с помощью команд SEL RB0, SEL RB1. Восьмиуровневый 16-разрядный стек с адресами 08..17 и ячейки ОЗУ 20..3FH используются только как ОЗУ данных.

Для записи и выборки данных из ОЗУ применяются два вида адресации: прямая и косвенная (регистрация). При прямой адресации в коде команды три младших бита определяют адрес РОН (регистры R0..R7 при выборе банка РОН0, регистры R'0..R'7 при выборе банка РОН1). При косвенной адресации место расположения адреса указывается в команде. Адрес ячейки ОЗУ может храниться либо в аккумуляторе, либо в регистрах



Рис. 3. Поле распределения памяти программ

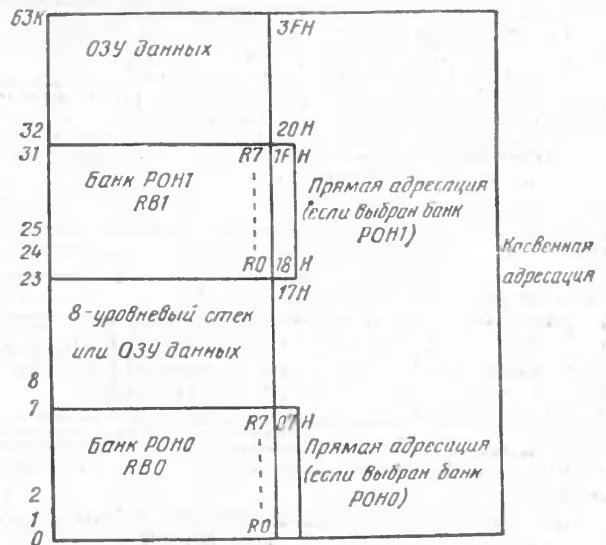


Рис. 4. Поле распределения памяти данных для БИС KM1816BE48

R0, R1 или регистрах R'0, R'1 в зависимости от выбранного банка регистров. С помощью косвенной адресации можно адресоваться к любой ячейке ОЗУ. Программист по своему усмотрению может заносить данные для хранения в любые неиспользованные ячейки-регистры банков P0H, стек, а также имеет доступ к любой из ячеек ОЗУ посредством косвенной адресации.

В микроЭВМ предусмотрена возможность расширения памяти данных путем подключения микросхем ОЗУ емкостью до 256 байт. Общая емкость ОЗУ в этом случае будет равна 324 байт. Обращение к внешней памяти данных осуществляется с помощью команд MOVX @R, A; MOVX A, @R. Стробится информация внешнего ОЗУ сигналами \bar{W} , \bar{R} .

В микроЭВМ имеется 27 линий ввода-вывода, 24 из которых объединены в три 8-разрядных порта (P0, P1, P2). Порт P0 (DB0...DB7) — двунаправленный, порты P1, P2 — квазидвунаправленные.

Порты P1 и P2 имеют идентичные характеристики. Данные, записанные на них, статически фиксируются и не изменяются до перезаписи. Как входы эти линии не фиксируются, т. е. входные данные должны присутствовать до считывания по команде приема данных.

Для использования портов P1, P2 в качестве входов микроЭВМ должна выдать «1» по соответствующим линиям P1, P2 по команде выдачи данных. Порты P1 и P2 устанавливаются в состояние «1» также после подачи сигнала CLR.

Порт P0 — это 8-разрядный двунаправленный порт с тремя состояниями и может использоваться или в качестве статически фиксированного выходного порта, или нефиксированного входного порта. Информация, выдаваемая портом P0 с помощью команд OUTL BUS, A и MOVX @R, A, сопровождается строб-импульсом \bar{W} . При записи информации в порт P0 с помощью команд INS A, BUS и MOVX A, @R вырабатывается строб-импульс \bar{R} .

Кроме операций ввода-вывода информации, предусмотрена возможность выполнения логических операций И, ИЛИ непосредственно на портах P0, P1 и P2 с помощью команд ANL P, #DATA; ORL P, #DATA; ANL BUS, #DATA; ORL BUS, #DATA.

Три линии ввода-вывода T0, T1, INT служат входами, проверяемыми командами условного перехода. Ли-

нию ввода-вывода T0 можно использовать для выдачи тактовых сигналов, а линию T1 — как вход счетчика внешних событий для таймера счетчика. Линию INT — для внешнего аппаратного прерывания.

Число линий ввода-вывода, можно увеличить, если использовать команды MOVD A, P; MOVD P, A; ANL D, P, A; ORLD P, A и соответствующую схему. При этом обмен информацией осуществляется через порт P2 (P20...P23) (рис. 5). Каждая пересылка состоит из двух 4-разрядных полубайтов. Первый полубайт содержит код операции и адрес порта, а второй — четыре бита данных A. Синхронизация осуществляется сигналом PROG. Переход сигнала PROG из состояния высокого уровня в состояние низкого уровня указывает, что на выводах P20...P23 находится код операции и адрес порта, а обратный переход означает, что на этих выводах находятся данные — содержимое четырех младших разрядов аккумулятора.

Устройство управления и синхронизации состоит из генератора, формирователя внутренних тактовых сигналов, формирователей сигналов состояний и режимов работы.

Встроенный генератор — это последовательная резонансная схема, работающая в диапазоне 1...6 МГц для КМ1816BE49 и 1...11 МГц для КМ1816BE39, КМ1816BE49. Выводы CR1, CR2 используются для подключения внешнего кварцевого генератора и LC-цепи (рис. 6), которые обеспечивают частотно-зависимую обратную связь и фазовый сдвиг для генератора. На выходы CR1 и CR2 можно подавать также сигналы от внешнего источника тактовых сигналов.

Формирователь внутренних тактовых сигналов делит частоту встроенного генератора на три и вырабатывает внутренние тактирующие сигналы, выдаваемые на вывод T0 по команде ENT0 CLK. С выхода делителя тактирующие сигналы поступают на вход делителя на пять, выходные сигналы которого определяют пять внутренних состояний микроЭВМ (S1...S5). Сигнал с одного из выходов делителя поступает на вывод ALE. Период

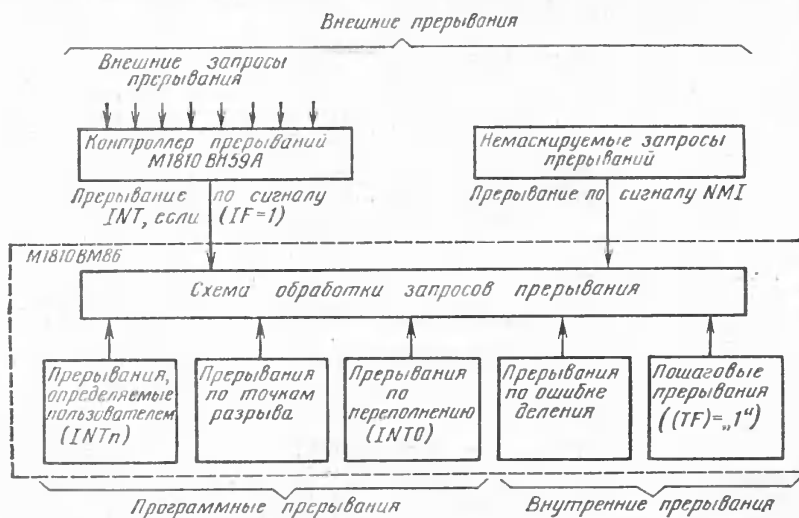


Рис. 5. Временная диаграмма работы с дополнительным портом

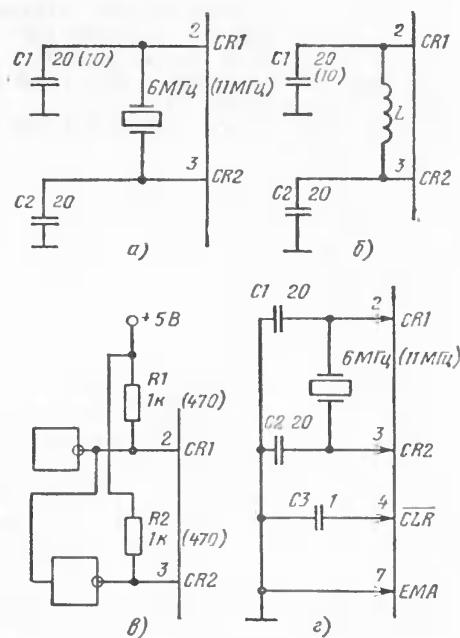


Рис. 6. Схемы подключения кварцевого генератора LC-цепи и внешнего источника тактовых сигналов.

В скобках приведены данные для БИС КМ1816BE39 и КМ1816BE49

следования сигнала ALE соответствует машинному циклу микроЭВМ.

Формирователи сигналов состояний и режимов работы. В качестве входных сигналов в формирователи используются сигналы EMA, PROG, TO, SS, CLR. Их комбинации позволяют реализовать основные режимы работы микроЭВМ.

Схема условных переходов предназначена для формирования сигналов управления вставленным программой при выполнении команд условных переходов. Переход осуществляется по содержимому аккумулятора («0» или не «0») и по состояниям одного из разрядов аккумулятора: триггера флага таймера-счетчика TT, входов T0, T1, INT, триггеров флага F0 и F1, триггера переноса C.

Условия перехода определяются соответствующими командами. Значения признаков C, F0, F1, TT устанавливаются программно, признаков T0, T1, INT — аппаратно.

Таймер-счетчик предназначен для подсчета внешних событий (используется внешний сигнал T1) и генерирования временных интервалов. Состоит из делителя на 32, счетчика, триггера флага.

На вход счетчика поступают импульсы с делителя на 32 частотой $F_{CR}/480$ (в режиме таймера) и с входа T1 (в режиме счетчика внешних событий). Содержимое счетчика устанавливается по команде MOVТ, А, а текущее значение проверяется по команде MOVА, Т. Запуск счетчика в режиме таймера осуществляется командой STRTT, а в режиме счетчика событий — командой STRT CNT. Остановка счетчика производится командой STOPT CNT.

Триггер флага таймера-счетчика устанавливается в состояние «1» при переходе счетчика из состояния FF в состояние 00. Триггер используется в командах условного перехода, а также для организации прерывания по таймеру-счетчику.

Схема прерывания. В микроЭВМ предусмотрено два вектора прерывания: один — аппаратный по входу INT (адрес 03H), другой используется как внутренний и как внешний (вход T1) по флагу таймера-счетчика (адрес 07H). Каждый из векторов прерывания может быть разрешен или запрещен программно командами ENI, DISI, ENI CNTI, DISI CNTI. Кроме того, по сигналу CLR прерывание запрещается до тех пор, пока оно не будет разрешено соответствующими командами (ENI, ENI CNTI).

При поступлении на вход сигнала INT («0» активный), если прерывание по входу INT разрешено, происходит обращение к программе обслуживания с адресом 03H. Как и при любом обращении к подпрограмме, содержимое счетчика команд и слово с информацией о состоянии программы (PSW) записывается в стек. То же происходит и при выработке флага таймера-счетчика. Однако в этом случае обращение производится к программе обслуживания по адресу 07H.

Программа обслуживания должна заканчиваться командой RETR (восстановление состояния и возврат к прерванной программе), так как до выполнения этой команды последующие запросы на прерывание не обслуживаются (прерывание запрещено).

Система прерывания в микроЭВМ — фиксированная по приоритету. Прерывание по входу INT имеет старший приоритет.

Регистр состояния программы (PSW) предназначен для хранения данных о состоянии микроЭВМ. Назначение разрядов PSW: 0..2 — разряды указателя стека (S0..S2); 3 — не используется (при чтении всегда «1»); 4 — разряд, указывающий на используемый банк рабочих регистров общего назначения; 5 — флаг пользователя (F0), используемый по команде условного перехода; 6 — разряд дополнительного переноса (AC), используемый для десятичной коррекции; 7 — перенос,

указывающий на переполнение аккумулятора после предыдущей операции (CY).

Регистр PSW может программно проверяться модифицироваться целиком и поразрядно. При прерываниях по входу INT и по флагу таймера-счетчика содержимое четырех разрядов (D4..D7) заносится в стек, а при возврате из программы прерывания по команде RETR содержимое этих разрядов восстанавливается.

Режимы работы

МикроЭВМ может работать в следующих режимах: проверки программной памяти; работы с внутренней и внешней памятью; пошагового выполнения команд; программирования внутренней памяти программы (для КМ1816ВЕ48). Режим работы устанавливается комбинацией входных и выходных сигналов.

Инициализация (сброс) микросхемы осуществляется сигналом CLR (активный — низкий уровень напряжения). Этот сигнал выполняет следующие функции: устанавливает счетчик команд и указатель стека (в PSW) в «0»; выбирает банк P0H0 (RBO) и банк внутренней программной памяти (MB0); устанавливает порт P0 в высокоимпедансное состояние (при EA=«0»); подготавливает порты P1 и P2 для приема информации; блокирует прерывание по входу INT и по таймеру-счетчику; останавливает таймер-счетчик; устанавливает флаги F0 и F1 в «0»; запрещает выдачу импульсов по выводу T0.

В режиме проверки программной памяти (рис. 7) контролируется правильность хранящейся в памяти информации, записанной при программировании, а также «чистота» памяти после стирания только для ИС КМ1816ВЕ48.

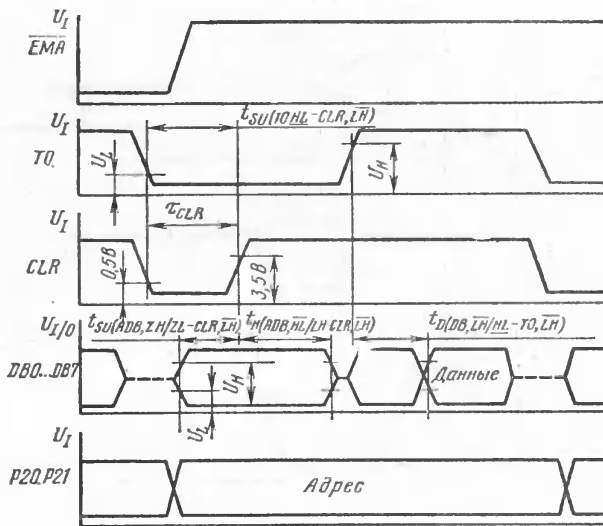


Рис. 7. Временная диаграмма работы микроЭВМ в режиме проверки памяти

Подаваемые на одноименные выходы микросхемы, сигналы выполняют следующие функции:

EMA — при подаче напряжения высокого уровня активизирует режим обращения к внутренней памяти для программирования;

T0 — обеспечивает режимы программирования (низкий уровень напряжения) и контроля (высокий уровень напряжения);

CLR — фиксирует выбранный адрес;

DB0...DB7 — организуют подачу адреса A0...A7 и данных;

P2—0, P2—1 — обеспечивают подачу адреса A8, A9.

При организации режима контроля памяти следует иметь в виду, что по выводам DB0...DB7 подаются адресные сигналы и выдаются данные для контроля. Поэтому при переходе к режиму контроля необходимо обеспечить высокоимпедансное состояние на данных выводах, которое исключает попадание на открытые выходы схемы напряжения адресных сигналов, поступающих от источника.

Режим работы с внутренней памятью устанавливается заданием низкого уровня напряжения на выводе ЕМА. Выполнение программы, хранящейся в памяти, начинается после ее инициализации с команды, расположенной по адресу 00H, так как счетчик команд по сигналу CLR=0 сбрасывается в нуль.

Для увеличения производительности микроЭВМ предусмотрено совмещение выполнения внутренних операций в одном цикле. Например, выполнение выбранной команды и подготовка следующего адреса команды происходят одновременно.

Режим работы микроЭВМ с внешней памятью (рис. 8) применяется при отладке программ и конт-

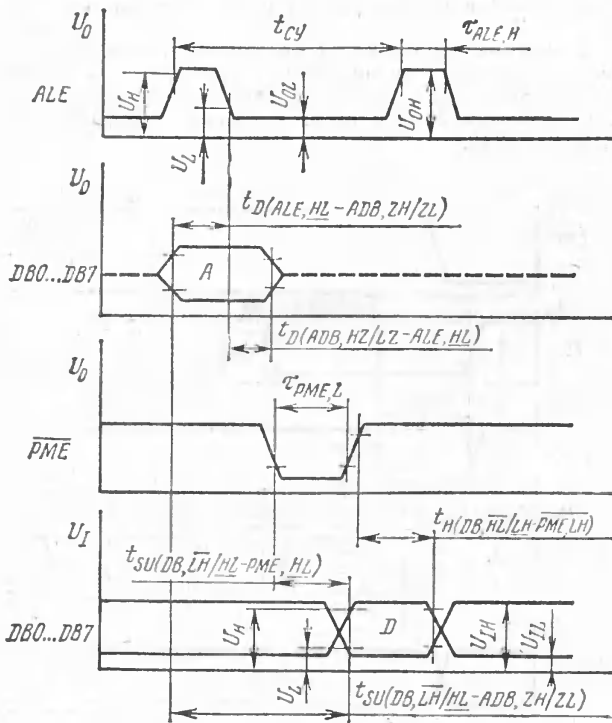


Рис. 8. Временная диаграмма работы микроЭВМ в режиме с внешней памятью программ

роле процессора микроЭВМ. Для этого внутренняя память команд отключается подачей на вывод ЕМА напряжения 5В.

Этот режим используется также, если внутренней памяти команд недостаточно. В этом случае можно совместить внутреннюю и внешнюю память команд общим объемом до 4096 байт (на вывод ЕМА подано напряжение низкого уровня) или только внешнюю память с

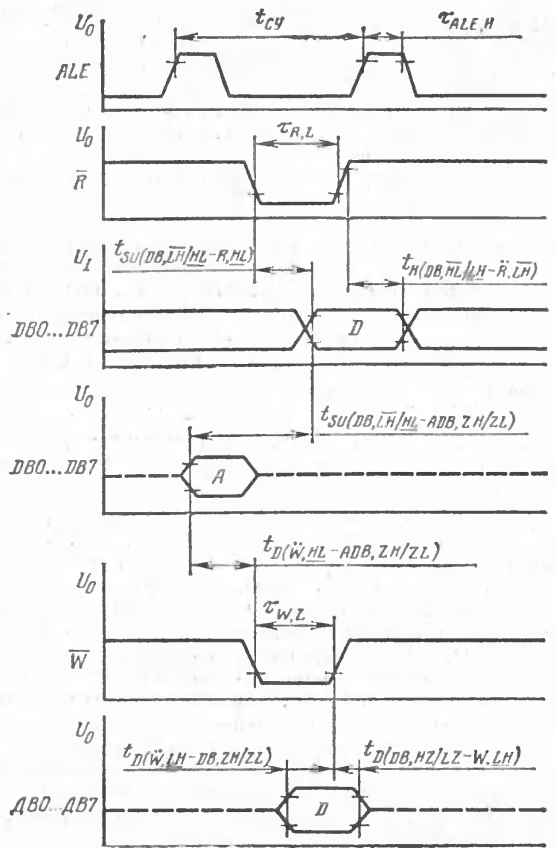


Рис. 9. Временная диаграмма работы микроЭВМ в режиме с внешней памятью данных

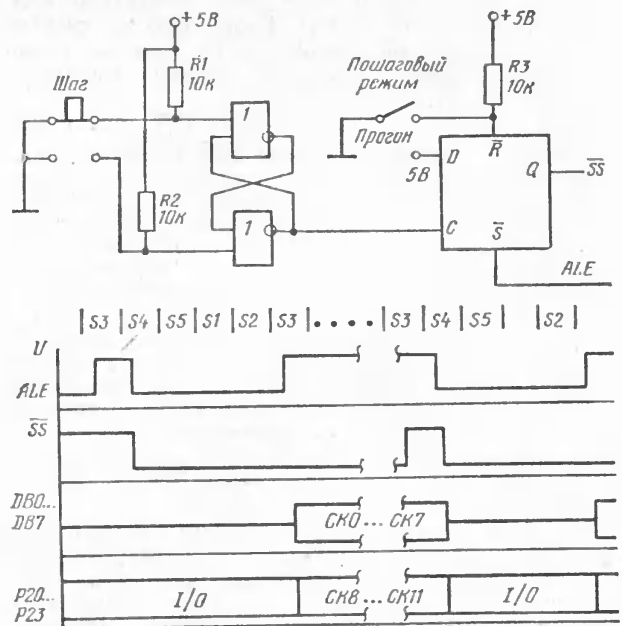


Рис. 10. Схема реализации режима пошагового выполнения команд и временная диаграмма работы микроЭВМ

максимальным объемом до 4096 байт (вывод ЕМА соединен с источником напряжения +5В). При работе с внешней памятью выдача младших разрядов адреса (A7...A0) на внешнюю память и прием кода команд из внешней памяти осуществляется через порт P0 (DB7...DB0). При этом адрес ЗУ фиксируется по сигналу ALE, а принимаются команды с микроЭВМ по сигналу PME. Старшие разряды адреса A11...A8 выдаются через порт P2 (P23...P20).

При увеличении объема памяти данных за счет подключения внешнего ОЗУ емкостью до 256 байт обмен данными между ОЗУ и микроЭВМ осуществляется через двунаправленный порт P0 с помощью команд MOVXA, @R; MOVX@R, A (рис. 9). При этом адрес данных фиксируется по сигналу ALE, а прием и выдача данных — по сигналам R и W.

Режим пошагового выполнения программы (рис. 10) необходим для отладки и проверки программы. В качестве управляющего входа микроЭВМ используется вывод SS и выходной сигнал ALE.

Пошаговый режим дает возможность остановки после выполнения команды независимо от того, одно- или двухцикловая команда. При подаче сигнала (низким уровнем напряжения) микроЭВМ завершает выполнение команды и делает останов; при этом на шинах порта P0 находится восемь младших разрядов, на шинах порта P2 (P23...P20) четыре старших разряда адреса следующей команды. Пошаговый режим целесообразно использовать при работе с внешней памятью. Когда микроЭВМ находится в состоянии останова, данные

ввода-вывода на выводах порта P0 и младших разрядов порта P2 отсутствуют. Эту информацию можно фиксировать внешним устройством по фронту сигнала ALE.

Режим программирования (рис. 11) в микросхеме КМ1816ВЕ48 применяется для записи информации во внутреннюю память программ.

Система команд

Базовая система команд включает 96 команд, 68 из них однобайтные. В двухбайтных командах первый байт содержит информацию о коде команды, второй байт — непосредственные данные или младшие разряды адреса следующей команды. Большинство команд (53) выполняются за один машинный цикл — 43 команды, в том числе 15 однобайтных, выполняются за два машинных цикла. Выполнение двухбайтных команд за два цикла связано с дополнительным обращением к программной памяти. Различают команды передачи данных, преобразования данных, передачи управления (табл. 3).

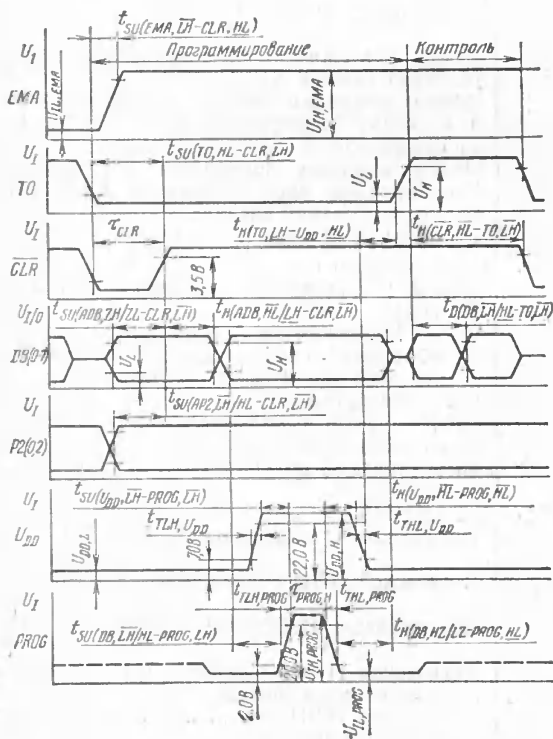


Рис. 11. Временная диаграмма работы микроЭВМ при программировании и контроле после программирования

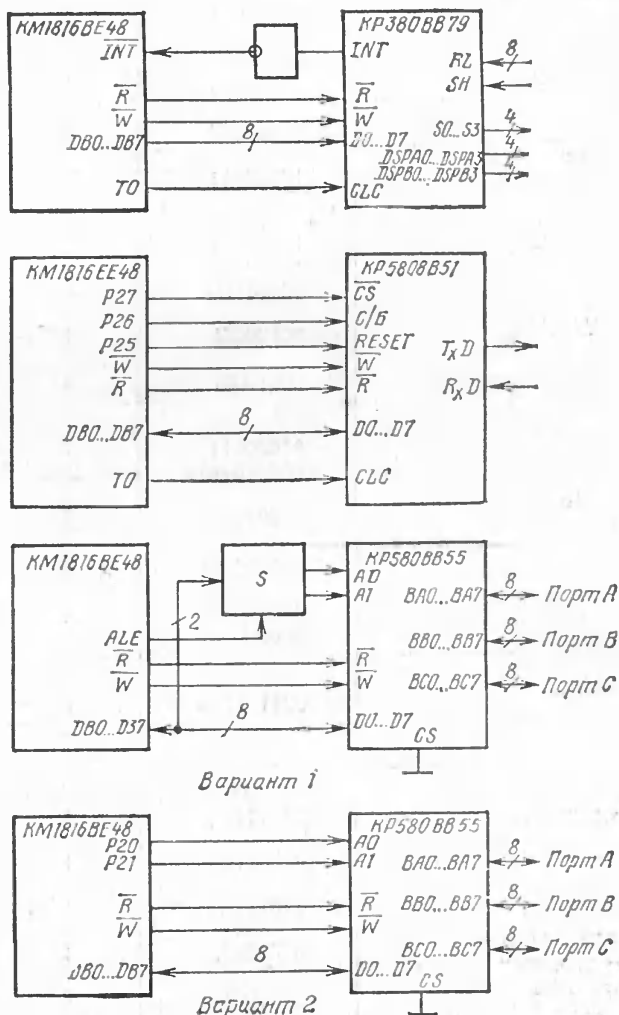


Рис. 12. Структурные схемы подключения БИС серии К580

Система команд однокристалльных микроЭВМ серии К1816

Мнемоника	Код операции, разряды 7, 6, 5, 4, 3, 2, 1, 0	Число циклов	Число байтов команды	Комментарий
1	2	3	4	5
I. КОМАНДЫ ПЕРЕДАЧИ ДАННЫХ				
MOVA, R _i (i=0...7)	11111(i) ₂	1	1	Пересылка содержимого РОН в А
MOVR _i , A (i=0...7)	10101(i) ₂	1	1	Пересылка содержимого А в РОН
MOVA, @R _i (i=0,1)	1111000i	1	1	Пересылка косвенно адресуемого байта памяти ЗУ в А
MOV@R _i , A (i=0,1)	1010000i	1	1	Пересылка содержимого А в косвенно адресуемый байт памяти ЗУ
MOVXA, @R _i (i=0,1)	1000000i	2	1	Пересылка косвенно адресуемого байта памяти внешнего ЗУ данных в А
MOVX@R _i , A (i=0,1)	1001000i	2	1	Пересылка содержимого А в косвенно адресуемый байт памяти внешнего ЗУ данных
MOVR _i , #d (i=0...7)	10111(i) ₂	2	2	Пересылка непосредственных данных в РОН
MOV@R _i , #d (i=0,1)	1011000i	2	2	Пересылка непосредственных данных в косвенно адресуемый байт памяти ЗУ
MOVA, #d	00100011	2	2	Пересылка непосредственных данных в А
MOVA, PSW	11000111	1	1	Пересылка содержимого PSW в А
MOVPSW, A	11010111	1	1	Пересылка содержимого А в PSW
MOVA, T	01000010	1	1	Пересылка содержимого регистра таймера-счетчика в А
MOVT, A	01100010	1	1	Пересылка содержимого А в регистр таймера-счетчика
MOVPA, @A	10100011	2	1	Пересылка в А содержимого косвенно адресуемого байта памяти ЗУ программ на текущей странице
MOVPA, @A	11100011	2	1	Пересылка в А косвенно адресуемого байта памяти ЗУ программ на с.
XCH A, R _i (i=0...7)	00101(i) ₂	1	1	Обмен содержимого А и РОН
XCH A, @R _i (i=0,1)	0010000i	1	1	Обмен содержимого А и косвенно адресуемого байта памяти ЗУ
XCH DA, @R _i (i=0,1)	0011000i	1	1	Обмен младшего полубайта содержимого А и косвенно адресуемого младшего полубайта памяти ЗУ
SWAR A	01000111	1	1	Обмен младших полубайтов содержимого А
INA, P _p (p=1,2)	000010(p) ₂	2	1	Стробируемый ввод данных из порта P1 (2) в А по окончании сигнала PROG
INS A, BUS	00001000	2	1	Стробируемый ввод данных из порта P0 в А по окончании сигнала R
OUTL, P _p , A (p=1,2)	001110(p) ₂	2	1	Вывод с фиксацией содержимого А в порт P1 (P2)
OUTL BUS, A	00000010	2	1	Вывод содержимого А в порт P0
MOVD A, P _p	0000 11nn	2	1	Стробируемый ввод данных из порта 4 (5...7) в младший полубайт А (с обнулением старшего полубайта А)
MOVD P _p , A	0011 11nn	2	1	Стробируемый вывод младшего полубайта А на порт-расширитель 4(5...7) (с сохранением старшего полубайта)
II. КОМАНДЫ ПРЕОБРАЗОВАНИЯ ДАННЫХ				
ANL A, R _i (i=0...7)	01011(i) ₂	1	1	Логическое И содержимого А и РОН
ORL A, R _i (i=0...7)	01001(i) ₂	1	1	Логическое ИЛИ содержимого А и РОН
XRL A, R _i (i=0...7)	11011(i) ₂	1	1	Исключающее ИЛИ содержимого А и РОН
ANL A, @R _i (i=0,1)	0101000i	1	1	Логическое И содержимого А и косвенно адресуемого байта памяти ЗУ
ORL A, #d	01000011	2	2	Логическое ИЛИ содержимого А с непосредственными данными
XRL A, #d	11010011	2	2	Исключающее ИЛИ содержимого А с непосредственными данными

1	2	3	4	5
ANL P _p , # d (p=1,2)	10010(p) ₂	2	2	Логическое И фиксированных данных порта P1 (P2) с непосредственными данными
ANL BUS, # d	10011000	2	2	
ORL P _p , # d (p=1,2)	100010(p) ₂	2	2	Логическое ИЛИ фиксированных данных порта P1 (P2) с непосредственными данными
ORL BUS, # d	10001000	2	2	
ANLD P _p , A	100111nn	2	1	Логическое ИЛИ фиксированных данных порта P0 с непосредственными данными
ORLD P _p , A	100011nn	2	1	Стробируемый вывод младшего полубайта A на порт-расширитель 4(5...7) (с сохранением содержимого A) по окончании сигнала через разряды 3...0 порта P2. По началу сигнала PROG выдача через разряды 3...0 порта P2 кода-признака логической операции И и адреса порта-расширителя 4(5...7)
RRA	01110111	1	1	Стробируемый вывод младшего полубайта A на порт-расширитель 4(5...7) (с сохранением содержимого A) по окончании сигнала PROG через разряды 3...9 порта P2. По началу сигнала PROG выдача через разряды 3...0 порта P2 кода-признака логической операции ИЛИ и адреса порта-расширителя 4(5...7)
RLA	11100111	1	1	Циклический сдвиг содержимого A вправо на 1 разряд без переноса
RRC A	01100111	1	1	Циклический сдвиг содержимого A влево на 1 разряд без переноса
RLC A	11110111	1	1	Циклический сдвиг содержимого A вправо на 1 разряд с переносом
INC A	00010111	1	1	Циклический сдвиг содержимого A влево на 1 разряд с переносом
INC R _i (i=0...7)	00011(i) ₂	1	1	Инкрементирование содержимого A
INC@R _i (i=0,1)	0001000i	1	1	Инкрементирование содержимого PОН
DEC A	00000111	1	1	Инкрементирование косвенно адресуемого байта памяти ЗУ
DEC R _i (i=0...7)	11001(i) ₂	1	1	Декрементирование содержимого A
DJNZR _i , ADDRESS (i=0,7)	11101(i) ₂	2	2	Декрементирование содержимого PОН
ADD A, R _i (i=0...7)	01101(i) ₂	1	1	Декрементирование содержимого PОН и тестирование его на нуль
ADDC A, R _i (i=0...7)	01111(i) ₂	1	1	Сложение содержимого PОН и содержимого A
ADD A, @R _i (i=0,1)	0110000i	1	1	Сложение содержимого разряда переноса, содержимого PОН и содержимого A
ADDC A, @R _i (i=0,1)	0111000i	1	1	Сложение косвенно адресуемого байта памяти ЗУ с содержимым A
ADDA, # d	00000011	2	2	Сложение содержимого разряда переноса содержимого A и косвенно адресуемого байта памяти ЗУ
ADDC, # d	00010011	2	2	Сложение непосредственных данных с содержимым A
CLR A	001001111	1	1	Сложение содержимого разряда переноса и непосредственных данных с содержимым A
CPL A	001100111	1	1	Обнуление A
DA A	01010111	1	1	Инвертирование содержимого A
CLR C	10010111	1	1	Десятичная коррекция
CPL C	10100111	1	1	Обнуление разряда переноса
CLR F0	10000101	1	1	Инвертирование содержимого разряда переноса
CLR F1	10100101	1	1	Обнуление флагового разряда F0
CPL F0	10010101	1	1	Обнуление флагового разряда F1
CPL F1	10110101	1	1	Инвертирование содержимого флагового разряда F0
ENI	00000101	1	1	Инвертирование содержимого флагового разряда F1
				Разрешение внешних прерываний

1	2	3	4	5
DIS I	00010101	1	1	Запрещение внешних прерываний
EN TCNTI	00100101	1	1	Разрешение прерываний по переполнению таймера-счетчика событий
DIS TCNTI	00110101	1	1	Запрещение прерываний при переполнении таймера-счетчика событий
STRT T	01010101	1	1	Запуск таймера
STRT CNT	01000101	1	1	Запуск счетчика отрицательных перепадов на выводе T1
STOP T CNT	01100101	1	1	Останов таймера-счетчика событий
ENTO CLK	01110101	1	1	Разрешение вывода импульсов синхронизации
SEL MB0	11100101	1	1	Выбор нулевого банка памяти программ
SEL MB1	11110101	1	1	Выбор первого банка памяти программ
SEL RB0	11000101	1	1	Выбор нулевого банка рабочих регистров памяти данных
SEL RB1	11010101	1	1	Выбор первого банка рабочих регистров памяти данных
NOP	00000000	1	1	Нет операции

III. КОМАНДЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ

CAL L ADDRESS	A ₁₀ , A ₉ , A ₈ , 10100	2	2	Вызов подпрограммы
RET	10000011	2	1	Возврат из подпрограммы обслуживания прерываний
RETR	10010011	2	1	Возврат из подпрограммы обслуживания прерываний с восстановлением содержимого
JMP ADDRESS	A ₁₀ , A ₉ , A ₈ , 00100	2	2	Безусловный переход по адресу, указанному во втором байте данной команде
JMP P@A	10110011	2	2	Безусловный переход по адресу, содержащемуся в A, внутри текущей страницы памяти программ
JC ADDRESS	11110110	2	2	Переход, если разряд переноса установлен в единицу
JNC ADDRESS	11100110	2	2	Переход, если разряд переноса установлен в нуль
JZ ADDRESS	11000110	2	2	Переход, если содержимое A равно нулю
JNZ ADDRESS	10010110	2	2	Переход, если содержимое A не равно нулю
JF0 ADDRESS	10110110	2	2	Переход, если флаговый разряд F0 установлен в единицу
JF1 ADDRESS	01101110	2	2	Переход, если флаговый разряд F1 установлен в единицу
JT0 ADDRESS	01101110	2	2	Переход, если уровень на входе T0 высокий
JNT0 ADDRESS	00100110	2	2	Переход, если уровень сигнала на входе T0 низкий («0»)
JT1 ADDRESS	01010110	2	2	Переход, если уровень на входе T1 высокий («1»)
JNT1 ADDRESS	01000110	2	2	Переход, если уровень на входе T1 низкий («0»)
JTF ADDRESS	00010110	2	2	Переход, если флаговый разряд таймера установлен в единицу
JNI ADDRESS	10000110	2	2	Переход, если на входе прерывания низкий уровень
JB _b ADDRESS (b=0...7)	(b) ₂ 10010	2	2	Переход, если указанный в коде операции разряд A установлен в единицу

Примечания. 1. Косвенная адресация байта внутренней памяти ЗУ данных выполняется по адресу в PОН (разряды 0...5).

2. Фиксация содержимого A при выводе в порт P0 возможна только при работе с внутренней памятью ЗУ программ.

3. При выполнении команд MOVDA, P_r, MOVDP, A, ANLDP, A ORLDP, A по началу сигнала PROG на разряды 2-3 порта P2 выводится соответствующий команде код (0, 10, 11 или 01, а на разряды 0...1 порта P2 — код номера порта-расширителя пп, уменьшенный на 4.

Указания по применению

Основные электрические параметры и режимы эксплуатации

Напряжение питания, В	4,75	5,25
Входное напряжение высокого уровня, В	2,0	
Входное напряжение высокого уровня, В:		

сигналов CR1, CR2, \overline{CLR}	3,8	
сигналов PROG, EMA,	21,5	24,5
Входное напряжение низкого уровня, В	-0,5	0,8
Входное напряжение низкого уровня, В:		
сигналов CR1, CR2, \overline{CLR}	-0,5	0,6
сигнала PROG	—	0,2

Входное напряжение низкого уровня сигнала ЕМА при программировании, В 4,5 5,25
 Тактовая частота, МГц 1,0 11(6,0 для КМ1816ВЕ48)
 Потребляемая мощность, мВт — 600
 Рабочий диапазон температур, °С —10 70

Микросхемы по входам и выходам совместимы со схемами ТТЛ серий. Для выводов CR1, CR2, CLR необходимо применять микросхемы ТТЛ серий с открытым коллектором, подключая резистор сопротивлением, обеспечивающим параметры входных сигналов для этих входов.

Двунаправленные и квазидвунаправленные выходы

микросхем рекомендуется подключать через специальные двунаправленные шинные формирователи серии К589 или др.

Структура двунаправленной шины DB0...DB7 (порт P0) микроЭВМ позволяет использовать в составе системы интерфейсные микросхемы микропроцессорного комплекта серии К580 (рис. 12). Применяются они для увеличения числа каналов ввода-вывода и их типов. В системе можно использовать последовательный интерфейс ввода-вывода КР580ВВ55А, контроллер КР580ВВ79. Обмен информацией в этом случае осуществляется так же, как при обращении к внешней памяти данных с помощью команд типа MOVX.

Статья поступила 26 ноября 1985 г.

УДК 621.3.049.77 : 681.3.06

В. А. Бобков, Ю. Н. Бурмистров, В. А. Кособрюхов, Ю. В. Уткин, Б. Н. Чернуха

ОДНОКРИСТАЛЬНЫЕ 4-РАЗРЯДНЫЕ МИКРОЭВМ СЕРИИ КР1820

МикроЭВМ серии К1820 выполнены по n-канальной МОП-технологии с поликремниевыми затворами на основе динамических логических элементов. Выпускаются в двух модификациях:

КР1820ВЕ1 — микроЭВМ с внешним ПЗУ (рис. 1). Применяется для отладки программного обеспечения, в изделиях с малой тиражностью или на этапе разработки и изготовления опытных партий изделий. Заключена в 40-выводной пластмассовый корпус типа DIP (рис. 2);

КР1820ВЕ2 — микроЭВМ с внутренним масочным ПЗУ программ емкостью 1К байт*. Выполнена в 28-выводном пластмассовом корпусе типа DIP (рис. 3).

* Структурная схема микроЭВМ КР1820ВЕ2 приведена в статье Крылова Е. И. Однокристальные микроЭВМ серии К1814, К1820, К1816. — Микропроцессорные средства и системы, 1985, № 2, с. 3—7.

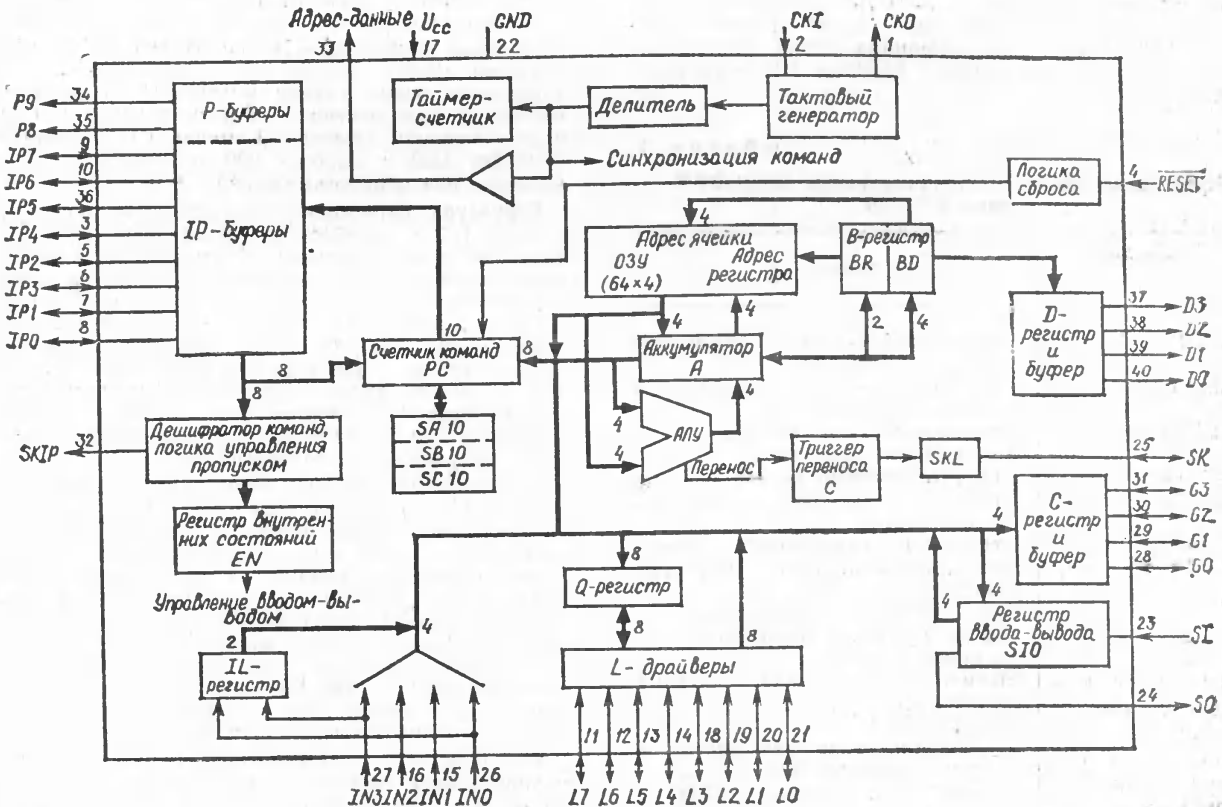


Рис. 1. Структурная схема однокристалльной микроЭВМ КР1820ВЕ1

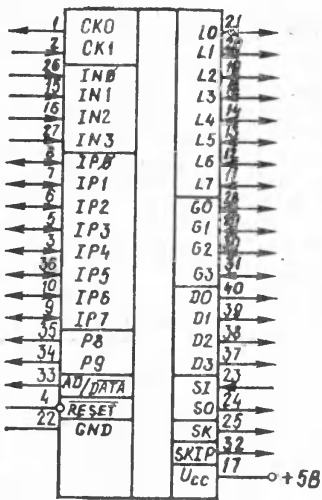


Рис. 2. Условное графическое изображение микроЭВМ КР1820ВЕ1

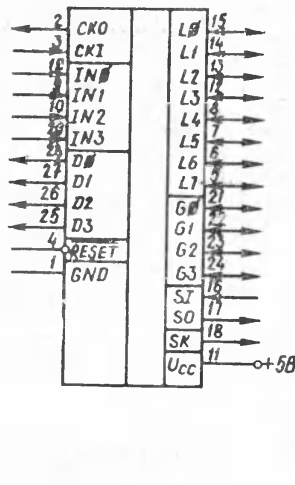


Рис. 3. Условное графическое изображение микроЭВМ КР1820ВЕ2

Кристаллы этих однокристалльных микроЭВМ содержат последовательный канал приема-передачи произвольного программируемого формата, блоки обработки слов и прерываний. Команды работы с таблицами обеспечивают высокую производительность микроЭВМ при работе в режиме микроконтроллера для управления различными устройствами и системами. Назначение выводов микроЭВМ показано в табл. 1.

Система команд (табл. 2) насчитывает 49 команд: 27 однобайтовых и 22 двухбайтовых. Двухбайтовые команды выполняются за два машинных цикла, однобайтовые, за исключением команд LQID и JID, — за один (рис. 4).

Таблица 1

Назначение выводов однокристалльных микроЭВМ серии КР1820

Наименование вывода	Назначение
L7 ... L0	8-разрядный двунаправленный порт ввода-вывода
G3 ... G0	4-разрядный двунаправленный порт ввода-вывода
D3 ... D0	4-разрядный порт вывода
IN3 ... IN0	4-разрядный порт ввода
SO	Последовательный выход генератора
SK	Контролируемый выход тактового генератора
AD/DATA	Строб, подтверждающий адрес
SKIP*	Выход подтверждения пропуска инструкции
SKI	Вход тактового генератора
SKO	Выход тактового генератора
RESET	Начальный запуск
Ucc	Напряжение источника питания микросхемы
GND	Общий вывод
IP7—IPO*	8 входов-выходов для адреса и данных внешнего ПЗУ
P8, P9*	Старшие адреса внешнего ПЗУ

* только для КР1820ВЕ1.

Формат	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
I	КОП							
II	КОП				y			
III	КОП		z		d-1			
IV	КОП	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆
V	КОП		a ₀	a ₁	a ₂	a ₃	a ₄	a ₅
VI	КОП		z		КОП			
VII	КОП						a ₄	a ₅
	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇
VIII	КОП							
	КОП		z		d			
IX	КОП							
	КОП				y			
X	КОП							

Рис. 4. Форматы команд: D₀...D₇ — разряды кода команды; y — 4-разрядное поле константы; a₀...a₉ — поле адреса перехода; z — 2-разрядное поле адреса регистра строк ОЗУ; d — 4-разрядное поле адреса регистра столбцов ОЗУ

Декодирование кодов команд и формирование управляющих воздействий на аппаратные блоки микроЭВМ осуществляются с помощью дешифратора команд и блока формирования условий. При выработке условий учитываются признаки, характерные для различных форматов команд (двух- или однобайтовые), и признаки, полученные в результате выполнения предыдущей команды (перенос, равенство нулю отдельных битов или слова, равенство двух слов).

Дешифратор команд выполнен на основе матрицы с логической функцией «И» (12 входов и 57 выходов). Схематически формируются 54 микрокоманды для управления всеми узлами микроЭВМ. В текущем машинном цикле дешифрируется последующая и выполняется текущая команда. Команда CLR A, записанная в ячейку ПЗУ с адресом 000, позволяет организовать конвейер при обработке команд.

Структура адресации. Адресное пространство ПЗУ разбито на 16 страниц по 64 слова. Каждые четыре страницы образуют блок объемов 256 слов. Такая организация памяти обусловлена особенностями выполнения однобайтовых команд передачи управления JP, JSRP, JID, LQID. Так, команда JP позволяет осуществить переход внутри каждой страницы или внутри страниц 2 и 3. Команда JSRP обеспечивает переход из любого адреса к подпрограмме, расположенной на странице 2. Благодаря этим командам память на страницах 2 и 3 может эффективно использоваться для хранения вложенных подпрограмм.

10-разрядный счетчик команд (PC) позволяет непосредственно адресовать 1024 ячейки ПЗУ. Новый адрес загружается в счетчик команд с каждым новым машинным циклом. Если выполняемая команда не является командой передачи управления, то содержимое счетчика увеличивается на единицу. Четыре старших разряда счетчика команд определяют номер страницы ПЗУ, шесть младших — номер слова в странице. Полную замену содержимого программного счетчика выполняют команды JMP и JSR. Разница между ними состоит в том, что при выполнении команды JSR происходит запись в верхний уровень стека адреса возврата.

Команды RET и RETSK, определяющие возврат из подпрограммы, обеспечивают запись из верхнего уровня стека в счетчик команд адреса возврата. Если возврат осуществляется по команде RETSK, то следующая за ней команда пропускается. Три уровня аппаратного

Система команд однокристальных микроЭВМ КР1820ВЕ1 и КР1802ВЕ2

Мнемоника	Код операции	Формат	Перемещение информации	Условие	Комментарий
	D ₇ ...D ₀				
1	2	3	4	5	6
Арифметические команды					
ASC	0011 0000	I	$A + C \rightarrow \text{ОЗУ} (B) \rightarrow A$ перенос $\rightarrow C$	Перенос	Двоичное сложение с переносом и пропуск следующей команды при формировании переноса
ADD	0011 0001	I	$A + \text{ОЗУ} (B) \rightarrow A$	Нет	Сложение содержимого аккумулятора с содержимым ячейки ОЗУ, без переноса
ADT	0100 1010	I	$A + 10 \rightarrow A$	Нет	Сложение числа 10 (1010) с содержимым аккумулятора, без переноса
AISC	0101 Y	II	$A + Y \rightarrow A$	Перенос	Сложение непосредственное, пропуск следующей команды при формировании переноса
CASC	0001 0000	I	$A + \text{ОЗУ} (B) + C \rightarrow A$ перенос $\rightarrow C$	Перенос	Вычитание содержимого аккумулятора из суммы содержимого ячейки ОЗУ и C-триггера, пропуск следующей команды при формировании переноса
CLRA	0000 0000	I	«0» $\rightarrow A$	Нет	Очистка аккумулятора
COMP	0100 0000	I	$\bar{A} \rightarrow A$	Нет	Формирование дополнения аккумулятора
NOP	0100 0100	I	Нет	Нет	Нет операции
RC	0011 0010	I	«0» $\rightarrow C$	Нет	Запись нуля в C-триггер
SC	0010 0010	I	«1» $\rightarrow C$	Нет	Запись единицы в C-триггер
XOR	0000 0010	I	$A \oplus \text{ОЗУ} (B) \rightarrow A$	Нет	«Исключающее ИЛИ» каждого бита содержимого аккумулятора и ячейки ОЗУ
Команды передачи управления					
JID	1111 1111	I	ИЗУ (PC9,8·A·M \rightarrow PC)	Нет	Команда косвенной адресации
JMP	0110 00a ₉ a ₈ a ₇ ...a ₀	VII	a \rightarrow PC	Нет	Передача управления по 10-разрядному адресу, указанному операндом

1	2	3	4	5	6
JP	1a ₆ a ₀	IV	a→PC	Нет	Переход в пределах страницы: только в пределах страниц 2 и 3
	или 11a ₅ a ₀	V	a→PC		В пределах текущей страницы
JSPR	10a ₅ ..a ₀	V	PC + 1→SA→SB→SC 0010→PC a→PC	Нет	Передача управления с любой страницы, кроме 2-й и 3-й, на ячейку страницы 2
JSP	0110 10 a ₇ a ₆ a ₇ ..a ₀	VII	PC + 1→SA→SB→SC a→PC	Нет	Переход к подпрограмме
RET	0100 1000	I	SC→SB→SA→PC	Нет	Возврат из подпрограммы
RETSK	0100 1001	I	SC→SB→SA→PC	•	Возврат из подпрограммы с пропуском команды
CAMQ	0011 0011 0011 1100	X	A→O _{7...4} OЗУ (B)→Q _{3...0}	Нет	Передача содержимого аккумулятора и ячейки ОЗУ в Q-регистр
CQMA	0011 0011 0010 1100	X	Q _{7...4} →OЗУ (B) Q _{3...0} →A	Нет	Передача содержимого R-регистра в ячейку ОЗУ и аккумулятор
LD	00r 0101	VI	OЗУ (B)→A BR⊕r→BR	Нет	Загрузка содержимого ячейки ОЗУ, адресуемой B-регистром в аккумулятор
LDD	0010 0011 00r d	VIII	OЗУ (rd)→A	Нет	Загрузка аккумулятора содержимым ячейки ОЗУ, прямо адресуемой полями r и d операнда
LQID	1011 1111	I	ПЗУ (PC _{0,8} A·M)→Q SB→SC	Нет	Загрузка Q-регистра содержимым ячейки ПЗУ, косвенно адресуемой
RMB 0 RMB 1 RMB 2 RMB 3	0100 1100 0100 0101 0100 0010 0100 0011	I	«0»→OЗУ (B) ₀ «0»→OЗУ (B) ₁ «0»→OЗУ (B) ₂ «0»→OЗУ (B) ₃	Нет	Запись «0» в разряд ячейки ОЗУ
SMB 0 SMB 1 SMB 2 SMB 3	0100 1101 0100 0111 0100 0110 0100 1011	I	«1»→OЗУ (B) ₀ «1»→OЗУ (B) ₁ «1»→OЗУ (B) ₂ «1»→OЗУ (B) ₃	Нет	Запись «1» в разряд ячейки ОЗУ
STII	0111 y	II	y→OЗУ (B) BD + 1→BD	Нет	Непосредственная запись в ячейку ОЗУ информации, находящейся в поле операнда. Состояние B-регистра увеличивается на единицу

1	2	3	4	5	6
X	00 r 0110	VI	$OЗУ(B) \leftrightarrow A$ $BR \oplus r \rightarrow BR$	Нет	Обмен содержимого аккумулятора и ячейки ОЗУ
XAD	0010 0011 10 r d	VIII	$OЗУ(r, d) \leftrightarrow A$	Нет	То же
XDS	00 r 0111	VI	$OЗУ(B) \leftrightarrow A$ $BD - 1 \rightarrow BD$ $BR \oplus r \rightarrow BR$	Пропуск при $BD - 1 = 0$	То же
XIS	00 r 0100	VI	$OЗУ(B) \leftrightarrow A$ $BD + 1 \rightarrow BD$ $BR + r \rightarrow BR$	Пропуск при $BD + 1 = 15$	То же

Команды обращения к регистрам

CAB	0101 0000	I	$A \rightarrow BD$	Нет	Запись содержимого аккумулятора в младшие разряды В-регистра
CBA	0100 1110	I	$BD \rightarrow A$	Нет	Запись содержимого В-регистра в аккумулятор
LBI	00 r d — 1 d = 0:9:15 0011 0011 10 r d d = 1:8	III VII	$r \cdot d \rightarrow BR \cdot BD$	* *	Непосредственная загрузка содержимого полей и операнда в ВR- и ВD-регистры
LEI	0011 0011 0110 y	IX	$y \rightarrow EN$	Нет	Непосредственная загрузка содержимого поля операнда в EN-регистр
XABR	0001 0010	I	$A \leftrightarrow BR$	Нет	Обмен содержимого В-регистра с аккумулятором
SKC	0010 0000	I		$C = 1$	Пропуск команды при $C = 1$
SKE	0010 0001	I		$A = OЗУ(B)$	Сравнение содержимого аккумулятора с содержимым ячейки ОЗУ
SKGZ	0011 0011 0010 0001	I		$G = 0$	Проверка состояния четырех G-линий и пропуск следующей команды, если все биты G-порта равны нулю
SKGBZ0 SKGBZ1 SKGBZ2 SKGBZ3	0011 0011 } 1-й байт 0000 0001 } 0001 0001 } 2-й байт 0000 0011 } 0001 0011 }	I		$G0 = 0$ $G1 = 0$ $G2 = 0$ $G3 = 0$	Проверка состояния одной из четырех G-линий
SKMBZ0 SKMBZ1 SKMBZ2 SKMBZ3	0000 0001 0001 0001 0000 0011 0001 0111	I		$OЗУ(B)_0 = 0$ $OЗУ(B)_1 = 0$ $OЗУ(B)_2 = 0$ $OЗУ(B)_3 = 0$	Проверка состояния битов ячейки ОЗУ, определяемых операндом команды

1	2	3	4	5	6
SKT	0100 0001	I		***	Проверка состояния внутреннего 10-разрядного таймера
Команды ввода-вывода					
ING	0011 0011 0010 1010	X	G→A	Нет	Ввод информации G-порта в аккумулятор
ININ	0011 0011 0010 1000	X	IN→A	Нет	Ввод информации с входов IN-порта в аккумулятор
INIL	0011 0011 0010 1001	X	IL3·IL0→A	Нет	Ввод содержимого старшего и младшего разряда регистра-защелки IL соответственно в старший и младший разряды аккумулятора
INL	0011 0011 0010 1110	X	L _{7...4} →OЗУ (B) L _{3...0} →A	Нет	Ввод информации L-порта в аккумулятор и в ячейку OЗУ
OBD	0011 0011 0011 1110	X	BD→D	Нет	Вывод содержимого младших разрядов B-регистра на D-выходы
OGI	0011 0011 0101 y	IX	y→G	Нет	Непосредственный вывод содержимого поля операнда в G-порт
OMG	0011 0011 0011 1010	X	OЗУ (B)→G	Нет	Вывод данных из ячейки OЗУ в G-порт
XAS	0100 1111	X	A ←→ SIO·C→SKL	Нет	Обмен содержимым аккумулятора и SIO-регистра, занесение содержимого C-триггера в SKL-регистр

Примечание: * — всегда пропуск после возврата; ** — пропуск последующих команд LВ1, пока не встретится любая другая команда; *** — пропуск следующей команды при переполнении счетчика временной базы таймера.

стека позволяют записать до трех адресов возврата, что обеспечивает эффективное функционирование вложенных подпрограмм.

При выполнении команд косвенной адресации I QID и JID происходит загрузка только восьми младших разрядов программного счетчика, что определяет обращение к новому слову на странице, находящемуся в одном из страничных блоков. Команда LQID обеспечивает запись в Q-регистр из ячейки ПЗУ, адресованной словом, состоящим из содержимого выбранной ячейки OЗУ и аккумулятора. При выполнении команды LQID теряется информация в нижнем уровне стека. По команде JID управление передается ячейке ПЗУ, по адресу, равному содержимому аккумулятора и выбранной ячейки, — OЗУ.

OЗУ микроЭВМ организовано в виде четырех регистров, каждый из которых состоит из 16 четырех-

разрядных ячеек (рис. 5). Адрес выбранного столбца запоминается в 4-разрядном BD-регистре, адрес одной из четырех строк заносится в BR-регистр.



Рис. 5. Структура OЗУ однокристальных микроЭВМ КР1820ВЕ1 и КР1820ВЕ2: заштрихованная область OЗУ адресуется байтовой командой LВ1

Команда LBI, обеспечивающая прямую адресацию ячейки памяти, имеет два формата: однобайтовый для ячеек с номерами 0 и 9...15 (заштрихованная область на рис. 5) и двухбайтовый для ячеек с остальными номерами. Это целесообразно учитывать при составлении карты памяти с целью экономии пространства ПЗУ. Данные из любой ячейки ОЗУ могут быть переданы в аккумулятор с помощью команды LDD. При этом содержимое BR- и BD-регистров не изменяется.

Арифметическое логическое устройство и аккумулятор. АЛУ выполняет операции сложения без учета и с учетом переноса, пересылки (CAB, XAS), логическую операцию «исключающее ИЛИ», а также операции инкрементирования и декрементирования BD-регистра в ходе выполнения операций, связанных с обращением к ОЗУ (STII, XDS, XAS). Результаты операций хранятся в аккумуляторе. Арифметические и логические операции выполняются по инструкциям ПЗУ над операндами, поступающими из ОЗУ, операндного поля команды и аккумулятора.

Единица переноса может образовываться при выполнении арифметических операций CASC, ASC, операций сравнения SKC, SKE, операций, связанных с изменением содержимого BD-регистра командами STII, X, XIS, операций обмена аккумулятора с SIO-регистром по команде XAS. Содержимое C-триггера (см. рис. 1) может быть непосредственно переписано в SKL-триггер и выведено на выход SK. Кроме того, в зависимости от содержимого регистра внутренних состояний EN выход SK может быть определен как выход синхросигналов для синхронизации приема и передачи последовательного порта SO и SI.

Регистр — SIO может работать как 4-разрядный последовательный регистр ввода-вывода (регистр сдвига) или как двоичный счетчик в зависимости от состояния младшего разряда EN-регистра (табл. 3). При EN0=0 он работает в режиме сдвига информации влево с каждым машинным циклом. Данные, поступающие на вход SI, записываются при этом в младший разряд SIO.

При EN0=1 регистр работает в режиме двоичного асинхронного счетчика, содержимое которого декрементируется с каждым переходом с высокого уровня на низкий на SI-входе. Длительность импульсов, поступающих на SI-вход должна быть не менее двух командных циклов. Кроме того, SIO-регистр можно использовать для временного хранения результатов вычислений. Старший разряд EN3 разрешает или запрещает выдачу информации на SO-выход.

Обработка прерываний. В качестве входа запроса прерывания используется IN1. Запрос прерывания полу-

чает подтверждение при соблюдении следующих условий:

разряд регистра EN1 установлен в единицу;
сигнал запроса прерывания удерживается на время, равное не менее чем два машинных цикла;
закончено выполнение текущей команды.

При подтверждении прерывания в стек заносится увеличенное на единицу значение счетчика команд (PC+1). При этом ранее записанные адреса возвратов опускаются на нижние уровни стека, а прежнее содержимое последнего уровня теряется (CK+1→SA→SB→SC). Разряд регистра EN1 сбрасывается, запрещая последующие прерывания.

В счетчик команд автоматически заносится адрес последнего слова на третьей странице ПЗУ (OFF), по которому должна быть записана команда NOP, являющаяся первой командой подпрограммы обработки прерывания. Для разрешения последующих прерываний непосредственно перед командой возврата RET, оканчивающей подпрограмму обработки прерывания, необходимо ввести команду LEI1.

Порты ввода-вывода. Ввод-вывод информации и управляющих сигналов осуществляется через пять портов G, L, IN, D и S (см. табл. 1).

Двунаправленный G-порт обеспечивает ввод информации со входов G0...G3 непосредственно в аккумулятор по команде INC. Вывод данных из поля операнда команды и адресованной ячейки ОЗУ осуществляется двумя командами OGI и OMG соответственно.

8-разрядный двунаправленный L-порт служит для вывода информации, находящейся в Q-регистре по команде LEI4. Благодаря наличию ограничителей выходного тока этот порт можно использовать для управления светодиодными индикаторами. Данные с входов L0...L7 принимаются в аккумулятор (четыре младших разряда) и адресованную ячейку ОЗУ по команде INL.

Данные со входов IN0...IN3 по команде ININ заносятся в аккумулятор. Кроме того, с помощью 2-разрядного регистра-защелки IL данные с внешних устройств могут приниматься в асинхронном режиме. Разряды IL-регистра устанавливаются в единичное состояние перепадом входного уровня с «Лог. 1» до «Лог. 0». Команда INIL переносит содержимое IL-регистра в аккумулятор и сбрасывает IL-регистр.

По команде OBD содержимое BD-регистра заносится в D-регистр и выводится на выходе D0...D3.

Последовательный порт ввода-вывода, состоящий из входа SI и выхода SO, обеспечивает синхронный обмен с периферийными устройствами. По команде XAS данные из аккумулятора обмениваются с данными в

Таблица 3

Зависимость режимов функционирования регистра SIO и последовательного порта ввода-вывода от состояния EN-регистра

EN3	EN0	Режим работы SIO-регистра	Состояние SI-входа	Состояние SO-выхода	Состояние SKL- и SK-выходов
0	0	Сдвиговый регистр	Вход сдвигового регистра	0	При SKL-1 SK — системная синхронизация. При SKL-0 SK-0
1	0	То же	То же	Последовательный вывод	То же
0	1	Двоичный счетчик	Вход двоичного счетчика	0	При SKL-1 SK-1 При SKL-0 SK-0
1	1	То же	То же	1	То же

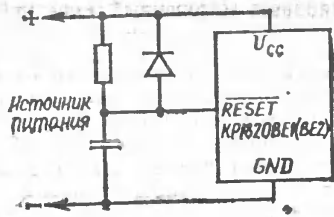


Рис. 6. Схема начального запуска микроЭВМ.

СИО-регистре и в последующих машинных циклах передаются на выход SO и принимаются на входе SI.

Начальный запуск микроЭВМ происходит автоматически при условии, что время нарастания напряжения питания не превышает 1 мс. В противном случае для надежного запуска необходимо ко входу RESET подключить RC-цепочку и диод (рис. 6). Постоянная времени RC-цепочки должна минимум в пять раз превышать время установления напряжения питания. Низкий уровень должен быть приложен ко входу RESET на время не менее трех командных циклов. По команде RESET обнуляется счетчик команд, очищаются аккумулятор и регистры BR, BD, D, EN и СИО. На вывод SK с каждым командным циклом выдается синхрипульс. Ячейки ОЗУ могут быть очищены только программным путем.

Синхронизация и основные характеристики. В микроЭВМ серии КР1820 предусмотрены режимы внешней и внутренней синхронизации. В первом — тактовые импульсы подаются на вход SK1, во втором — кварцевый резонатор или LC-контуры подключаются между выводами SKO и SK1. Время машинного цикла микроЭВМ в 16 раз больше периода внешней и внутренней синхронизации.

Основные электрические характеристики и условия эксплуатации микроЭВМ приведены ниже.

	Мин.	Макс.
Входное напряжение низкого уровня, В	-0,6	1,2
Входное напряжение высокого уровня, В	2,4	U_{cc}
Выходное напряжение низкого уровня, В	—	0,8
Время машинного цикла, мкс	16	4
Напряжение питания U_{cc} , В	5	9
Ток потребления, мА	—	35
Рабочий диапазон температур, °С	-45	+80

Пример сложения двоично-десятичных чисел. Программа сложения положительных 12-разрядных десятичных чисел представлена ниже. Одно слагаемое находится в первом регистре ОЗУ, другое в нулевом. Каждая десятичная цифра занимает одну ячейку ОЗУ, младший разряд находится в нулевой ячейке, старший — в двенадцатой. Программа занимает 10 ячеек ПЗУ и выполняется за 450 мкс.

После выполнения программы в нулевом регистре ОЗУ содержится сумма двух 12-разрядных десятичных чисел.

Применение микроЭВМ. При решении задач управления измерительными приборами часто возникает необходимость организовать структуру устройства с использованием индикатора и клавиатуры. Пример непосредственного подключения светодиодного индикатора на четыре знака к матрице и клавиатуры (матрица 4×4 кнопок) приведен на рис. 7. Для приема информации с клавиатуры и организации индикации использованы IN-, L- и D-порты. Для подключения других внешних устройств оставлены свободными G-порт и порт последовательного ввода-вывода S.

BCDADD :	LBI	1,0	: Адресация младшего разряда первого числа
	RC		: Сброс переноса
ADDL :	LDI		: Передача содержимого ячейки в аккумулятор, модификация содержимого для обращения к другому слагаемому
	AISC6		: Десятичная коррекция
	ASC		: и сложение соответствующих ячеек ОЗУ
	XISI		: Передача результата сложения в нулевой регистр ОЗУ модификации BR-и BD-регистров для перехода к следующей ячейке
	CBA		: Передача содержимого BD-регистра в аккумулятор
	AISC3		: Проверка сложения старших двенадцатых разрядов
	JP	ADDL	: Сложение очередного разряда
	RET		: Возврат к основной программе

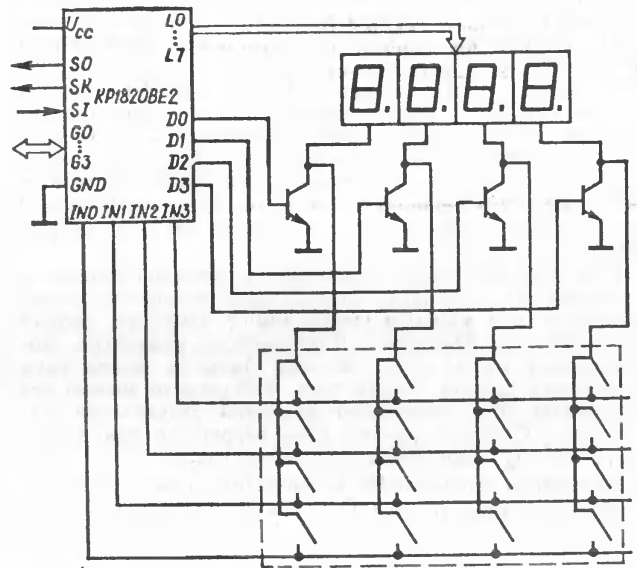


Рис. 7. Схема подключения индикатора и клавиатуры

МикроЭВМ серии КР1820 дают возможность использовать различные методы аналого-цифрового преобразования. Рассмотрим простой и дешевый способ, основан-

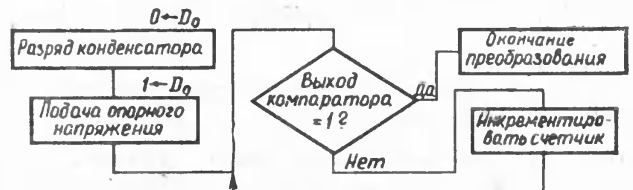


Рис. 8. Алгоритм аналого-цифрового преобразования с помощью микроЭВМ серии КР1820

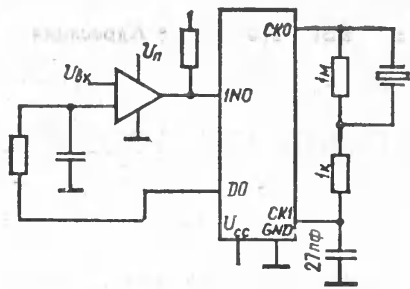


Рис. 9. Схема АЦП на основе однокристалльной микроЭВМ

ный на заряде конденсатора (рис. 8, 9). Реализация более сложных алгоритмов преобразования, например преобразования по методу двойного интегрирования с компенсацией нуля, позволяет построить АЦП, имеющие 10- и 12-разрядную точность.

Однокристалльные микроЭВМ серии К1820 нашли применение в автомобильном радиоприемнике с синтезатором частоты. МикроЭВМ управляет синтезатором, обеспечивает настройку в УКВ-, СВ- и ДВ-диапазонах, запоминание частот настройки до пяти станций в каждом диапазоне, автоматический поиск станций, бесшумную настройку и индикацию частоты принимаемой станции.

Однокристалльные микроЭВМ могут успешно использоваться для построения автомобильного эконометра. В этом случае при движении автомобиля можно вычис-

лять и индицировать мгновенный и средний расходы бензина, среднюю скорость, время нахождения в пути, расход бензина с момента выезда и истинное время. Прибор позволит водителю оптимизировать скоростной режим и существенно сэкономить горючее.

Примером использования микроЭВМ в измерительной технике может служить метанометр индивидуального пользования, предназначенный для измерения концентрации горючих газов в шахтных выработках и других газоопасных объектах и помещениях. Алгоритм работы метанометра предусматривает периодическое включение датчика, подсчет концентрации горючего газа, сравнение измеренных значений с уставками, генерацию многорежимной тревожной сигнализации. Перед началом цикла измерения проверяется напряжение питания батарей и в случае отклонения от нормы подается звуковой сигнал. МикроЭВМ следит за разрядом батарей питания и определяет оптимальный режим их зарядки.

Для отладки программного обеспечения разработаны кросс-ассемблер и математическая модель микроЭВМ, написанные на языке Паскаль с включением фрагментов на ассемблере. Программа математической модели отражает архитектуру и систему команд микроЭВМ серии К1820.

Скомпонованная со стандартным отладчиком языка высокого уровня, она позволяет осуществить как пошаговый, так и автоматический режим отладки программ с возможностью останова в произвольной точке. Совместная отладка аппаратных и программных средств производится с помощью блока схемного эмулятора, соединяющегося с отлаживаемым устройством гибким кабелем.

Статья поступила 6 ноября 1985 г.

ПАМЯТКА АВТОРУ

Общие требования

1. Предпочтением для публикации в журнале пользуются статьи, в которых кратко и ясно излагаются конкретные аппаратные и программные решения в области разработки и внедрения микропроцессорных средств, а также систем автоматизации на их основе.

2. При подготовке выполненной работы к публикации авторам следует стремиться к тому, чтобы язык статьи был понятен возможно более широкому кругу специалистов, заинтересованных во внедрении микропроцессорной техники.

3. Статья должна быть хорошо иллюстрирована: принципиальные, структурные или функциональные схемы, монтажные схемы или фотонегативы печатной платы; фотографии (слайды) общего вида и отдельных узлов блоков (фото с дисплея ЭВМ); структура программной системы; тексты избранных фрагментов программ, подпрограммы и т. д.

Оформление рукописи

1. Объем статей (включая аннотацию, рисунки, 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; 3 страницы — для рекламного материала или информационного сообщения.

2. Рукопись представляется в двух экземплярах, напечатанных через два интервала на одной стороне листа стандартного размера с полями слева шириной 30 мм. Оба экземпляра должны быть подписаны всеми авторами.

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 на графические работы.

5. Тексты программ с АЦПУ должны быть четко и контрастно отпечатаны на белой бумаге. При невозможности изготовить с представленной авторами АЦПУ выдачи качественного клише — программы не публикуются.

6. В качестве иллюстраций следует присылать цветные слайды (предпочтительно на широкой пленке), которые могли бы быть опубликованы на цветной вкладке или обложке журнала, и их черно-белые варианты (для иллюстраций в тексте).

7. 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 (служебные и домашние).

8. Допускается не более четырех соавторов одной статьи.

ОДНОКРИСТАЛЛЬНЫЙ ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ 16-РАЗРЯДНЫЙ МИКРОПРОЦЕССОР KM1810VM86

Современные требования к средствам вычислительной техники обусловили необходимость разработки микропроцессорного комплекта, ориентированного на решение сложных задач в реальном масштабе времени, обладающего повышенным быстродействием, увеличенной длиной обрабатываемых слов, и в то же время архитектурно и программно совместимого с комплектом БИС серии KP580. Для решения этой задачи разработан базовый микропроцессорный комплект серии K1810, включающий в настоящее время: СБИС однокристалльного 16-разрядного микропроцессора KM1810VM86, БИС программируемого контроллера приоритетных прерываний KP1810VM59, системного контроллера KP1810VK88, генератора тактовых сигналов KP1810 GF84, шинных формирователей KP1810VA86/87, буферных регистров KP1810KP82/83, арбитра системной шины KP1810VB89. Однокристалльный высокопроизводительный 16-разрядный микропроцессор KM1810VM86, наряду с программной и аппаратной совместимостью с периферийными и контроллерными БИС серии KP580, обладает возможностью эффективной работы с языками высокого уровня, большим объемом адресуемой памяти и устройств ввода-вывода, возможностью обработки различных типов данных, разнообразными способами адресации памяти, гибкой и мощной системой команд, развитой структурой прерываний, наличием программных и аппаратных средств для работы в микропроцессорных системах.

Указанные свойства микропроцессора обеспечиваются совокупностью архитектурных, структурных и схемотехнических решений, а также применением современной технологии МДП БИС на п-канальных МДП транзисторах с пропорциональным масштабированием размеров элементов. Эта технология обеспечивает реализацию схемотехнических решений с четырьмя уровнями пороговых напряжений МДП транзисторов на одном кристалле, что существенно повышает плотность компоновки и обеспечивает высокие характеристики соотношения параметров быстродействия — мощность при однокристалльной конструкции СБИС.

Микропроцессор KM1810VM86 реализован по традиционной архитектуре, при которой осуществляется последовательная выборка информации из памяти программ и памяти дан-

ных, и характеризуется рядом структурных решений, среди которых прежде всего следует выделить:

развитую регистровую структуру, что существенно уменьшает количество обращений к памяти;

конвейерный принцип выполнения команд с предварительной выборкой, обеспечивающий максимальную пропускную способность канала;

распределенное аппаратно-программное устройство управления; обширный 16-разрядный канал адреса данных с разделением во времени;

многофункциональное использование выводов схемы, позволяющее при существующих конструктивных ограничениях адаптировать процессор к уровню сложности разрабатываемой системы.

Сохраняя все особенности регистровой структуры однокристалльных 8-

разрядных микропроцессоров, СБИС KM1810VM86 представляет собой двухадресную машину, обеспечивающую следующие комбинации источника-приемника операндов: регистр-регистр, регистр-память, память-регистр, память-память (только для операций над строками); непосредственный операнд-регистр и непосредственный операнд-память.

Одной из отличительных особенностей данного микропроцессора, позволяющей частично решить противоречие между стремлением к увеличению функций микропроцессора и ограничением на число выводов корпуса СБИС, является возможность аппаратной перестройки внутренней структуры схемы управления и синхронизации. При этом разработчик может выбрать подмножество выходов (входных) управляющих сигналов в соответствии со степенью сложности проектируемой микропро-

Технические характеристики микропроцессора KM1810VM86

Представление чисел	дополнительный код с фиксированной запятой
Система команд	135 базовых команд; безадресная, однодвухадресная
Типы обрабатываемых данных	биты, двоично-десятичные упакованные и распакованные числа, байты, слова, строки до 64К байт длиной
Число программно-доступных регистров	14 шестнадцатиразрядных
Объем адресуемой памяти, байт	1М (1.048.576)
Число адресуемых устройств ввода-вывода	64К/64К'
Число способов вычисления адреса операнда в памяти	24
Число типов прерывания	до 256
Тактовая частота, МГц	до 5
Максимальное быстродействие (операций/с типа «регистр-регистр»)	до 2,5 Мли
Напряжение питания, В	+5±5%
Потребляемая мощность, Вт	не более 1,75
Уровни входных логических сигналов, В	
U_{OH}	не менее 2,4
U_{OL}	не более 0,45
Нагрузочная способность, мА	
I_{OH}	0,4
I_{OL}	2,0
Уровни входных логических сигналов, В	
U_{IH}	не менее 2,0
U_{IL}	не более 0,8
Емкость нагрузки, пФ	до 100
Конструкция	металлокерамический 40-выводной корпус типа DIP

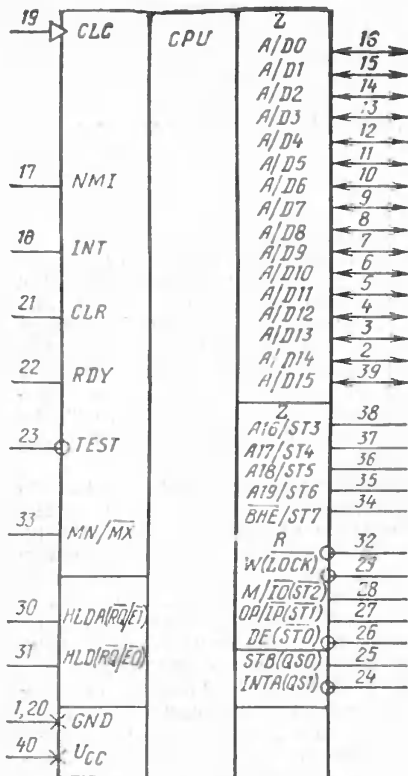


Рис. 1. Условное графическое обозначение микропроцессора

цессорной системы. Системная «настройка» микропроцессора на соответствующий режим работы обеспечивается с помощью специального вывода выбора режима конфигурации.

Условное графическое обозначение микросхемы КМ1810ВМ86 приведено на рис. 1, назначение выводов микросхемы — в табл. 1.

Структурная организация

Структурная схема микропроцессора КМ1810ВМ86 ориентирована на параллельное выполнение функций выборки и выполнения команд и состоит из устройств: сопряжения с каналом обработки и устройства управления и синхронизации (рис. 2).

Устройство сопряжения с каналом (УСК) обеспечивает функции, связанные с формированием 20-разрядного физического адреса памяти, выборкой команд и операндов из памяти, формированием очереди команд, запоминанием результатов выполнения команд в памяти. В соответствии с этим в состав УСК входят: шесть 8-разрядных регистров очереди команд; четыре 16-разрядных сегментных регистра; 16-разрядный регистр адреса (указателя) команды; 16-разрядный регистр обмена; 16-разрядный сумматор адреса,

Назначение выводов микропроцессора

Номер вывода	Обозначение вывода	Назначение вывода
1, 20	GND	Общий
2—16, 39	A/D0 ... A/D15	Канал адреса/данных
17	NMI	Немаскируемый запрос прерывания
18	INT	Маскируемый запрос прерывания
19	CLC	Тактовый импульс
21	CLR	Установка
22	RDY	Готовность
23	$\overline{\text{TEST}}$	Проверка
32	$\overline{\text{R}}$	Чтение
33	MN/MX	Минимальный (максимальный) режим
34	$\overline{\text{BHE/S17}}$	Разрешение передачи по старшей половине канала данных D15 ... D8 сигнал состояния 7
35—38	A19/ST6, A18/ST5, A17/ST4, A16/ST3	Канал адреса / сигналы состояния
40	U _{cc}	Напряжение источника питания +5 В
Минимальный режим		
24	$\overline{\text{INTA}}$	Подтверждение прерывания
25	STB	Строб адреса
26	$\overline{\text{DE}}$	Разрешение передачи данных
27	OP/IP	Выдача / прием данных
28	M/I $\overline{\text{O}}$	Память / внешнее устройство
29	$\overline{\text{W}}$	Запись
30	HLDA	Подтверждение захвата
31	HLD	Захват
Максимальный режим		
24, 25	QS1, QS0	Сигналы состояния очереди команд
23—28	$\overline{\text{ST0}}, \overline{\text{ST1}}, \overline{\text{ST2}}$	Сигналы состояния цикла канала
29	$\overline{\text{LOCK}}$	Канал занят
30, 31	$\overline{\text{RQ/E1}}$ RQ/E0	Запрос / разрешение доступа к магистрали

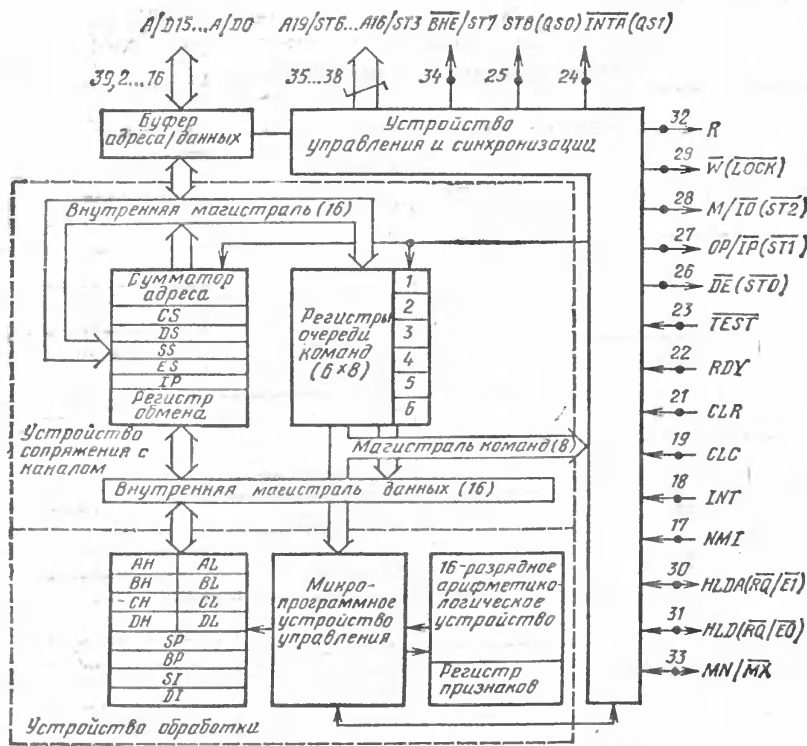


Рис. 2. Структурная схема микропроцессора

Устройство обработки (УО) предназначено для выполнения операций по обработке данных. Команды, выбранные из памяти и записанные в регистры очереди команд УСК по запросу от УО, поступают через 8-разрядную магистраль команд на микропрограммное устройство управления, которое декодирует команды и вырабатывает соответствующую последовательность микрокоманд, управляющую процессом выполнения текущей операции. УО не имеет непосредственной связи с внешней магистралью системы и обменивается данными через регистр обмена с УСК. В состав УО входят: 16-разрядное АЛУ; восемь 16-разрядных регистров общего назначения; регистр признаков состояния микропроцессора.

Для программиста программно-доступными функциональными частями микропроцессора являются регистры: общего назначения (используемые для хранения операндов и результатов выполнения команд), сегментные (используемые для хранения базовых адресов текущих сегментов памяти), адреса команд (IP) и признаков (рис. 3).

Старшие и младшие восемь разрядов группы регистров данных могут быть адресованы раздельно. В этом случае они образуют набор из восьми 8-разрядных регистров общего назначения AH, AL, BH, BL, CH, CL, DH, DL, причем регистрам AH, BH, CH, DH соответствуют старшие восемь разрядов, а регистрам AL, BL, CL, DL — младшие восемь разрядов группы регистров данных.

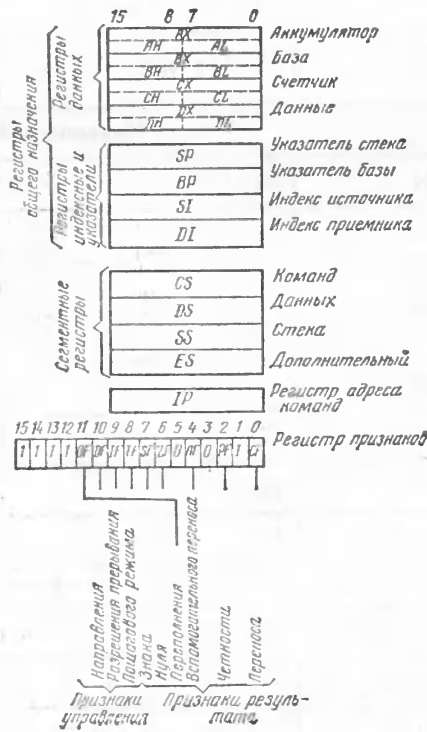


Рис. 3. Программно-доступные регистры микропроцессора

Программы, написанные для микропроцессора, «рассматривают» 1 Мбайт памяти как группу сегментов, определяемых конкретным применением. Каждому сегменту соответствует непрерывная и отдельно адресуемая область памяти. Сегменты могут следовать друг за другом непрерывно, с интервалом или могут перекрываться. Максимальное число следующих непрерывно друг за другом сегментов емкостью 16 байт равно 65536, а емкостью 65536 байт равно 16. Программный сегмент можно настроить на адрес любой ячейки и любого слова сегмента.

Микропроцессор позволяет независимо адресовать четыре программных сегмента в памяти, называемых текущим сегментом команд, текущим сегментом стека и текущим дополнительным сегментом. Сегментирование памяти совместно с позиционно-независимыми командами передачи управления позволяет создавать динамически перемещаемые программные модули.

Физически область памяти для микропроцессора организуется в два банка по 512К байт — старший банк (D15...D8) и младший банк (D7...D0). Для адресации ячеек памяти в каждом банке используются разряды A19...A1 канала адреса микропроцессора. Байт данных с четным адресом пересылается по линиям D7...D0 канала данных, а байт данных с нечетным адресом — по линиям D15...D8 канала данных. Микропроцессор вырабатывает два сигнала A0 и BHE, используемые для выбора соответствующего банка памяти.

Младший банк, содержащий четно адресуемые байты, выбирается, когда сигнал A0 соответствует «Лог. 0». Старший банк, содержащий нечетно адресуемые байты, выбирается, когда сигнал BHE соответствует «Лог. 0». Команды всегда выбираются из памяти как слова, независимо от того, по четному или нечетному адресу производится чтение команды.

Выбор режима конфигурации

Режим минимальной конфигурации (вывод MN/MX подключен к шине питания) предназначен для применения микропроцессора в простых однопроцессорных системах. В этом режиме микропроцессор непосредственно вырабатывает все необходимые сигналы управления обменом с памятью и внешними устройствами в соответствующей временной последовательности, а также обеспечивает простой доступ к системному каналу.

лу по запросу от контроллера прямого доступа к памяти, используя сигналы HL и HLDA.

Режим максимальной конфигурации (вывод MN/MX подключен к шине «корпус») предназначен для применения микропроцессора в сложно одно- и многопроцессорных системах. При работе в этом режиме структура схемы управления и синхронизации частично перестраивается и изменяются функции восьми выводов микропроцессора. Функции управления каналом выполняются схемой системного контроллера БИС KP1810BK88, которая декодирует три сигнала состояния $\overline{ST0} \dots \overline{ST2}$, поступающие из микропроцессора, и выдает расширенный набор сигналов управления обменом. Остальные пять выводов используются для координации совместной работы микропроцессора с другими процессорами в мультипроцессорных системах.

Принцип работы

Для разработчика системы действия, выполняемые микропроцессором, представляют собой последовательность циклов канала по обмену информацией с памятью или периферийными устройствами. Все типы циклов канала могут быть объединены в два базовых цикла — цикл чтения и цикл записи (рис. 4). Каждый цикл канала состоит как минимум из четырех машинных тактов T1...T4 (машинный такт равен периоду тактовой частоты). В такте T1 на канал адреса / данных всегда выдается адресная информация. В этом же такте вырабатывается сигнал STB, позволяющий идентифицировать начало цикла канала, который используется как строб для занесения адресной информации во внешний регистр адреса.

В такте T2 переключается направление работы канала адреса / данных. Данные по каналу передаются в тактах T3 и T4. Длительность цикла канала можно удлинить с помощью управляющего сигнала RDY, позволяющего синхронизировать скорость работы внешней памяти со скоростью работы микропроцессора введением в цикл канала между тактами T3 и T4 дополнительных тактов ожидания $T_{ож}$. В течение тактов ожидания данные на канале остаются неизменными. Между тактом T4 текущего цикла и тактом T1 следующего цикла канала процессор может вводить дополнительные «холостые» такты, предназначенные для выполнения внутренних действий. Моменты введения этих тактов и их число зависят от состояния очереди команд и выполняемой команды в УО.

Прерывание. В соответствии с требованиями областей применения

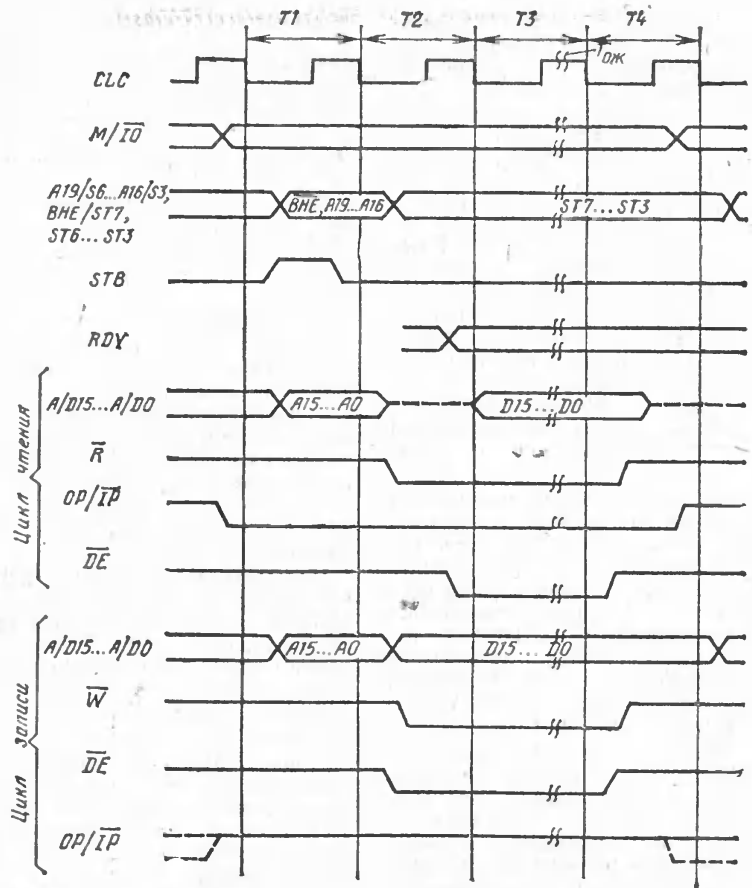


Рис. 4. Диаграмма базового цикла канала для минимального режима

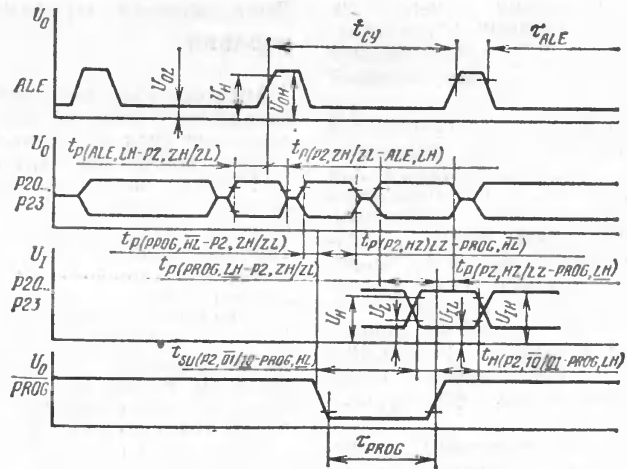


Рис. 5. Структура прерываний микропроцессора

микропроцессор обеспечивает возможность реакции на внешние события (что является одним из главных атрибутов операционных систем реального времени) и на различные особые ситуации, возникающие в процессе выполнения команд, используя

векторную структуру прерывания. Он позволяет обрабатывать 256 типов прерываний с номерами 0...255, которые делятся на внешние аппаратные, внутренние аппаратные и программные, организованные по приоритетному принципу (рис. 5).

По любому прерыванию управление передается программе (процедуре) обслуживания прерывания посредством вектора прерывания, выбираемого из таблицы векторов прерывания, располагаемой в области памяти по адресам 0000₁₆ до 003 FE₁₆. Таблица векторов связывает номер прерывания и соответствующую процедуру, предназначенную для обработки запроса прерывания. Каждый вектор представляет собой два 16-разрядных слова, располагаемых в памяти в соседних ячейках, причем в ячейке памяти с меньшим адресом (четным) содержится смещение (новое значение программного счетчика), а в ячейке с большим адресом — база сегмента логического адреса процедуры обслуживания прерывания.

Запросы на внешние прерывания воспринимаются и обрабатываются микропроцессором, как правило, после выполнения текущей команды. Однако для обеспечения защиты программы от ошибок при выполнении пары последовательных команд, изменяющих компоненты логического адреса (смещение и базу сегмента), прерывания обрабатываются только после выполнения обеих команд.

Запросы на маскируемые прерывания от внешних устройств должны поступать на входы БИС программируемого контроллера прерываний KP1810BH59A, который формирует номер вектора прерывания и общий сигнал запроса прерывания INT, поступающий на одноименный вход микропроцессора. Обработка маскируемого прерывания в микропроцессоре зависит от значения признака разрешения прерывания IF, управление которым осуществляется командами STI, разрешающей прерывание, и CLI, запрещающей прерывание.

Немаскируемый запрос прерывания поступает по входу NMI микропроцессора и обычно используется для прерывания работы микропроцессора при «катастрофических» событиях, таких как пропадание питания, обнаружение ошибки памяти и др. Обработка немаскируемого запроса прерывания, внутренних аппаратных и программных прерываний не зависит от значения признака разрешения прерывания, и соответствующий номер вектора прерывания формируется внутренне.

К внутренним аппаратным прерываниям относятся прерывание по ошибке деления и пошаговые прерывания.

Прерывание по ошибке деления происходит во время выполнения команд деления (DIV, IDIV), если результат деления превышает максимально допустимую величину или при делении на нуль.

Пошаговые прерывания позволяют программисту просто и эффективно

организовать покомандное выполнение программы с выводом содержимого всех внутренних регистров микропроцессора после выполнения каждой команды программы. Выработка пошагового прерывания зависит от состояния признака пошагового режима TF.

Программные прерывания делятся на прерывания по точкам разрыва, прерывания, определяемые пользователем, и прерывания по переполнению.

Прерывание по точкам разрыва генерируется по однобайтной команде INT3. Точка разрыва в общем случае — это любое место в программе, где нормальное ее выполнение приостанавливается для выполнения некоторых специальных действий. Обычно данное прерывание используется в программных средствах отладки для распечатки содержимого ячеек памяти, регистров и т. д.

Прерывание по переполнению генерируется по однобайтной команде INTO и зависит от значения признака переполнения OF. Данное прерывание используется обычно для обнаружения переполнений в случае выполнения операций над целыми числами.

В прерываниях, определяемых пользователем, номер прерывания задается вторым байтом кода команды INT. Данный вид прерывания может быть использован для вызова системных процедур и (или) процедур пользователя в любом месте программы для увеличения эффективности работы программных средств.

Возможности мультипроцессорования

Архитектурная особенность микропроцессора — наличие аппаратно-программных средств, позволяющих упростить построение мультипроцессорных систем на его основе. В таких системах можно использовать процессоры двух типов: независимые (выполняющие собственные потоки команд) и сопроцессоры (специализированные процессоры, аппаратно реализующие команды сложных процедур и выбирающие «свои» команды из общего, с центральным процессором потока команд). Аппаратно-программные средства мультипроцессорования микропроцессора обеспечивают решение проблемы координации работы мультипроцессорной системы в соответствии с ее определением [1]: требование разделения ресурсов (арбитраж канала); взаимные исключения (синхронизация процессов).

Разделение общих ресурсов во времени между процессорами выполняется с помощью двух двунаправленных линий «запрос-разрешение до-

ступа к магистрали» микропроцессора, организованных по приоритетному принципу; с помощью внешней схемы арбитра системной шины БИС KP1810BH59 или их комбинацией, если процессоры имеют доступ к нескольким совместимым каналам. В любом случае механизм разделения каналов (арбитража) функционирует независимо от программного обеспечения.

Существует несколько приемов, используемых для взаимного исключения [2]: блокировка памяти; операция «проверка и установка»; «семафоры». Требуемое взаимное исключение методом блокировки памяти осуществляется с помощью однобайтного префикса LOCK и выходного сигнала LOCK (блокировка), управляемого этим префиксом. Префикс LOCK позволяет обеспечить блокировку доступа к памяти на все время выполнения команды, следующей за ним. Однако сигнал LOCK является лишь информационным, а принятие решения о блокировке обращений к памяти возлагается на арбитра системного канала.

Эффективный метод синхронизации процессов — метод семафоров [3, 4] (семафор — это переменная, указывающая, что ресурс-буфер или регистр-указатель доступен или занят), который в микропроцессоре реализуется комбинацией префикса LOCK с командой обмена XCHG.

При работе в максимальном режиме предусмотрены аппаратные средства для подключения сопроцессоров, обеспечивающих выполнение специализированных операций и расширяющих систему команд микропроцессора. Специальная команда ESC микропроцессора инициирует выполнение в сопроцессоре 64 команд, выбираемых из общего с универсальным микропроцессором потока команд. При размещении перед каждой сопроцессорной командой однобайтной команды WAIT обеспечивается синхронизация параллельного выполнения операций в сопроцессоре с операциями в микропроцессоре. Аналогичные методы расширения систем команд широко используются в настоящее время в универсальных больших высокопроизводительных вычислительных машинах.

Система команд

Система команд микропроцессора содержит 135 базовых команд, которые могут быть разделены на команды пересылки данных, арифметические, поразрядной обработки данных, обработки строк, передачи управления, управления процессором.

Команды пересылки данных предназначены для пересылки операнда источника на место операнда-прием-

ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ НА БАЗЕ МИКРОЭВМ «ИСКРА226»



Комплекс из рентгеновского электронного микроскопа и микроЭВМ «Искра 226», количественно анализируя электронно-микроскопическую информацию, позволяет получать принципиально новую информацию о внутренней трехмерной структуре исследуемых объектов на микронном и субмикронном уровнях без разрушения этих объектов.

Аппаратура микрофотографии на базе микроЭВМ «Искра 226» (фото слева)

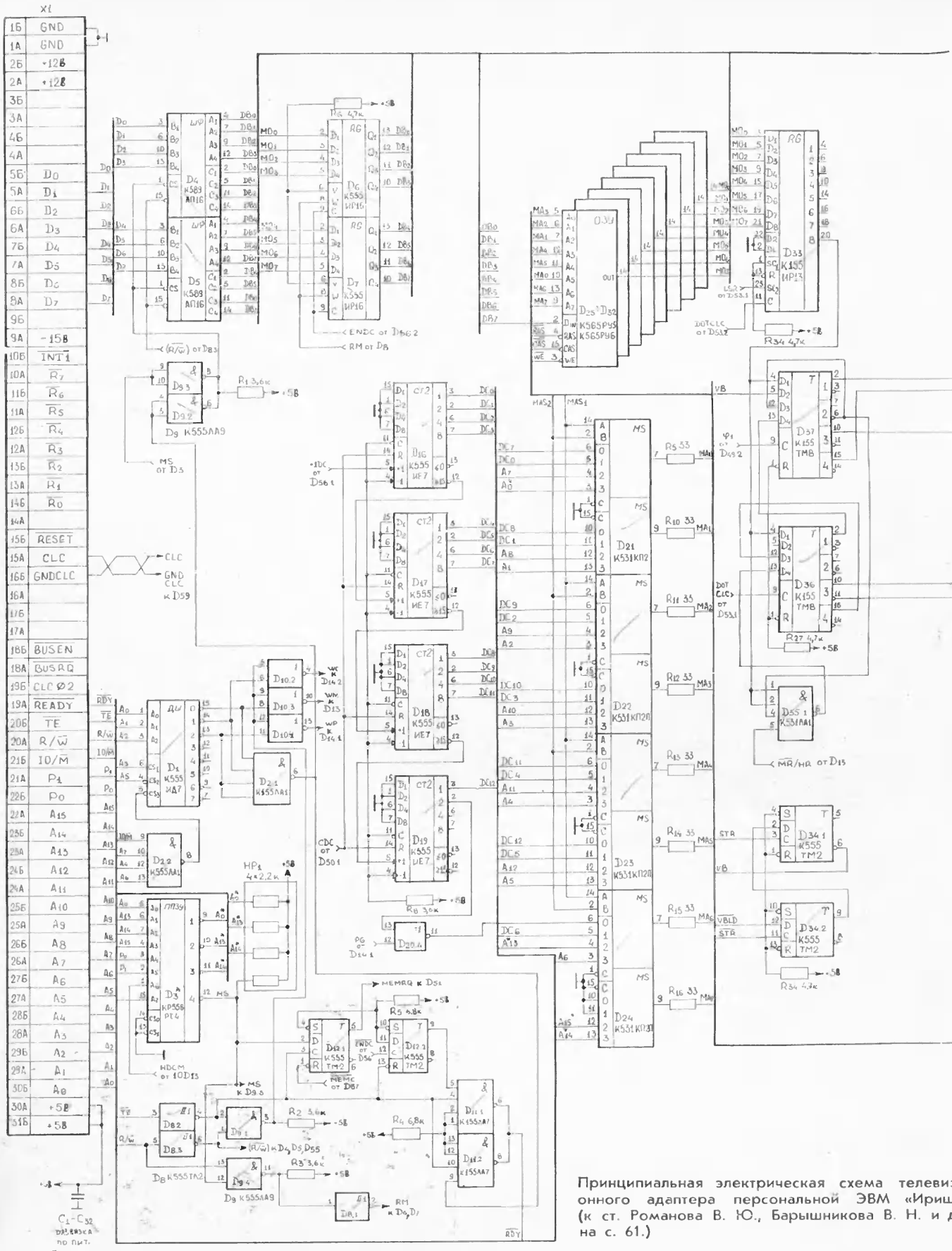
Система количественного анализа изображений, получаемых с оптического микроскопа (фото справа, вверху)

Система анализа изображений, получаемых со сканирующего электронного микроскопа (фото справа, в центре)

Электронный микроскоп РЭМ-100У и система обработки изображений на базе микроЭВМ «Искра 226» (фото справа, внизу)

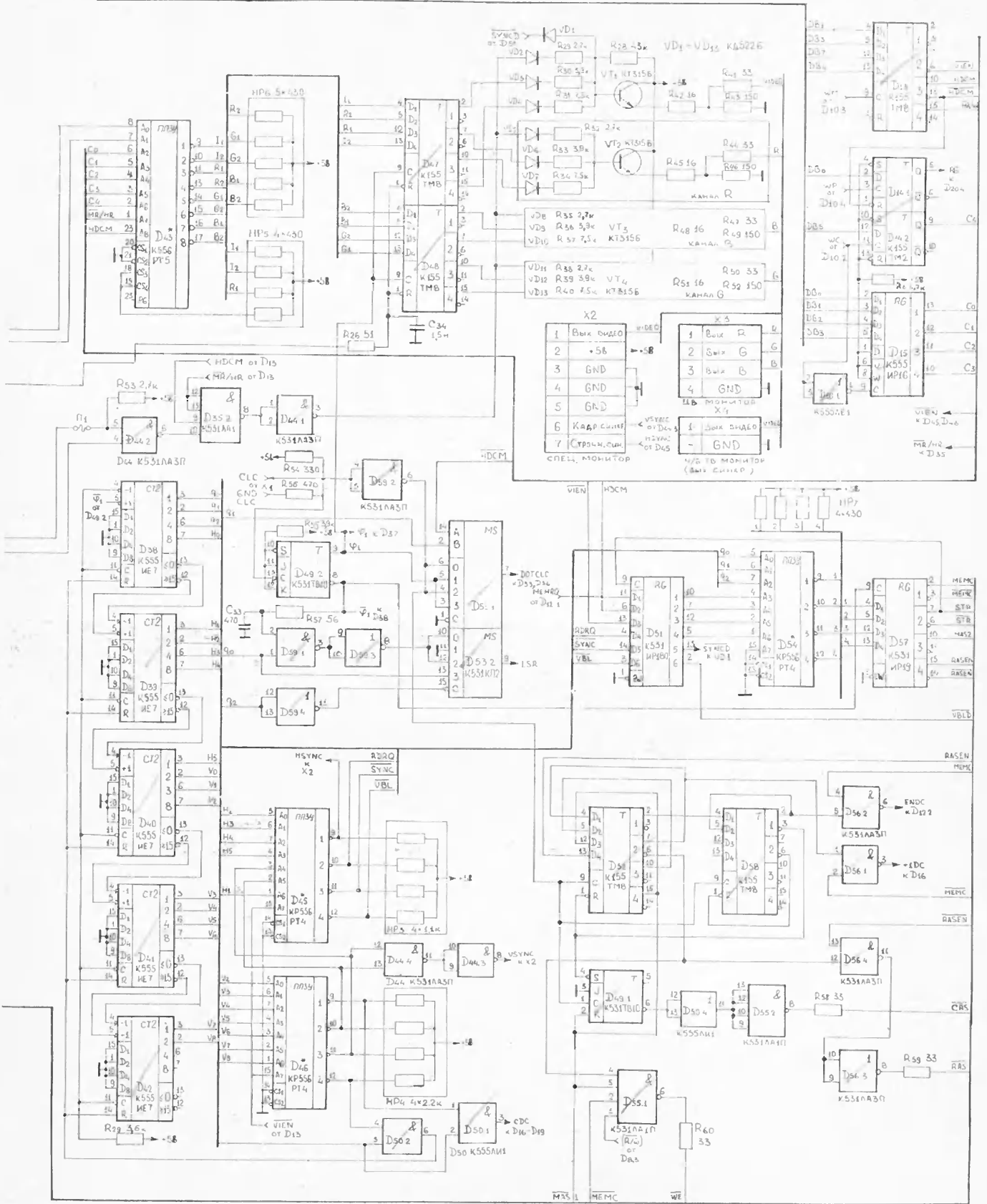
(См. статью Сасова А. Ю. на с. 55.)





Принципиальная электрическая схема телевизионного адаптера персональной ЭВМ «Ириша» (к ст. Романова В. Ю., Барышника В. Н. и др. на с. 61.)

С1 - С32
 D1 - D32
 по п. 1.
 Сг - 4 x 47 мФ



Персональная ЭВМ ПК 8001

Выполнена на одной плате
Размещается в корпусе клавиатуры

Содержит модули:

■ процессор на базе микро-
процессора КР580ИК80А

■ ПЗУ емкостью 16 или 64К
байт с интерпретатором языка
Бейсик

■ ОЗУ емкостью 16 или 64К
байт

■ интерфейсы бытового маг-
нитофона, клавиатуры, НГМД,
локальной сети, параллельный
и последовательный интерфей-
сы

■ контроллеры цветного и ал-
фавитно-цифрового графиче-
ского дисплеев

■ таймер

■ контроллер прерываний

■ два двухканальных АЦП
ПЭВМ ПК8001 может использо-
ваться в качестве интеллекту-
альной клавиатуры, терминала,
бытовой, школьной или про-
фессиональной ЭВМ, графиче-
ской станции. [к ст. Велихова
Е. П., Персианцева И. Г. и др.
на с. 34.]



ника. Существует 14 команд пересылки данных, которые делятся на четыре группы: команды пересылки данных общего назначения, ввода-вывода, пересылки логического адреса и пересылки признаков. В группу команд пересылки данных общего назначения включена команда XLAT, используемая для перекодировки символов одного кода в другой.

Арифметические команды обеспечивают выполнение основных арифметических операций (сложение, вычитание, умножение и деление) над 8- и 16-разрядными порядковыми и целыми двоичными числами, упакованными и распакованными двоично-десятичными числами. В эту группу включены также команды преобразования форматов данных — двоичного числа в двоично-десятичное и наоборот; преобразование байта в слово, слова в двойное слово.

Команды **поразрядной обработки данных** предназначены для выполнения логических операций булевой алгебры, операций линейных (арифметических и логических) и циклических сдвигов вправо и влево на один или n -разрядов.

Команды **обработки строк** обеспечивают обработку (пересылку, сравнение, запись в память, загрузку в аккумулятор) одного элемента строки за одно обращение к памяти. Эти команды совместно с префиксом повторения позволяют организовать аппаратные циклы для обработки строк длиной до 64К байт, выполняемых намного быстрее, чем с помощью программных циклов.

Команды **передачи управления** обеспечивают организацию переходов в программе и разделены на 4 группы: безусловные переходы, вызовы и возвраты; условные переходы; команды управления циклами и команды прерывания. Предусмотрено две разновидности команд передачи управления: обеспечивающие передачу управления в текущем сегменте команд (внутрисегментный переход) и передачу управления произвольному сегменту команд, который после этого становится текущим (межсегментный переход). При этом адрес перехода может задаваться как прямо, так и косвенно с использованием стандартных режимов адресации.

Внутрисегментные прямые вызовы и переходы осуществляются относительно содержимого программного счетчика, что позволяет программисту организовать позиционно-независимые программы.

Команды условного перехода обеспечивают ветвление в программах в соответствии с состоянием признаков в пределах 256-байтной области относительно программного счетчика.

Команды управления циклами позволяют эффективно выполнять многократно повторяющиеся циклы в программах в пределах 256 байт относи-

тельно программного счетчика. Выход из цикла возможен: содержимое счетчика циклов равно нулю или по условию.

Команды **управления процессором** обеспечивают управление состоянием определенных разрядов регистра признаков, а также используются для синхронизации работы процессора с внешними событиями и процессорами.

Адресация операндов в памяти. Команды микропроцессора обеспечивают выполнение операций над одним или двумя операндами. Результат операции записывается по адресу любого из операндов. В зависимости от типа команды операнды можно расположить в программно-доступных регистрах, непосредственно в коде команды, в памяти и регистрах ввода-вывода. Непосредственные данные могут быть типа байта или слова; операнды в программно-доступных регистрах — типа байта или слова, а для команд умножения и деления — типа двойного слова; операнды в памяти — типа байта, слова, двойного слова, а в регистрах ввода-вывода — типа байта и слова.

Для указания операнда в программно-доступных регистрах используется регистровая и неявная регистровая адресация. При регистровой адресации номер 8- или 16-разрядного регистра, содержащего операнд, указывается в поле кода команды. При регистровой неявной адресации номер регистра определяется неявно самим кодом команды. Указание операнда в коде команды осуществляется непосредственной адресацией.

Для адресации операндов в памяти используется прямая, косвенная регистровая, стековая, строковая и табличная адресация. Адрес операнда в памяти состоит из двух 16-разрядных компонент: базы сегмента и смещения относительно базы сегмента, называемого исполнительным адресом.

База сегмента хранится в сегментном регистре. Исполнительный адрес — расстояние в байтах от начала сегмента, содержащего операнд, до операнда и интерпретируется как порядковое число в пределах 0...65536. Составными частями при вычислении исполнительного адреса операнда являются: смещения в команде (8- или 16-разрядное), содержимое базового регистра ВХ или ВР, содержимое индексного регистра SI или DI.

При **прямой адресации** исполнительным адресом операнда является смещение в команде.

При **косвенной регистровой адресации** способ вычисления исполнительного адреса операнда задается в коде команды. Имеются четыре типа косвенной регистровой адресации: через содержимое базового или ин-

дексного регистра ВХ, ВР, SI, DI; через сумму содержимого базового или индексного регистра ВР/ВХ/SI/DI и смещение в команде; через сумму содержимого базового ВР/ВХ и индексного SI/DI регистров; через сумму содержимого базового ВР/ВХ, индексного SI/DI регистров и смещения в команде.

При **стековой адресации** исполнительный адрес операнда находится в регистре указателя стека SP, причем после чтения операнда из стека содержимое регистра увеличивается на два, а перед записью операнда в стек содержимое регистра уменьшается на два.

При **строковой адресации** исполнительный адрес операнда находится в регистре индекса-источника SI и (или) регистре индекса-приемника DI. После обращения к операнду содержимое указанного регистра увеличивается или уменьшается в зависимости от значения признака направления DF. Такая адресация используется только в командах обработки элементов строк данных.

При **табличной адресации** исполнительный адрес операнда определяется как сумма содержимого регистров ВХ и AL, причем после обращения к операнду содержимое регистра AL изменяется на содержимое памяти по исполнительному адресу. Данный вид адресации используется только в команде XLAT.

Рассмотренные особенности микропроцессора KM1810BM86 позволяют в сочетании с БИС базового комплекта серии K1810, контроллерами и периферийными БИС серии KP580 реализовать широкий спектр средств вычислительной техники: от одноплантных высокопроизводительных контроллеров до иерархических мультипроцессорных систем.

ЛИТЕРАТУРА

1. Энслоу Ф. Г. Мультипроцессорные системы и параллельные вычисления. — М.: Мир, 1976. — 383 с.
2. Цикритзис, Беристай Ф. Операционные системы. — М.: Мир, 1977. — 336 с.
3. Шоу А. Логическое проектирование операционных систем. — М.: Мир, 1981. — 360 с.
4. Архитектура многопроцессорных вычислительных систем / Под ред. В. И. Тимохина. — Л.: ЛГУ, 1981. — 103 с.

Статья поступила 26 ноября 1985 г.

УДК 681.322.1—181.5

Е. П. Велихов, И. Г. Персианцев, А. Т. Рахимов, Н. Н. Рой, А. В. Скурихин, О. А. Щербаков

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР В СИСТЕМЕ АВТОМАТИЗАЦИИ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА

Современное состояние физических исследований, в частности в области физики плазмы, предъявляет столь высокие требования к экспериментальным установкам, что становится невозможным обойтись без их комплексной автоматизации. Для решения этой задачи в отделе физики плазмы Института ядерной физики МГУ создан трехуровневый комплекс средств автоматизации управления физическим экспериментом. Комплекс представляет собой наращиваемую иерархическую структуру с распределенным интеллектом и состоит из следующих устройств:

микроЭВМ ПК 8001, которая может быть отнесена к классу ПЭВМ, обладающая развитыми средствами передачи, обработки и отображения информации;

встраиваемые одноплатные ЭВМ для управления отдельными, достаточно крупными блоками установки, такими как газовый тракт, высоковольтные энергетические установки и т. д.;

интеллектуальные исполнительные устройства, управляющие конкретными узлами установки и контролирующие их работу.

Составные части комплекса объединяются приборной локальной сетью. Такая управляющая структура может расширяться как внутри указанных трех уровней иерархии, так и наращиванием иерархических уровней сверху с привлечением вычислительных машин, относящихся к более высоким классам.

Основу управляющего комплекса составляет ПЭВМ, главные конструктивные и функциональные характеристики которой и будут рассмотрены ниже. Конструктивно ПЭВМ выполнена на одной двусторонней плате, размещенной в корпусе клавиатуры, что обеспечивает существенно более высокую надежность и технологичность по сравнению с многоплатными конструкциями. Компоновка платы позволяет просто, без доработок конструировать машину в трех возможных вариантах. Мини-

мальная конфигурация ПЭВМ занимает центральную часть полноформатной платы и включает следующие модули.

Процессор на базе микропроцессора КР5801К80А, работающего с тактовой частотой 2,5 МГц. При выборе микропроцессора для данной машины в первую очередь учитывалась возможность реализации программной совместимости с самыми популярными зарубежными персональными компьютерами.

ПЗУ объемом 16 или 64К байт, содержащее интерпретатор языка Бейсик, написанный специально для данной машины. Интерпретатор с точки зрения пользователя является достаточно полным подмножеством языка Бейсик фирмы Микрософт для таких, например, компьютеров, как IBM PC, TRS-80, и удовлетворяет требованиям стандарта MSX. Скорость выполнения программ на ЭВМ ПК 8001 меньше скорости выполнения тех же программ на ЭВМ IBM PC/XT всего на 25%. При включении минимальная версия ПЭВМ выходит непосредственно в Бейсик, а периферийными устройствами служат бытовой магнитофон и телевизор.

Интерфейс бытового магнитофона задает две скорости записи информации: 500 и 2400 бод. Скорость 500 бод обеспечивает совместимость ПК 8001 на уровне носителя с персональным компьютером TRS-80, где она является стандартной. В ПК 8001 предусмотрена, кроме того, предварительная цифровая статистическая обработка входного аналогового сигнала. Рабочая скорость 2400 бод позволяет загрузить весь объем памяти ПЭВМ приблизительно за три минуты. Режим блочной записи с коррекцией ошибок гарантирует вероятность сбоя не более 10^{-9} , т. е. одна ошибка на 100 Мбайт, что позволяет в полной версии машины использовать бытовой магнитофон в качестве стриммера электронного эмулятора накопителя на гибких магнитных дисках. Каждый блок данных в используемом

формате включает системную информацию о принадлежности блока к конкретному файлу и расположении его внутри файла. Это дает возможность визуализировать поиск необходимых данных на ленте, отображая принадлежность текущей записи на экране дисплея в реальном масштабе времени, а также, в случае сбоя, повторять считывание только необходимого блока. На одну кассету МК-60 при скорости 500 бод можно записать 200К байт, а при скорости 2400 бод — 1 Мбайт информации. Интерфейс магнитофона включает также реле, с помощью которого программно осуществляется пуск и останов мотора магнитофона через вход «кратковременный стоп», что поддерживается стандартным программным обеспечением.

Интерфейс алфавитно-цифрового монитора генерирует стандартный композитный видеосигнал, что позволяет использовать в качестве дисплея бытовой телевизор. На экране отображаются 64 символа в 16 строках текста и одновременно 128×48 графических блоков. Предусмотрен также режим отображения тех же символов, но более крупного размера (32 символа в 16 строках и 64×48 графических блоков). Знакогенератор интерфейса реализован аппаратно, что обеспечивает достаточно высокую скорость обработки текстовой и графической информации и дает возможность отображать 256 различных символов в матрице 8×16 точек. В числе этих символов прописные и строчные буквы русского и латинского алфавитов и графические символы. Таким образом, графика низкого разрешения является достаточно быстрой (для заполнения всего экрана достаточно 10 мс). Любой из символов может отображаться инверсно (темный на светлом фоне). Предусмотрены возможность инверсии неиспользуемой части телевизионного раstra и аппаратная поддержка гашения всего раstra. Гашение происходит в том случае, когда к включенной машине не обращаются дольше заданного интервала времени (например, 10 мин). Для восстановления изображения достаточно нажать на любую клавишу.

Интерфейс клавиатуры обслуживает клавиатуру, которая представляет собой матрицу из замыкателей любого типа. Устранение дребезга контактов, возможность нажатия следующей клавиши при неотпущенной предыдущей и другие сервисные функции реализованы программно. Можно использовать клавиатуру из 64 или 128 клавиш. В полноформатной клавиатуре (128 клавиш) есть отдельное поле управления курсором, цифровое поле и поле программируемых клавиш. Программа обеспечивает ввод букв как русского, так и латинского алфавитов. Расположение клавиш

может соответствовать стандартной клавиатуре отечественных терминалов или пишущей машинки с русским или латинским шрифтами. В составе клавиатуры имеется также пьезо-керамический излучатель звука. Высота тона и длительность звучания задаются программно.

Параллельный интерфейс позволяет подключить к ПЭВМ 8-разрядные периферийные устройства, такие, например, как графопостроитель, печатающее устройство, программаторы ПЗУ и ПЛМ.

Интерфейс локальной сети обеспечивает связь ПЭВМ с аналогичными компьютерами или периферийными устройствами для организации управляющих, информационных и обучающих систем.

Помимо перечисленных устройств на плате имеется разъем, на который выведены все важные сигналы ПЭВМ. Через него можно подключать, в частности, отладочный модуль, реализующий пошаговый режим и другие сервисные функции.

Оперативная память ПЭВМ может составлять в зависимости от типа используемых микросхем либо 64К байт (полная конфигурация), либо 16К байт (минимальная конфигурация). Примечательно, что после включения происходит самотестирование основных систем и определение объема работоспособной оперативной памяти, которая может быть целиком занята программой пользователя, так как остальные подсистемы ПЭВМ имеют собственные банки памяти (полный объем памяти на плате может достигать 325К байт).

Описанные устройства составляют минимальную конфигурацию ПЭВМ. Минимальное центральное поле платы можно дополнить участком, расположенным справа от него и реализующим контроллер графического дисплея с разрешением 512×256 точек в 16 возможных цветах или градациях яркости. При этом становится возможным отображать на экране 85 символов в 25 строках с программируемыми шрифтами в матрице 6×10 точек. Приняты определенные меры, ускоряющие обработку цветной графической информации более чем на порядок по сравнению с традиционным подходом. К ним относятся возможности обращения к дисплею во время прямого хода развертки и аппаратная поддержка алгоритмов работы с цветом, основанная на применении ПЛМ. На выходе блока цветной графики размещена просмотревая таблица, входными сигналами которой являются сигналы трех цветовых плоскостей и сигнал алфавитно-цифрового монитора, поступающий из центральной части платы.

Такая конфигурация ПЭВМ позволяет работать как с двумя отдельными мониторами

(алфавитно-цифровым и графическим), так и объединять эти две системы на одном мониторе. При этом просмотрная таблица обеспечивает произвольное переопределение цветов всех источников сигнала, в том числе их выключенные. С выхода просмотрной таблицы TTL-сигнал подается на цветной монитор, который может иметь вход управления яркостью, а также на цифроаналоговый преобразователь, генерирующий монохромный композитный видеосигнал с 16 возможными градациями яркости. Схемотехника графических банков памяти позволяет без каких-либо аппаратных переключений при полном сохранении всех графических возможностей реализовать электронный «квазидиск» емкостью 144К байт. Это достигается путем простой замены микросхем емкостью 16К байт микросхемами емкостью 64К байт. В такой конфигурации ПЭВМ может работать с дисковой операционной системой, загружаемой с магнитофона.

Кроме того, минимальное поле платы можно дополнить участком, расположенным слева от центрального и содержащим следующие устройства:

параллельный интерфейс, расширяющий 8-разрядный порт минимальной конфигурации до 16-разрядного с сохранением совместимости по разъему;

последовательный интерфейс с опторазвязкой, обеспечивающий работу в стандартах «20 мА токовая петля» (активная или пассивная), и RS-232C. Выбор стандарта осуществляется установкой перемычек на плате;

трехканальный таймер. Первый канал таймера генерирует тактовые частоты, задающие

скорость обмена информацией по последовательному интерфейсу. Второй — используется для аппаратной поддержки работы излучателя звука. Третий — задает прерывания центрального процессора с периодом 0,01 с, что позволяет реализовать в ПЭВМ часы реального времени с соответствующим временным разрешением и многозадачный режим работы;

контроллер прерываний, управляющий режимами обслуживания всех периферийных устройств и систем. Источниками запросов на прерывания являются локальная сеть, контроллер НГМД, таймер, приемник и передатчик последовательного интерфейса и устройства, подключаемые к расширительному разъему;

два простейших двухканальных аналого-цифровых преобразователя, предназначенных для подключения двух внешних устройств управления курсором типа «джойстик»;

интерфейс НГМД, позволяющий работать с 8-, 5-, 3,5-дюймовыми одно- или двусторонними дисками с одинарной или двойной плотностью записи.

Интерфейс гибкого диска дает возможность работать с поставленной на ПЭВМ дисковой операционной системой, полностью соответствующей операционной системе CP/M-80. ПЭВМ ПК 8001 в зависимости от числа установленных в ней микросхем может использоваться в качестве интеллектуальной клавиатуры, терминала, бытового, школьного или профессионального компьютера, графической станции или управляющей микроЭВМ.

Статья поступила 21 августа 1985 г.

РЖ АВТ-85

7Б355. МикроЭВМ, выполненная в виде интегральной схемы совместно с операционной системой. Microcomputer and CP/M on one chip. „Electron. Eng.” (Gr. Brit.), 1985, 57, № 697, 13 (англ.)

В настоящее время микроЭВМ типа Z80, обеспечиваемая операционной системой CP/M, выполняется на отдельной ИС, что стало возможно благодаря внедрению процессора операционной системы типа S83. Устройство S83 является модернизированным вариантом микроЭВМ типа AMI. Фирмы AMI и Zilog осуществили совместную разработку системы, состоящей из 8-разрядного процессора микроЭВМ типа Z80 и процессора S83. Фирма AMI использовала процессор фирмы Zilog в качестве дополнительной микроячейки. В устройстве применяется ПЗУ на 8К байт для дешифрируемой информации, а

также динамическое ЗУ произвольного доступа.

РЖ АВТ-85

9Б424. Система шин и конструкция корпусов для микроЭВМ на основе 32-разрядных МП. Gehäuse und Aufbau-systeme für 32-bit-Mikroprozessor-Anwendungen. 1985, 93, № 3.

Приводится краткое описание системных шин VME и Multibus II, применяемых для создания систем на основе 32-разрядных МП. Обе шины имеют структуру, не зависящую от типа МП, и могут быть использованы для систем с 8-, 16- и 32-разрядными МП. В отличие от VME шина Multibus II имеет мультиплексную структуру и работает в синхронном режиме. Для проектирования ВС применяются следующие размеры плат: VME 100×160 мм и 233,4×160 мм с 96-контактными разъемами; Multibus — 100×220 мм и 233,4×

×220 мм также с 96-контактными разъемами.

Приводится описание конструкций корпуса аппаратных средств для построения ВС на основе шин VME и Multibus плат стандартных размеров и аппаратных модулей, производимых фирмой Schroff GmbH (ФРГ).

РЖ АВТ-85

7Б356. Области применения персональных ЭВМ. Тахара Фумио. „Comput. Rept.”, 1984, 24, № 11, 18—22 (яп.).

Рассматриваются настоящие и будущие области применения персональных ЭВМ. Анализируются требования пользователей к персональным ЭВМ. В настоящее время выдвигаются главные функции ПЭВМ: упрощенный доступ к БД разного уровня и разного профиля и упрощение подключения ЭВМ к центру обработки данных.

УДК 681.3.02

А. Г. Алексенко, А. В. Гапопенко, А. Д. Иванников, И. Д. Курилов

РАЗРАБОТКА И ОТЛАДКА МИКРОПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЦИФРОВЫХ СИСТЕМ НА ОСНОВЕ СЕКЦИОНИРОВАННЫХ МИКРОПРОЦЕССОРОВ

При высоких требованиях к производительности специализированных микропроцессорных систем используются комплекты микропрограммируемых секционированных БИС. Высокое быстродействие таких комплектов, возможность построения обрабатывающих блоков любой разрядности, гибкость микропрограммного управления позволяют создать цифровые системы с характеристиками, лучшими, чем у систем на основе однокристалльных микропроцессоров. Однако разработка и отладка систем на основе микропрограммируемых процессорных секций весьма сложна. Универсальный микроассемблер ГНОМ (Генератор Объектного Микрокода) и система СЛОТ (Система Логической ОТладки) позволяют разрабатывать и отлаживать микропрограммное обеспечение для систем на основе секционированных микропрограммируемых БИС с любым форматом микрокоманд. Микроассемблер ГНОМ и система СЛОТ реализованы на мини-ЭВМ СМ-4 в операционных системах РАФОС (ГНОМ) и ДОС КП (ГНОМ и СЛОТ) на алгоритмическом языке ФОРТРАН.

Микроассемблер ГНОМ написан с перекрытиями. Суммарный объем всех хранящихся на диске программных модулей вместе с таблицами составляет 88К байт. Одновременно в оперативной памяти могут находиться программные модули и таблицы объемом не более 24К байт. Микроассемблер ГНОМ может быть использован для разработки микропрограмм в том случае, если диапазон адресации не превышает 2^n , где n — разрядность слова микрокоманды, которая, в свою очередь, не может быть более 128.

Общий объем программ системы СЛОТ без учета моделей БИС составляет около 130К байт. Объем библиотеки моделей ИС зависит от сложности последних и количества типов используемых ИС. Так, примерный объем библиотек моделей ИС серий К1804 и К155 составляет около 300К байт. Система СЛОТ функционирует на мини ЭВМ как два взаимно-

действующих задания и требует для своей работы 112К байт ОЗУ сверх памяти, занимаемой резидентными средствами ДОС КП (около 96К байт).

Универсальный микроассемблер

На первом этапе работы микроассемблер настраивается на используемые форматы микрокоманд, на втором этапе символические микропрограммы транслируются в объектные.

Настройка проводится с помощью следующих операторов. Оператор LENGTH K, где K — целое число без знака, указывает количество двончных разрядов в слове микрокоманды. Операторы FORMAT и FIELD описывают поля и субполя в микрокоманде и имеют вид:

FORMAT <ИМЯ ФОРМАТА> ИМЯ₁ V₁ [V₂]₁, ..., ИМЯ_n V_n [V₂]_n;

FIELD <ИМЯ ПОЛЯ> ИМЯ₁ V₁ [V₂]₁, ..., ИМЯ_n V_n [V₂]_n;

где ИМЯ_i — имя i -го поля или субполя; V₁, V₂ — выражения, задающие количество разрядов в i -м поле или субполе и его значение по умолчанию.

Для присвоения числовым значениям полей или субполей мнемонических обозначений используется оператор

ASSIGN <ИМЯ ПОЛЯ ИЛИ СУБПОЛЯ> МНЕМ₁=V₁, ..., МНЕМ_k=V_k;

где МНЕМ_i ($i=1, \dots, k$) — мнемоническое обозначение; V_i ($i=1, \dots, k$) — выражения, значения которых присваиваются указанным мнемоникам.

Если мнемоническое обозначение присваивается числовой величине, использование которой не связано с определенным полем или субполем, то имя поля или субполя опускается. Мнемонические обозначения, определенные в операторе ASSIGN, сохраняют свои значения и на этапе трансляции символических микропрограмм. Оператор SET ИД₁=V₁, ..., ИД_l=V_l позволяет присвоить идентификаторам ИД_i ($i=1, \dots, l$) значения выражений

V_1 ($i=1, \dots, 1$), причем эти значения внутри секции настройки можно переопределять. На этапе трансляции они не сохраняются.

Для установления связей между значениями полей или субполей используется оператор IF <ИМЯ ФОРМАТА> ИМЯ₁= V_1 ИМЯ₂= V_2 . Этот оператор указывает, что поле ИМЯ₂ принимает значение выражения V_2 всякий раз, когда в поле ИМЯ₁ записывается значение выражения V_1 . Последним в секции настройки записывается оператор END.

Все имена, мнемоники и идентификаторы состоят из русских и латинских букв, цифр и знаков подчеркивания, причем первым символом может быть буква или символы ?, @, а общее количество символов не должно быть больше 80. Присвоенные имена, мнемоники и идентификаторы заносятся во внутренние таблицы, которые на этапе трансляции располагаются в ОЗУ ЭВМ. В секции настройки может быть задано: не более 20 форматов, причем общее количество символов в именах форматов должно быть не более 200; не более 600 различных полей и субполей, причем общее количество символов в именах полей и субполей должно быть не более 10 тысяч; не более 200 мнемоник, связанных с конкретными полями или субполями, причем общее количество символов в таких мнемониках должно быть не более 5 тысяч; не более 50 мнемоник, не связанных с каким-либо полем или субполем, причем общее количество символов в таких мнемониках должно быть не более 1 тысячи.

При вычислении выражений действия выполняются с целыми числами без знака, причем предполагается, что их разрядность равна разрядности слова микрокоманды. Язык настройки имеет свободный формат, операторы отделяются друг от друга точкой с запятой. В любом месте, где возможен пробел, допускается запись комментариев, отделяемых с двух сторон знаками /* и */.

Трансляция символических микрокоманд в объектные осуществляется на втором этапе работы микроассемблера. При этом также используется ряд псевдокоманд.

Полная запись символической микрокоманды имеет вид:

МЕТКА: АДРЕС: <ИФ> ИП₁= V_1 , ИП₂= V_2 , ..., МО₁, МО₂, ..., [ИС],

где

МЕТКА — необязательный символический адрес микрокоманды;

АДРЕС — абсолютный адрес микрокоманды. Необязателен, но если он указан, то микрокоманда размещается по данному адресу, а его значение присваивается метке, если микрокоманда имеет метку;

ИФ — имя формата, указанное в одном из операторов FORMAT секции настройки;

V_1, V_2 — выражения, значения которых присваиваются полям с именами ИП₁, ИП₂;

МО₁, МО₂ — мнемонические обозначения, задающие значения определенных полей. Эти обозначения должны быть определены в секции настройки с указанием полей, которые они задают;

ИС — имя строки (фрагмента микрокоманды), заданное псевдокомандой STRING.

Последняя псевдокоманда позволяет присвоить часто встречающимся фрагментам микрокоманд некоторое имя, а затем ссылаться на этот фрагмент по имени: STRING <ИС> ИП₁= V_1 , ИП₂= V_2 , ..., МО₁, МО₂, Всего может быть использовано до 50 имен строк, причем общее количество символов в этих именах не должно превышать 500.

Правила записи меток в символической микропрограмме аналогичны правилам записи имен в секции настройки. Всего в микропрограмме может быть использовано не более 100 меток, причем общее количество символов в метках не должно превышать 700—3n, где n — количество меток.

Псевдокоманда ORG B позволяет занести в счетчик размещения микрокоманд значение выражения B.

Псевдокоманда SET, имеющая такой же формат, что и в секции настройки, позволяет присваивать и повторно присваивать идентификаторам численные значения.

Для записи в ячейку ЗУ с разрядностью, равной разрядности слова микрокоманды, константы, равной значению выражения B, используется псевдокоманда МЕТКА: АДРЕС: WORD B, в которой метка и адрес могут отсутствовать.

Псевдокоманда DS V_1 (V_2) резервирует равное значению выражения V_1 количество ячеек ЗУ и заполняет эти ячейки константами, равными значению выражения V_2 , если последнее указано в псевдокоманде. Псевдокоманда DS может иметь метку и заданный адрес.

В конце символической микропрограммы ставится псевдокоманда END.

Символические микропрограммы обрабатываются микроассемблером в два прохода. Это позволяет задавать значения идентификаторов ниже того места, где они используются в выражениях. В выражениях в отличие от секции настройки может присутствовать текущее значение счетчика размещения, обозначаемое D.

Организация системы отладки микропрограмм

Для автоматической генерации эмулятора, т. е. модели технических средств микропроцессорной системы, по заданной принципиальной схеме используется библиотечный принцип. Модели БИС, объединенные в библиотеку, в точности соответствуют корпусам БИС и адекватно описывают логические сигналы на выводах, а также изменение содержимого программно доступных регистров. Структура модели не обязательно соответствует внутренней структуре БИС, что обуславливает функциональный характер моделей. Работа моделей БИС в общей модели технических средств согласуется с помощью передаваемых логических сигналов, т. е. работа технических средств моделируется на логическом уровне.

Система СЛОТ позволяет проследить во времени изменение содержимого регистров и ячеек ЗУ, а также сигналов в выбранных точках схемы. Сигналы моделируются в двоичной логике с учетом состояния с высоким выходным сопротивлением. По желанию разработчика сигналы на шинах или содержимое многоразрядных регистров могут моделироваться как один многозначный сигнал или переменная. При моделировании предполагается, что между двумя соседними по времени изменениями входных сигналов, в том числе

синхросигналов, все переходные процессы заканчиваются и схема приходит в устойчивое состояние. Задержки всех элементов считаются одинаковыми, сигналы в схеме распространяются в соответствии со степенью достижения от источников. Этот подход позволяет получить квазивременные логические диаграммы сигналов. Их вполне достаточно для отладки микропрограммного обеспечения, представляемого либо в виде наборов логических сигналов, поступающих на моделируемые блоки, либо в виде состояний ячеек ЗУ.

Модель технических средств — это некоторая структура данных (рис. 1), содержащая схему соединения БИС, имена программных моделей последних, значения сигналов в узлах схемы, содержимое внутренних регистров БИС. В списке контактов указывается их тип: входной, выходной, двунаправленный, допускающий объединение на общую линию, по схеме проводного И или ИЛИ.

Список узлов схемы включает предыдущее и новое значения сигналов, что позволяет моделировать работу ИС, реагирующих на фронты входных сигналов, тип узла (аналогичен типу контактов), количество изменений сигнала в текущей итерации. Обращение к сигналу узла схемы, как и к регистру ИС, возможно либо по имени, если оно присвоено, либо по номеру элемента и номеру регистра

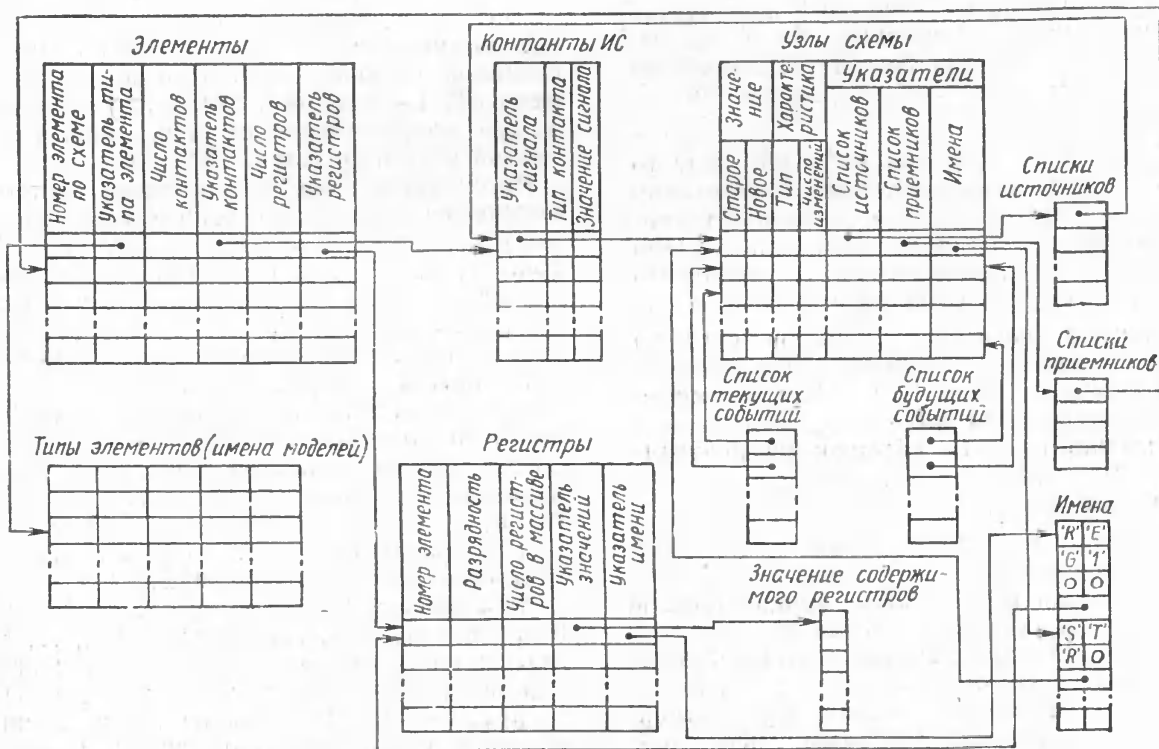


Рис. 1. Структура данных системы моделирования СЛОТ

или контакта, подключенного к данному узлу. Каждому узлу схемы соответствует свой список источников, т. е. контактов БИС, сигналы на которых позволяют вычислить результирующий сигнал в узле, зависящий от типа объединяемых контактов (объединение на общую линию, проводные И, ИЛИ).

Моделирование состоит из следующих шагов.

1. Выбрав новые значения входных сигналов, записать их в список узлов. Номера строк списка узлов с новыми значениями входных сигналов поместить в список текущих событий.

2. Для каждого элемента из списка текущих событий моделировать работу всех БИС и СИС, входы которых подключены к данному узлу, записать новые значения на выходе БИС и СИС в таблицу контактов. При несовпадении нового сигнала на контакте БИС с сигналом в соответствующем узле записать номер узла в список будущих событий.

3. Для всех узлов схемы, на которые есть указатели в списке будущих событий, переписать новое значение сигнала в старое, сформировать в соответствии с функцией объединения узла новое значение сигнала. Если старое значение сигнала равно новому, то удалить указатель на сигнал из списка будущих событий; если нет, то увеличить количество изменений сигнала на единицу. Если количество изменений сигнала больше заданной величины, то завершить моделирование по закливанию.

4. Если список будущих событий пуст, то значения сигналов вычислены. Необходимо приравнять нулю количество изменений сигналов в таблице узлов и перейти к п. 1. При непустом списке будущих событий переписать его в список текущих событий и перейти к п. 2.

Содержимое регистров, как и сигналы в таблице контактов, изменяется при обращении к модели БИС или СИС. После каждого такта моделирования на экран дисплея или печать можно вывести содержимое внутренних регистров БИС и значения сигналов в любых узлах схемы.

Приведенная структура данных позволяет легко модифицировать схему, удаляя, добавляя или заменяя БИС и СИС, просматривать содержимое регистров и значения сигналов. На языке управления работой системы СЛОТ можно, кроме того, описать схему соединения БИС и СИС и задать входные воздействия либо в виде логических диаграмм, либо в виде микропрограмм, генерируемых микроасемблером ГНОМ.

Заключение

Система функционально-логической отладки СЛОТ совместно с библиотекой моделей микропроцессорных БИС и СИС позволяет по заданной принципиальной схеме автоматически получить для контрольного выполнения и отладки микропрограмм эффективный эмулятор технических средств микропроцессорной системы. Реализация микроасемблера ГНОМ и системы СЛОТ на мини-ЭВМ СМ-4 доступна широкому кругу разработчиков микропроцессорной аппаратуры и дискретной логики. Настраиваемый на различные форматы микрокоманд микроасемблер позволяет разрабатывать микропрограммы в удобных для пользователя мнемонических обозначениях.

ПРИЛОЖЕНИЕ

Отладка микропрограммного обеспечения цифрового фильтра на базе серии К1804

При цифровой обработке сигналов, управлении технологическими процессами, а также во многих других случаях широко используются цифровые фильтры, алгоритм работы которых определяется выражением

$$y_n = \sum_{i=0}^N a_i x_{n-i} + \sum_{i=1}^M b_i y_{n-i};$$

где x_n , y_n — значения входного и выходного сигналов цифрового фильтра в момент времени nT , T — период квантования по времени; a_i , b_i — коэффициенты рекурсивной и рекурсивной частей фильтра.

Рассмотрим один из вариантов построения цифрового фильтра на основе микропрограммируемых процессорных секций К1804ВС1 (рис. 2). На рис. 3, 4 для большей наглядности приведены принципиальные схемы блоков цифрового фильтра не более чем шестого порядка, обрабатывающего четырехразрядные величины.

Блоки входной памяти и памяти обратной связи аналогичны друг другу и построены на основе набора регистров (рис. 3), в которых хранятся значения входного или выходного сигналов x_n или y_n (элемент D2), x_{n-1} или y_{n-1} (элемент D3) и т. д. При поступлении тактового сигнала сдвига СДВ (он отмечает период выдачи управляющего воздействия y_n) величины x_n , x_{n-1} , ... или y_n , y_{n-1} , ... сдвигаются на один регистр вниз. При поступлении поля микрокоманды S_0 — S_2 на вход дешифратора D1 на одном из его выходов появляется сигнал низкого уровня, разрешающий выдачу содержимого одного из регистров на шину D0...D3.

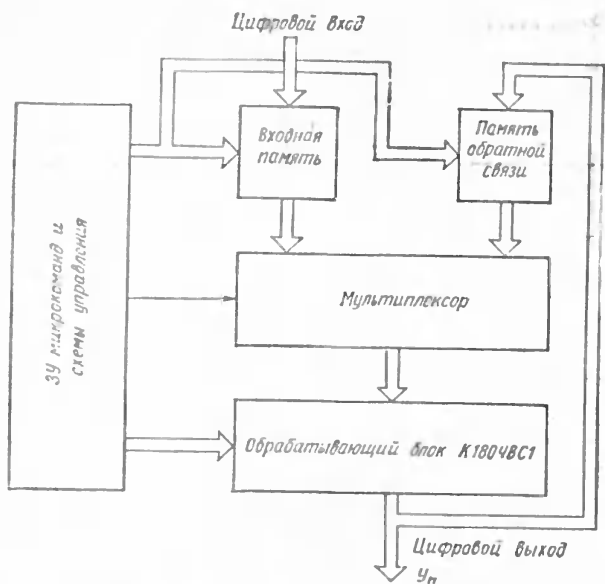


Рис. 2. Структурная схема цифрового фильтра на базе К1804

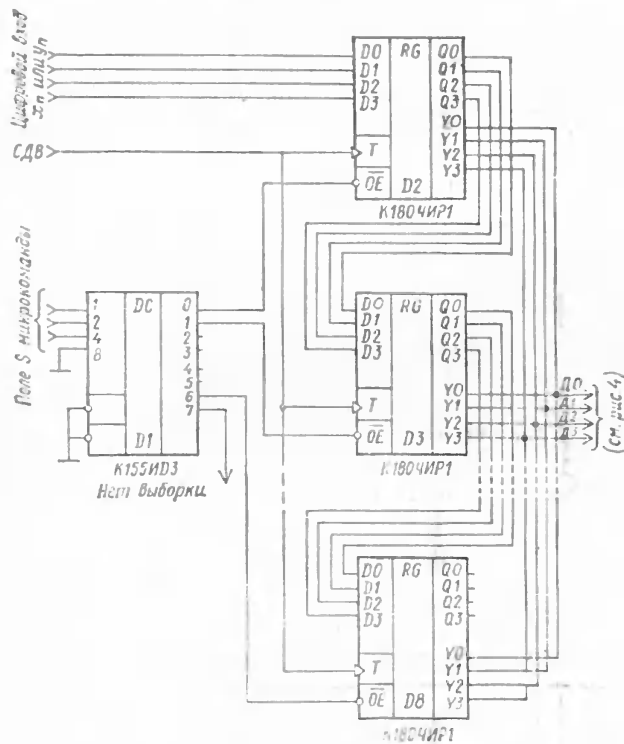


Рис. 3. Принципиальная схема блоков памяти

Данные с блоков входной памяти или памяти обратной связи через мультиплексор (D1, D2, рис. 4), управляемый полем M микрокоманд, поступают на вход данных обрабатывающего блока К1804ВС1. Во время вычисления нерекурсивной части фильтра $M=0$,

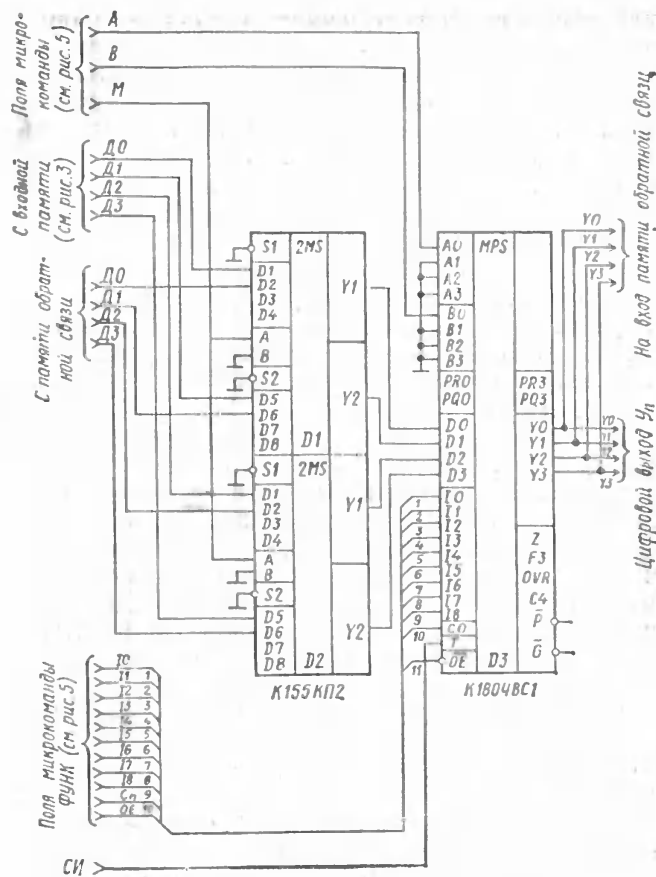


Рис. 4. Принципиальная схема обрабатывающего блока и мультиплексора цифрового фильтра

соответствующая сумма формируется в регистре R0. При вычислении рекурсивной части $M=1$, информация на вход данных поступает с памяти обратной связи, сумма рекурсивной части формируется в R1. При поступлении каждого синхрипульса СИ выполняется одна микрокоманда. Период тактовых сигналов сдвига СДВ должен быть достаточным для выполнения микропрограммы, реализующей заданный цифровой фильтр.

Блок управления (рис. 5) содержит регистр начального адреса (К155ТМ5), в

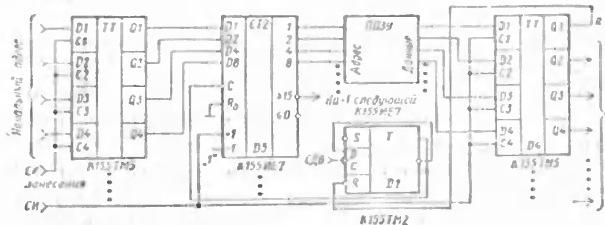


Рис. 5. Принципиальная схема блока управления цифрового фильтра

Алгоритм вычисления суммы произведений

$$\sum_{i=0}^2 a_i x_{n-i}$$

$x_n=1010$ $a_0=0100$	$x_{n-1}=0011$ $a_1=1011$	$x_{n-2}=0101$ $a_2=0111$	Сумма частных произведений
0000 0000 +1010 0000	0011 0011 +0000 0011	0101 0101 +0101 0000	1000 1000 +1111 0011
0101000 + 0100001 + 0100011 = 1101100			

который внешними схемами до начала работы блока заносится начальный адрес микропрограммы нужного цифрового фильтра. При поступлении сигнала низкого уровня с триггера D1 начальный адрес заносится в счетчик адреса D3. Поступающий импульс сдвига СДВ устанавливает триггер D1 в состояние «Лог. 1», и счетчик D3 вырабатывает последовательность адресов микрокоманд. Разрядность адреса зависит от требуемого объема ППЗУ микрокоманд (на рис. 5 показано четыре разряда адреса). Микрокоманды с выхода ППЗУ поступают в регистр микрокоманд D4, откуда соответствующие поля $I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0$, C_n , OE, A, B, M, $S_2 S_1 S_0$, R подаются на обрабатывающий блок (I_0-I_8 , C_n , OE, A, B), мультиплексор (M), дешифраторы входной памяти и памяти обратной связи (S_0-S_2). Одноразрядное поле R во всех микрокомандах, кроме последней, имеет значение 1. Нулевое значение поля R последней микрокоманды сбрасывает триггер D1, что приводит к занесению в счетчик D3 начального адреса микропрограммы и блокирует увеличение содержимого D3. При этом до момента поступления импульса СДВ из ППЗУ постоянно будет считываться первая микрокоманда, выдающая рассчитанное U_n на выход фильтра.

При вычислении рекурсивной и нерекурсивной частей фильтра будем суммировать не произведения $a_i x_{n-i}$ (нижняя строка табл. 1), а частные произведения (правый столбец табл. 1). При таком алгоритме количество операций сокращается с уменьшением единичных разрядов коэффициентов фильтра. В связи с этим при кодировании коэффициентов будем использовать цифры 1, 0, -1. Так, $a_2=7$ вместо 0111 может быть представлен как 100-1. При значении разряда -1 из формируемой суммы частных произведений необходимо вычесть произведение.

В табл. 2 приведены операции обрабатывающего блока и соответствующие значения

управляющих разрядов, используемые при реализации цифровых фильтров. Поскольку значение поля M во всех микрокомандах, а значение поля A во всех микрокомандах, кроме сложения рекурсивной и нерекурсивной частей, совпадают со значением поля B, то при настройке микроассемблера (рис. 6) задаются соответствующие условные операторы. Каждому набору коэффициентов фильтра соответствует своя микропрограмма, длина которой определяет максимально возможную

```

НАСТРОЙКА МИКРОАССЕМБЛЕРА   ГНОМ
LENGTH 10;
FORMAT   ФУНК 10,
          OE 1[1], A 1, B 1, M 1, S 3[7], R 1[1];
ASSIGN   <ФУНК>  OUT=0110001000B,
                  ZERO=0110010000B,
                  R_PL_D=0110001010B,
                  R_MN_D=0110011011B,
                  R_PL_D_SD=1110001010B,
                  R_MN_D_SD=1110011011B,
                  SD=1110001000B,
                  PLUS=01100000100B;

IF B=1 M=1;
IF B=0 M=0;
IF B=1 A=1;
IF B=0 A=0;
END;
    
```

Рис. 6. Листинг настройки микроассемблера

Таблица 2

Значения полей микрокоманды, определяющих функцию обрабатывающего блока

Действие	Мнемоническое обозначение	$I_8 I_7 I_6$	$I_5 I_4 I_3$	$I_2 I_1 I_0$	C_n
Выдать содержимое регистра	OUT	0 1 1	0 0 0	1 0 0	0
Записать в регистр ноль	ZERO	0 1 1	1 0 0	1 0 0	0
К содержимому регистра прибавить данные	S_PL_D	0 1 1	0 0 0	1 0 1	0
От содержимого регистра отнять данные	R_MN_D	0 1 1	0 0 1	1 0 1	1
К содержимому регистра прибавить данные, результат сдвинуть	R_PL_D_SD	1 1 1	0 0 0	1 0 1	0
От содержимого регистра отнять данные, результат сдвинуть	R_MN_D_SD	1 1 1	0 0 1	1 0 1	1
Сдвинуть содержимое регистра	SD	1 1 1	0 0 0	1 0 0	0
Сложить содержимое двух регистров	PLUS	0 1 1	0 0 0	0 0 1	0

частоту дискретизации (период импульсов сдвига СДВ). Так, для фильтра четвертого порядка с заданными коэффициентами (табл. 3) микропрограмма (рис. 7) содержит 17 микрокоманд, что при частоте синхронимпульсов СИ 1 МГц ограничивает максимальную частоту дискретизации фильтра величиной 58 кГц.

Таблица 3

Пример кодирования коэффициентов и последовательность их использования при реализации фильтра четвертого порядка

Коэффициент	Десятичное значение	Кодирование			
		2 ³	2 ²	2 ¹	2 ⁰
a ₀	+1	0	0	0	1
a ₁	+7	1	0	0	-1
a ₂	+10	1	0	1	0
a ₃	+8	1	0	0	0
a ₄	+4	↓ 0	↓ 1	↓ 0	↓ 0
b ₁	-1	0	0	0	-1
b ₂	-5	0	-1	0	-1
b ₃	-7	-1	0	0	1
b ₄	-2	↓ 0	↓ 0	↓ -1	↓ 0

Микрокоманда	Микроассемблер	ГНОМ	Трансляция
1	1880F	1	/*МИКРОПРОГРАММА ЦИФРОВОГО ФИЛЬТРА ЧЕТВЕРТОГО ПОРЯДКА*/
2	1880F	2	ORG 0;
3	1880F	3	/*ВЫДАЧА YN*/
4	1880F	4	OUT, OE=0;
5	1880F	5	/*ОБНУЛЕНИЕ R0 И R1*/
6	1880F	6	ZERO, B=0;
7	1880F	7	ZERO, B=1;
8	1880F	8	/*ВЫЧИСЛЕНИЕ РЕКУРСИВНОЙ ЧАСТИ В R0*/
9	1880F	9	R_PL_D, B=0, S=1;
10	1880F	10	R_PL_D, B=0, S=2;
11	1880F	11	R_PL_D, SD, B=0, S=3;
12	1880F	12	R_PL_D, SD, B=0, S=4;
13	1880F	13	R_PL_D, SD, B=0, S=2;
14	1880F	14	R_PL_D, B=0, S=0;
15	1880F	15	R_MN_D, B=0, S=1;
16	1880F	16	/*ВЫЧИСЛЕНИЕ РЕКУРСИВНОЙ ЧАСТИ В R1*/
17	1880F	17	R_MN_D, SD, B=1, S=3;
18	1880F	18	R_MN_D, SD, B=1, S=2;
19	1880F	19	R_MN_D, SD, B=1, S=4;
20	1880F	20	R_MN_D, SD, B=1, S=2;
21	1880F	21	R_MN_D, B=1, S=1;
22	1880F	22	R_MN_D, B=1, S=2;
23	1880F	23	R_PL_D, B=1, S=3;
24	1880F	24	/*ВЫЧИСЛЕНИЕ YN В R0*/
25	1880F	25	YN В R0*/
26	1880F	26	PLUS, A=1, B=0, R=0;
27	1880F	27	END;

Рис. 7. Листинг трансляции микропрограммы цифрового фильтра

Закодировав принципиальную схему цифрового фильтра (рис. 3—5) и задав импульсы СДВ и СИ, отладим микропрограмму рассматриваемого фильтра, используя систему

СЛОТ. Обозначим выходные сигналы D3 (рис. 5) как АДРЕС, выходные сигналы D4 (рис. 5) как МКМ, входы D0...D3 элемента D3 (рис. 4) как ДАН. Зададим вывод на печать сигналов СДВ, СИ, АДРЕС, МКМ, ДАН и содержимого регистра R0 элемента D3 (рис. 4). В качестве начальных состояний выберем $y_n=0101$ (содержимое R0), $x_n=0111$ (цифровой вход входной памяти), $x_{n-1}=1000$ (содержимое D2, рис. 3), $x_{n-2}=0101$ (содержимое D3, рис. 3), $x_{n-3}=0010$ (содержимое D4, рис. 3), $x_{n-4}=0100$ (содержимое D5, рис. 3), триггер D1 (рис. 5) в состоянии 0. Состояния ячеек ПЗУ (рис. 5) определим в соответствии с кодами микрокоманд (рис. 7). Проведя моделирование с помощью СЛОТ, получим квазивременную диаграмму, представленную на рис. 8. По этой диаграмме

СДВ	СИ	СЛОТ АДРЕС	МКМ	ДАН	R0
0	0	0	1880F	X	5
1	0	0	1880F	X	5
0	1	1	1880F	X	5
0	0	1	1880F	X	5
0	0	2	1C88F	X	5
0	0	2	1C88F	X	0
0	1	3	1C8FF	X	0
0	0	3	1C8FF	X	0
0	1	4	18A83	B	0
0	0	4	18A83	B	0
0	1	5	18A85	5	0
0	0	5	18A85	5	0
0	1	6	38A87	2	0
0	0	6	38A87	2	0
0	1	7	38A89	4	0
0	0	7	38A89	4	0
0	1	8	38A85	5	0
0	0	8	38A85	5	0
0	1	9	18A81	7	0
0	0	9	18A81	7	0
0	1	A	19869	8	0
0	0	A	19869	8	0

Рис. 8. Распечатка результатов моделирования

можно одновременно проследить за изменениями сигналов в нужных точках схемы и содержимым регистров, в данном случае R0. Видно, что ход выполнения микропрограммы соответствует ожидаемому.

Телефон для справок о приобретении документации по системам ГНОМ и СЛОТ для секционированных БИС: 297-20-52, доб. 2-98 (Москва).

Статья поступила 15 марта 1985 г.

РАЗРАБОТКА ИНТЕРАКТИВНЫХ ПРОГРАММ НА ОСНОВЕ ПРИНЦИПА НЕПОСРЕДСТВЕННОГО РЕДАКТИРОВАНИЯ ИНФОРМАЦИИ

В организации интерактивной работы человека на ЭВМ в последние годы происходит существенный сдвиг в направлении от командно-языкового диалога и общения с ЭВМ на «почти естественном» языке к системам, максимально использующим качественно новые возможности видеотерминала. Суть нового — в терминологии [1] объектно-ориентированного — взаимодействия с ЭВМ выражается лозунгом «Вы видите то, что имеее» (What You See Is What You Get), а его пионерами, по-видимому, являются специалисты из Xerox Palo Alto Research Center. Новый подход применяется в современных экранных редакторах и текстовых процессорах, интерактивных графических системах, системах Smalltalk [2], Lisa [3], Gavilan [4], Macintosh [5], в комплексе программных средств ВЕД АН СССР для поддержки интеллектуального человеко-машинного интерфейса [6] и др. Команды человека при этом обычно ассоциированы с элементарными нажатиями на клавиши или перемещениями «мыши», а на экране изображаются состояния объектов, с которыми работает человек, а не тексты команд системы. Эти же идеи в рамках «метафоры редактирования» используются для редактирования файлов на физическом уровне, работы с оглавлением файлов и т. п. [7, 8].

Ниже объектно-ориентированное взаимодействие с ЭВМ излагается в терминах, сложившихся в практике разработки интерактивных программ на механико-математическом факультете МГУ. Основу изложения составляют аналогия «Человек в Мире объектов» и принцип непосредственного редактирования информации. «Объект» при этом понимается в максимально широком, житейском смысле этого слова (в противоположность узкоспециальному программистскому), а ключевой термин «непосредственное редактирование» — редактирование без системы-посредника — означает, что диалоговая система «сама по себе» во внешнем интерфейсе никак не проявляется (т. е. она прозрачна для пользователя). Таким образом, с точки зрения пользователя он вообще ни с кем ни на каком языке никакого диалога не ведет, а просто нажимает на кнопки и наблюдает на экране происходящие изменения.

«Человек в Мире объектов», принцип непосредственного редактирования и критерий непосредственности

Мы полагаем, что работа человека на ЭВМ состоит в целенаправленном изменении некоторого объекта, который имеет внутреннюю структуру и содержание, а кроме того, внешнее (обычно алфавитно-цифровое или графическое) представление. Такими объектами могут быть: текст программы, модель физической системы или явления, база данных, над которой работают какие-то прикладные программы, реальный физический объект, которым управляет ЭВМ, и т. п. Любой процесс «ручного» или «самостоятельного» создания, изучения, изменения объекта, выполняемый человеком в интерактивном режиме, мы будем называть «редактированием», а соответствующую программную систему — «редактором».

Принцип непосредственного редактирования информации состоит в следующем:

в процессе работы с редактором человек должен представлять себя находящимся в том же «пространстве», что и объект редактирования (это пространство вместе с объектом редактирования мы далее будем называть «Миром»); на экране всегда должны изображаться редактируемая часть «Мира» и «положение» человека в нем; это «положение» должно определять аргументы большинства команд; внешний вид редактора должен отражать существенные для работы человека понятия «Мира» и соответствовать его интуитивным представлениям, но не должен содержать понятий, связанных с конкретной реализацией «Мира» или реализацией самого редактора.

Таким образом, в отличие от устаревшего телетайпного режима диалога, при котором отображаются набираемые человеком команды и информационные сообщения системы об их выполнении, в непосредственном режиме изображается состояние объекта, а не тексты команд. Кроме того, использование «положения» человека в «Мире» для определения аргументов команд означает, что традиционная схема «команда — аргумент, команда — аргумент, ...» заменяется последовательностью «аргумент — команда — команда — команда — ...». Именно так устроен диалог в современных высокоинтерактивных компьютерных играх.

Основным преимуществом разработки систем на основе аналогии «Человек в Мире объектов» и принципа непосредственного редактирования является то, что сама система в качестве посредника между человеком и объектом как бы исчезает, а вместе с ней исчезают и диалоги между человеком и системой в традиционном смысле этого слова. Эффект «исчезновения» системы-посредника вызван задействованием правого полушария (образно-геометрическое восприятие) и разгрузкой левого (абстрактно-логическое) [9, 10]. С точки зрения человека, си при таком взаимодействии просто имеет набор инструментов для «ручного» изменения «Мира», в то время как при традиционном подходе человек, наоборот, ведет диалог с некоторой системой-посредником, а уже система-посредник меняет объект или изображает его. Непосредственный интерфейс позволяет человеку максимально использовать свою интуицию, знания, жизненный опыт и опыт ориентировки в пространстве, т. е. опыт непосредственного, чувственного восприятия «Мира». В результате, после приобретения необходимых моторных навыков, человек получает возможность работать за терминалом, обдумывая лишь содержательные вопросы о том, что надо делать, и не тратя практически никаких сознательных усилий на обдумывание того, как этого достичь.

Таким образом, как это ни парадоксально звучит, в идеальной диалоговой системе вообще не должно быть никаких диалогов в традиционном смысле этого слова, т. е. диалогов человека с системой-посредником. Этот критерий непосредственности системы легко проверить на практике. В частности, если в системе хоть какой-нибудь акт взаимодействия с человеком может быть без изменений проведен по телефону (или с помощью телетайпа), то такая система не удовлетворяет критерию непосредственности и, следовательно, может быть улучшена,

К сожалению, широко эксплуатируемое в настоящее время программное обеспечение обычно не является непосредственным — строчные редакторы, операционные системы, системы взаимодействия с базами данных и информационно-поисковые системы, автоматизированные обучающие системы и т. д., как правило, имеют специальный язык для взаимодействия с пользователем. Такой язык может быть простым, сложным или даже «близким к естественному»; может содержать или не содержать средства выдачи справочных сообщений, режим «подсказок», аппарат «умолчаний» и пр.; может зависеть или не зависеть от «уровня пользователя», т. е. от того, насколько хорошо, с точки зрения системы, пользователь вел себя в предыдущие сеансы работы, но обычно он есть — язык для проведения диалога между человеком и системой.

Даже в некоторых наиболее «революционных» современных системах (например, Lisa [3], Gavilan [4]) человек часто вынужден вести диалог с системой посредником, хотя и в новой форме — в форме графического или текстового меню. В одном меню выбирается картинка или текст, соответствующий желаемому действию (например, команде EDIT), а в другом указывается объект-аргумент (файл, который надо редактировать). Смена или наложение меню на экране, «скачки» курсора из одного меню в другое и сохранение пары (объект, действие) или, в традиционной терминологии, пары (команда, аргумент) не позволяют забыть о существовании команд и не создают иллюзии присутствия человека в «Мире» своих объектов (в данном случае файлов). Заметим, что наличие меню действий (команд) на экране противоречит и «метафоре письменного стола» — в реальной жизни человек находит и правит нужный документ, а не отмечает желаемое действие в специальной бумажке со словами «EDIT», «COPY» и др.

Принцип непосредственного редактирования нарушается и в большинстве экранных редакторов, текстовых процессоров и интерактивных графических систем — даже в этих системах на экране кроме состояния редактируемых объектов часто изображаются еще и тексты или меню команд. Типичны «диалоги в малом» (например, при вводе координат новой точки в трехмерной графической системе), «подмена Мира» (например, введение понятий, описывающих часть текста, уместившуюся в памяти ЭВМ, в экранном редакторе), отсутствие естественных для человека и наличие реализационных понятий и команд (например, «номер строки», «сборка мусора» и др.). Критику нескольких текстовых редакторов и графических систем можно найти в [11]. Здесь же подчеркнем, что, поскольку основу принципа составляет учет «человеческого фактора», принцип применим к организации взаимодействия человека с ЭВМ в любых существенно интерактивных системах.

Если терминал алфавитно-цифровой

Диалог человека с прикладной системой, построенной на основе аналогии «Человек в Мире объектов», является не последовательностью обменов «сообщениями», а процессом одновременного редактирования «Мира» человеком и системой с постоянной визуализацией представления «Мира» на экране терминала. При работе с такой системой любой запрос человека на перемещение (изменение) распадается на две части: на перемещение (изменение) в «Мире» и на визуализацию — перемещение (изменение) в представлении «Мира» на экране терминала. Если для взаимодействия с системой используется алфавитно-цифровой терминал, то представление «Мира» — обязательно текстовое, т. е. алфавитно-цифровое.

Для человека сам «Мир» и его текстовое представление образуют единое целое, с теми лишь оговорками, что некоторые части текста, соответствующие еди-

ным объектам «Мира», могут быть неразрывны, непроходимы или неизменяемы, а некоторые запросы человека, «атомарные» с точки зрения «Мира», могут быть составными с точки зрения представления «Мира» или вызываться нажатием нескольких клавиш. Например, при переименовании файла непосредственным образом надо набрать новое имя на месте старого. При этом запросом на переименование является акт выхода за границы зоны имени файла, а не элементарные нажатия клавиш в процессе набора нового имени. Естественно, что перемещения и изменения в текстовом представлении «Мира» (и тем самым в самом «Мире») должны осуществляться с помощью команд обычного (непосредственного) редактора текстов. В этом смысле набор доступных человеку команд в значительной степени вообще не зависит от того, с каким «Миром» идет работа, а вопросы внешнего интерфейса сводятся к выбору структуры и представления «Мира».

Ниже приведены примеры, частично пересекающиеся с [7, 8] и иллюстрирующие применение принципа.

Непосредственный редактор текстов. При формализации интуитивных представлений человека о тексте наиболее адекватно, видимо, представление текста линейным двусторонним списком строк (Л2-списком) с одним указателем (соответствующим положению человека в «Мире») и с возможностями ссылки из одной части текста в другие части того же текста или в другие тексты. Существование, что Л2-список строк допускает движение указателя вверх и вниз по тексту без каких-либо ограничений, вплоть до начала (конца) текста. При этом представлении отсутствуют понятия, описывающие часть текста в памяти ЭВМ. Разумный набор основных понятий для редактора текстов общего назначения — это, видимо, набор (символ, пробел, слово, строка, абзац), где строка с внешней точки зрения — это вектор символов с элементарными операциями перемещения влево-вправо, вставки-удаления, замены символа до или за указателем. Каждый символ должен занимать одну позицию на экране, чтобы можно было задействовать интуитивные представления человека, сложившиеся при работе на бумаге. В частности, это означает, что клавиша табуляции должна восприниматься как команда перемещения, а не как команда «вставить один символ — символ табуляции».

При работе с редактором все действия должны производиться по нажатию на клавиши, а на экране должно отображаться окрестность указателя в тексте (т. е. фрагмент окружающего человека «Мира»). При перемещениях по тексту видимый человеком фрагмент «Мира» должен меняться без дополнительных команд.

Наконец, макрокоманды, если они есть в редакторе, также должны вводиться в непосредственном режиме, т. е. нажатия на клавиши должны обрабатываться немедленно (человек должен видеть результат, а не текст). Таким образом, первый раз макрокоманда будет выполнена в процессе ее ввода, а в целом работа с макрокомандой будет организована не как последовательность копирования — использования, а как запоминание образца по ходу работы, а затем его повторение.

Приведенные замечания можно рассматривать как конструктивную критику ряда экранных редакторов для СМ-4.

Непосредственная работа с файлами. При работе на ЭВМ человек часто выполняет разнообразные операции с файлами: создание, переименование, изменение атрибутов (например, прав доступа), редактирование, просмотр, распечатка, компиляция, сборка, выполнение и т. д. Как правило, для выполнения таких команд человек должен набрать строку на специальном командном языке. Следовательно, он должен знать синтаксис этого языка и посвящать значительную часть своего времени «разговору» с ОС, сначала формулируя желаемые действия, а затем — запросы, которые покажут, что же произошло (например, для просмотра оглавления файлов после переименования).

«Мир», с которым при выполнении таких действий имеет дело человек, — это «Мир файлов в целом» или «Мир», где элементарные объекты — описатели файлов. Обычно (в том числе и в экранных файловых мониторах) описатель содержит символическое имя файла, даты создания и последнего изменения, число занимаемых файлом блоков на диске, права доступа, местоположение файла и пр. Никакой информации о содержимом файла, кроме, быть может, указания на тип хранящейся информации (программа на Фортране, объектный модуль, листинг и т. п.), описатель не содержит, ибо эта информация для ОС бесполезна.

Однако если идти не от устройства оглавления в ОС, а от потребностей человека, то описатель в первую очередь должен содержать информативную информацию о содержимом файла (программа решения такой-то задачи, текст такой-то статьи) и лишь во вторую — какие-то необходимые для работы характеристики (символьное имя, число занимаемых блоков и пр.).

Порядок файлов в оглавлении обычно соответствует в ОС порядку записей о файлах в физическом оглавлении на диске

и меняется при операциях с файлами, а пользователь может упорядочить оглавление в лексикографическом порядке имен, по датам создания и т. п. С точки зрения аналогии «Человек — Мир» все это — просто полный хаос, в котором объекты (описатели файлов) самопроизвольно меняют свое положение, после операции перемещаются, куда им вздумается, и в котором невозможно навести никакой разумный порядок. Ясно, что человек значительно лучше ориентируется в «Мире», где он может располагать объекты (описатели файлов) так, как он — человек — считает нужным, и в котором эти объекты остаются на своих местах, пока он — человек — их сам куда-нибудь не перенесет. Заметим, что операция упорядочения такого «Мира» в лексикографическом порядке не имеет ни малейшего смысла.

Таким образом, непосредственный файловый монитор должен выглядеть, как и непосредственный редактор текстов, с той лишь разницей, что «текст» в нем — это последовательность «человеческих» описателей файлов. Операции над таким текстом должны приводить к соответствующим операциям над файлами. Например, файл должен удаляться командами редактора, удалением описателя файла, переименование или изменение атрибутов — подводом курсора к нужному месту и набором нового имени или новых значений атрибутов поверх старых и т. п. Соответственно для выполнения действий над файлами необходимы клавиши «распечатать», «выполнить», «внутри» (для перехода в подоглавление или входа в редактор) и пр.

Такой подход существенно повышает производительность и комфортабельность труда человека за терминалом и экономит ресурсы ЭВМ. Например, описатели файлов в «тексте» могут содержать в невидимых частях информацию о местоположении файла на диске. Это позволяет прямо выполнять операции с файлом, минуя характерный для традиционных систем поиск файла в оглавлении.

Непосредственная ОС. Принцип непосредственного редактирования можно применить и к другим сторонам взаимодействия человека с ОС (редактирование файлов на физическом уровне, изучение конфигурации оборудования, загрузка, выгрузка и изменение характеристик драйверов, регистрация, просмотр и изменение прав пользователей, управление планированием и приоритетами задач и др.). Причем, если говорить о преобразовании к непосредственному виду существующей ОС, т. е. считать «Миры» уже определенными, то задача не является чересчур сложной. Например, в системе РАФОС СМ ЭВМ вместо сотни можно обойтись всего четырьмя командами непосредственного редактирования соответствующих «Миров» (устройства и драйверы, оглавление файлов, тексты, физический уровень).

Непосредственные редактирование-компиляция и интерпретация-отладка программ. Попытаемся применить принцип непосредственного редактирования к разработке программ. Первый вопрос, который возникает, — что такое программа и какова структура этого «Мира»? В традиционных системах, где программа сначала создается и редактируется, а потом компилируется, «Мир» — это просто исходный текст. Поэтому при создании, например, конструкции «если-то-иначе-конеч-если», человек должен ввести в текст соответствующие символы и строчки. Информация о том, что эти символы образуют единую конструкцию, остается в голове человека. Позже лексический и синтаксический анализаторы компилятора будут тратить ресурсы, пытаясь выделить эти «если-то-иначе-конеч-если» из последовательности символов и сформировать из них единую конструкцию. Такие непроизводительные затраты труда как человека (преобразование одной конструкции в последовательность символов), так и ЭВМ (обратное преобразование), не говоря уже о появляющихся возможностях для ошибок, явно говорят, что «Мир» сформулирован неверно и не соответствует представлению человека. Правильный «Мир» должен содержать основные конструкции языка программирования именно как неделимые объекты (что сделано, например, в [2, 12]) и активно пропагандируется сейчас в зарубежных работах, посвященных редактированию-компиляции программ. Непосредственный редактор может генерировать такие конструкции по нажатию на специальные кнопки. Желательно и внешне (с помощью шрифта и цвета) различать объекты «Мира», имеющие разную природу (локальные и глобальные переменные, формальные параметры, имена подпрограмм и пр.), и сделать такое различие синтаксически значимым.

Вторая сторона проблемы — непосредственные прокрутка и отладка программ. Отличительная особенность здесь — наличие двух «Миров»: «Мира» самой программы или ее текста и «Мира» объектов программы, который возникает и меняется в процессе выполнения.

Для изменения любого из этих двух «Миров» и переключения из одного «Мира» в другой экран терминала удобно разделить на две области (два «окна»), отображающие части двух «Миров». Перемещения в «Мире» программы в процессе ее выполнения и изменения в «Мире» объектов должны немедленно визуализироваться на экране.

Базовые средства реализации непосредственного интерфейса

Из сказанного следует, что есть целый класс задач, общий для всех непосредственных АЦ-систем: хранение текста, обработка запросов и человека и приклад-

ной программы на перемещения и изменения в тексте, визуализация текста на экране терминала, все другие действия, связанные с текстом, а также формирование и передача прикладной программе запросов человека на перемещения и изменения в «Мире». Естественно возложить все эти функции на универсальную базовую систему, чтобы прикладная программа занималась только содержательной деятельностью, связанной с «Миром». Такая базовая система была создана и получена название «непосредственный драйвер текста «НДТ-83» [13] — она существенно повысила производительность труда программиста,

Заключение

Взаимодействие человека с ЭВМ имеет свою специфику, и надо максимально использовать эту специфику для повышения эффективности труда человека, а не имитировать диалог двух разумных существ со всеми вытекающими отсюда психологическими и естественно-языковыми проблемами. Не Smalltalk, Lisa, Macintosh и др., а современные высокоинтерактивные компьютерные игры являются теми наиболее прогрессивными системами, у которых надо заимствовать организацию интерфейса с пользователем, а также терминологию и методологию разработки этого интерфейса. У игр с высокой степенью динамики принципиально не бывает не только «директив», но и выбора команды из меню. Игроку просто некогда переключаться на команды — потеря контроля над «Миром объектов» даже на мгновение может привести к проигрышу. После «вживания в игру», т. е. после приобретения необходимых моторных навыков, игрок перестает замечать клавиши и последовательности их нажатий — он начнет мыслить в терминах «Мира», его внимание и интеллект целиком задействуются в игре. По-видимому, возможно «жить» в своем «Мире» и непосредственно воздействовать на него — одна из основных причин бума компьютерных игр.

Достаточно создать такие условия профессиональному конструктору — и он будет жить в «Мире» своих изделий, даже не замечая, что существует САПР. Достаточно создать такие условия профессиональному программисту — и он сможет жить в «Мире» своих программ, даже не замечая, что существует ОС. И хотя он — профессионал — вроде бы и раньше на взаимодействие с ОС никаких особых усилий не тратил, производительность его труда резко возрастет.

Таким образом, при реализации экранных систем, систем многооконного взаимодействия, интерактивных графических систем и др. к актам диалога типа «запрос — ответ», «ввод (именно ввод, а не редактирование) параметров», «выбор команды из меню» и т. п. надо относиться как к остаткам мусора, которые во время уборки заметают в угол. Конечно, иногда выгоднее написать фрагмент такого диалога, чем решать вопросы о том, что же у нас за «Мир», как его изображать на экране и что с ним можно делать. Однако следует отдавать себе ясный отчет, что любой такой фрагмент — это все-таки мусор.

Непосредственное редактирование объектов на ЭВМ можно организовать уже сейчас на серийно выпускаемом отечественном оборудовании. Хотя отсутствие цвета, графики, звукоинтегрирующих возможностей не лучшим образом скажется на представлении «Мира», организация непосредственного взаимодействия возможна и требует не больше ресурсов, чем традиционные системы.

За последние годы на механико-математическом факультете МГУ был разработан ряд непосредственных программ и систем для ЭВМ СМ-4 (ОС RSX11M) с алфавитно-цифровыми терминалами: непосредственный драйвер текста «НДТ-83», интегрированная программная система «Микромир-85» (симбиоз текстового ре-

См. продолжение на стр. 58

УДК 621.317 : 681.3.62

Г. Я. Мирский

ВСТРОЕННАЯ МИКРОПРОЦЕССОРНАЯ СИСТЕМА — ОСНОВА СОВРЕМЕННОГО СРЕДСТВА ИЗМЕРЕНИЯ

По широте и эффективности применения микропроцессоров одно из первых мест занимает контрольно-измерительная техника. Для современного этапа ее развития характерно коренное изменение принципов построения средств измерения: их органической частью стала микропроцессорная система (МС).

Встроенная в измерительный прибор, МС расширяет его возможности, придает новые качества: многофункциональность, резкое повышение точности, упрощение и облегчение управления, гибкость, самокалибровку, повышение надежности и экологичности, миниатюрность конструкции, а также позволяет осуществить автоматизацию статистической обработки результатов измерений. Появились полностью автоматизированные, программно управляемые приборы, радикально повышающие производительность труда, позволяющие решать задачи, которые ранее даже не ставились [1]. Подобные приборы называют интеллектуальными.

Современные цифровые средства измерения, как правило, строят на основе МС. Возможности, особенности таких приборов, а также пути достижения этих возможностей достаточно подробно рассмотрены в [1], причем особое внимание уделено способам улучшения метрологических характеристик. При проектировании микропроцессорных средств измерения предусматривают в случае необходимости подключение прибора к стандартной интерфейсной шине (каналу общего пользования). Это позволяет осуществлять дистанционное управление приборами, организовывать их в измерительную систему, которую в процессе эксплуатации можно наращивать, дополнять новыми функциональными блоками или, если возникает такая потребность, заменять в ней одни блоки другими. Система получается функционально гибкой.

Известно [1, 2], что каждому прибору, который может работать в системе, объединенной интерфейсом, присущи приборные и интерфейсные

функции. Осуществление двух групп функций сказывается и на конструктивных решениях таких приборов, называемых системными.

На характер приборных функций программно-управляемого системного прибора оказывает влияние необходимость взаимодействия прибора с интерфейсом. У таких приборов различают три приборные функции: формирования сигнала измерительной информации, программного управления, запроса на обслуживание.

Формирование сигнала измерительной информации, т. е. информации о значении измеряемой физической величины, — основная функция, определяемая назначением прибора. Она включает все слагаемые измерительной процедуры, выполняемой прибором: масштабные преобразования исследуемого сигнала (усиление, ослабление), воспроизведение единицы, преобразование исследуемого сигнала к виду, удобному для сравнения, операцию сравнения с единицей, фиксацию результата сравнения, отображения результата измерения, запоминания его, статистической обработки и т. п.

Программное управление прибора заключается в выполнении программы, хранящейся в ПЗУ и частично в ОЗУ микропроцессорной системы. Системные приборы допускают дистанционное программное управление, осуществляемое через интерфейс.

Запрос на обслуживание предполагает сообщение системе о состоянии прибора: готовность к программному управлению, началу работы, передаче результатов измерений, наличии повреждения, ошибочных измерениях и т. п.

Другая часть системного прибора, предназначенная для выполнения **интерфейсных функций**, содержит узлы сопряжения приборных и интерфейсных функций, а также узлы сопряжения с интерфейсом. На выходе прибора имеется разъем, необходимый для подключения к стандартному интерфейсу. Обычно около разъема приводятся надписи, указывающие перечень выполняемых интерфейсных функций. Внутри прибо-

ра предусмотрены интерфейсные карты — ИКАР. Их структура зависит от выполняемых данных прибором интерфейсных функций. В настоящее время для построения ИКАР выпускаются специализированные БИС.

Системное назначение прибора оказывает определенное влияние на приборные функции. Это относится главным образом к алгоритмам преобразования информации.

Набор интерфейсных функций, выполняемых конкретным прибором, определяется задачами, которые должен решать этот прибор, и условиями его применения. Каждая из десяти интерфейсных функций осуществляется с помощью соответствующей логической схемы [2, 3]. Конструктивно набор таких схем объединен в универсальное устройство сопряжения с магистралью. Помимо логических схем оно содержит схемы согласования уровней сигналов и дешифратор команд.

В измерительных приборах, содержащих МС, с помощью последних возможно управление осуществлением интерфейсных функций. Такое использование внутриприборной МС особенно целесообразно при большом числе интерфейсных присоединений и сложности операции сопряжения. Однако организация микропроцессорного управления интерфейсом требует хорошо обоснованного выбора микропроцессора (прежде всего по быстродействию и возможности прерывания), детального анализа функционирования интерфейса, а также рационального программирования работы МС. В [3] подчеркивается, что при осуществлении интерфейса с микропроцессором нельзя руководствоваться общими соображениями. В каждом конкретном случае должны быть учтены реализуемые интерфейсные функции, характеристики применяемых приборов и микропроцессоров, элементная база и т. п. Наиболее распространенными видами электронных измерительных приборов, содержащих микропроцессоры, являются цифровые вольтметры, частотомеры, осциллографы. Их особенности, принципы построения и выполня-

емые функции описаны в [1]. В настоящее время выпускаются микропроцессорные приборы, измеряющие и многие другие электрические величины или параметры сигналов. Кратко рассмотрим принципы построения и работы, а также возможности некоторых из таких приборов.

Микропроцессорный фазометр

Цифровой фазометр, содержащий микропроцессорную систему (рис. 1), помимо общих преимуществ по сравнению с прибором, выполненным по схеме с жесткой логикой работы и измеряющим средний (за интервал измерения) фазовый сдвиг, обладает еще рядом специфически «фазометрических» достоинств. Одно из них заключается в том, что прибор

Числа p и N передаются из счетчиков 1 и 2 в МС, где вычисляется отношение p/N . После умножения отношения на 360 получается искомое значение фазового сдвига, которое отображается на дисплее.

Микропроцессорная система дает возможность пользователю выбрать любой (i -й) период, за который требуется найти фазовый сдвиг, а также наблюдать флуктуации фазовых сдвигов.

Чтобы определить разрешающую способность фазометра в зависимости от значения f частоты исследуемого сигнала, необходимо выразить погрешность дискретности в градусах фазового сдвига. Несложные преобразования показывают, что максима-

га, равная умноженному на 360 отношению значений частоты f исследуемого сигнала и частоты $F_{сч}$ следования счетных импульсов [4].

Таким образом, при непосредственном использовании рассмотренного принципа высокая точность измерения может быть получена только на низких и инфранизких частотах. Чтобы расширить частотный диапазон прибегают к предварительному гетеродинному преобразованию частоты исследуемых напряжений.

Повышение разрешающей способности фазометра и точности измерения малых фазовых сдвигов достигается применением метода интерполяции, эффективного при измерении малых интервалов времени [1, 4].

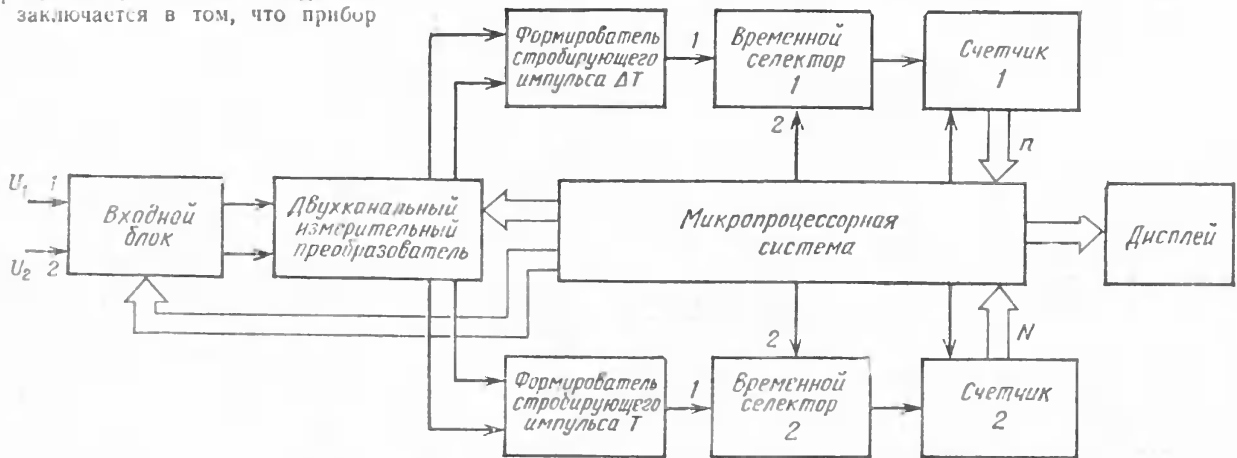


Рис. 1. Структурная схема фазометра

позволяет измерять фазовые сдвиги за один период исследуемого напряжения.

Рассмотрим принцип измерения фазового сдвига в течение одного периода (рис. 2). Синусоидальные напряжения u_1 и u_2 , фазовый сдвиг между которыми необходимо измерить, преобразуются в короткие однополярные импульсы (рис. 2, а-в). Из первой пары импульсов 1 и 2 в приборе формируется стробирующий импульс длительностью ΔT (рис. 2, г). С помощью временного селектора 1 он заполняется счетными импульсами, подаваемыми из МС с частотой следования $F_{сч}$. Число p импульсов, поступающих в счетчик 1 (рис. 2, д), пропорционально интервалу ΔT (коэффициент пропорциональности $F_{сч}$).

Параллельно формируется стробирующий импульс длительностью T , равной периоду исследуемого синусоидального напряжения. Этот стробирующий импульс заполняется посредством временного селектора 2 счетными импульсами, следующими с той же частотой $F_{сч}$ (рис. 2, а, в, е, ж). Число импульсов N , сосчитанных счетчиком 2 (рис. 2, ж), пропорционально периоду T ,

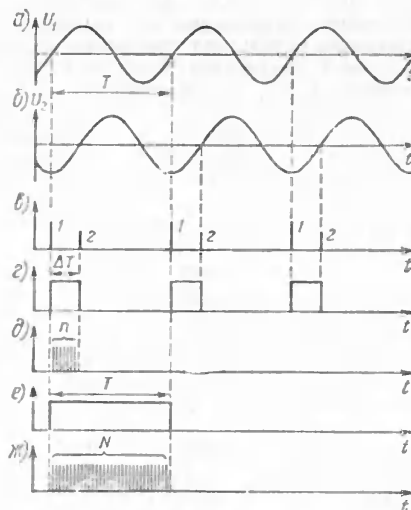


Рис. 2. Принцип действия фазометра

мальной абсолютной погрешности дискретности, составляющей плюс-минус единицу младшего разряда счета, соответствует абсолютная погрешность измерения фазового сдви-

Описанным фазометром можно измерять и средние фазовые сдвиги за k периодов. При этом в счетчике 1 накапливаются числа импульсов, заполняющих k интервалов ΔT , а счетчик 2 фиксирует число импульсов, попадающих в интервал kT . Значение k задается программой работы микропроцессорной системы.

Наличие в составе прибора МС создает возможность определения статистических характеристик фазовых флуктуаций: математического ожидания, дисперсии, среднеквадратического отклонения.

Широкодиапазонный автоматический частотомер на основе микропроцессорной системы и синтезатора частоты

В [1] описаны два цифровых частотомера: сервисный, измеряющий частоты до 200 кГц, и прибор повышенной точности, рабочий диапазон которого простирается от 0,01 Гц до 320 МГц. Современная частотозмерительная техника располагает более широкодиапазонными приборами, содержащими встроенную МС и син-

тезатор частоты. Подобные приборы позволяют без перестроек измерять частоты, значения которых заключены в очень широком диапазоне, например от 10 Гц до 18 ГГц или от 10 Гц до 32 ГГц.

Максимальными характеристиками обладает широкодиапазонный частотомер с программируемым синтезатором частот, управляемым МС и выполняющим роль гетеродина (рис. 3) [5].

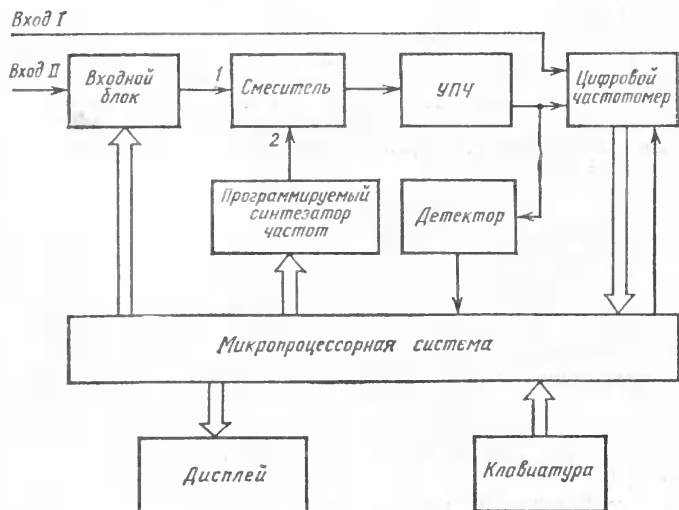


Рис. 3. Структурная схема широкодиапазонного частотомера

Синтезатор частот — это устройство, преобразующее сигнал с фиксированным значением частоты, вырабатываемый образцовым (высокостабильным) источником, в сигнал, дискретные значения частоты которого устанавливаются с требуемым шагом в определенном диапазоне частот. В схеме частотомера переход от одного значения частоты к другому осуществляет МС.

Для нахождения значения частоты в поддиапазоне от 10 Гц до 300 МГц напряжение исследуемого сигнала подается на вход I. Измерение выполняет непосредственно цифровой частотомер. При более высоких значениях частоты $f_{изм}$ выход источника сигнала соединяется со входом II прибора. Работа схемы заключается в следующем.

Напряжение исследуемого сигнала поступает через входной блок на вход 1 смесителя, ко входу 2 которого подводится напряжение выходного сигнала программируемого синтезатора частот. Микропроцессорная система изменяет значение частоты $f_{снт}$ выходного сигнала синтезатора до такого значения $f'_{снт}$, при котором разность значений измеряемой

частоты $f_{изм}$ и n -й гармоники выходного сигнала синтезатора $nf_{снт}$ примет значение $f'_{пр}$, находящееся в полосе пропускания усилителя промежуточной частоты (УПЧ). При этом детектор, включенный на выходе УПЧ, формирует сигнал-команду, по которой МС прекращает дальнейшую перестройку частоты синтезатора. Значение $f'_{пр}$ измеренный цифровым частотомером и результат измерения направляется в память системы,

ной частоты выходного сигнала синтезатора $f'_{снт}$ на Δf (причем значение Δf во много раз меньше значения $f'_{пр}$), и оно становится $f''_{снт} = f'_{снт} - \Delta f$. Измерительная процедура повторяется. На выходе УПЧ появляется сигнал разностной частоты, значение которой $f''_{пр}$ измеряется частотомером. Результат измерения направляется в МС, которая вычисляет номер гармоники как отношение модуля разности значений $f'_{пр}$ и $f''_{пр}$ к значению Δf , а также сравнивает значения $f'_{пр}$ и $f''_{пр}$. Если $f'_{пр} > f''_{пр}$, то, как несложно установить, и $f_{изм} > nf'_{снт}$. При $f''_{пр} < f'_{пр}$ соответственно $f_{изм} < nf'_{снт}$.

Мостовой измеритель параметров компонентов электронных схем

На рис. 4 представлен микропроцессорный прибор для измерения индуктивности катушки L , емкости конденсатора C , сопротивления резистора R и добротности колебательного контура Q с автоматическим переключением пределов измерений и автоматической подстройкой. Для подобных приборов характерны высокая точность измерений (относительная погрешность не более 0,1%) в широком динамическом диапазоне и простота управления — передняя панель с малым числом управляющих клавиш.

Работа прибора заключается в следующем. К выходным зажимам в качестве одного из плеч мостовой схемы подключают компонент (например, резистор), параметр которого (сопротивление) требуется измерить. Нажатием соответствующих клавиш задают вид измеряемого параметра (в данном случае клавиша, обозначенная буквой R) и выбирают значение частоты напряжения переменного тока, питающего мостовую схему. Так, если в приборе предусмотрена возможность установки значений частоты 100 Гц, 1 кГц и 10 кГц, то, выбирая одно из трех значений, учитывают особенности ис-

Искомое значение частоты $f_{изм}$ определяется как сумма или разность значений $nf'_{снт}$ и $f'_{пр}$ в зависимости от того, больше или меньше $f_{изм}$, чем $f'_{пр}$.

Определение числа n и того, как вычислять $f_{изм}$ (в виде суммы или разности значений $nf'_{снт}$ и $f'_{пр}$), производится автоматически по следующей методике. Микропроцессорная система уменьшает значение основ-

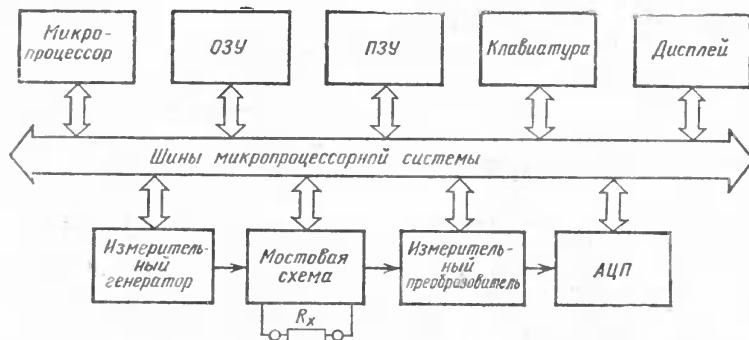


Рис. 4. Структурная схема измерителя параметров компонентов схем

следуемого компонента: в общем случае устанавливают 1 кГц, при измерении емкости электролитического конденсатора и большой индуктивности — 100 Гц, а в случае измерения малых емкостей и индуктивностей — значение 10 кГц.

После нажатия клавиши включения прибора микропроцессор производит опрос передней панели (положения клавиш), включает измерительный генератор и устанавливает указанное соответствующей клавишей значение частоты его выходного сигнала, подводимого к одной диагонали мостовой схемы, считывает программу, хранимую в ПЗУ и определяющую последовательность операций. В общем случае мостовая схема оказывается разбалансированной и на второй ее диагонали имеется напряжение разбаланса. Оно преобразуется в напряжение постоянного тока и поступает на вход АЦП. Числовой эквивалент напряжения разбаланса передается с выхода АЦП в МС, которая регулирует мостовую схему до наступления состояния баланса. Один из вариантов схемного решения регулировки мостовой схемы заключается в том, что в качестве ее плеч используются цифровые управляемые сопротивления, подобные тем, которые составляют основу ЦАП. Значение сопротивления плеча определяется числом цифрового кода, подаваемого на входы управляемого сопротивления из МС. В случае необходимости осуществляется переключение поддиапазонов измерений.

После достижения баланса мостовой схемы МС проводит необходимые вычисления для получения результата измерения, преобразует его в «дисплейный формат» и выводит на дисплей (обычно жидкокристаллический). Прибор может выполнять многократные измерения и последующую их статистическую обработку.

Важной особенностью микропроцессорного мостового измерителя параметров компонентов является автоматическое определение погрешности, создаваемой остаточным сопротивлением и индуктивностью выводов используемого компонента (при измерении низких импедансов) или паразитной емкостью (при измерении высоких импедансов), а также программно-управляемая подстройка с целью исключения указанной погрешности.

Измеритель амплитудно-частотных характеристик

Он представляет собой осциллографический прибор, предназначенный для визуального наблюдения и исследования АЧХ четырехполюсников. Широко известны аналоговые измерители АЧХ, состоящие из генератора с модулирующей частоты по линейному закону (генератора качающей-

ся частоты), осциллографического индикатора и устройства калировки оси частот. Подобные измерители, как и аналоговые анализаторы спектра, имеют ряд недостатков:

- трудность точной установки частоты и получения высокой разрешающей способности при качании частоты генератора;

- недостаточно высокая линейность оси частот экрана, обусловленная нестрогой линейностью пилообразного напряжения, применяемого для линейно-частотной модуляции гетеродина, и нелинейностью модуляционной характеристики;

- несовпадение (смещение) моментов начала циклов качания генератора; невысокая точность измерения значений напряжения (мощности), соответствующих характерным точкам осциллограммы АЧХ;

- неудобство и значительная погрешность измерения частоты; отсутствие возможности объективного сопоставления данных, получаемых в ходе исследований, и регистрации результатов измерений.

От указанных недостатков свободны цифровые измерители с встроенными микропроцессорами, полностью автоматизирующие процедуру снятия АЧХ четырехполюсника и обладающие всеми достоинствами современных цифровых измерительных приборов. Структурная схема одного из вариантов подобных измерителей приведена на рис. 5.

Работу схемы поясним в предположении, что испытуемый четырехполюсник — усилитель. В соответствии с программой, хранимой в ПЗУ,

микропроцессор направляет слово данных (управляющий код) через интерфейс вывода 1 на входы ЦАП. На его выходе появляется напряжение, значение которого определяет значение частоты сигнала, вырабатываемого генератором. Сначала с помощью мультиплексора подключается вход АЦП через измерительный преобразователь к выходу генератора. После аналого-цифрового преобразования на выходе АЦП образуется цифровой эквивалент амплитуды напряжения генератора. Полученное число передается через интерфейс ввода в ОЗУ. Затем подключается вход АЦП к выходу испытуемого усилителя. Вновь проводится аналого-цифровое преобразование (напряжения $U_{\text{вых}}$), результат которого сравнивается в МС с числом, хранимым после первого аналого-цифрового преобразования. В итоге сравнения вырабатывается слово данных, посылаемое микропроцессором через интерфейс выхода 2 на аттенуатор. Вносимое им затухание изменяется таким образом, что значения выходного напряжения $U_{\text{вых}}$ усилителя и напряжения генератора $U_{\text{г}}$ становятся равными. Это означает равенство затухания аттенуатора и коэффициента усиления усилителя. Слово данных, определяющее затухание аттенуатора, хранится в ОЗУ. Значение коэффициента усиления может быть выведено на дисплей, причем в случае необходимости возможно выражение коэффициента передачи испытуемого четырехполюсника в логарифмических единицах.

В соответствии с заданной прог-

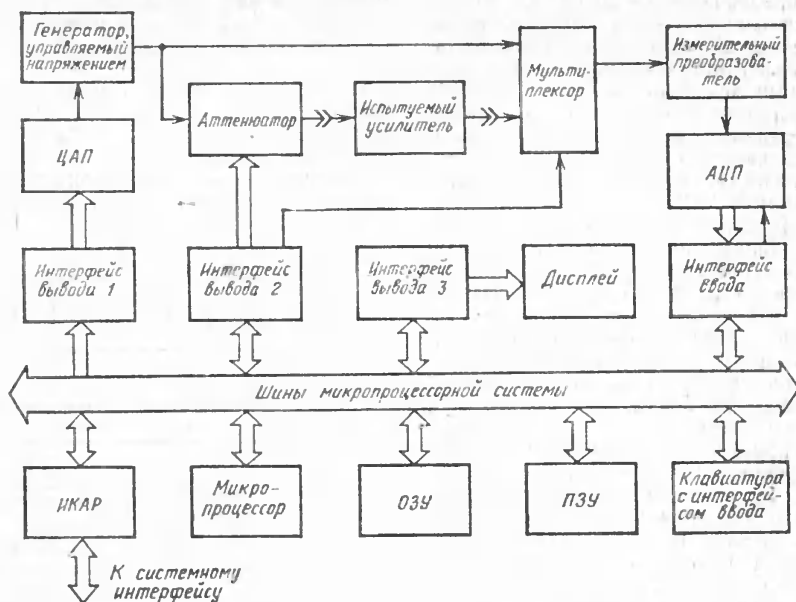


Рис. 5. Структурная схема цифрового измерителя АЧХ четырехполюсника

раммой последовательно устанавливается совокупность значений частоты выходного сигнала генератора, и процедура многократно повторяется. Шаг перестройки частоты определяется характеристикой генератора и числом разрядов ЦАП. На дисплее может быть отображена как совокупность числовых значений коэффициента передачи четырехполосника, так и кривая, отображающая его АЧХ.

Наличие интерфейсной карты (ИКАР) позволяет включить цифровой измеритель АЧХ через приборный интерфейс в измерительную систему. При этом появляется возможность дистанционного управления работой прибора.

Цифровые автоматические измерители АЧХ четырехполосников строятся также на основе синтезатора частоты. Управляемый МС синтезатор вырабатывает сигналы определенной сетки значений частоты. Он выполняет роль измерительного генератора. В остальном работа такого прибора аналогична работе рассмотренного измерителя АЧХ.

Измерительный генератор

Все более широкое распространение получают измерительные генераторы сигналов с встроенной МС. Такие приборы отличаются широкими возможностями и высокими метрологическими характеристиками. Обладая всеми общими преимуществами микропроцессорных приборов, измерительные генераторы сигналов имеют ряд специфических свойств, обусловленных назначением — служить источниками сигналов различных форм и частот.

Прежде всего отметим, что программируемые измерительные генераторы по сравнению с обычными приборами этой группы выделяются расширенными возможностями управления при относительно простой передней панели. Требуемое значение частоты выходного сигнала набирается на клавиатуре, и МС, опросив клавиатуру, автоматически устанавливает набранное значение с высокой точностью. Имеется возможность автоматического одно- или многократного приращения значения частоты в установленных пределах через заданный интервал времени. В приборах, содержащих генераторы с механической перестройкой частоты (на что затрачивается время от 0,5 до 10 с), предусматривается подача команды Ожидание, по которой на время перестройки затемняется дисплей, отображающий параметры сигнала.

Наличие в приборе памяти для запоминания значений частоты позволяет многократно выбирать из памяти хранимые там данные и автоматически повторять установку частоты по заданной программе.

Возможна ситуация, когда требуе-

мое значение частоты находится близко к конечному значению поддиапазона. Так как поддиапазоны переключаются автоматически, то небольшое изменение частоты может повлечь за собой переход на другой поддиапазон. Избежать этого позволяет работа в режиме «удержания частоты», устанавливаемого нажатием соответствующей клавиши.

Большие удобства при испытании ряда устройств создает автоматическое изменение глубины амплитудной модуляции или девиации частоты при частотной модуляции по определенной программе и в том числе свипирования, т. е. линейного качания частоты.

Многие из отмеченных достоинств измерительных генераторов гармонических сигналов в разной мере присутствуют (с учетом специфики) генераторам импульсных сигналов.

Существенно расширяют возможности измерительного генератора автоматическая установка требуемого уровня выходного сигнала и поддержание неизменного значения при перестройке частоты в определенной области, а также калибровка уровня и отображение его значения в различных видах: абсолютных единиц, относительных единицах, децибелах (по отношению к принятому за единицу значению). Во многих приборах предусмотрено автоматическое выполнение диагностических процедур.

Говоря о высоких метрологических характеристиках измерительных генераторов, содержащих МС, следует отметить высокую точность установки частоты (у некоторых генераторов разрешающая способность 0,05 % и выше), очень хорошую стабильность установленного значения (например, нестабильность менее 0,001 %), малые нелинейные искажения синусоидального сигнала, а также высокую точность фиксации длительности и частоты следования импульсов.

Наконец, необходимо подчеркнуть, что все программно-управляемые генераторы могут быть подключены к интерфейсной шине, в результате чего возможно не только ручное (с передней панели) управление прибором, но и дистанционное, осуществляемое контроллером.

Рассмотрим несколько видов измерительных генераторов, содержащих МС.

Генераторы синусоидальных сигналов, формируемых методом аппроксимации. Сначала кратко остановимся на принципе формирования. Наиболее простой вид аппроксимации — ступенчатая. Она предполагает представление (замену) синусоидального напряжения напряжением ступенчатой формы, весьма мало отличающейся от синусоидальной кривой (рис. 6, а).

Сущность ступенчатой аппроксимации можно пояснить следующим образом. Аппроксимируемое синусоидальное напряжение дискретизируют по времени (равномерная дискретизация с шагом ΔT) и в интервале времени, разделяющем два соседних момента t_i и t_{i+1} , заменяют синусоидальное напряжение напряжением постоянного тока — ступенькой, высота которой равна значению аппроксимируемого напряжения в момент t_i . Таким образом, реально вместо кривой синусоидальной формы получается ступенчатая кривая.

При данном периоде T синусоидального напряжения число p ступенек, приходящихся на один период, определяется шагом дискретизации и равно отношению периода к шагу. Если же задано число ступенек, то изменение шага дискретизации приводит к изменению периода формируемого напряжения.

Очевидно, что ступенчатая кривая тем лучше аппроксимирует синусоиду (тем меньше погрешность аппроксимации), чем больше число ступеней p . Когда это число достаточно велико, сформированное ступенчатое напряжение можно рассматривать как низкочастотное синусоидальное напряжение, искаженное в небольшой степени высокочастотной аддитивной помехой. Сравнительно просто можно осуществить высококачественную фильтрацию, резко ослабляющую высшие гармоники, т. е. получить синусоидальное напряжение, характеризующееся очень малым коэффициентом гармоник.

На рис. 6, б приведена упрощенная структурная схема устройства, формирующего ступенчатую кривую. Кварцевый генератор вырабатывает периодическую последовательность коротких импульсов с периодом следо-

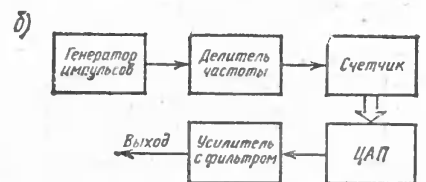
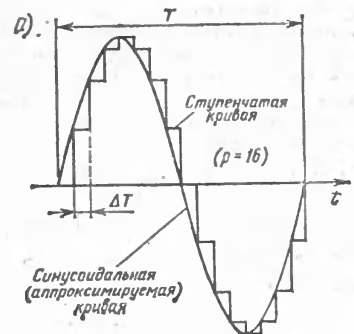


Рис. 6. К принципу ступенчатой аппроксимации

вания $T_{кв}$. На выходе делителя частоты с регулируемым коэффициентом деления q получается последовательность импульсов с периодом следования, в q раз большим, чем $T_{кв}$. Он задает шаг дискретизации. Импульсы поступают в счетчик емкостью p . Кодовая комбинация, определяемая числом i импульсов, накопленных в счетчике, передается в функциональный цифроаналоговый преобразователь. Последний вырабатывает напряжение, соответствующее числу i . Таким образом, формируются p ступенек. После накопления p импульсов счетчик переоплачивается и сбрасывается на нуль. С приходом $(p+1)$ -го импульса начинается формирование нового периода ступенчатой кривой.

Частоту формируемого сигнала при фиксированном числе p ступеней регулируют, изменяя шаг дискретизации ΔT , что достигается изменением коэффициента деления q делителя частоты.

Применение микропроцессорной системы упрощает аппаратную часть прибора (рис. 7). В этом случае достаточно иметь линейный ЦАП и таблицы значений, определяющие форму сигнала (например, синусоидальную), «защитые» в ПЗУ. Изменение частоты сигнала достигается изменением скорости выборки значений из памяти. Очевидно, что стабильность частоты формируемого сигнала определяется стабильностью тактового генератора МС.

Широкодиапазонный генератор синусоидальных сигналов. Подобные генераторы выполняются на основе сочетания синтезатора частоты и МС (рис. 8). Так как синтезатор может генерировать сигналы с широкой сеткой частот, то становится очевидной возможность создания широкодиапазонного генератора при высокой разрешающей способности перестройки частоты.

Для пояснения принципа действия программно-управляемого измерительного генератора, синтезирующего гармонические сигналы в широком

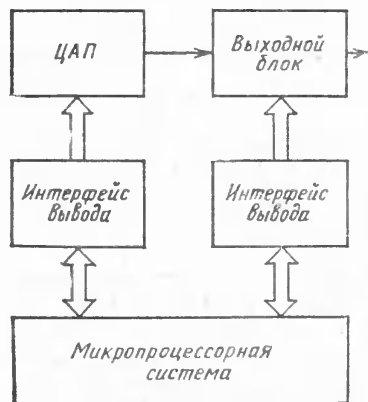


Рис. 7. Структурная схема генератора синусоидальных сигналов

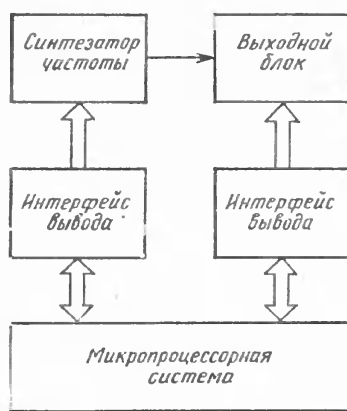


Рис. 8. Структурная схема широкодиапазонного измерительного генератора

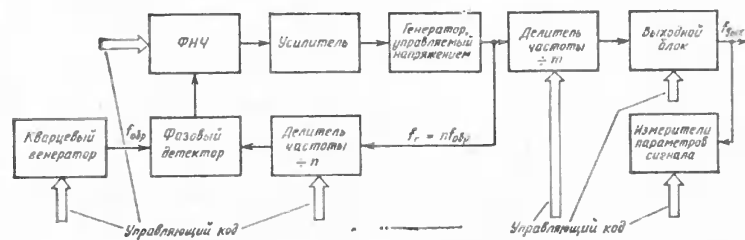


Рис. 9. Структурная схема программно-управляемого генератора

диапазоне частот, воспользуемся его структурной схемой, детализирующей устройство синтезатора (рис. 9). В этой схеме синтезатор частоты состоит из кварцевого генератора, вырабатывающего сигналы образцовой частоты $f_{обр}$, фазового детектора, ФНЧ, усилителя, генератора, управляемого напряжением, и делителя частоты с коэффициентом деления n . Как видно из схемы, кварцевый генератор и делитель частоты — программно-управляемые блоки. На их цифровые входы подается управляющий код из МС.

Генератор, управляемый напряжением, вырабатывает напряжение синусоидального сигнала, значение частоты f_r которого в n раз выше значения $f_{обр}$. Напряжение передается через делитель частоты с коэффициентом деления n на фазовый детектор, вырабатывающий напряжение рассогласования. Оно поступает через ФНЧ и усилитель на генератор и поддерживает точное значение частоты f_r вырабатываемого им сигнала. Сигнал передается через делитель частоты с коэффициентом деления m (изменяемым кодом, поступающим из МС) и выходной блок. Таким образом, значение частоты сигнала, снимаемого с выхода измерительного генератора, равно значению $f_{обр}$, умноженному на отношение m/n . Изме-

няя программным путем значения m , n и $f_{обр}$, можно автоматически перестраивать в широких пределах частоту выходных сигналов измерительного генератора с малым шагом (высокой разрешающей способностью), решать задачу автоматического приращения значения частоты и т. п.

Автоматическая регулировка уровня выходного сигнала, изменение единиц измерения, вида представления результата измерения достигается подачей управляющего кода из МС на цифровые входы выходного блока и измерителей параметров сигнала.

Из синтезированных сигналов синусоидальной формы могут быть сформированы и сигналы других форм. Таким образом, можно построить программируемый генератор сигналов специальных форм (функциональный генератор).

Генератор периодической последо-

вательности прямоугольных импульсов. Принципы построения подобных программно-управляемых генераторов разнообразны. Один из них иллюстрирует структурная схема, представленная на рис. 10.

Принцип формирования прямоугольного импульса требуемой длительности заключается в следующем. Запускающий сигнал поступает через цифровую схему задержки на вход R триггера и перебрасывает его из состояния 0 в состояние 1. На выходе триггера образуется положительный перепад напряжения u , следовательно, с этого момента на вход 1 временного селектора подан разрешающий потенциал. Через временной селектор на вход счетчика начинают проходить счетные импульсы, подводимые ко входу 2 селектора от генератора. В счетчик, общая емкость которого C , предварительно вводится из МС число A , причем разность m чисел C и A зависит от того, какую длительность прямоугольного импульса требуется получить. После того, как в счетчик поступит m счетных импульсов и накопленное в нем число будет равно сумме чисел m и A , т. е. числу C , счетчик переполнится. Образующийся на выходе счетчика сигнал переполнения подается на вход S триггера, возвращая его в исходное состояние 0. Напряжение на выходе триггера резко падает, со

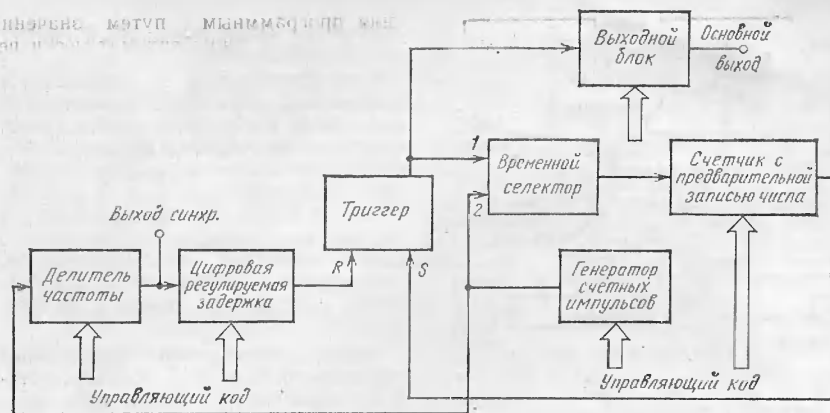


Рис. 10. Структурная схема генератора прямоугольных импульсных сигналов

хода 1 временного селектора снимается разрешающий потенциал и прохождение счетных импульсов в счетчике прекращается.

Сформированный на выходе триггера прямоугольный импульс, длительность которого в m раз больше периода $T_{сч}$ следования счетных импульсов, передается через выходной блок на основной выход измерительного генератора. Для изменения длительности импульса необходимо изменить либо число m , т. е. число A ,

вводимое МС в счетчик, либо период $T_{сч}$. Регулировка частоты следования импульсов достигается изменением коэффициента деления частоты с помощью управляющего кода, вырабатываемого МС. Аналогично изменится продолжительность задержки. Амплитудой выходного сигнала также управляет МС, посылая управляющие сигналы на цифровую шину выходного блока.

Возможны и иные варианты построения программно-управляемых генераторов.

1. Мирский Г. Я. Микропроцессоры в измерительных приборах.— М.: Радио и связь, 1984.— 160 с.
2. Интерфейс для программируемых приборов в системах автоматизации эксперимента / П. И. Гореликов, А. Н. Домарацкий, С. Н. Домарацкий и др.; Под ред. Л. С. Ситникова.— М.: Наука, 1981.— 262 с.
3. Науман Г., Майлинг В., Шербина А. Стандартные интерфейсы для измерительной техники: Пер. с нем./ Под ред. А. С. Бондаревского.— М.: Мир, 1982.— 304 с.
4. Мирский Г. Я. Радиоэлектронные измерения.— М.: Энергия, 1975.— 600 с.
5. Bologn A. Ein neues Verfahren zur Frequenzmessung von Mikrowellensignalen unter Verwendung von Mikroprozessoren.— Nachr. Electron., 1980, 39, № 5, s. 149—152.
6. Ikeda H., Takatsuka H., Tamamoto H. A fully automatic measuring device for frequency characteristics of operational amplifiers.— EMM—IC'81: Elec. and Electron Measur. and Test Instruments Conf. and Exps., Citawa, 1981, Sept. 22—24.— New York, 1981, p. 172—175.

Статья поступила 21 мая 1985 г.

УДК 681.326—181.4

А. Ю. Сасов

МИКРОТОМОГРАФИЯ И ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ НА МИКРОЭВМ «ИСКРА 226»

Комплекс РЭМ (растровый электронный микроскоп) — микроЭВМ (см. фото на вкладке), ориентированный на количественный анализ рентгеновской и электронномикроскопической информации, позволяет получать принципиально новую информацию о структуре исследуемых объектов на микроном и субмикронном уровнях.

1. Комплекс РЭМ — микроЭВМ

Основной принцип создания аппаратуры сопряжения РЭМ с микроЭВМ — обеспечение максимальной программной гибкости системы. Этого можно достичь оптимальной конфигурацией аппаратной части.

В комплексе РЭМ — микроЭВМ РЭМ используется практически без переделок. Он связан с блоком сопряжения (рис. 1), установленным в микроЭВМ и использующим ее источники питания. При совпадении адреса, выдаваемого на информацион-

ную магистраль ввода-вывода (ИМВВ), с адресом блока сопряжения, заданным аппаратно (при помощи ИС К155ЛА2 с необходимым числом инверторов на входах, показанных штриховой линией на рис. 1) узел связи по команде «Установить связь» (УС) на магистрали команд (МК) подключает блок сопряжения к микроЭВМ. При этом перебрасывается триггер адреса блока сопряжения и формируется сигнал «Команда выполнена» установкой в 0 второго разряда магистрали состояний (МС). Такой же ответ будет сформирован на все запросы до получения команды «Установить связь» с «чужим» (для данного блока) адресом. Команды, поступающие от микроЭВМ, дешифруются на ИС К155ИД3. Такой простейший узел связи используется и в остальных блоках для «Искры 226» (см. ниже).

Блок сопряжения РЭМ с микроЭВМ содержит АЦП на основе РПП К155ИР17; два счетчика на 10 бит с ЦАП, формирующие аналоговые напряжения разверток РЭМ; регистр на К155ТМ5 с ЦАП для формирования выходного видеосигнала и четыре триггера, переключающих режимы работы с помощью реле, установленных в РЭМ.

АЦП работает методом последовательных приближений; время преобразования — 10 мкс, причем по команде «Выдать байт» (ВБ) с микроЭВМ выдается информация предыдущего цикла оцифровки, а затем запускается новый цикл. Это делает работу АЦП «прозрачной» для процессора микроЭВМ.

Во время совместной работы РЭМ с микроЭВМ электронный зонд устанавливается цифровыми разверт-

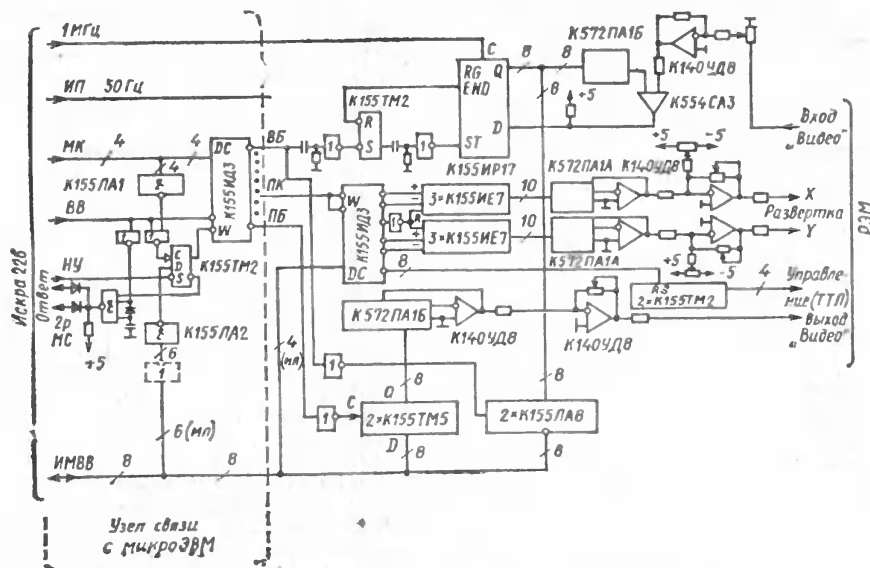


Рис. 1. Схема блока сопряжения РЭМ с микроЭВМ

ками в необходимые точки объекта и сигнал с помощью АЦП оцифровывается. Сканирование зонда по объекту и луча по экрану РЭМ происходит синхронно, поэтому формировать изображение на экране РЭМ можно с помощью тех же цифровых разверток и ЦАП. Изображение формируется при перемещении зонда не только по растру, но и по любой программно-заданной кривой.

Автор работал с микроЭВМ «15ВСМ5», «Электроника Т3-16», «Электроника Д3-28», «Искра 1256», «Искра 226» (исп. 1—3,6). Наиболее

подходит микроЭВМ «Искра 226» в исполнении 3: с накопителями на гибких и жестких магнитных дисках, накопителем на магнитной ленте, печатающим устройством, графопостроителем.

Отличительная особенность микроЭВМ «Искра 226» — возможность вывода на встроенный дисплей не только знаковой, но и графической информации. Для расширения возможностей комплекса печатающее устройство «Роботрон 1154» переделано для вывода как знаковой (прописных и строчных букв русского и

латинского алфавитов), так и графической информации.

Устройство ввода с телекамеры существенно расширило возможности микроЭВМ в обработке изображений, негативов, фотографий реальных объектов, видеозаписей кадров оптического микроскопа и т. д. Устройство ввода с телекамеры состоит из монитора, содержащего схемы селектора синхроимпульсов, развертки и видеоусилитель, и собственно интерфейса ввода с телекамеры.

Автор использовал самодельный монитор на базе ЭЛТ 16ЛК2Б, установленный внутри «Искры 226» справа над блоком питания (на месте КНМЛ); можно использовать и любой серийный монитор.

Интерфейс ввода с телекамеры (рис. 2), кроме узла связи с микроЭВМ, содержит счетчик положения и строки курсора, аналоговый компаратор положения и цифровой компаратор номера строки, устройство выборки-хранения (УВХ) и АЦП на РПП К155ИР17. Эксперименты показали, что время выборки УВХ 1100СК2 300... 500 нс, поэтому блок может вводить изображения формата 128×128 точек, а при небольшой программной коррекции — 256×256 точек. АЦП формирует 8-битовый код, из которого 7 старших бит выдаются на микроЭВМ, а младший бит индицирует готовность к выдаче информации. Программно изображение вводится не по строкам, а по столбцам, так как за время ввода одной телевизионной строки (64 мкс) блок успевает оцифровать сигнал одной точки изображения и провести обмен с микроЭВМ. Все изображение вводится за несколько секунд.

Для расширения возможностей отображения информации разработан цветной полутонный дисплей (рис. 3), содержащий узел связи с микроЭВМ типа «Принять команду» (ПК) может (в зависимости от содержания команды) устанавливаться цвет или изменяться положение точки записи. По команде «Принять байт» (ПБ) в память записываются четыре старших бита (интерпретируются как яркость) и прибавляется единица в счетчиках курсора. Это позволяет организовывать вывод изображения как подачу команды установки курсора в нача-

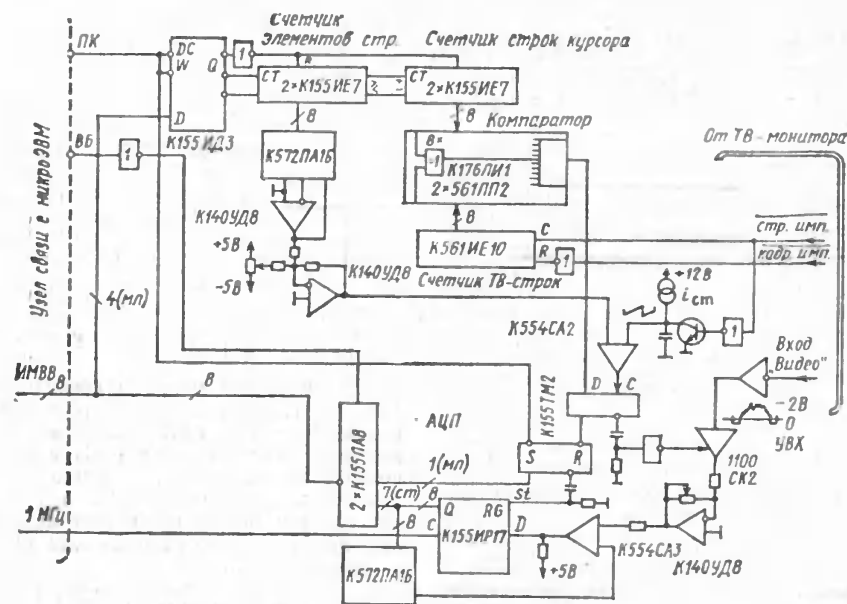


Рис. 2. Схема интерфейса ввода с телекамеры

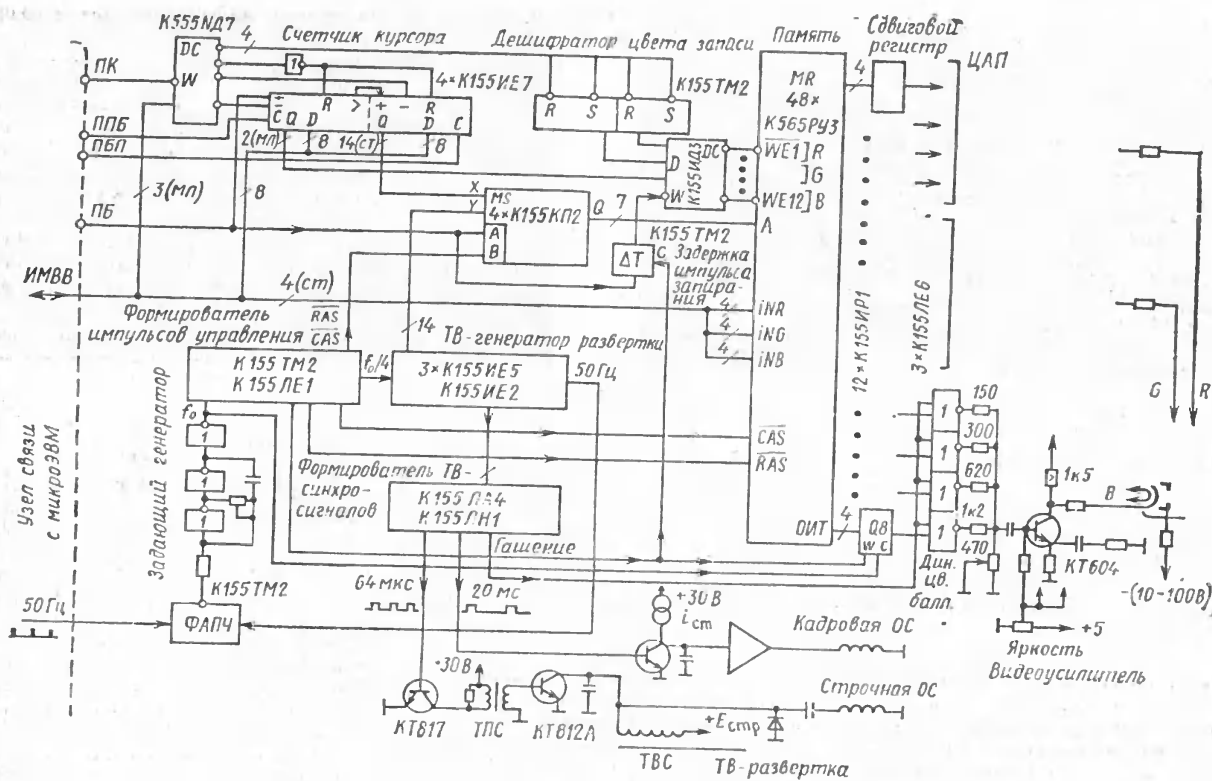


Рис. 3. Схема цветного полутонного дисплея

ло координат и 256² команд ПБ. Элементы аналоговых цепей зависят от типа ЭЛТ. В нашем случае используется ЭЛТ 40ЛК1Ц.

Для решения специальных задач разработан ряд нестандартных устройств. Одно из них — интерфейс ввода информации с рентгеновских пропорциональных счетчиков, состоящий из усилителя, дискриминатора и счетчика импульсов (16 бит) с программно-управляемым временем счета. Другое специализированное устройство — четырехканальный двунаправленный коммутатор последовательно канала протокола С2, позволяющий использовать последовательный интерфейс «Искра 015—85» для обмена с тремя стандартными терминалами и с одним аналогичным устройством — через телефонную линию. Кроме того, для микроЭВМ «Искра 226» разработаны программаторы ППЗУ К556РТ5, Р573РФ2 и РФ5. Блок питания микроЭВМ «Искра 226» позволяет подключать к нему все специализированные устройства и блоки сопряжения.

Цифровая обработка изображений

Обработку изображений условно делят на три части [1]: улучшение и реставрация изображений; анализ, описание и классификация; кодирова-

ние и передача изображений. При улучшении изображений повышается субъективное качество информации и она преобразуется для машинной обработки. Реставрация — восстановление изображений с коррекцией искажений (обычно — аппаратных).

Простейшее улучшение изображений — это линейные операции преобразования яркости и контрастности аналоговыми методами. Объективная характеристика изображения при преобразованиях — гистограмма яркости. Для автоматического анализа целесообразно приводить гистограмму яркости к линейному виду, для наилучшего наблюдения — к гиперболическому (с учетом нелинейности зрительной системы человека). По выборке 128×128 точек разработанные программы определяют гистограмму яркости изображения, вычисляют индивидуальную нелинейную функцию преобразования для приведения гистограммы к заданному виду и проводят это преобразование на аналоговых и цифровых развертках. Для борьбы с шумами, если о характеристиках шума нет достаточной информации, разработанная программа фильтрует шумы дискретной сверткой при анализе девяти соседних точек каждого элемента изображения.

Использование цветных полутонных дисплеев открывает широкие воз-

можности для решения задач электронной микроскопии. На микроЭВМ «Искра 226» реализованы следующие программы:

Цветовое контрастирование изображений. Преобразование «интенсивность — цвет» производится с заданной непрерывной или дискретной шкалой, а также по закону, заданному пользователем.

Цветокодирование топографии или состава образцов в метрологических режимах РЭМ. При анализе состава определенные цвета привязываются к коэффициенту отражения электронов на эталонных образцах, а при переходе на исследуемый объект отображается цветовая карта распределения состава на поверхности.

Вывод на одном изображении в разных цветах различных сигналов об объекте (состав, топография, электрически активные области, катодолюминесценция и т. д.). Изображения в разных режимах РЭМ считываются последовательно в ЭВМ, а воспроизводятся одновременно на цветном дисплее.

Цветоделение стереопар. Одно из изображений объекта после обработки в микроЭВМ выводится на дисплей в синем цвете, другое — после наклона образца — в красном цвете. Благодаря светофильтрам наблюдается объемный эффект.

Сравнение изображений

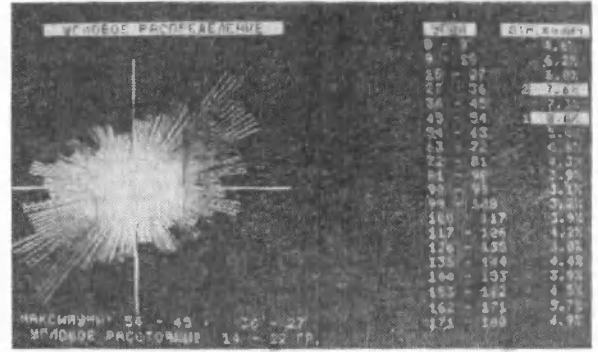
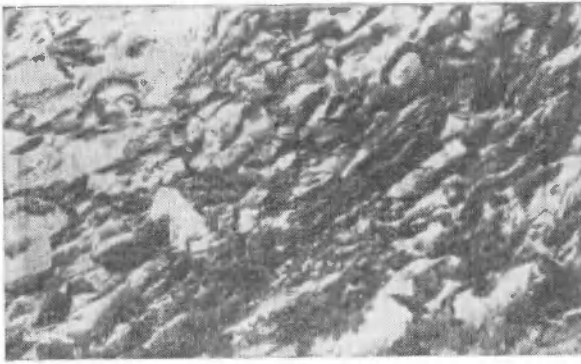


Рис. 4. Микроструктура ленточной глины (а) и результаты ориентационного анализа (б)

путем их отображения в дополнительных цветах.

Все программы обработки цветных изображений строятся на основе программных блоков обмена РЭМ с микроЭВМ и с дисплеем массивами 208×250 точек (с запоминанием в ЭВМ) или 256×256 точек (без запоминания).

Реставрация изображений — это в основном исключение аппаратных искажений. В электронной микроскопии эти искажения связаны с конечным размером электронного зонда РЭМ и области генерации сигналов в образце. Электронный зонд имеет примерно гауссово распределение плотности тока, и реставрация, направленная на повышение пространственного разрешения, сводится к нахождению обратного преобразования изображения. Такое преобразование увеличивает вклад высокочастотных пространственных составляющих для улучшения передачи мелких деталей изображения (инверсная фильтрация). Чтобы избежать появления чрезмерного высокочастотного шума, коррекция ограничивается с учетом спектральных характеристик шума (Винеровская фильтрация). Программа реставрации РЭМ-изображений путем Винеровской фильтрации производит

быстрое двумерное преобразование Фурье массива 128×128 точек, умножение в Фурье-пространстве на специально вычисленную корректирующую функцию и обратное быстрое преобразование Фурье. Время реставрации изображения на микроЭВМ «Искра 226» — около 6 мин. Максимально достигнутое разрешение РЭМ — 1,2–1,5 нм.

Анализ, описание, классификация изображений. Результат анализа изображений — это обычно ряд числовых данных или зависимостей характеристик изображения. Анализу подвергаются полутоновые изображения и бинарные (содержащие две градации яркости). Бинарные изображения получают из полутоновых на основе априорных сведений в режиме диалога оператора с микроЭВМ. При исследовании микроморфологии РЭМ выделяет поры или частицы на изображении. Разработаны следующие программы морфологического анализа по РЭМ-изображениям.

Экспресс-анализ (время работы — 3 с). Вычисляется общая площадь и периметр пор (или частиц), средние площадь, периметр и диаметр одной поры (частицы), пористость, распределение по размерам. Анализируется изображение 128×128 точек на основе статистических закономерностей [2].

Полный анализ для поля 128×128 точек (~1 мин). Кроме перечисленных параметров, определяются число пор (частиц), связанных пор (контактов между частицами), средние размеры и форма пор (частиц). В процессе вычислений индивидуально в ЭВМ анализируется каждая пора обходом по границе со сглаживанием по 5 точкам, анализом локального радиуса кривизны границы и последующим исключением проанализированной части структуры.

Полный анализ для поля 1024×1024 точки (20–40 мин). Крупные поры анализируются при разбиении 128×128 точек, мелкие — прохождением на этом же изображении 64 непересекающихся полей по 128×128 точек.

Анализ по сопряженным поверхностям. Для исключения из анализа ложных пор, являющихся следами от частиц, оставшихся на другой стороне раскола образца, анализируется суммарное изображение двух половин раскола. Точное совмещение производится на цветном дисплее при выводе разных поверхностей раскола в разных цветах. Выходная информация аналогична получаемой в программе 2.

Полутоновые изображения исследуются на основе яркостных или про-

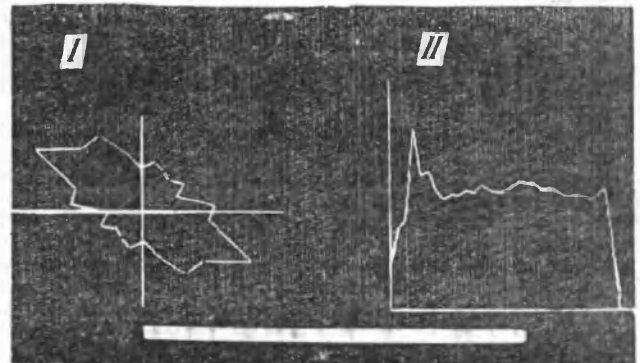
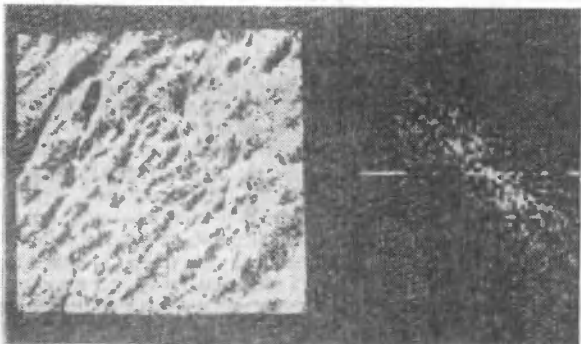


Рис. 5. Микроструктура и ее двумерный Фурье-спектр (а); ориентационные и гранулометрические характеристики этого образца (б)

страйственно-спектральных признаков. Программа строит розу ориентации структурных элементов изображения (рис. 4) на основе исследования локальных амплитуд и направления градиента сигнала в каждой точке поля 128×128 по алгоритму, изложенному в работе [3]. Наиболее полная информация об образце — Фурье-спектр изображения — центрально симметричен, причем радиальная составляющая соответствует гранулометрии структуры, а угловая — ориентации.

Программа анализа и классификации изображений по пространственно-спектральным признакам включает в себя быстрое двумерное преобразование Фурье, интегрирование в Фурье-пространстве по угловым и секторным сегментам и сравнение полученных характеристик для ряда образцов (рис. 5).

Задачи кодирования изображений встречаются в РЭМ и в системах обработки изображений при передаче информации и при создании банков данных. Объем информации при создании банков данных уменьшается в зависимости от назначения банка. Для идентификации изображений человеком необходимо существенное сжатие (до 1 бита на точку). Эффективные методы сжатия — это дельта-модуляция на изотропных развертках [4, 5] и двухканальные методы преобразования [1].

Обработка изображений в кодах процессора «Искра 226» дала выигрыш в быстродействии в 60...100 раз по сравнению с аналогичными программами на языке Бейсик.

Микротомография: аппарат, программное обеспечение, примеры

Микротомография — метод исследования внутренней трехмерной структуры микрообъектов без их разрушения. Основу метода составляет математический аппарат восстановления изображений по проекциям [6]. Проекционные данные можно получить при просвечивании объекта каким-либо слабопоглощаемым излучением: рентгеновским или инфракрасным. Для высокого пространственного разрешения источник излучения должен иметь малые размеры и создается установкой под острофокусированным электронным зондом РЭМ наклонной мишени. Сканирование зонда по мишени создает перемещающийся точечный источник рентгеновского излучения. Оно проходит через объект исследования (частично поглощаясь) и регистрируется рентгеновским пропорциональным счетчиком. Сигнал со счетчика поступает на микроЭВМ «Искра 226» через специализированный

интерфейс, о котором говорилось в первой части статьи. Для получения проекционных данных под разными углами объект поворачивается прецизионным шаговым двигателем (программно-управляемым) той же микроЭВМ «Искра 226». Алгоритм нахождения трехмерного изображения объекта по проекционным данным — обратное проецирование исходных данных, коррекция в Фурье-пространстве с помощью некоторой функции и обратное двумерное преобразование Фурье, а иногда (более простой путь) — предварительная одномерная обработка проекционных данных с последующим обратным проецированием без перехода в Фурье-пространство. Подробно алгоритмы реконструкции по проекциям изложены в работах [6, 7].

Программное обеспечение микротомографии для микроЭВМ «Искра 226» состоит из трех подпрограмм:

1. Подпрограмма проверки и юстировки аппаратной части управляет шаговым двигателем поворота объекта, получением изображения проекции образца для юстировки геометрии системы сбора данных и т. д.

2. Подпрограмма алгоритма свертки с обратным проецированием собирает данные (120 проекций по 128 точек) и, получив очередную проекционную строку, запускает поворот шагового двигателя вращения образца. За время поворота (~ 1 с) производится свертка с обратным проецированием данных и прием следующей проекции. Выходное изображение — массив 80×80 точек по 2 байта. Из таких «слоев» строится трехмерное изображение объекта. Так как только не-

большая часть из полученных 32 000 градаций яркости несет полезную информацию, уровни яркости и конт-



Рис. 6. Внешний вид головы жука (а) и сечения, полученные на микроЭВМ «Искра 226» методами микротомографии (б)

растности на экране устанавливаются в режиме диалога.

3. Подпрограмма реконструкции на основе преобразования Фурье. Сбор информации аналогичен предшествующей программе, реконструкция аналогична программе реставрации изображений (прямое и обратное быстрые дискретные преобразования Фурье).

Для повышения скорости реконструкции все программы написаны в кодах процессора «Искра 226». Кроме того, многие преобразования подготовлены в виде таблиц (преобразование декартовых координат в полярные и др.), находящихся на диске и считываемых в микроЭВМ перед началом реконструкции. Необходимый объем памяти микроЭВМ для восстановления изображения 80×80 точек — около 50К байт. В настоящее время разработан специализированный вычислительный интерфейс микромографии для еще более быстрой реконструкции изображений 128×128 точек при объеме используемой памяти «Искры 226» около 35К. Этот интерфейс содержит таблицы преобразований и некоторые вычислительные блоки, что позволяет распараллелить вычисления на микроЭВМ.

Наиболее эффективны подобные исследования для биологических, медицинских, геологических объектов, изделий микроэлектроники и т. д.

Например, внешний вид (в нестандартном режиме работы РЭМ) головы жука (рис. 6, а) представлен в четырех (рис. 6, б) сечениях этого объекта, полученных восстановлением по проекциям без разрушения (снимки с экрана ЭВМ). Положение сечений реконструкции указано на рис. 6, а. Практически никакие методы препарирования и изучения не могут дать столь достоверной информации о внутреннем строении образца (в данном случае — головы жука), так как размер объекта — около 600 мкм, а точность восстановления до 8 мкм).

Для анализа более крупных объектов, но соответственно с худшим пространственным разрешением целесообразнее использовать информацию не с РЭМ, а с рентгенотелевизионного микроскопа. Аппаратная и программная части системы микромографии изменяются при этом незначительно.

Рассмотренные в данной работе аппаратура и программы для обработки РЭМ-изображений на микроЭВМ серии «Искра» успешно внедрены во ВНИИМедполимеров, ИФТТ, на Сумском заводе электронных микроскопов и др.

За справками по вопросам передачи документации и дублирования аппаратуры обращаться по адресу: 119899, МГУ, Ленинские горы, Геологический факультет.

ЛИТЕРАТУРА

1. Прэтт У. Цифровая обработка изображений.— М.: Мир, 1982, т. 1, 2.
2. Serra J. Lectures on image analysis by mathematical morphology.— Tokyo Noko University, 1976, p. 1—125.
3. Tovey N. K. Digital computer technique for orientation analysis of micrographs of soil fabric.— J. Microsc., v. 120, p. 303—315.
4. Sasov A. Yu., Sokolov V. N., Rau E. I. Special form scan for SEM based on microprocessor and its application.— SEM 82, Chicago III, AMF O'Harre, p. 17—27.
5. Данин Б. Н., Рау Э. И., Савин Д. О., Сасов А. Ю., Спивак Г. В. Изотропные развертки в электронно-зондовых приборах и системах обработки изображений.— Изв. АН СССР. Сер. физ., 1983, № 6, с. 1103.
6. Хермен Г. Восстановление изображений по проекциям (основы реконструктивной томографии).— М.: Мир, 1983, с. 3—349.
7. Рентгенотехника: справочник.— М.: Машиностроение, 1980, кн. 2, с. 319—329.

Статья поступила 20 августа 1985 г.

Окончание. Начало см. на стр. 44

дктора, текстового процессора, редактора файлов на физическом уровне и файлового монитора с простейшими инструментальными возможностями), непосредственный TURE, непосредственный редактор каталога пользователей, ряд практикумов для нужд учебного процесса (например, альфа-практикум, позволяющий новичку, впервые в жизни увидевшему терминал, за полчаса ввести и отладить две-три простейшие линейные программы) и др. [12, 13].

За справками обращаться в лабораторию вычислительных методов механико-математического факультета МГУ по понедельникам с 10 до 15 ч (телефон: 139-39-57).

ЛИТЕРАТУРА

1. Ершов А. П. Об объектно-ориентированном взаимодействии с ЭВМ.— Микропроцессорные средства и системы, 1985, № 3, с. 2.
2. Smalltalk-80. Special Issue — BYTE, 1981, v. 6, N 8.
3. Bruce D. Lisa's Alternative Operation System.— Comp. Des., 1983, v. 22, N 9, p. 159—166.
4. Lemmons P. The Gavilan Mobile Computer.— BYTE, 1983, v. 8, N 6, p. 74—92.

5. Bruce F. Webster. The Macintosh.— BYTE, 1984, v. 9, N 8, p. 238—251.
6. Брябрин В. М. Интеллектуальный интерфейс на основе персональной ЭВМ.— Микропроцессорные средства и системы, 1984, № 3, с. 38—44.
7. Fraser W. A Generalized Text Editor.— Comm. of the ACM, 1980, v. 23, N 3, p. 154—158.
8. Борковский А. Б. Многооконное текстовое взаимодействие с персональной ЭВМ.— Микропроцессорные средства и системы, 1984, № 4, с. 47—50.
9. Манин Ю. И. Математика и физика.— М.: Знание. Сер. Математика и кибернетика, 1979, № 12.
10. Вейценбаум Дж. Возможности вычислительных машин и человеческий разум. От суждений к вычислениям.— М.: Радио и связь, 1982.
11. Лебедев Г. В. Непосредственное редактирование алфавитно-цифровой и графической информации.— М.: ВИНТИ АН СССР, № 7325—84 деп.
12. Варсановьев Д. В., Кушниренко А. Г., Лебедев Г. В. Е-практикум — программное обеспечение школьного курса информатики и вычислительной техники.— Микропроцессорные средства и системы, 1985, № 3, с. 27—32.
13. Кушниренко А. Г., Варсановьев Д. В., Дымченко А. Г., Лебедев Г. В. Практическое программирование. Проектирование и разработка диалоговых систем. Нетрадиционный подход.— М.: Изд-во МГУ, 1985.

Статья поступила 15 октября 1985 г.

ИНТЕЛЛЕКТУАЛЬНЫЙ ТЕРМИНАЛ НА БАЗЕ ДВК-1

При реализации интеллектуального терминала (ИТ) на базе ДВК-1 возникает ряд проблем, связанных с его конструктивными и структурными особенностями. В состав ДВК-1 входят одноплатная микроЭВМ МС 1201.1 и алфавитно-цифровой дисплей 15ИЭ-00-013*. Плата микроЭВМ установлена в корпусе дисплея и содержит процессор с системой команд микроЭВМ «Электроника 60» и выходом на магистральный параллельный интерфейс МПИ, а также подключенные к МПИ ОЗУ объемом 32К слов, контроллеры накопителя на гибких магнитных дисках, байтового параллельного интерфейса (БПИ) и последовательного ввода-вывода (ПВВ). На плате микроЭВМ имеется панель для установки дополнительных ППЗУ емкостью до 4К слов. Адреса ППЗУ могут отображаться на любое место адресного пространства памяти с дискретностью 4К. Начальный адрес отображения ППЗУ в памяти ДВК задается с помощью перемычек.

Для организации бездискового ИТ на базе ДВК-1 необходимо:

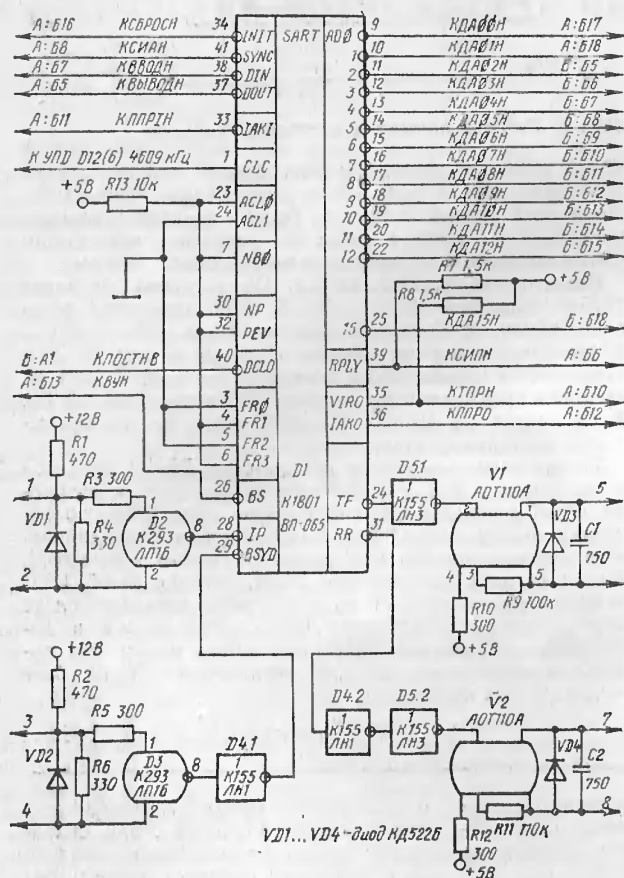


Рис. 1. Канал связи с центральной ЭВМ

* Попов А. А., Хохлов М. М., Глухман В. Л. Диалоговые вычислительные комплексы «Электроника НИЦ-80-20». — Микропроцессорные средства и системы, 1984, № 4, с. 61—64.

реализовать канал связи между ИТ и центральной ЭВМ;

при включении питания ИТ обеспечить запись в ОЗУ микроЭВМ программ загрузки и эмуляции терминала; реализовать средства управления системным таймером.

Канал связи с центральной ЭВМ может быть реализован на основе БПИ, но длина линии связи в этом случае не должна превышать 10 м. Кроме того, исключается возможность подключения автономного печатающего устройства или других дополнительных устройств.

В предлагаемом интерфейсе, реализованном на основе матричной БИСК1801 ВП1-065, предусмотрена возможность квитирования обмена в обоих направлениях передачи данных (рис. 1). Адреса регистров и векторов прерывания являются сменными по группам и задаются состоянием входов 23 и 24. В положении, указанном на схеме, установлен адрес 176560 и вектор 360. Входы 3..6 задают скорость обмена в диапазоне 50...57 600 бод. На схеме указано состояние входов для скорости передачи 9600 бод. Состояние сигнала на выходе 30 разрешает формирование бита паритета и контроля паритета — на схеме в положении «отсутствие контроля». Сигнал на входе 7 — в состоянии «Лог. 1» — задает формат посылки в линии длиной 8 бит.

Информация от внешнего передатчика поступает на вход 28. Сигнал на выходе 31 — готовность ПВВ принимать данные. Если на выходе 31 «Лог. 1», то буферный регистр ПВВ заполнен и передача информации из внешнего передатчика приведет к потере предыдущего байта. Если на нем «Лог. 0», то буферный регистр БИС пуст и схема готова принять данные из линии. БИС передает данные в линию на выход 27 в зависимости от состояния сигнала на входе 26. Если на нем «Лог. 0», то внешний приемник готов к приему и БИС передает очередной байт данных, если «Лог. 1», то ПВВ ждет готовности внешнего приемника. Входы 28 и 29 подключаются к линии через оптронную развязку К293ЛП11, выходы 31 и 27 — через инверторы и оптронную пару АОТ110А. Для увеличения скорости обмена свыше 9600 бод АОТ110А необходимо заменить на АОТ123.

Для реализации средств начальной загрузки ИТ также существует несколько возможностей. Можно использовать панель с ПЗУ, содержащим программы загрузки и эмуляции терминала. Однако в этом случае область ОЗУ емкостью 4К слов исключается из адресного пространства. Учитывая, что программы начальной загрузки и связи занимают около 512 слов, такое решение следует признать нерациональным.

Память целесообразно организовать по принципу теневого ПЗУ (рис. 2). Программы загрузки и эмуляции терминала записаны в микросхемах ПЗУ D6, D7, К556РТ5. На микросхемах D1, D3, D8.2 реализован алгоритм взаимодействия с МПИ. Селектор адреса D3 настроен на адреса нулевого банка ОЗУ. Схема задержки, собранная на VT и D2, поддерживает на D3.9.4 состояние «Лог. 1», в течение времени $T = 0.4...0.5$ с, тем самым подключая ПЗУ в адресное пространство микроЭВМ вместо нулевого банка ОЗУ. Для работы схемы устанавливается режим старта по вектору 24.

После включения питания начинает работать программа, записанная в ПЗУ с адреса, который находится в ячейке 24. Она перисписывает программы загрузки

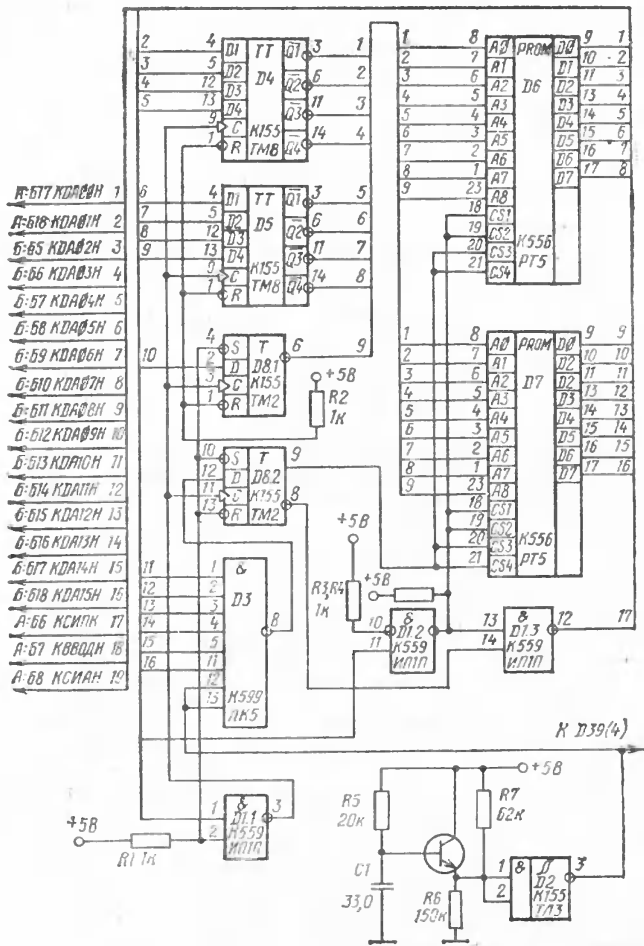


Рис. 2. Теневое ПЗУ

и эмуляции терминала в ОЗУ из ПЗУ и выполняет команду «останов». Чтобы начать работу, необходимо нажать на клавиатуре дисплея клавишу «Р», продолжив тем самым выполнение программы. Останов необходим, чтобы исключить возможные сбросы при отключении ПЗУ и включении в адресное пространство ОЗУ. Переключение происходит при подаче «Лог.0» на D39.4. Переключатель SA3 (4) на плате микроЭВМ должен быть установлен в положение 0.

В ДВК-1 не предусмотрены средства управления системным таймером, хотя микроЭВМ вырабатывает все необходимые сигналы. Режим управления системным таймером можно реализовать либо как в микроЭВМ «Электроника 60М» — установкой тумблера «включено-выключено», либо введением регистра команд и состояний РКС (рис. 3), обеспечивающего

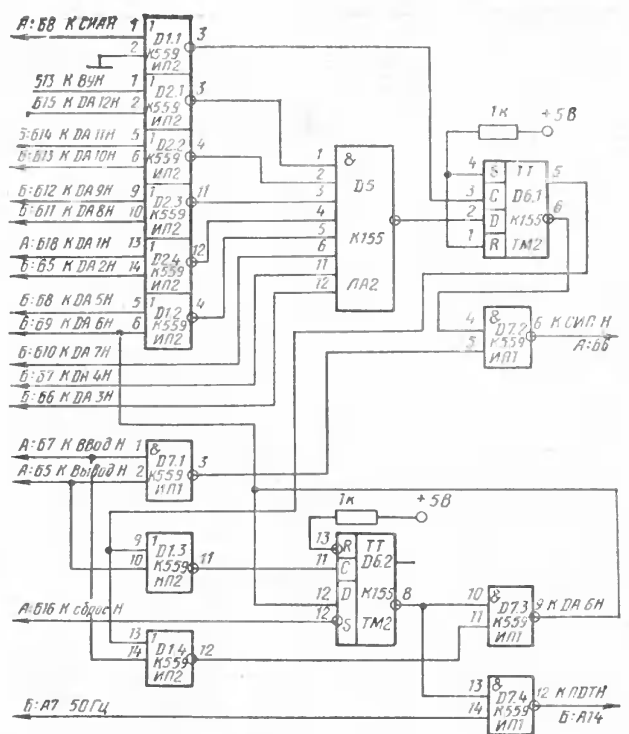


Рис. 3. Регистр команд и состояний таймера

программное включение и выключение таймера путем установки или сброса бита разрешения прерывания. Последний способ является более предпочтительным, поскольку не требует работ по установке коммутационных элементов, а также является более гибким.

Селектор адреса D2, D1.1,2, D5 настроен на адрес 177546. Элементы D1.3,4, D7.1,3, D3.1 реализуют алгоритм обмена с Q-шиной. Триггер D6.2 — РКС (06) — маска прерывания по таймеру. Если на выходе 06 D6.2 «Лог. 1», то сигнал 50 Гц поступает на вход К ПРТ Н, вызывая прерывания по вектору 100 с частотой 50 Гц. Если сигнал на выводе 06 D6.2 «Лог. 0», то прерывания по таймеру запрещены.

Техническая реализация ИТ на базе ДВК-1 сводится к установке в нем дополнительного разъема, к которому через специально разработанную гнемонтажную печатную плату подводится МПИ. Разъем устанавливается непосредственно над левым разъемом микроЭВМ. Канал связи с центральной ЭВМ, теневое ПЗУ, РКС таймера располагаются на одинарной печатной плате конструкции «Электроника 60М», вставляемой в дополнительный разъем МПИ. Эта плата может без доработок использоваться для реализации ИТ на базе «Электроника 60М».

Статья поступила 28 июля 1985 г.

РЖ АВТ-85

9Б128. Стратегия соединения микроЭВМ с большими ЭВМ. 1985, 14, № 3.

За последние месяцы связь микроЭВМ с большим ЭВМ стала важным элементом рынка ВТ. При анализе программного обеспечения (ПО) связи необходимо учитывать следующие аспекты: степень доступности, пропускная способность каналов, удобство пользователей, методы управления доступом. ПО связи можно раз-

бить на 4 категории: имитаторы терминалов, функции которых ограничены связью с редактором текстов на больших ЭВМ при низкой скорости передачи (до 120 символов/с); копирование экрана в файл также при связи с редактором, но при большей скорости (400—500 символов/с); имитаторы устройств, требующие выполнения программ как на большой ЭВМ, так и на микроЭВМ; пакеты, обеспечивающие доступ к БД.

Предложена организация связи, объединяющая преимущества 3-го и

4-го подходов. Она обеспечивает доступ к дискам большой ЭВМ, сокращение затрат на внешнюю память в 20 раз, защиту файлов большой ЭВМ от ошибок пользователей микроЭВМ, поддержку виртуальных внешних устройств. Предполагается широкое использование методов искусственного интеллекта для обеспечения доступа пользователей микроЭВМ к распределенным БД, а также использование преимуществ локальных сетей с файловыми системами больших ЭВМ.

Первая публикация в нашем журнале (№ 3, 1985 г.) краткого описания персональной ЭВМ «Ириша» вызвала много откликов. Авторы писем единодушно в своей просьбе продолжить разговор об «Ирише». Отвечая на запросы читателей, мы публикуем подробную информацию о структуре системной магистрали ПЭВМ и об одном из основных ее модулей — телевизионном адаптере.

В последующих номерах журнала расскажем об остальных функциональных модулях ПЭВМ.

УДК 681.322.1

В. Ю. Романов, В. Н. Барышников, М. А. Воронов, Ф. И. Паначев

ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ ПЕРСОНАЛЬНОЙ ЭВМ «ИРИША»

ПЭВМ «Ириша» состоит из системного блока с источником питания, выносной клавиатуры, цветного или черно-белого телевизионного монитора и ряда периферийных устройств*. Системный блок в наиболее простом исполнении (вариант А) насчитывает два модуля. Каждый такой модуль представляет собой печатную плату размером 150×230 мм, снабженную арматурой для монтажа. Питание блока осуществляется от источника постоянного тока с напряжением 20...27 В. ПЭВМ с системным блоком варианта А предназначена для работы в школьных кабинетах информатики и вычислительной техники (КИВТ).

Внутренний источник питания системного блока преобразует входное напряжение в необходимый для работы модулей ПЭВМ набор напряжений: +5 В (5 А), +12 В (0,3 А), -12 В (0,1 А). Источник питания смонтирован на передней несущей панели корпуса, изготовленной из алюминиевого сплава, являющейся теплоотводом для силовых элементов. В рабочем состоянии он закрыт съемной декоративной крышкой. Такая конструкция обеспечивает удобный доступ к элементам для регулировки и ремонта. Модули расположены в системном блоке вертикально, что обеспечивает достаточно хорошие условия для естественного охлаждения элементов. Более сложный по исполнению системный блок (вариант Б) состоит из восьми модулей и сетевого источника питания (220 В) с выходными напряжениями: +5 В (10 А), +12 В (3 А), -15 В (0,5 А). В нем предусмотрена установка дополнительного трансформатора со специальными экранными обмотками для питания аналоговых схем с возможностью гальванической развязки. Для принудительного охлаждения элементов источника питания и модулей

используется вентилятор. ПЭВМ с блоком варианта Б рассчитан на работу в системах автоматизации научного эксперимента.

Технические характеристики различных моделей ПЭВМ «Ириша» приведены в табл. 1.

Внутренняя магистраль ПЭВМ

Обмен данными между модулями ПЭВМ осуществляется через системную магистраль и стандартные разъемы ГРПМ1-61, одноименные выводы которых соединены между собой. Таким образом, режим работы модулей не зависит от места подключения к магистрали. Структура магистрали и разводка сигналов на разъем ГРПМ1-61 показаны на рис. 1. Назначение линий магистрали — в табл. 2.

Линии, помеченные на рис. 1 звездочкой, зарезервированы для разводки питания к специальным устройствам с аналоговыми цепями, требующими гальванической развязки (для работы системного блока ПЭВМ варианта А не используются).

Модули по отношению к магистрали могут быть пассивными приемопередатчиками информации или активно работающими устройствами, формирующими сигналы на адресной шине и линиях синхронизации. Магистраль допускает подключение нескольких активных модулей, но только один из них выполняет обработку прерываний, разрешает захват магистрали другими активными модулями, формирует сигналы опорных частот.

Системная магистраль ПЭВМ — асинхронная. Цикл обмена информацией начинается по инициативе активного устройства. Пассивное устройство сообщает о завершении приема или передачи информации установкой в низкий уровень сигнала READY. На рис. 2 и табл. 3 показаны временные диаграммы и временные соотношения при выполнении циклов обмена данными.

Обработка запросов от активных модулей на захват магистрали выполняется активным устройством (роль активного устройства играет процессор). Запросы це-

* Барышников В. Н., Воронов М. А., Главный А. В., Паначев Ф. И., Романов В. Ю., Титов О. Ф. Персональная ЭВМ «Ириша» для кабинетов информатики и вычислительной техники — Микропроцессорные средства и системы, 1985, № 3, с. 5-8.

Технические характеристики моделей ПЭВМ «Ириша»

	Модель	Состав	Объем ОЗУ, К байт	Объем ПЗУ, К байт	Число свободных мест для дополнительных модулей	Программное обеспечение	Назначение
	1	2	3	4	5	6	7
Системный блок (вариант А)	ПК01	Процессор, ТВ-адаптер, клавиатура, ч/б или цветной монитор, игровые пульта, индивидуальный выпрямитель, бытовой магнитофон	48	16	2	ФОРТ, монитор в ПЗУ Бейсик (28 К байт с МЛ)	ПЭВМ индивидуального пользования
	ПК02	Процессор, ТВ-адаптер, согласователь интерфейсов, клавиатура, НГМД (203 мм), принтер (УВВПЧ-30-004), ч/б или цветной монитор, игровые пульта, индивидуальный выпрямитель	48	16	1	ФОРТ, монитор в ПЗУ, ОС «Ириша» (аналог ОС 1800), Бейсик, Паскаль, Фортран, Си и другие	ПЭВМ индивидуального пользования
	У01	Процессор, ТВ-адаптер, клавиатура, ч/б или цветной монитор, адаптер локальной сети, игровые пульта	48	16	2	Бейсик, Паскаль через локальную сеть	ПЭВМ на рабочем месте учащегося в составе КИВТ
	ПК03	Процессор, ТВ-адаптер, контроллер НГМД, дополнительное ОЗУ, клавиатура, ч/б или цветной монитор, НГМД (133 мм), принтер (УВВПЧ-30-004), игровые пульта	48/128	28/64	—	ФОРТ, монитор в ПЗУ, ОС «ИРИША» (в варианте с 64 К байт ПЗУ — Бейсик ПЗУ), Бейсик, Фортран, Паскаль, Си и другие	ПЭВМ индивидуального пользования (профессиональный вариант)
Системный блок (вариант Б)	ИВК01	Процессор, ТВ-адаптер, ПЗУ, модули связи с объектом, специальная клавиатура, ч/б или цветной монитор	48	48	5	ФОРТ, монитор в ПЗУ, специальные программы	Встроенная ПЭВМ для систем обработки данных аналитических приборов
	ИВК02	Процессор, ТВ-адаптер, дополнительное ОЗУ, дополнительное ПЗУ, модули связи с объектом, клавиатура, НГМД (133 или 203 мм), принтер (УВВПЧ-30-004) ч/б или цветной монитор	112	28/64	4	ФОРТ, монитор в ПЗУ, ОС «ИРИША», Бейсик, Паскаль, Си, Фортран и другие	Универсальный измерительно-вычислительный комплекс для автоматизации научных исследований

редаются по линии BUSRQ, а сигнал разрешения на захват — по линии BUSEN (рис. 3).

Конструктивно системная магистраль выполнена как печатная плата размером 105×110 мм с четырьмя разъемами ГРПМ1-61ГП2-В, установленными с шагом

23 мм. Плата магистрали монтируется на обратной стороне несущей панели источника питания. Выходные напряжения источника питания подаются непосредственно на соответствующие линии магистрали. Предусмотренная в ПЭВМ «Ириша» возможность организа-

Назначение линий системной магистрали

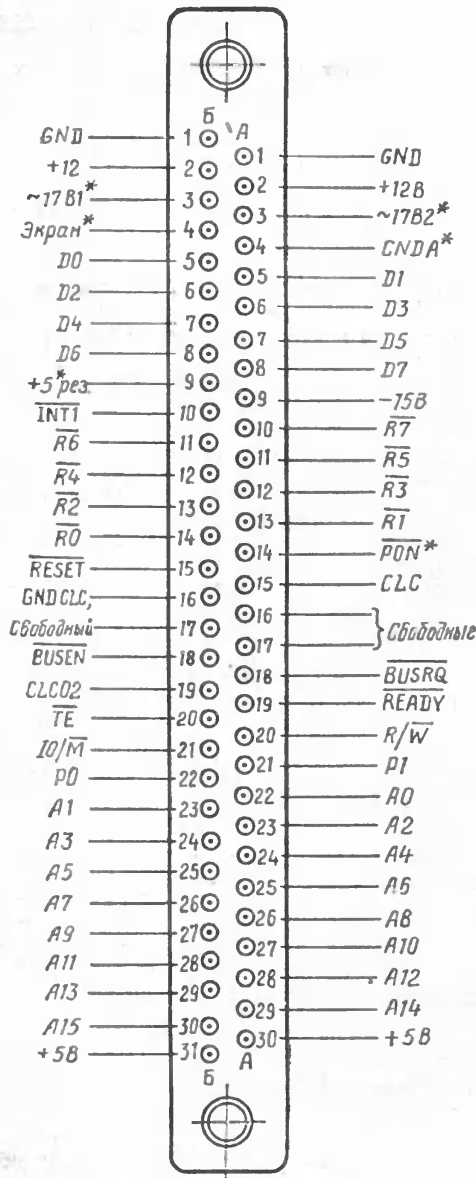


Рис. 1. Структура магистрали и разводка сигналов на разъемах ГРПМ1-61

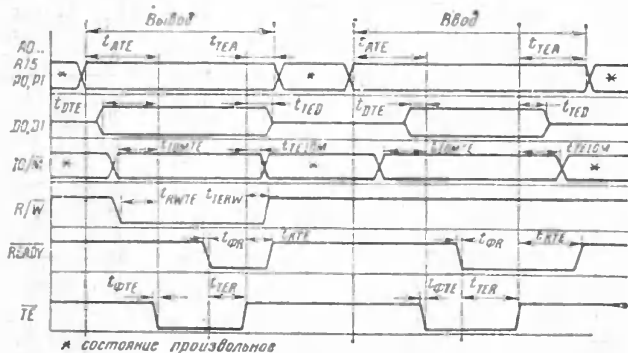


Рис. 2. Временные диаграммы режима обмена данными между модулями

Линия	Назначение
D0 ... D7 A0 ... A15, P0, P1 IO/M	Двунаправленная шина данных Адресная шина
R/W	Сигнал обращения к ВУ (низкий уровень — к памяти)
TE	Направление передачи данных (высокий уровень — к процессору)
READY	Запись данных
BUSRQ	Готовность
BUSEN	Запрос магистрали
INT1, R0 ... R7	Разрешение захвата магистрали
CLC	Запрос прерывания
GNDCLC	Системная опорная частота с отдельной линией заземления
CLC02	Фаза 2 синхронизации процессора
PON*	Наличие всех напряжений (вырабатывается источником питания, сбрасывается при нарушении питания)
RESET	Общий сброс
~17B1* ~17B2* Экран* GND* +5* рез.	Дополнительные линии питания, экран и заземление питания, модулей, требующих гальванической развязки
+5B, +12B, -15(-12)B, GND	Батарейное питание для специальных модулей
	Линии питания модулей и земля
	сигнальная

Таблица 3

Временные соотношения операций обмена данными

Параметр	Наименование	Норма, нс	
		Мин.	Макс.
tATE	Опережение адреса относительно TE	200	
tDTB	Опережение данных относительно TE	200	
tIOMTE	Опережение сигнала IO/M относительно TE	200	
tRWTE	Опережение сигнала R/W относительно TE	200	
tФTE, tФR	Длительность спада		50
tTER	Задержка схемы TE после установки сигнала READY	350	
tTEA, tTED	Время съема адреса и данных после съема TE	100	
tTEIOM, tTERW	Время съема сигналов IO/M и R/W после съема сигнала TE	100	
tRTE	Время съема сигнала READY	100	

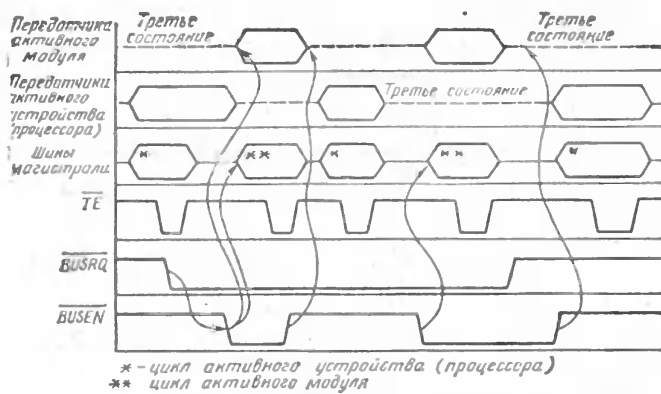


Рис. 3. Последовательность появления сигналов на захват шин магистрали

ции работы нескольких активных модулей одновременно позволяет монтировать в ней несколько процессоров, специальное устройство пересылки данных и т. д.

Модуль телевизионного адаптера

Модуль предназначен для вывода на черно-белый или цветной телевизионные мониторы алфавитно-цифровой и графической информации.

Модуль вырабатывает полный телевизионный сигнал, который может быть непосредственно подан на вход «Видео» стандартного черно-белого телевизора или монитора. Бытовые черно-белые телевизоры, не имеющие входа «Видео», требуют небольшой доработки. Для подключения цветного монитора модуль вырабатывает, кроме полного видеосигнала, сигналы R, G, B, управляющие интенсивностью каждого из первичных цветов кинескопа. Обычные цветные телевизоры требуют доработки до RGB-монитора. Наиболее просто такая переделка может быть произведена в телевизорах марок «Электроника Ц-432» (4УПИЦТ-25-IV-2), «Юность Ц-404», «Шилялис

Ц-410» (УПИЦТ-32-10), «Рубин Ц-201», 202, 208 (УПИМЦТ-61-С-2).

Модуль допускает подключение специальных мониторов, не имеющих селектора синхронимпульсов. Для них вырабатываются отдельные сигналы кадровой и строчной синхронизации. Амплитуда на нагрузке 75 Ом по выходам «Видео» и R, G, B должна быть не менее 1 В. В табл. 4, 5 приведены основные параметры модуля при работе с черно-белыми и цветными мониторами.

Принцип работы и описание схемы модуля

Модуль телевизионного адаптера обеспечивает формирование на черно-белом или цветном мониторах различных графических изображений в трех основных режимах: монохромном среднего разрешения (режим 1), цветном среднего разрешения (режим 2), монохромном высокого разрешения (режим 3).

В первых двух режимах изображение формируется на матрице, состоящей из 320 точек по горизонтали и 200 точек по вертикали. В монохромном режиме высокого разрешения размер рабочего поля составляет 640×200 точек соответственно.

В модуле используется принцип отображения части содержимого встроеной памяти на экране телевизионного монитора. В процессе работы происходит непрерывное считывание и преобразование данных в последовательность битов. «Лог. 1», записанная в памяти, вызывает засветку соответствующей точки на экране. Схема обработки видеосигнала из указанной последовательности вырабатывает сигналы управления яркостью и интенсивностью цветов, которые после смешения с синхросигналом поступают на выходы модуля. Таким образом осуществляется циклическая регенерация изображения на экране, создается эффект статической картины. Объем памяти, необходимый для работы в режиме 1, составляет 8000 байт, а в режимах 2, 3 — 16 000. Общий объем памяти, устанавливаемый в модуль, может быть больше требуемой для регенерации изображения и зависит от типа примененных микросхем. При установке микросхем К565РУ6 с любым буквенным индексом общий объем ОЗУ модуля

Таблица 4

Основные параметры модуля телевизионного адаптера при работе с черно-белым телевизионным монитором

Режим	Число градаций яркости	Размер рабочего поля, точек	Число рабочих страниц	Требуемый объем памяти, К байт	Максимальное время доступа к памяти, мкс	Коэффициент прозрачности для внешнего процессора*, %, не менее
1	2	3	4	5	6	7
Монохромный среднего разрешения	2	320×200	2	8	1	77
Цветной (полутоновый) среднего разрешения	4	320×200	1	16	40	57
Монохромный высокого разрешения	2	640×200	1	16	40	57
Работа с погашенным изображением	—	—	—	—	1	93,7

Примечание. Параметры приведены для частоты системного СГС=16 МГц

* Коэффициент прозрачности = $\frac{\text{Число циклов, необходимых модулю для регенерации изображения и содержимого памяти}}{\text{Максимально возможное число циклов}} \times 100\%$

Основные параметры модуля телевизионного адаптера при работе с цветным телевизионным монитором

Режим	Число основных цветов	Число цветов фона	Число рабочих палитр	Число рабочих страниц	Размер изображения, точек
Монохромный среднего разрешения	Один из восьми	Один из четырех	—	2	320/200
Цветной среднего разрешения	4	Один из восьми	2	1	320/200
Монохромный высокого разрешения	1	Один из шестнадцати	—	1	640/200

равняется 16К байт. Из этого объема только 384 ячейки могут использоваться процессором для своих нужд. При установке микросхем K565PY5 этот излишек составит более 48К байт (49 536 байт), которые являются основной оперативной памятью процессора в минимальной конфигурации ПЭВМ.

Основу модуля телевизионного адаптера составляет ОЗУ, в котором хранится информация об изображении на экране (рис. 4). Считываемый из ОЗУ при регенерации изображения поток байтов преобразуется сдвиговым регистром в последовательность битов. С выхода этого регистра сигнал поступает на схему обработки видеосигнала (рис. 5).

В блоке обработки видеосигнала два канала. Канал 1 работает в монохромном и цветном режимах среднего разрешения. Канал 2 — в режиме графики высокого разрешения. Переключение каналов производится сигналами HDCM и MR/HR, записываемыми в регистр управления модулем D13 (Здесь и далее по тексту нумерация микросхем приводится по рис. 6 в принципиальной схеме, представленной на цветном развороте.) Топология платы ТВ-адаптера приведена на рис. 7. Сигнал HDCM переключает тактовую частоту сдвигового регистра D33: если этот сигнал установлен в состояние «Лог. 1», то тактовая частота равна 16 МГц; если в состоянии «Лог. 0», то 8 МГц.

Ниже приведены комбинации этих сигналов для различных режимов работы модуля.

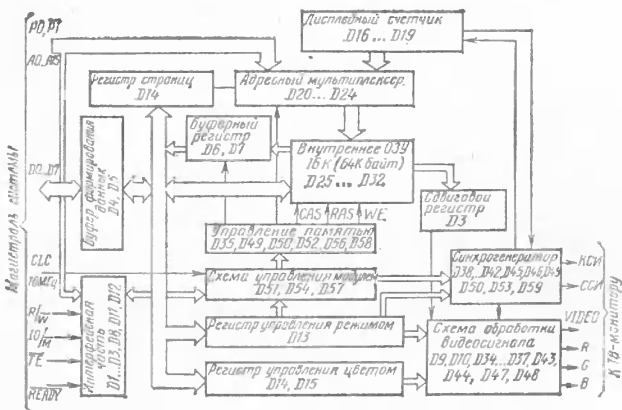
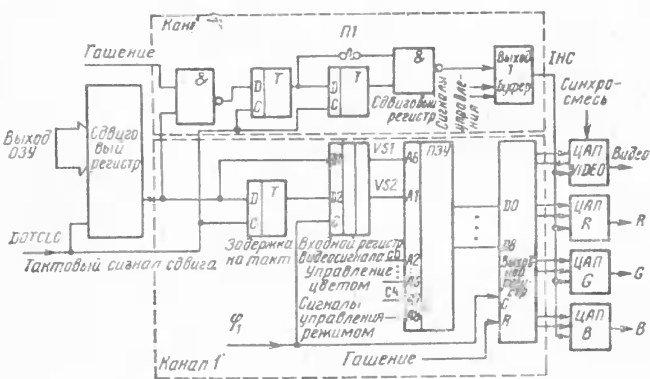


Рис. 4. Структурная схема модуля ТВ-адаптера

Режим	Сигнал	
	HDCM	MR/HR
Монохромный среднего разрешения, 320×200 точек	0	X
Цветной среднего разрешения, 320/200 точек, четыре цвета	1	0
Монохромный высокого разрешения, 640×200 точек	1	1

X — состояние бита безразлично.



При работе канала 1 в режимах 1, 2 информация из сдвигового регистра с помощью соответствующей схемы задерживается на такт. Задержанный и прямой сигналы запоминаются во входном регистре видеосигнала. В качестве стробирующего импульса этого регистра используется сигнал внутренней синхронизации ф1 (8 МГц). В режиме 1 частота сигнала сдвига DOTCLC равна ф1 и на выходе входного регистра видеосигнала формируются два, сдвинутых на такт, информационных сигнала, из которых используется только один. В режиме 2 частота сигнала сдвига DOTCLC вдвое больше частоты сигнала внутренней синхронизации ф1. В этом режиме используются оба выходных информационных сигнала, причем один из них содержит информацию о четных битах байта считанной из ОЗУ информации, а второй — о четных. Эти соседние биты байта используются для кодировки цвета засвечиваемой точки.

Оба информационных видеосигнала, а также сигналы управления цветом и режимом работы поступают на вход ПЗУ D43. С его помощью они преобразуются в сигналы управления яркостью и сигналы управления

Рис. 5. Схема устройства обработки видеосигнала

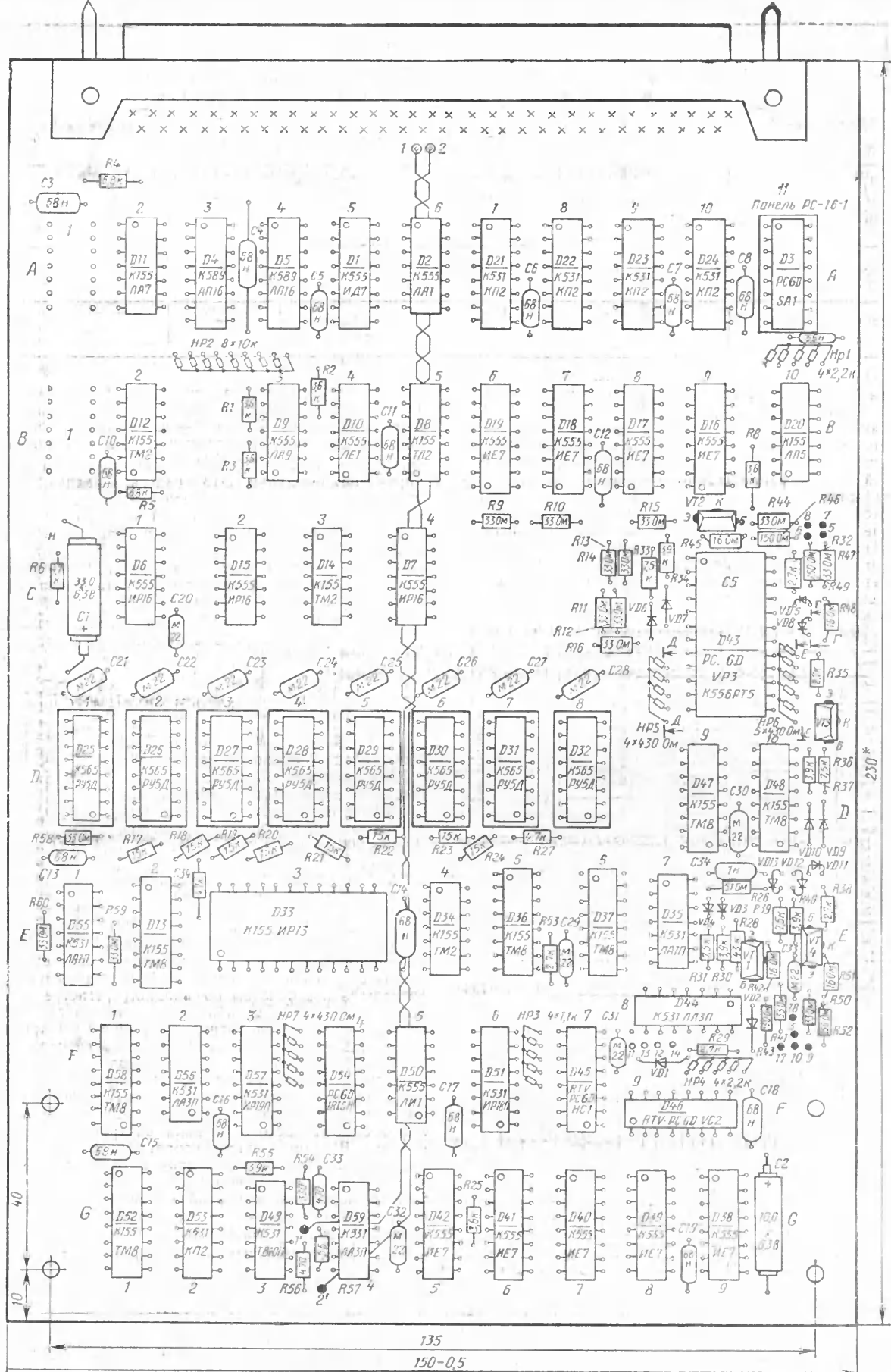


Рис. 6. Монтажная схема ТВ-адаптера

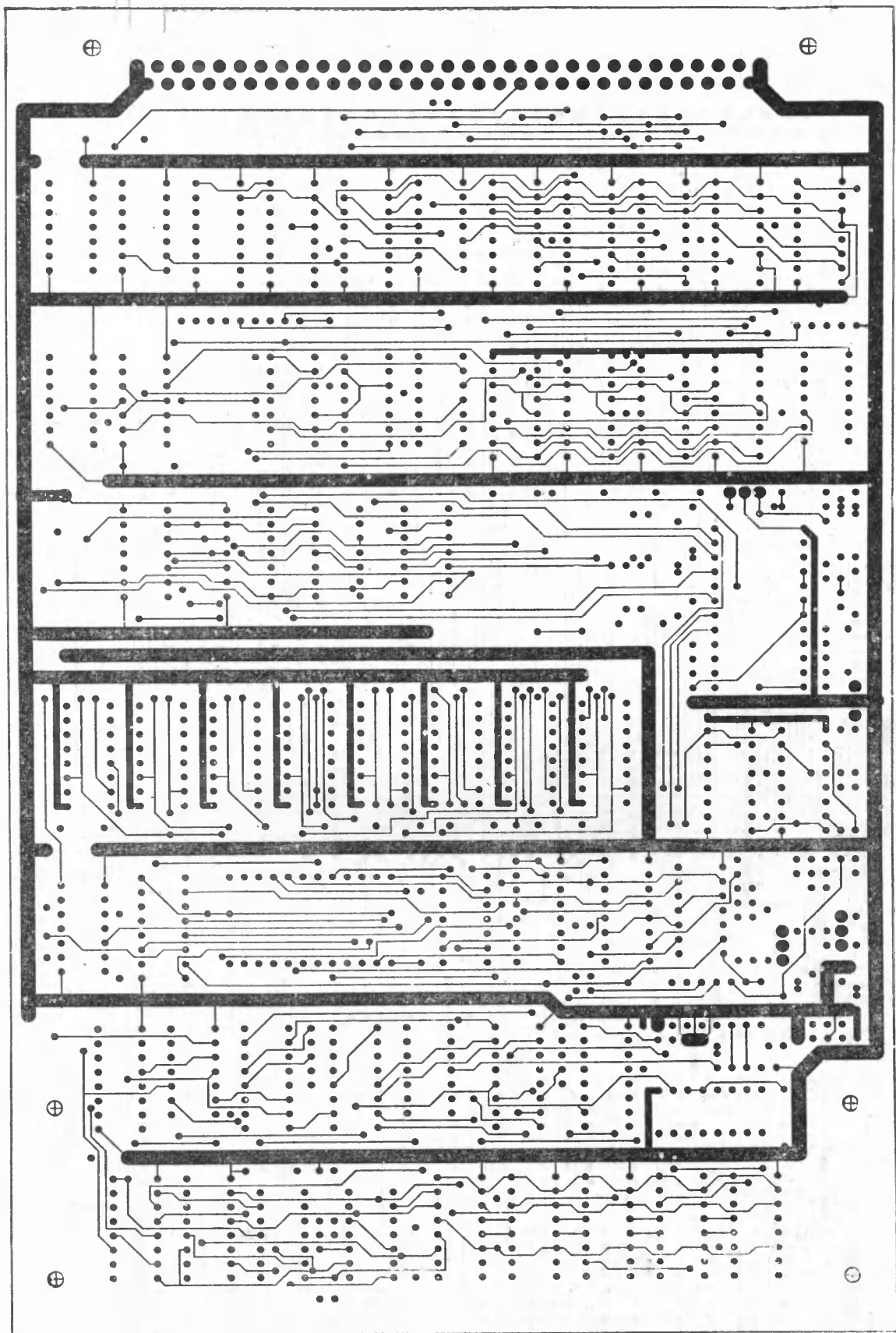


Рис. 7,а. Топология — лицевая сторона

оп. 4

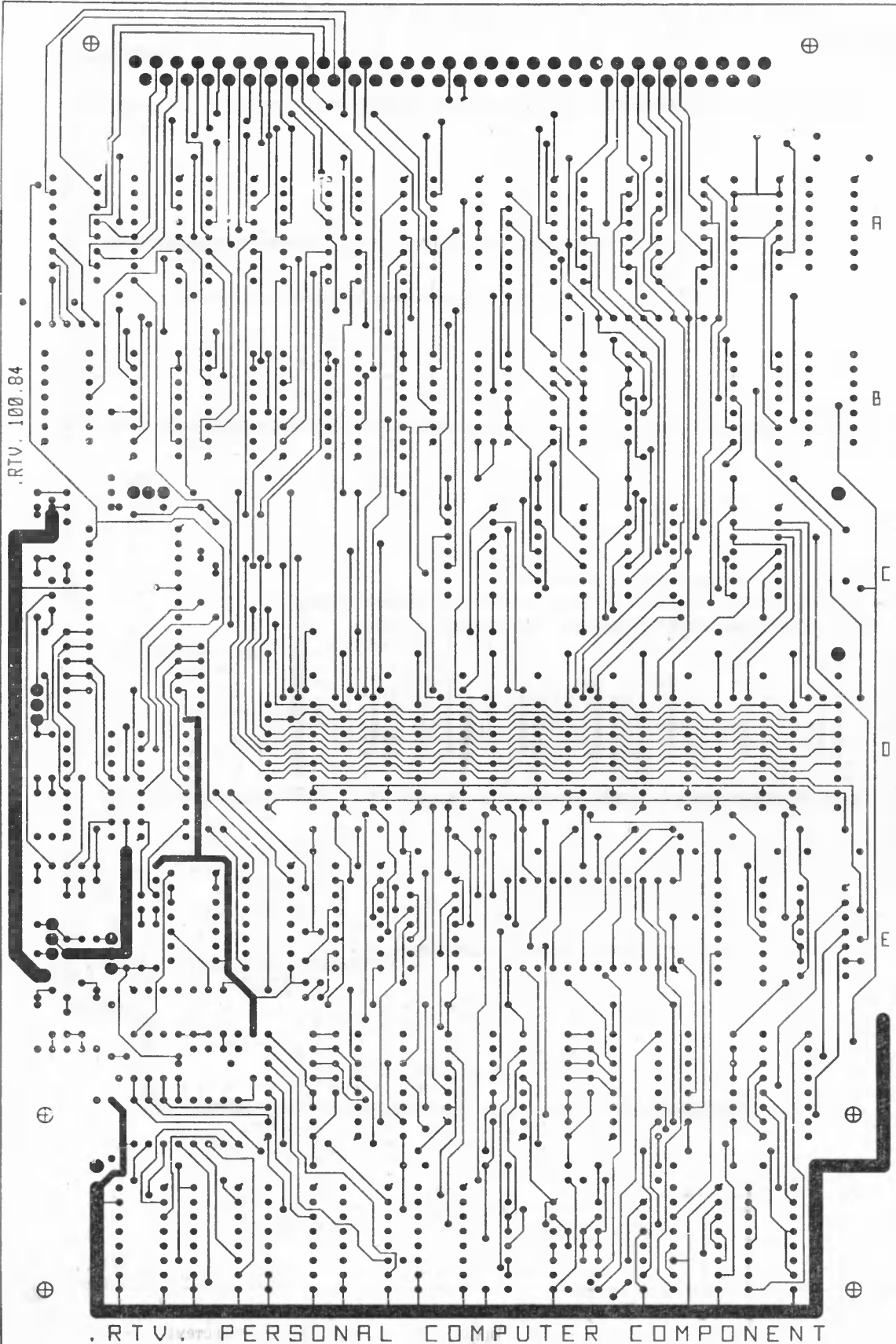


Рис. 7.б. Топология — обратная сторона

первичными цветами R, G, B. Выходной регистр служит для исключения переходных характеристик в ПЗУ схемы обработки видеосигнала. «Прошивка» этого ПЗУ (КР556РТ5) приведена на рис. 8.

Канал 2 схемы обработки видеосигнала используется только в режиме графики высокого разрешения. Он состоит из сдвигового регистра с логикой управления включением и гашением изображения. Этот канал может работать в двух вариантах, определяемых положением перемычки П1. Если эта перемычка разомкнута, то включается нормальный режим, когда в строке содержится 640 точек. Однако для поддержания этого режима необходимы мониторы с расширенной полосой пропускания. Если в качестве монитора используется обычный телевизор, то этот режим может быть применен для получения более качественного графического изображения со сглаживанием линий за счет сдви-

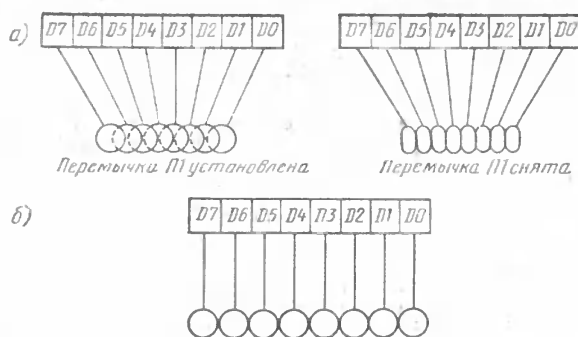


Рис. 9 Процесс засвечивания точек в строке в режимах высокого (а) и среднего (б) разрешения

ADDRESS X=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
- DATA (HEX) -																
00X	00	FF	00	FF	00	C3	00	C3	00	33	00	33	00	F3	00	F3
01X	00	0F	00	0F	00	CF	00	CF	00	3F	00	3F	00	FF	00	FF
02X	80	FF	80	FF	80	C3	80	C3	80	33	80	33	80	F3	80	F3
03X	80	0F	80	0F	80	CF	80	CF	80	3F	80	3F	80	FF	80	FF
04X	10	FF	10	FF	10	C3	10	C3	10	33	10	33	10	F3	10	F3
05X	10	0F	10	0F	10	CF	10	CF	10	3F	10	3F	10	FF	10	FF
06X	54	FF	54	FF	54	C3	54	C3	54	33	54	33	54	F3	54	F3
07X	54	0F	54	0F	54	CF	54	CF	54	3F	54	3F	54	FF	54	FF
08X	00	FF	00	FF	00	C3	00	C3	00	33	00	33	00	F3	00	F3
09X	00	0F	00	0F	00	CF	00	CF	00	3F	00	3F	00	FF	00	FF
0AX	80	FF	80	FF	80	C3	80	C3	80	33	80	33	80	F3	80	F3
0BX	80	0F	80	0F	80	CF	80	CF	80	3F	80	3F	80	FF	80	FF
0CX	10	FF	10	FF	10	C3	10	C3	10	33	10	33	10	F3	10	F3
0DX	10	0F	10	0F	10	CF	10	CF	10	3F	10	3F	10	FF	10	FF
0EX	54	FF	54	FF	54	C3	54	C3	54	33	54	33	54	F3	54	F3
0FX	54	0F	54	0F	54	CF	54	CF	54	3F	54	3F	54	FF	54	FF
10X	E2	E2	E2	E2	E2	CD	E2	CD	E2	CD	E2	CD	E2	CD	E2	CD
11X	04	E2	CD	FF	44	E2	CD	FF	14	E2	CD	FF	5A	E2	CD	FF
12X	AA	E2	CD	FF	6A	E2	CD	FF	9A	E2	CD	FF	5A	E2	CD	FF
13X	A6	E2	CD	FF	66	E2	CD	FF	69	E2	CD	FF	5A	E2	CD	FF
14X	30	32	0D	2F	40	32	0D	2F	10	32	0D	2F	50	32	0D	2F
15X	04	32	0D	2F	44	32	0D	2F	14	32	0D	2F	54	32	0D	2F
16X	AA	32	0D	2F	6A	32	0D	2F	9A	32	0D	2F	5A	32	0D	2F
17X	A6	32	0D	2F	66	32	0D	2F	6E	32	0D	2F	5A	32	0D	2F
18X	00	00	00	00	80	80	80	80	20	20	20	20	A0	A0	A0	A0
19X	08	08	08	08	68	68	68	68	28	28	28	28	A8	A8	A8	A8
2AX	00	00	00	00	6A	6A	6A	6A	9A	9A	9A	9A	5A	5A	5A	5A
2BX	A0	A6	A6	A6	66	66	66	66	96	96	96	96	56	56	56	56
2CX	00	00	00	00	80	80	80	80	20	20	20	20	A0	A0	A0	A0
2DX	08	08	08	08	68	68	68	68	28	28	28	28	A8	A8	A8	A8
2EX	00	00	00	00	6A	6A	6A	6A	9A	9A	9A	9A	5A	5A	5A	5A
2FX	A6	A6	A6	A6	66	66	66	66	96	96	96	96	56	56	56	56

Рис. 8. Карта прошивки PC.GD.VP3 ПЗУ D43 схемы обработки видеосигнала

га на половину ширины линии (перемычка П1 должна быть установлена). Рис. 9 поясняет этот процесс.

Выходные сигналы каналов 1 и 2 поступают на входы ЦАП, управляющих яркостью и интенсивностью цвета. В режимах 1 и 2 канал 2 не используется. При работе в режиме 3 канал 1 вырабатывает только сигналы для окраски фона. Цвет засвечиваемых точек в этом случае всегда белый. Кроме того, во всех режимах работы в ЦАП яркости канала «замешивается» синхросмесь от синхрогенератора.

Синхрогенератор состоит из счетчика, выходные сигналы которого обрабатываются двумя микросхемами ПЗУ (D45, D46, КР556РТ4). Прошивки этих ПЗУ определяют частоту кадровой и строчной разверток, длительность синхронимпульсов и размер рабочей части раstra. На рис. 10, а, б приведены прошивки ПЗУ синхрогенератора для разрешения 320x200 точек (общее число строк в кадре равно 244). Кроме того, синхрогенератор формирует запросы на регенерацию памяти в периоды, когда выдача информации на экран не производится, а также сигналы управления дисплейным счетчиком. Счетчик, собранный на микросхеме К555ИЕ7 (D16...D19), вырабатывает адреса для считывания данных из памяти при регенерации изображения. В конце каждого кадра содержимое счетчика обновляется.

В интерфейсную часть входит селектор внутренних адресов, регистров управления, памяти и двунаправленный буфер — формирователь шины данных. Селектор выполнен на основе ПЗУ D3. Прошивки

ADDRESS (HEX)	X=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
- DATA (HEX) -																	
0X	6	0	2	0	0	0	2	0	0	0	2	0	0	8	4	4	4
1X	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
2X	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3X	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4X	0	0	2	0	0	0	2	0	0	0	2	0	0	2	0	1	1
5X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8X	0	2	0	0	0	2	0	0	0	2	0	0	0	8	2	0	0
9X	0	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	2
AX	0	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	2
BX	0	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	2
CX	0	2	0	0	0	2	0	0	0	2	0	0	0	2	1	1	1
DX	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
EX	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
FX	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ADDRESS (HEX)	X=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
- DATA (HEX) -																	
0X	8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1X	8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2X	8	2	2	2	7	7	7	7	7	7	7	7	7	7	7	7	7
3X	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4X	8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
5X	8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
6X	8	2	2	2	7	7	7	7	7	7	7	7	7	7	7	7	7
7X	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8X	8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
9X	8	2	2	2	6	6	6	6	6	6	6	6	6	6	6	6	6
AX	8	2	2	2	6	6	6	6	6	6	6	6	6	6	6	6	6
BX	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CX	8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
DX	8	2	2	2	6	6	6	6	6	6	6	6	6	6	6	6	6
EX	8	2	2	2	6	6	6	6	6	6	6	6	6	6	6	6	6
FX	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рис. 10. Карты прошивок RTV.PC.GD.VC2 ПЗУ D46 (а) и RTV.PC.GD.HC1 D45 (б) синхрогенератора

ПЗУ приведены на рис. 11, а, б. Младший разряд адресной шины также обрабатывается селектором адресов, что обеспечивает совместимость с картой распределения памяти графического контроллера IBM PC. Схема управления работой модуля осуществляет анализ запросов на обращение к внутренней памяти со стороны внешних по отношению к модулю устройств, запросов от синхрогенератора на выдачу данных для их индикации и запросов на регенерацию содержимого памяти.

Эта схема выбирает запрос, который будет удовлетворяться в первую очередь, и разрешает схеме управления ОЗУ выработать в соответствии с запросом определенную последовательность синхронизирующих сигналов. При одновременном поступлении запросов наивысший приоритет имеет запрос на регенерацию изображения на экране. Этот узел реализован на ос-

а) ADDRESS (HEX)	X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
- ДАТА (HEX) -																	
0X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2X	1	9	5	D	3	B	7	F	1	9	5	D	3	B	7	F	
3X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6X	1	9	5	D	3	B	7	F	1	9	5	D	3	B	7	F	
7X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

б) ADDRESS (HEX)	X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
- ДАТА (HEX) -																	
0X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2X	1	9	5	D	3	B	7	F	1	9	5	D	3	B	7	F	
3X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6X	1	9	5	D	3	B	7	F	1	9	5	D	3	B	7	F	
7X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рис. 11. Карта прошивки PC.GD.SA1 ПЗУ D3 селектора адреса для внутренней памяти объемом 16К байта (а) и 64К байта (б)

адрес (HEX)	X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
- ДАТА (HEX) -																	
0X	0	0	4	0	0	0	4	0	B	B	5	8	B	B	5	8	
1X	0	0	4	0	0	0	4	0	B	B	5	8	B	B	5	8	
2X	0	0	4	0	0	0	4	0	B	B	5	8	B	B	5	8	
3X	0	0	4	0	0	0	4	0	B	B	5	8	B	B	5	8	
4X	0	0	4	0	2	2	4	0	B	B	5	0	2	2	4	8	
5X	2	2	4	0	2	2	4	0	2	2	4	0	2	2	4	0	
6X	0	0	4	0	2	2	4	0	B	B	5	0	2	2	4	8	
7X	0	0	4	0	2	2	4	0	B	B	5	0	2	2	4	8	
8X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
AX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
BX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
CX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
DX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
EX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
FX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Рис. 12. Карта прошивки PC.GD.IRISH ПЗУ D54 схемы управления работой модуля

нове двух регистров D51 (K531IP18П), D57 (K531IP19П) и ПЗУ D54 (KP556PT4). Прошивка ПЗУ приведена на рис. 12.

Распределение циклов обращения к внутренней ОЗУ определяется, кроме того, установленным режимом работы Синхрогенератор совместно со схемой управления модулем обеспечивает максимально прозрачный доступ к внутренней памяти. Таким образом, если не требуется регенерации изображения, например при обратном ходе строчной и кадровой разверток, то при обращении к памяти процессор или другое активное устройство получает ближайший цикл для обмена информацией. В период обратного хода кадровой развертки в нерабочей части кадра схема управления производит регенерацию содержимого памяти. Если изображение на экране погашено (один из битов VE

или GS в управляющем байте установлен в состояние «Лог.0»), то производится только регенерация памяти и ее прозрачность соответственно возрастает. На рис. 13, 14 показаны распределения типов памяти при работе модуля в различных режимах.

Работа с модулем телевизионного адаптера

С точки зрения пользователя, составляющего программы для работы с модулем, он представляется как устройство, занимающее три адреса в области устройств ввода вывода и некоторую зону в адресном пространстве памяти. Эта зона в зависимости от установленного объема внутренней памяти имеет протяженность от 16 до 64К байт. Записывая или стирая информацию в дисплейной области памяти, можно син-

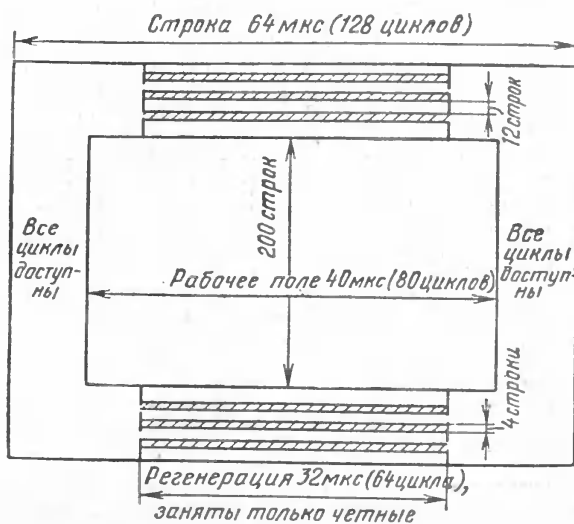


Рис. 13. Распределение времени доступа к памяти: в режимах 2, 3 заняты все 80 циклов, в режиме 1 — только четные

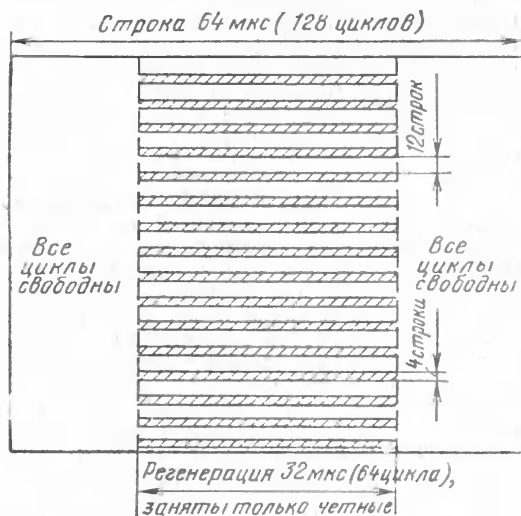


Рис. 14. Время доступа к памяти при гашении изображения

тезировать изображение на экране монитора. При указанных на рис. 12 прошивках ПЗУ селектора адресов область дисплейной памяти располагается, начиная с адреса 20 000. На рис. 15 показана карта распределения внутренней памяти при работе модуля в различных режимах.

Переключение режимов, выбор рабочих цветов точек и фона осуществляется записью соответствующих команд в регистры управления, имеющих следующие адреса обращения: D8 (Hex) — регистр управления режимом, D9 (Hex) — регистр управления цветом, DA (Hex) — регистр выбора рабочих страниц (рис. 16). По фундаментальному назначению битов первые два регистра аналогичны графическому контроллеру IBMPC.

В режиме 1 на экране отображается одна из двух страниц объемом 8000 байт, причем в верхнем левом углу находится первый байт записанного изображения. (Старший бит соответствует первой засвечиваемой точке.) Через регистр управления цветом можно задавать окраску засвечиваемых точек и фона. Структура регистра управления цветом при работе в режиме 1 показана на рис. 17. Засвечиваемые точки могут быть окрашены в один из семи цветов на цветном фоне.

Регистр выбора страниц используется только в режиме 1 и позволяет выбирать области отображаемой на экране памяти (две страницы). Это дает возможность подготавливать информацию в невидимой странице и затем записью соответствующего байта вызвать ее на экран.

При работе в режиме 2 каждая точка на экране может быть окрашена в один из трех цветов заранее выбранной палитры (две палитры). Требуемый объем дисплейной памяти — 16 000 байт. Включение этого режима производится передачей кода 8A (Hex) в регистр управления режимом. Окраска засвечиваемой

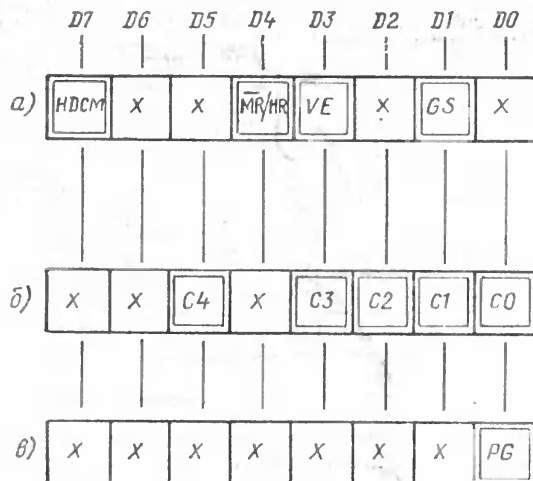
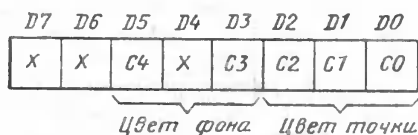


Рис. 16. Структура регистров управления режимом (а), цветом (б), выбором страниц (в)



C4	C3	Цвет фона	C2	C1	C0	Цвет точки
0	0	Черный	0	0	0	Белый
0	1	Синий	0	0	1	Синий
1	0	Зеленый	0	1	0	Зеленый
1	1	Малиновый	0	1	1	Голубой
			1	0	0	Красный
			1	0	1	Малиновый
			1	1	0	Желтый
			1	1	1	Белый

Рис. 17. Работа регистра управления цветом в режиме 1

точки задается комбинацией битов считываемого из внутренней памяти байта (рис. 18). Из рисунка следует, что каждый байт управляет засветкой только четырех точек и для строки размером в 320 точек необходимо 80 байт.

Выбор рабочей палитры осуществляется изменением пятого бита (C4) регистра управления цветом, а младшие четыре бита C0...C3 этого регистра используются для задания цвета фона (рис. 19). Число окрасок фона, включая черную, равно 16.

Если вместо цветного монитора к ТВ-адаптеру подключить черно-белый, то можно работать с полутоновым изображением. Яркость точек в этом случае также зависит от комбинации соседних битов считываемого байта (см. рис. 18). Число градаций яркости равно четырем.

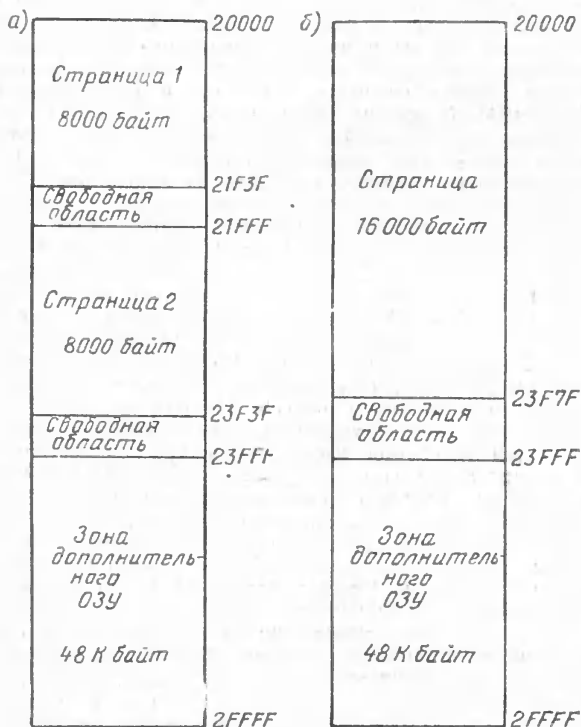


Рис. 15. Распределение внутренней памяти в режиме 1 (а) и в режимах 2, 3 (б)

D7	D6	D5	D4	D3	D2	D1	D0
СС1	СС0	СС1	СС0	СС1	СС0	СС1	СС0
Первая точка		Вторая точка		Третья точка		Четвертая точка	

а)

СС1	СС0	Палитра 0 (C4=0)	Палитра 1 (C4=1)	Черно-белый монитор
0	0	Цвет фона	Цвет фона	Выключение луча
0	1	Зеленый	Голубой	Малая яркость
1	0	Красный	Малиновый	Средняя яркость
1	1	Коричневый	Белый	Большая яркость

б)

Рис. 18. Последовательность засветки точек (а) и схема кодировки цвета (яркости) (б) в режиме 2

D7	D6	D5	D4	D3	D2	D1	D0
X	X	C4	X	C3	C2	C1	C0

Выбор палитры

C3	C2	C1	C0	Цвет фона
0	0	0	0	Черный
0	0	0	1	Синий
0	0	1	0	Зеленый
0	0	1	1	Голубой
0	1	0	0	Красный
0	1	0	1	Малиновый
0	1	1	0	Коричневый
0	1	1	1	Темно-серый
1	0	0	0	Светло-серый
1	0	0	1	Светло-синий
1	0	1	0	Светло-зеленый
1	0	1	1	Светло-голубой
1	1	0	0	Светло-красный
1	1	0	1	Светло-малиновый
1	1	1	0	Желтый
1	1	1	1	Белый

Рис. 19. Работа регистра управления цветом в режиме 2

Режим 3 включается записью в регистр управления режимом кода 9А (Hex). Каждый бит в байте отвечает за свечение соответствующей точки. Рабочее поле при этом составляет 640×200 точек, в каждой строке размещается 80 байт (символов). Требуемый объем памяти, как и в режиме 2, составляет 16 000 байт. Он определяется кодовой комбинацией, засылаемой в регистр управления цветом. На рис. 20 приведено назначение битов регистра управления цветом в режиме 3 и кодировка цветов фона для этого режима.

При записи в регистр управления режимом кода, содержащего нули в одном из битов VE или GS (например, запись кода 00), изображение на экране гасится, но возможность обмена с внутренней памятью остается (см. рис. 16). Более того, поскольку в этом случае запросы на считывание информации для изображения на экране блокированы, то процессор получает все возможные циклы внутренней памяти. При погашенном изображении производится только регенерация содержимого памяти и коэффициент прозрачно-

сти ее возрастает до 0,93. Период регенерации установлен равным 1 мс. Это сделано для того, чтобы можно было устанавливать микросхемы памяти ОЗУ D25...D32 с любым быстродействием. Режим работы с погашенным изображением может быть полезным для ускорения работы расчетных задач.

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	C3	C2	C1	C0

C3	C2	C1	C0	Цвет фона
0	0	0	0	Черный
0	0	0	1	Темно-синий
0	0	1	0	Темно-зеленый
0	0	1	1	Темно-голубой
0	1	0	0	Темно-красный
0	1	0	1	Темно-малиновый
0	1	1	0	Темно-коричневый
0	1	1	1	Серый
1	0	0	0	Черный
1	0	0	1	Светло-синий
1	0	1	0	Светло-зеленый
1	0	1	1	Светло-голубой
1	1	0	0	Светло-красный
1	1	0	1	Светло-малиновый
1	1	1	0	Желтый
1	1	1	1	Светло-серый

На черно-белом мониторе серый фон

Рис. 20. Работа регистра управления цветом в режиме 3

Модуль телевизионного адаптера не имеет специальных средств для отображения символьной информации. Текст на экране формируется с помощью процессора путем записи в соответствующие места дисплейной памяти байтов с изображением символов. Знакогенератор, с которым работает процессор при выводе символа, может располагаться либо в ОЗУ, либо в ПЗУ ПЭВМ. В первом случае пользователь имеет возможность легко заменять знакогенератор в соответствии со своими требованиями. Это может быть полезным при смене русского алфавита на любой другой.

Число символов в строке определяется размером матрицы, на которой синтезируется изображение подпрограммой рисования символов. ПЭВМ «Ириша» использует матрицы 7×8 точек, и, таким образом, максимальное число символов в строке равно 40 для режимов 1 и 2 и 80 символов — для режима 3. Число строк текста на экране также определяется подпрограммой рисования символов. На экран можно вывести до 20 строк, т. е. каждое знакоместо занимает по вертикали 10 строк растра. Из них восемь строк используются знакогенератором, в две последние (нижние) — для генерации изображения курсора, внешний вид которого задается программно. Самая трудоемкая операция при работе с текстом — «ролик». Она требует перезаписи всего содержимого дисплейной области внутренней памяти. Специальные подпрограммы работы с дисплейной памятью позволяют выполнять эту операцию в режиме 1 менее чем за 0,1 с, а в режимах 2 и 3 — менее чем за 0,25 с.

Все операции по выводу на экран символьной информации выполняются специальной программой, записанной в ПЗУ, которое устанавливается в модуле процессора ПЭВМ. Эта программа эмулирует работу обычного алфавитно-цифрового терминала, поддерживая, кроме того, ряд дополнительных функций, например работу с «окнами», операцию уширения символов и т. д.

Статья поступила 21 ноября 1985 г.

УСТРОЙСТВО ПРИОРИТЕТНОГО ПРЕРЫВАНИЯ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

При управлении сложными, быстродействующими объектами в режиме циклического опроса, когда программно проверяется состояние элементов автоматики, быстродействие процессора в ряде случаев недостаточно.

Режим прерывания программ, при котором периферийные устройства, прервав выполнение основной программы, дают запросы на обслуживание их соответствующей подпрограммой, рационально использует машинное время и существенно расширяет функциональные возможности микроЭВМ.

У разработанного согласно рекомендациям предприятия-изготовителя [1] блока приоритетного прерывания (БПП) [2] схема выделения запроса с наивысшим приоритетом обслуживания громоздкая. Это конструктивно ограничивает разработки систем управления при увеличении количества уровней приоритета. Применение БИС типа КР580ВН59, К589ИК14 и др. требует дополнительных аппаратных и программных затрат из-за различия процедур обработки прерываний в «Электронике 60» и упомянутых БИС.

Устройство приоритетного прерывания (УПП), созданное авторами, существенно повышает количество обрабатываемых уровней прерывания за счет более совершенной схемы выделения запроса с наивысшим приоритетом.

Сигналы электроавтоматики (ЭА) объекта управления (рис. 1) через ключи (К), управляемые регистром масок (РМ), — его состояние изменяется программно, — поступают на регистр памяти заявок (РП).

Аварийные сигналы электроавтоматики (ЭАА) поступают в РП на управляющие входы С соответствующих триггеров Д-типа, устанавливая на их выходах «Лог. 1».

Заявки на прерывание из РП через регистр фиксации состояния (РФ), выполненный на триггерах Д-типа, поступают на входные шины узла выделения заявки с наивысшим приоритетом (УВЗНП). РФ обеспечивает постоянство входной информации УВЗНП при выполнении операций по прерыванию программы за счет фиксации состояния триггеров РФ на время воздействия сигнала «Ввод Н» на управляющие входы С. В отсутствие сигнала «Ввод Н» информация в РФ изменяется в соответствии с состоянием триггеров РП. При наличии заявки на прерывание в РП и РФ

УВЗНП формирует сигнал требования прерывания «ТПРН», выделяя из числа одновременно поступивших заявок с наивысшим приоритетом, а также очищает триггер соответствующей заявки в РП.

Узел адреса вектора прерывания (УА), приняв из УВЗНП сигнал по шине, соответствующей заявке с наивысшим приоритетом, формирует адрес вектора прерывания. Устройство управления (УУ), получив сигнал «ТПРН», через узел обмена (УО) передает требование на прерывание в центральный процессор (ЦП). УО состоит из шинных формирователей, обеспечивающих обмен информацией и управляющими сигналами с каналом микроЭВМ, дешифратора адреса и регистра адреса. В дальнейшем ЦП и УПП обмениваются информацией в соответствии с процедурой прерывания программы, принятой в «Электронике 60М».

Адреса векторов прерывания программы выбираются с учетом наличия свободных зон в поле адресов и зависят от архитектуры системы управления.

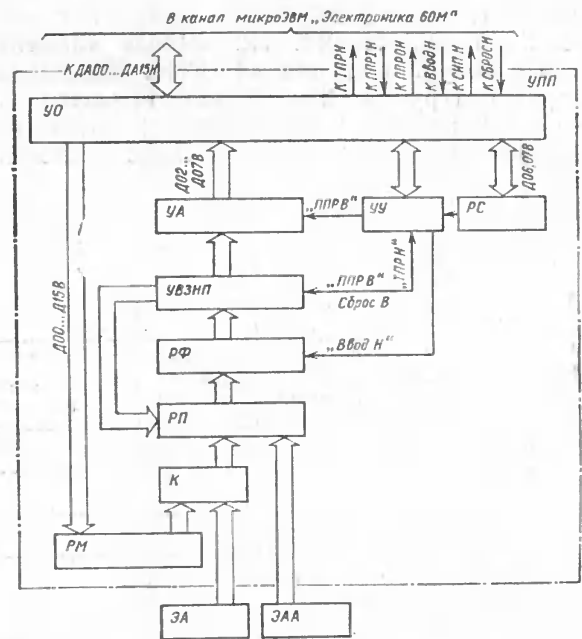


Рис. 1. Структурная схема устройства приоритетного прерывания (УПП)

Приоритет заявок на прерывание, обрабатываемых в УПП, зависит от места конструктивной установки модуля связи с объектом в канале микроЭВМ в соответствии со схемой распространения сигнала «ППР» по каналу [1]. Регистр состояния (РС) позволяет программно блокировать работу УПП, управляя состоянием шестого разряда РС в соответствии с принятой в микроЭВМ процедурой обработки прерываний. Состояние шестого РС и седьмого (информирующего о наличии требования на прерывание в УПП) разрядов программно доступно ЦП.

В электрической схеме УВП на n уровней (рис. 2), примененной в описываемом УПП, следует выделить три группы элементов. Элементы «И» (1.1...1. n) и «Исключающее ИЛИ» (2.1...2. n) выявляют запрос с наивысшим приоритетом.

В исходном состоянии потенциал выходов РФ — «Лог. 1». Такой же потенциал поступает на вход «1» и соответствующие входы элементов 1.1, 2.1, формирующие сигнал запроса с наивысшим приоритетом на прерывание. Потенциалы выходов РФ не препятствуют прохождению потенциала «Лог. 1» по всей цепочке элементов 1.1...1. n . Вследствие равенства потенциалов на входах элементов 2.1...2. n , на их выходах — потенциалы «Лог. 0».

Приход запросов на прерывание программы устанавливается на соответствующих им выходах РФ «Лог. 0».

Это блокирует распространение потенциала со входа «1» (начиная с выхода элемента группы 1, соответствующего запросу с наивысшим приоритетом). На выходе элемента 1. n формируется сигнал «ТПРН». На выходе элемента (группы 2 — соответствующего запросу с наивысшим приоритетом) вследствие разницы потенциалов на его входах устанавливается потенциал «Лог. 1».

Остальные элементы группы 2 не изменяют состояния выходов. Выходные сигналы элементов группы 2 инициируют формирование адреса вектора прерывания программы в УА и готовят к работе соответствующий элемент группы 3. После передачи вектора прерывания в ЦП после прихода сигнала «ППРВ» элементы 3.1...3. n обеспечивают сброс триггера обрабатываемой заявки на прерывание в РФ.

На время выполнения операций по прерыванию программы РФ фиксирует первоначальную информацию от РФ. Поэтому поступление новой заявки на прерывание либо сброс триггера РФ во время передачи вектора не влияют на формирование адреса вектора прерывания и обмен информацией с ЦП. УПП устанавливается в исходное состояние по сигналу «К СБРОС И» от ЦП.

Схема УПП довольно просто наращивается со стороны уровней и с высшим, и с низким приоритетом. Сравним схемные решения УВЗНП в УПП и БПП [2] по количеству используемых ИМС. В УПП применяются ИС типа К155ЛП5 и К155ЛИ1, а в БПП — ИС типа К155ЛП9. Эффективность применения схемы УВП (рис. 3) также нарастает с увеличением количества уровней приоритета. Незначительная длина внутрисхемных связей УВП и их регулярный характер упрощают разработку платы печатного монтажа.

Применительно к конструктиву «Электроника 60» с размером плат 240×280 мм и максимальной вместимостью 120 корпусов возможна реализация 44 уровней прерывания без маскирования либо 34 уровней прерываний, половина которых может быть маскируемой.

В разработанном авторами УПП из восьми уровней прерывания (в соответствии с техническими требованиями для объекта управ-

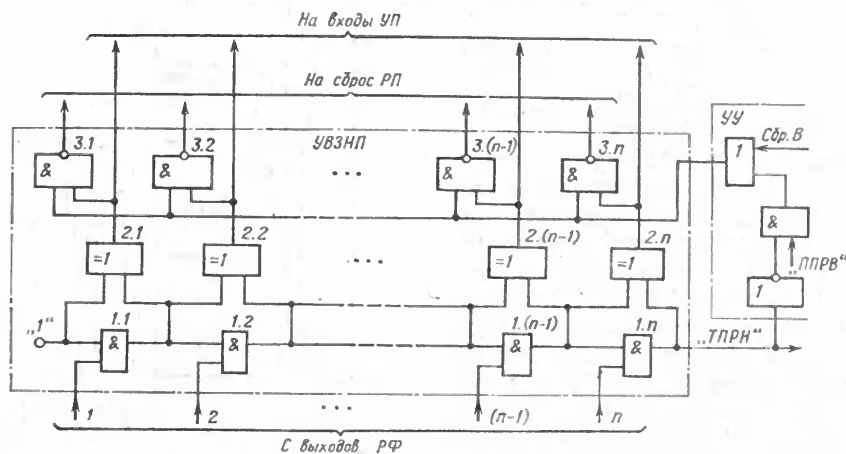


Рис. 2. Электрическая схема узла выделения заявки с наивысшим приоритетом (УВЗНП)

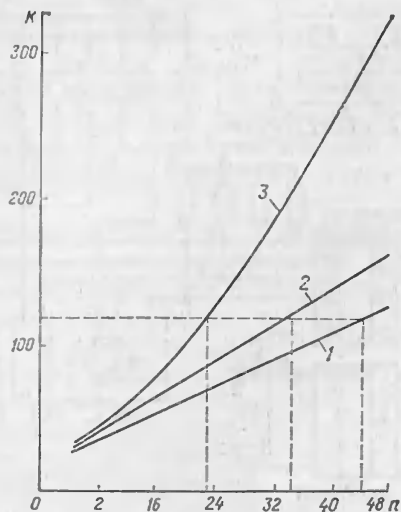


Рис. 3 График зависимости количества элементов (К) в УПП (1, 2) и БПП (3) от числа (n) обрабатываемых им уровней прерывания

ления) четыре маскируемые. Количество элементов УПП (без учета УО, общего для всех

устройств и занимающего 14 элементов) — 26. УПП размещено на части платы модуля связи с объектом. Этот модуль входит в систему ЧПУ обрабатывающим координатно-револьверным центром ОЦ КО126Ф4, изготавливающим детали типа панелей. Система управления разработана на базе устройства ЧПУ 2С85—62 с микроЭВМ «Электроника 60».

Благодаря УПП основные технические характеристики обрабатывающего центра (скорость перемещения — до 30 м/мин, производительность — 150 отв/мин) соответствуют мировому уровню.

Телефон для справок о приобретении технической документации или аппаратуры: 72-87-83 (Харьков).

ЛИТЕРАТУРА

1. Центральный процессор М2. — Техническое описание и инструкция по эксплуатации 3.852.ТО.1979 г.
2. Гойхман П. А., Удовченко А. Б., Вульфсон В. Л. Блок приоритетного прерывания. — Информ. листок № 83—060. — Харьков: ХЦНТИ, 1983.

Статья поступила 19 сентября 1985 г.

ПЗУ — РАЗНООБРАЗИЕ ПРИМЕНЕНИЙ

УДК 681.32 : 621.31 + 621.398 + 621.37

Д. А. Лукьянов

ПЗУ — УНИВЕРСАЛЬНЫЙ ЭЛЕМЕНТ ЦИФРОВОЙ ТЕХНИКИ

В предыдущем выпуске «Учебного центра» [1] были рассмотрены принципы работы программируемых ПЗУ (ППЗУ) и аппаратура, необходимая для занесения информации в эти микросхемы. В микропроцессорной технике ППЗУ наиболее часто применяются в качестве памяти программного обеспечения, не разрушающегося при выключении питания МП-устройства. Так хранятся прикладные программы и программы-загрузчики встраиваемых в аппаратуру микросистем, а также библиотеки программ стандартных математических операций в универсальных мини- и микроЭВМ. При построении программного обеспечения, пригодного для занесения в ППЗУ, необходимо учитывать ряд особенностей. Жесткое программное обеспечение должно иметь отдельные области кода программы и данных только для чтения (RO), компонуе-

мые в область ПЗУ микросистемы, и область рабочей памяти (RW), для которой используется зона ОЗУ. Кроме того, все подпрограммы такого ПО должны быть повторно-входными, что проще всего обеспечивается, если для передачи данных между подпрограммами используется стек. Желательно, чтобы код прикладных программ был позиционно-независимым — это облегчает настройку и эксплуатацию МП-средств.

Заметим, что не все компиляторы языков программирования позволяют получать объектные модули, пригодные для работы в ППЗУ, поэтому выбранный компилятор должен быть испытан до написания программ прикладного ПО. Эти вопросы достаточно хорошо освещены в литературе [2, 3].

Однако программируемые ПЗУ и ПЛМ могут найти значительно более широкое приме-

нение в микроэлектронной аппаратуре, заменяя узлы из многочисленных вентилях, соединенных нерегулярным образом. При этом достигается гибкость перестройки созданного прибора, что особенно существенно в мелкосерийном производстве. Ниже мы рассмотрим ряд таких применений.

Функции, которые мы выбираем

Будем считать, что ПЗУ — «черный ящик», имеющий вход — N линий адреса и выход — M линий данных (рис. 1). При задании адреса (A) на выходе ПЗУ появляется комбинация логических уровней напряжения (D), вообще говоря, различная для всех 2^N адресов и определенная на этапе занесения в ИС информации с помощью программатора. Следовательно, ПЗУ можно описать M -разрядной логической функцией $D=f(A)$, определенной на поле N -разрядных переменных, что дает прекрасный способ производить таблично заданные функциональные преобразования.

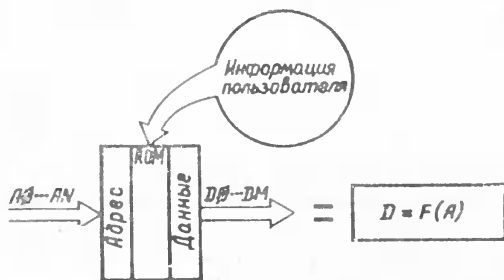


Рис. 1. Функциональные преобразования с ПЗУ

Простейший случай такого применения — преобразователь 4-разрядного двоичного кода в стилизованные изображения цифр шестнадцатеричной нотации на семисегментном индикаторе (рис. 2).

ТТЛ-ПЗУ удобно применять в качестве дешифратора адреса в микропроцессорных устройствах, особенно тогда, когда адресное

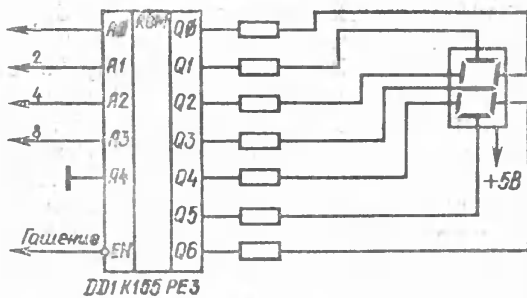


Рис. 2. Преобразователь кода с ПЗУ

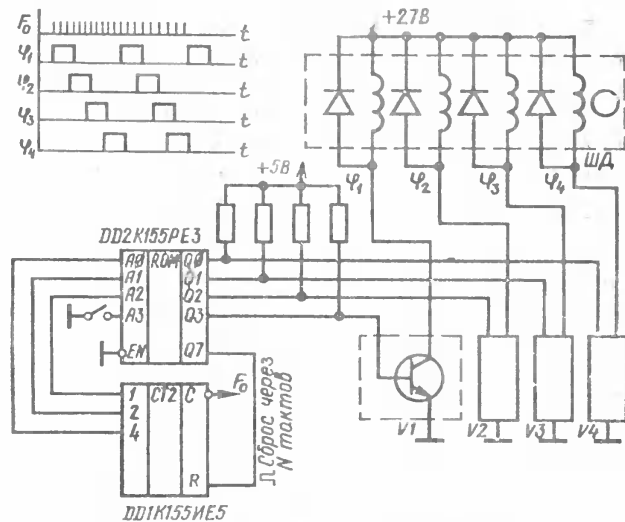


Рис. 3. Блок управления шаговым приводом

пространство МП необходимо разбить на различные по размеру зоны, занимаемые блоками ОЗУ, ПЗУ и устройствами ввода-вывода. При этом адресные входы ПЗУ подключаются к старшим разрядам шины адреса МП, а его объем определяет минимальную зону адресов, которую можно выделить одному устройству во время программирования ПЗУ дешифратора.

Для формирования сложных функций времени необходимо с заданной частотой перебирать адреса ячеек ПЗУ с помощью подходящего двоичного счетчика, «проигрывая» запись необходимой функции. Так, проще всего построить узел управления многофазным шаговым приводом (рис. 3), где ПЗУ К155РЕ3 совместно с микросхемой К155ИЕ5 выполняет функции распределителя импульсов тока.

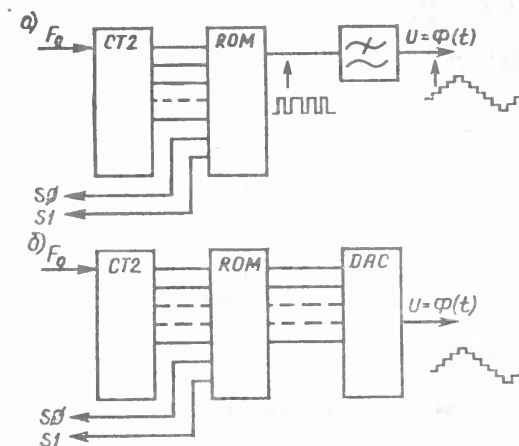


Рис. 4. Методы получения аналоговых функций по коду в ПЗУ: широтно-импульсный (а), кодово-импульсный (б).

Во многих случаях в радиотехнической аппаратуре требуются аналоговые сигналы, частота и форма которых поддерживается с высокой точностью. Для преобразования цифровой информации, снимаемой с выхода ППЗУ, в аналоговое напряжение или ток можно воспользоваться одним из двух методов (рис. 4).

Использование записи информации в широтно-импульсном представлении иллюстрируется на примере узла передатчика модема, применяемого для обмена числовой информацией по телефонным линиям (рис. 5). Напомним, что модемы преобразуют входной двоичный сигнал в двухтональный сигнал звуковой частоты. Для простоты предположим, что уровню «Лог. 1» соответствует сигнал частоты 1400 Гц, а «Лог. 0» — 2100 Гц. Для исключения помех в других телефонных каналах сигналы, формируемые модемом, должны быть максимально близки к синусоидальным. В нашем примере ППЗУ DD2 разбито на две «страницы», каждая из которых содержит таблично-заданный синус, причем полный цикл

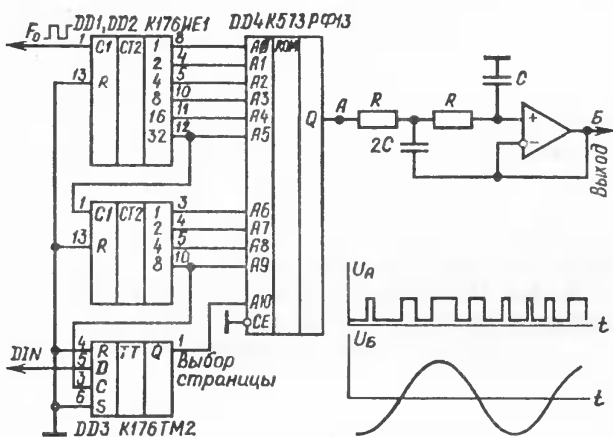


Рис. 5. Структура ШИМ-передатчика модема

перебора адреса для первой страницы соответствует двум периодам частоты 1400 Гц, а для второй — трем периодам частоты 2100 Гц, из-за чего объем страниц совпадает. Управляя номером страницы, легко изменить частоту тона на выходе модема. Триггер DD3 служит для синхронизации переключения страниц, соответствующей переходу выходного сигнала через ноль.

Аналогично (рис. 6) может быть построен блок управления ключами инвертора (преобразователя постоянного напряжения в переменное синусоидальное), которые широко применяются в агрегатах гарантированного питания [4]. Здесь ППЗУ может заменить доста-

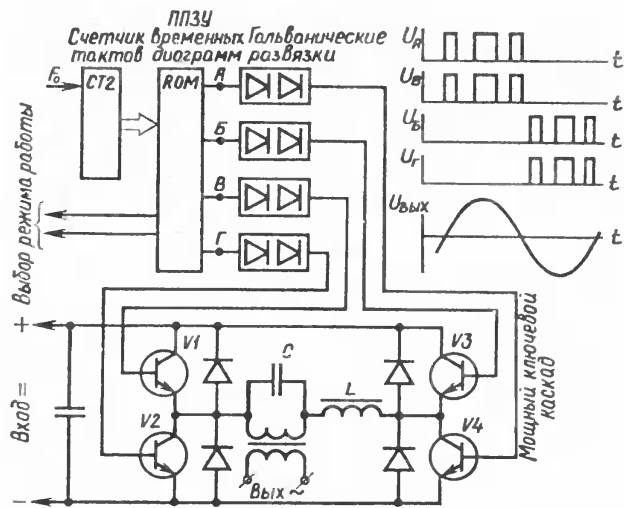


Рис. 6. Структура мощного высокочастотного инвертора и временные диаграммы коммутации напряжений

точно сложный формирователь коммутирующих импульсов, причем на нескольких страницах ППЗУ могут быть записаны алгоритмы коммутации напряжений пускового и рабочих режимов инвертора. При трехфазном выходном напряжении инвертора хранение в ПЗУ таблицы управления ключами получает особое преимущество, обеспечивая независимый от частоты и точно поддерживаемый сдвиг фаз, равный 120 градусам.

Другой способ преобразования цифрового сигнала в аналоговый реализуется в простом прецизионном синтезаторе низкочастотных сигналов с помощью БИС ЦАП (рис. 7). Четы-

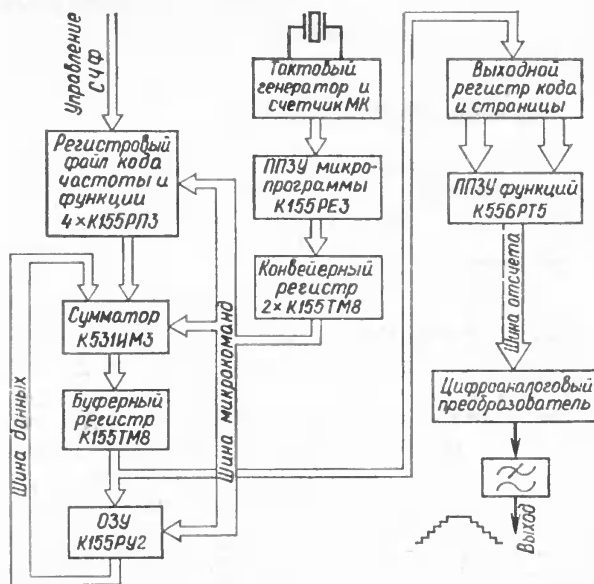


Рис. 7. Структура низкочастотного прецизионного синтезатора сигналов

рехразрядный сумматор и ОЗУ в этом приборе образуют последовательный арифметический процессор, микропрограммно эмулирующий 24-разрядный накопитель фазы, полный цикл работы которого занимает 1 мкс. Синтез необходимой частоты происходит распространением методом аккумуляции фазы [5] с дискретностью установки частоты 0,01 Гц. Величина элементарного приращения фазы за один цикл микропрограммы задается управляющим микропроцессором по входной шине в регистровый файл кода частоты и функций, играющий роль «почтового ящика» между быстродействующим специализированным процессором синтезатора и основным, работающим достаточно медленно. Восемь старших разрядов кода фазы и код номера страницы ППЗУ, содержащей таблично-заданную форму выходного напряжения, последовательно пере-сылаются в выходной буферный регистр. Считанная из заданной этим кодом области ППЗУ, информация преобразуется в аналоговое напряжение с помощью ЦАП и поступает на вход синтезатора.

Точность задания функции определяется разрядностью ППЗУ и ЦАП, а минимальная длительность шага — быстродействием ЦАП (для массовых ЦАП типа КР572ПА1 и К594ПА1 длительность шага составляет 3 и 1 мкс соответственно). Для сглаживания ступенек выходного напряжения необходим фильтр с частотой среза, равной высшей частоте генерируемого сигнала. Синтезатор можно применить в многофункциональной измерительной аппаратуре, например, в качестве цифрового свипгенератора, а также в качестве генератора тона в цифровых электромзыкальных инструментах.

Память для микропрограмм

Понятие микропрограммирование в сложившемся виде соответствует исполнению команды (в виде внешнего воздействия на автомат) как последовательности микроопераций. Микропрограммный автомат обычно содержит счетчик микроадреса, ПЗУ микрокоманд и исполнительное устройство (ИУ). Как правило, микрокоманды (МК) не дешифрируются в ИУ, и каждый их разряд соответствует одной элементарной функции автомата. Отдельные поля МК могут управлять счетчиком, т. е. разрешать условные или безусловные переходы в микропрограмме. Функции ИУ чаще всего выполняет АЛУ. Пугающая потребителя сложность программирования МП-автоматов — одна из причин сдержанного интереса к секционированным микропроцессорам серий К1802 и К1804.

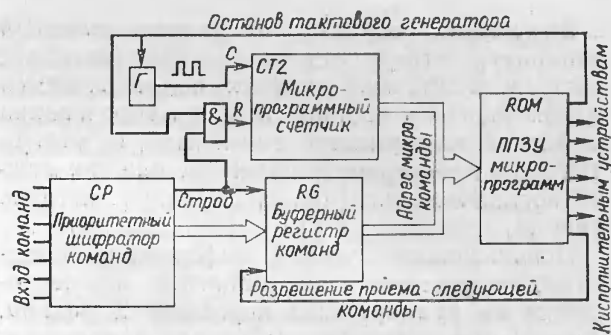


Рис. 8. Микропрограммный «релейный» автомат

Однако ИУ в виде АЛУ или МП может не быть совсем, и простейший микропрограммный автомат сведется к узлу из счетчика МК и ПЗУ. Так могут быть созданы простые и дешевые контроллеры, эффективно решающие задачи автоматизации, например контроллер, предназначенный для замены релейной автоматики (рис. 8).

Подобные контроллеры целесообразно применять прежде всего там, где группа внешних воздействий должна инициировать сложную и жестко привязанную ко времени последовательность сигналов управления. Они способны заменить сложные системы из триггеров и реле времени в технологическом оборудовании. Повышенное быстродействие, которое чаще всего ассоциируется с микропрограммными автоматами, здесь может оказаться несущественным, причем время исполнения каждого шага микропрограммы может варьироваться от долей микросекунды до десятков часов. Последовательность сигналов, генерируемая на выходах ППЗУ микропрограмм, после соот-

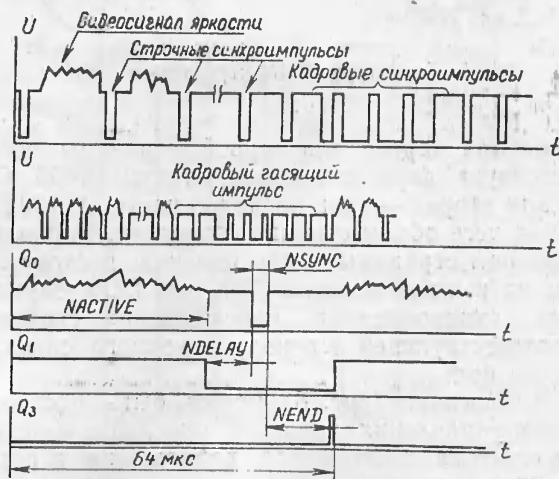


Рис. 9. Временные диаграммы видеосигналов и работы ТВ-синхрогенератора

ветствующего усиления используется для управления ИУ.

Другой пример — ТВ-синхрогенератор с возможностью получения испытательных изображений. Стандартный телевизионный сигнал (рис. 9) представляет собой смесь строчных и кадровых синхроимпульсов, импульсов гашения обратного хода луча по строкам и кадрам и яркостного сигнала [6]. Для правильной работы телевизионной техники временные соотношения компонент видеосмеси оговорены стандартами и должны поддерживаться с высокой точностью. По этой причине синхрогенератор в виде логического блока на ИМС малой и средней степени интеграции оказывается весьма сложным. Проще реализовать его в виде МП-автомата (рис. 10), состоящего из счетчика элементов разложения вдоль строки DD1, DD2, DD3, счетчика строк DD5, DD6,

DD7, ППЗУ строчного сигнала DD4 и ППЗУ кадрового сигнала DD8. Минимальный временной интервал элементов разложения, определяющий точность взаимного размещения компонентов синхросмеси, определяется периодом тактового генератора с кварцевой стабилизацией частоты. Полный синхросигнал снимается с выхода Q0 DD3.

Формирование синхросмеси происходит так: в исходном состоянии счетчик элементов считает импульсы тактового генератора и считывает содержимое ППЗУ DD4, начиная с адреса 0. Первые NACTIVE ячеек ППЗУ соответствуют активной части строки (полезное изображение) и содержат код 0111 — никаких импульсов. По окончании активной части на выходе Q1 DD3 устанавливается низкий уровень — такой код в этом разряде записан далее во все ячейки ППЗУ, и начинается строчный гасящий импульс. Затем через NDELAY тактов появляется низкий уровень на выходе Q 0 ППЗУ. Нулевой код в этом разряде записан в NSINC последовательных ячейках и определяет длительность строчного синхроимпульса. По его окончании гасящий импульс продолжается еще NEND тактов, что соответствует полному периоду строки. В следующей ячейке в разряде Q3 записана «1», которая сбрасывает счетчик строк, обеспечивая безусловный переход на нулевой адрес «микропрограммы».

Аналогично работают счетчик и ППЗУ кадров. Особенность заключается в том, что на время передачи кадрового синхроимпульса полярность и длительность строчных синхроимпульсов изменяются. Для этого ППЗУ DD4 содержит два варианта программы синхросигналов, переключаемые кадровым синхроимпульсом. Стандарт разложения изображения и временные соотношения контроллера зависят только от содержимого ППЗУ.

На основе ППЗУ можно создать генератор испытательных ТВ-изображений, если занести необходимую информацию в неиспользуемые разряды Q2 ППЗУ, или графический растровый дисплей, если выходы счетчиков элементов разложения и строк использовать для адресации ячеек ОЗУ, содержимое которого выводится на телевизионный экран. Схему программируемого синхрогенератора с применением ППЗУ, обеспечивающего формирование чересстрочной развертки в различных телевизионных стандартах, можно найти в работе [6].

Коррекция ошибок данных

При хранении и передаче информации в ЭВМ по ряду причин могут возникать ошиб-

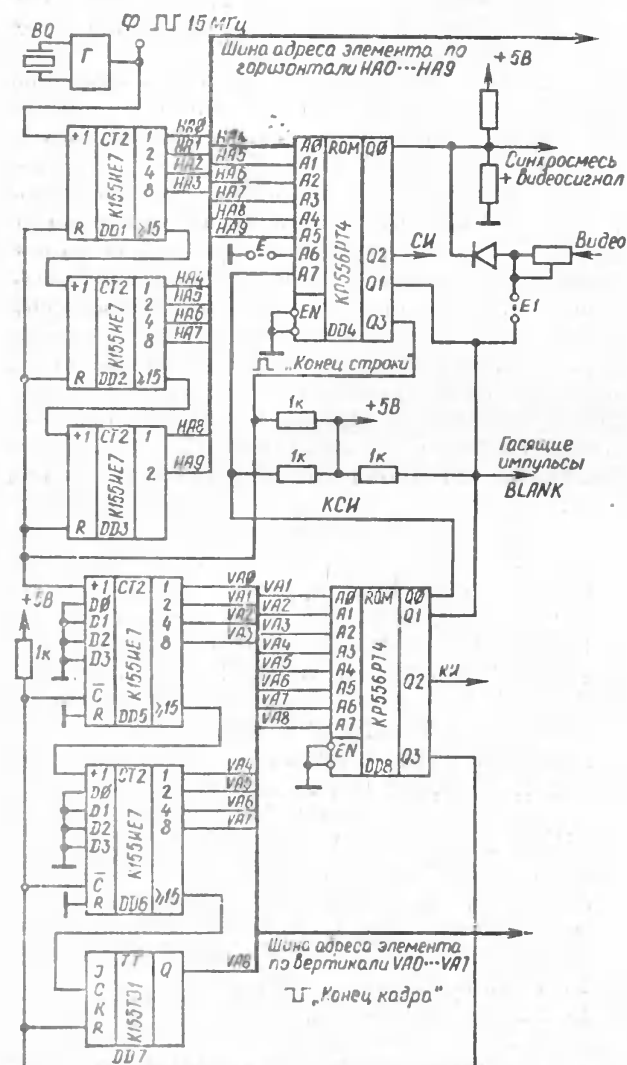


Рис. 10. Принципиальная схема ТВ-синхрогенератора

ки, поэтому в микроЭВМ повышенной надежности необходимы средства обнаружения и исправления возникающих ошибок. Одним из таких методов является знакомый читателю контроль по четности. Для этого вместе с двоичными данными передается дополнительный разряд четности, значение которого равно сумме по модулю 2 всех «единиц» данных. К сожалению, он позволяет обнаружить искажение только одного бита.

Расширением этого метода являются коды Хэмминга, впервые предложенные более 30 лет назад. Рассмотрим их применение на примере корректора одиночных ошибок в 8-разрядных словах (байтах) ОЗУ. Для коррекции ошибок вместе с N-разрядными словами полезной информации передаются K контрольных разрядов, причем $2^K \geq N+K+1$, и при N=8 K не должно быть менее 4. Значение каждого контрольного разряда (КР) C0...C3, как и при контроле по четности, вычисляется как сумма по модулю 2, но только по части информационных разрядов (табл. 1). Затем по принятой информации значения КР вновь вычисляются на приемной стороне и сравниваются с приня-

тым их значением — несовпадение указывает на ошибку.

Обратим внимание: уравнения Хэмминга составлены так, что при искажении одного информационного разряда изменяются не менее двух контрольных разрядов, а по их сочетанию можно однозначно вычислить искаженный разряд. Это позволяет отличить ошибки передачи полезной информации от ошибок КР: в последнем случае изменится только один КР. Код Хэмминга позволяет обнаружить (но не исправить) множественные ошибки, если отличия в КР таковы, что их нельзя получить из приведенных в табл. 1 уравнений в предположении изменения значения только одного информационного разряда.

Блок диагностики и коррекции ошибок по Хэммингу работает так: на входе устройства, например ОЗУ, по байту полезной информации вычисляются КР и вместе с информацией записываются в ячейки. Ясно, что каждое слово ОЗУ должно содержать N+K разрядов. При считывании ОЗУ по информационным разрядам вновь вычисляются КР и сравниваются с их значениями, считанными из памяти. Совпадение КР означает, что информация считана правильно.

При несовпадении возможно несколько вариантов. Если не совпадает только один из четырех КР, то искажены могут быть только сами КР, а информация считана правильно. Такая ошибка игнорируется. Если не совпадает более одного разряда, необходима проверка на допустимость получившегося несовпадения. Если проверка покажет, что несовпадение вызвано искажением одного информационного разряда, сочетание несовпадающих разрядов (синдром) S0...S3 укажет на искаженный разряд. Этот разряд следует инвертировать, чтобы исправить ошибку. Наконец, все оставшиеся варианты соответствуют многократной (некорректируемой) ошибке, о чем следует информировать центральный процессор. Возможность исправлять даже только однократные ошибки увеличивает надежность работы ОЗУ не менее чем в 200...1000 раз [7].

Изложенный алгоритм проще всего реализовать аппаратно с помощью ППЗУ (рис. 11). Здесь ППЗУ DD1 генерирует контрольные разряды в соответствии с табл. 1, а прочитанные из ОЗУ DD2 значения этих разрядов сравниваются с вычисленными в ППЗУ DD7 с помощью компаратора DD3, на выходе которого образуется синдром. Для обработки синдрома служат два идентичных ППЗУ DD4 и DD6, содержащих две страницы кодов.

Нулевая страница включает коды, превращающие DD6 в дешифратор искаженных информационных разрядов. Уровень «Лог. 1», появившийся на одном из выходов этой ИМС,

ТАБЛИЦА 1

ИСХОДНЫЕ УРАВНЕНИЯ ДЛЯ ВЫЧИСЛЕНИЯ СИНДРОМОВ

1	C0 =	A0 + A1 + A2 + A3 + A4 + A5 + A6 + A7	1
1	C1 =	A0 + A1 + A2 + A3 + A4 + A5 + A6 + A7	1
1	C2 =	A0 + A1 + A2 + A3 + A4 + A5 + A6 + A7	1
1	C3 =	A2 + A3 + A4 + A5 + A6 + A7	1

* HUMP * U61.02 Д.Л. * 04-НОУ-85 *

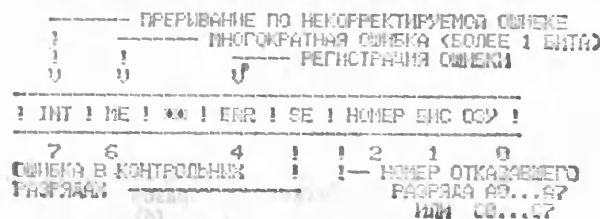
ТАБЛИЦА СОДЕРЖИМОГО ППЗУ ГЕНЕРАТОРА СИНДРОМОВ КР556FT4

0000:	00 07 03 04 09 0E 0A 0D 0A 0D 09 0E 03 04 00 07
0010:	08 0C 08 0F 02 05 01 06 01 06 02 05 08 0F 08 0C
0020:	0C 08 0F 08 05 02 06 01 06 01 06 02 0F 08 0C 08
0030:	07 09 04 03 0E 09 0D 0A 0D 0A 0E 09 04 03 07 0D
0040:	0D 0A 0E 09 04 03 07 0D 07 0D 04 03 0E 09 0D 0A
0050:	05 01 05 02 0F 08 0C 08 0C 08 0F 03 05 02 06 01
0060:	01 06 02 05 08 0F 08 0C 08 0C 08 0F 02 05 01 06
0070:	0A 0D 09 0E 03 04 00 07 00 07 03 04 09 0E 0A 0D
0080:	0E 09 0D 0A 07 0D 04 03 04 03 07 0D 0D 0A 0E 09
0090:	05 02 06 01 0C 08 0F 03 0F 08 0C 08 06 01 05 02
00A0:	02 05 01 06 08 0C 08 0F 03 0F 08 0C 01 06 02 05
00B0:	09 0E 0A 0D 0D 07 03 04 03 04 0D 07 0A 0D 09 0E
00C0:	03 04 00 07 0A 0D 09 0E 0D 0E 0A 0D 00 07 03 04
00D0:	08 0F 08 0C 01 06 02 05 02 05 01 06 08 0C 08 0F
00E0:	0F 08 0C 08 06 01 05 02 05 02 06 01 0C 08 0F 08
00F0:	04 03 07 0D 0D 0A 0E 09 0E 09 0D 0A 07 0D 04 03

СОДЕРЖИМОЕ ППЗУ ДИАГНОСТИКИ И КОРРЕКЦИИ К1558PZ

0000:	00 00 00 02 00 00 00 01 00 04 03 10 20 40 80 00
0010:	00 00 09 11 0A 10 10 0B 12 13 14 15 16 17 10

НАЗНАЧЕНИЕ РАЗРЯДОВ ВЫХОДА ППЗУ ДИАГНОСТИКИ:



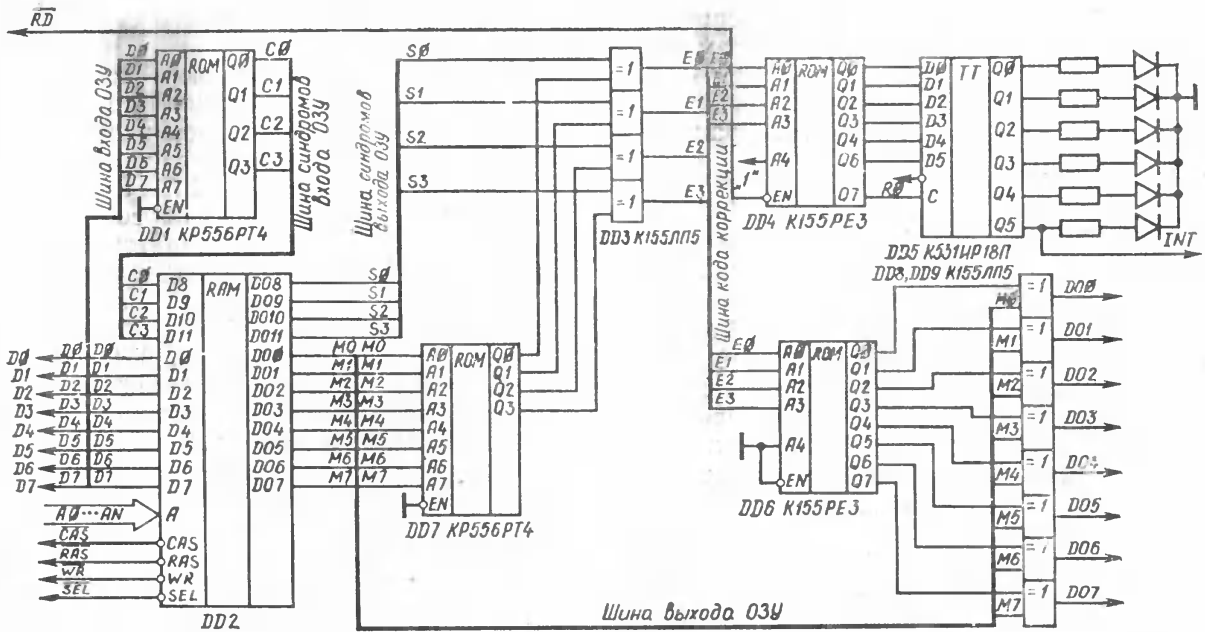


Рис. 11. Структура блока диагностики и коррекции ошибок по Хэммингу

приводит к инверсии соответствующего разряда с помощью DD8, DD9.

Первая страница использована в DD4 и содержит информацию о характере появившейся ошибки для индикации. Светодиодный индикатор позволяет высветить номер отказавшего разряда и тип ошибки. При возникновении многократной ошибки корректор по линии INT вызывает прерывание. Описанный вариант корректора можно применять не только совместно с ОЗУ, но и при организации линий передачи данных повышенной надежности. Вся обработка информации, связанная с диагностикой и коррекцией, проходит аппаратно, занимая дополнительно 100...200 нс, и полностью «прозрачна» для центрального процессора.

Вычисления в реальном времени

Для обработки сигналов в реальном масштабе времени необходимы процессоры со значительным быстродействием на достаточно сложных математических операциях. Если разрядность обрабатываемых данных невелика (8—12 бит), то эти операции нетрудно реализовать табличным способом с помощью ППЗУ.

Простейший вариант табличного умножителя $M \times N$ — это ППЗУ с $M+1$ входами адре-

са, содержащее значения произведений для всех возможных состояний сомножителей (рис. 12). Время вычислений здесь совпадает с временем выборки и для массовых ППЗУ составляет 100...500 нс. Однако для реализации умножения 8×8 необходимо ППЗУ объемом 64К слов. Для сокращения объемов таблиц применяются комбинации с операциями типа сложения, которые способны выполнить простейший процессор. Например, если воспользоваться алгебраическим тождеством, для умножения 8×8 потребуются ППЗУ объемом всего 512 слов, а последовательность опе-

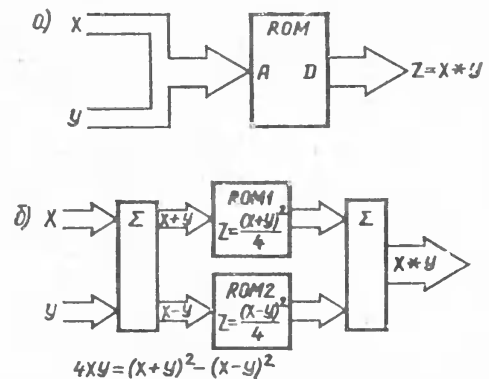


Рис. 12. Быстрые умножители на основе ППЗУ: непосредственное табличное вычисление произведения (а); использование алгебраических преобразований для сокращения объема таблиц (б)

раций, изображенных на рис. 12, б, способен выполнить простейший микропрограммный процессор за время 300...500 нс [8]. В результате общее время вычисления увеличится незначительно при значительном сокращении затрат на аппаратуру.

Наглядность кодирования

Для занесения информации в ППЗУ чаще всего используются файлы форматов BIN (отображения памяти), INTELLEC HEX (формат загрузчика ОС ISIS-II) и LDA (формат абсолютного загрузчика ПЛОС СМ ЭВМ). В этих случаях информация представляется в специальном виде, и если ППЗУ содержит программу, полученную не с помощью одного из кросс-трансляторов, ее ручная подготовка крайне затруднена.

Для документирования и подготовки данных можно использовать форматы разнообразных распечаток (DUMP), которые подготавливаются и редактируются системным редактором имеющейся ОС. Однако для перечисленных выше применений трансляцию таблицы содержимого ППЗУ в DUMP-формат разработчик обязан произвести вручную. На этом этапе теряется наглядность замысла разработчика,

ТАБЛИЦА 2

```

.ENABL LC/ABS ; MACRO-11/RAFOS
.NLIST MD/OND
;***** СБОРЩИК *****
.MACRO FROM X ; FROM <ТЕКСТ> -
.NLIST SRC ; - ДИРЕКТИВЫ ЗАДАЧИ
.BYTE =0 ; РАБОЧАЯ ПЕРЕМЕННАЯ
.IRP Y,<X> ; ОБРАБОТКА ТЕКСТА
.BYTE =*BYTE!Y
.ENDR
.BYTE *BYTE ; ТЕКСТ ОБРАБОТАН
.NLIST SRC ; ЗАНЕСЕНИЕ СОБРАННОГО
.ENDM FROM ; БАЙТА В ВИДНОМ
.NLIST HEX ; ФАЙЛ.
;***** ОПРЕДЕЛЕНИЯ БИТ *****
NOSYNC = 1 ; ВЫСОКИЙ УРОВЕНЬ НА 00
NOBLANK = 1*2 ; ВЫСОКИЙ УРОВЕНЬ НА 01
RESET = 1*10 ; ИМПУЛЬС СБРОСА НА 03
NORMAL = 7 ; АКТИВНАЯ ЧАСТЬ КАРА (0111)
;***** ПРОГРАММА ДЛЯ ПЗУ *****
; ОПРЕДЕЛЕНИЯ :
NACTIVE=256./20 ; ЧИСЛО ЭЛЕМЕНТОВ В СТРОКЕ
NDELAY =64./20 ; ОСТАЛЬНЫЕ ПАРАМЕТРЫ СМ.
NSYNC = 1 ; РИС. 9 В СТАТЬЕ
NEND = 120./20
; КОДИРОВАНИЕ :
= 0 ; НАЧАЛЬНЫЙ АДРЕС КОДОВ

.REPT NACTIVE ; ИСПРАВЛЕНИЕ
FROM <NSYNC,NOBLANK>
.ENDR

.REPT NDELAY ; ПРОМЕНУСК ДО
FROM <NSYNC> ; СИНХРОИМПУЛЬСА
.ENDR

FROM <0> ; СИНХРОИМПУЛЬС

.REPT NEND ; КОНЕЦ СТРОКИ
FROM <NSYNC>
.ENDR

FROM <RESET> ; СБРОС СЧЕТЧИКА

```

.END

появляются ошибки, отсутствует возможность документирования.

Для исключения ошибок и получения возможности работы с понятным текстом можно воспользоваться имеющимся в доступной ОС макроассемблером. В операционной системе РАФОС на ЭВМ типа СМ и «Электроника 60» такую возможность обеспечит ассемблер МАКРО-11. Для его адаптации необходимо предварительно написать макроопределение-сборщик и определить мнемонические имена для отдельных бит и наиболее часто встречающихся их комбинаций (табл. 2). Так получится простейший транслятор, преобразующий мнемонический текст-описание содержимого ППЗУ в двоичный файл. Подробнее об адаптации макроассемблеров можно прочитать в работе [9]. Для управления адресом трансляции используется имеющийся в МАКРО-11 механизм («точка»).

Теперь с содержимым ППЗУ можно работать на уровне «словесного» описания, а также пользоваться комментариями, директивами повторения и условной трансляции, встроенными в макроассемблер. Все это существенно облегчает подготовку данных и снижает вероятность ошибки. Полученный текст вместе с макроопределением обрабатывается МАКРО-11, как обычная программа, и получается листинг и LDA-файл. Последний преобразуется в формат отображения памяти с помощью программ CONVRT и MIXUP (из пакета PROMER ИОФ АН СССР) или аналогичных и передается в программатор. Текст программы для строчного ППЗУ ТВ-синхрогенератора приведен в табл. 2. Описанный прием применим и для макроассемблера М80 ОС-1800 на ЭВМ, совместимых с СМ-1800.

ЛИТЕРАТУРА

1. Учись работать с ПЗУ. — Микропроцессорные средства и системы, 1985, № 3, с. 71—90.
2. Григорьев В. Л. Программное обеспечение микропроцессорных систем. — М.: Энергоатомиздат, 1983. — 208 с.
3. Микропроцессоры: системы программирования и отладки / Мясников В. А., Игнатьев М. Б., Кочкин А. А., Шейнин Ю. Е. — М.: Энергоатомиздат, 1985. — 272 с.
4. Источники вторичного электропитания / Под ред. Ю. И. Конева — М.: Радио и связь, 1983, с. 186—223.
5. Гнатек Ю. Р. Справочник по цифроаналоговым и аналого-цифровым преобразователям. — М.: Радио и связь, 1982, с. 255—257.
6. Иванов В. Программируемый телевизионный синхрогенератор. — Радио, телевидение, электроника, 1985, № 8, с. 7—10.
7. Trebar A. Soft error detection and correction board. — Practical Electronics, 1983, № 9, p. 32—37.
8. Greaves D. J. Digital polyphonic keyboard. — Electronics and Wireless World, v. 91, № 1595 (Sep. 1985), p. 37—40.
9. Лукьянов Д. А. Как написать кросс-ассемблер. — Микропроцессорные средства и системы, 1985, № 4, с. 35—41.

Статья поступила 12 ноября 1985 г.

ПЗУ ВМЕСТО ПРОИЗВОЛЬНОЙ ЛОГИКИ

Широко известно использование программируемых ПЗУ в качестве энергонезависимой памяти, которая необходима для хранения программных процедур перезапуска микропроцессорной системы (МС) и объектного кода рабочих программ микроконтроллеров. Но ППЗУ, кроме того, успешно применяются для хранения различного рода табличной информации в преобразователях кодов и генераторах функций, а также микрокода в МС с микропрограммным управлением.

Ниже приводится несколько практических примеров использования биполярных ППЗУ с плавкими перемычками вместо схем произвольной логики. Такие схемы часто встречаются при разработке МП-систем и обычно реализуются на основе стандартных ИМС малой и средней степеней интеграции. ППЗУ достаточно хорошо заменяют эту дискретную логику, сопрягающую различные БИС в законченную МС.

Контроллер шины. На рис. 1 представлена обобщенная схема МС на базе МП К1810ВМ86. Уникальным свойством этого микропроцессора является возможность выбора с помощью входа MN/MX одного из двух его рабочих режи-

мов, в наибольшей степени подходящего для конкретного применения. В соответствии с этим изменяется логика контроллера шины. Минимальный режим ($MN/\overline{MX}=1$) оптимизирован для малых МС с одним процессором. Микропроцессор обеспечивает управление системной шиной типа И41 (рис. 1). Максимальный режим ($MN/\overline{MX}=0$) предназначен для создания мультипроцессорных систем различной конфигурации. В этом режиме МП формирует промежуточную локальную шину, для преобразования которой в системную магистраль И41 требуется более сложная логика. Представленная на рис. 2 схема такого контроллера содержит ряд триггерных и комбинационных схем.

Замена комбинационной логики контроллера на БИС ППЗУ позволяет сократить число используемых кристаллов на плате центрального процессора. При необходимости это же ППЗУ можно использовать и для других целей, например для формирования сигнала подтверждения XACK от главного контроллера прерываний, обычно располагаемого на одной плате с МП (см. рис. 2).

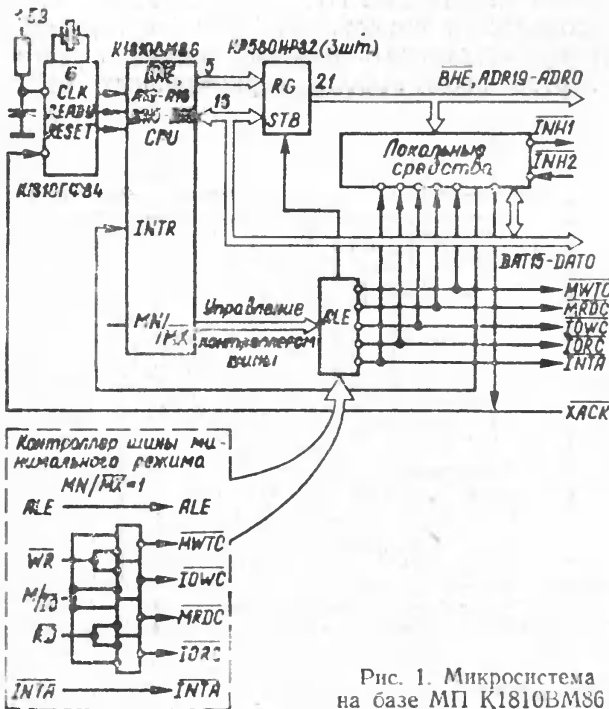


Рис. 1. Микросистема на базе МП К1810ВМ86

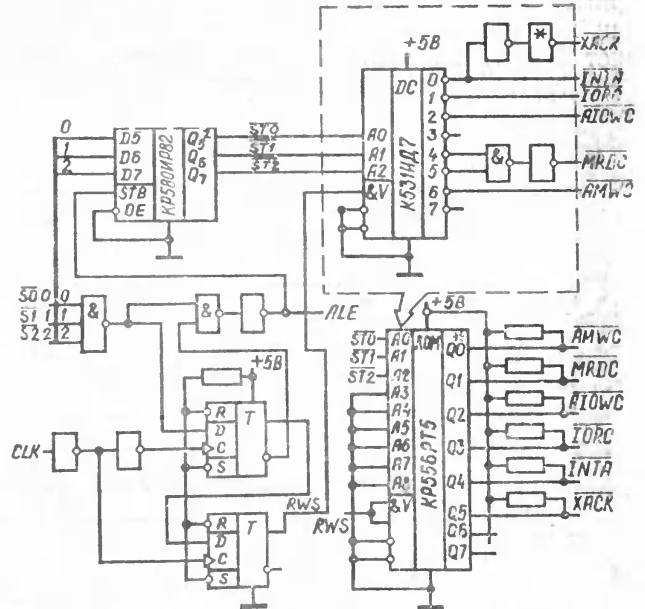


Рис. 2. Контроллер шины максимального режима ($MN/\overline{MX}=0$)

Интерфейсная логика шины. Один из практически возможных вариантов подключения к 16-разрядной внутримодульной магистрали типа И41 микросхем ПЗУ, ОЗУ статического типа, а также периферийных БИС ввода-вывода приведен на рис. 3. Конфигурации такого типа могут быть использованы для построения одноплатных микроЭВМ на базе МП К1810ВМ86.

Такой интерфейс шины включает логику выборки кристаллов, которая генерирует сигналы выборки БИС \overline{CS} и \overline{CE} . Оперативная память и область ввода-вывода содержат по два 8-разрядных банка каждый. Выбор банка осуществляется сигналами \overline{BHE} (H-банк) и \overline{ADRO} (L-банк), в результате чего реализуется обмен либо байтами, либо словами. Кроме этого в состав интерфейса шины входит логика подтверждения операции обмена, формирующая сигнал \overline{XACK} , и логика запрета, основанная на сигналах $\overline{INH1}$ и $\overline{INH2}$. Сигналы подтверждения \overline{XACK} и запрета $\overline{INH1}$ и $\overline{INH2}$ формируются по схеме «монтажное ИЛИ».

В схеме использованы типовые микросхемы памяти и ввода-вывода. Кристаллы ПЗУ так же, как и периферийные БИС, имеют линию \overline{OE} или \overline{RD} для стробирования выхода при чтении данных. Интерфейсом такого типа обладают УФРПЗУ серии К573 и ПЗУ серии КР556, а также периферийные микросхемы серии КР580 и К1810. Вместе с этим большинство микросхем ОЗУ статического типа, например К541РУ1/РУ2, К537РУ3, не имеют линии для приема команды чтения. Это усложняет логику их подключения к магистралям типа И41. Другой класс микросхем, например

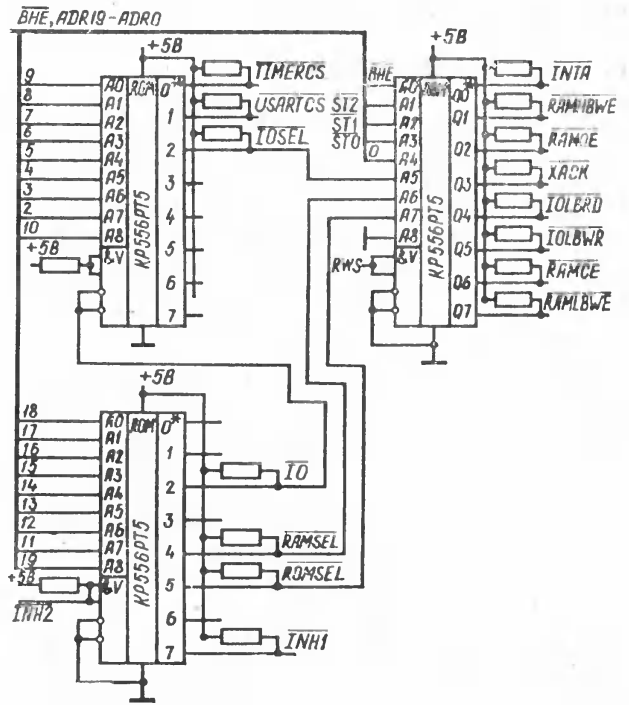


Рис. 4. Логика сопряжения БИС в микросистеме

К537РУ8, имеющий вместе с линией выборки кристалла вход \overline{OE} для команды чтения, подключается к шине по аналогии с приборами ввода-вывода.

Такую интерфейсную логику шины произвольного типа целесообразно заменить на несколько биполярных ППЗУ. В результате этого появляется возможность реконструировать интерфейс, изменить базовые адреса памяти и портов ввода-вывода периферийных БИС

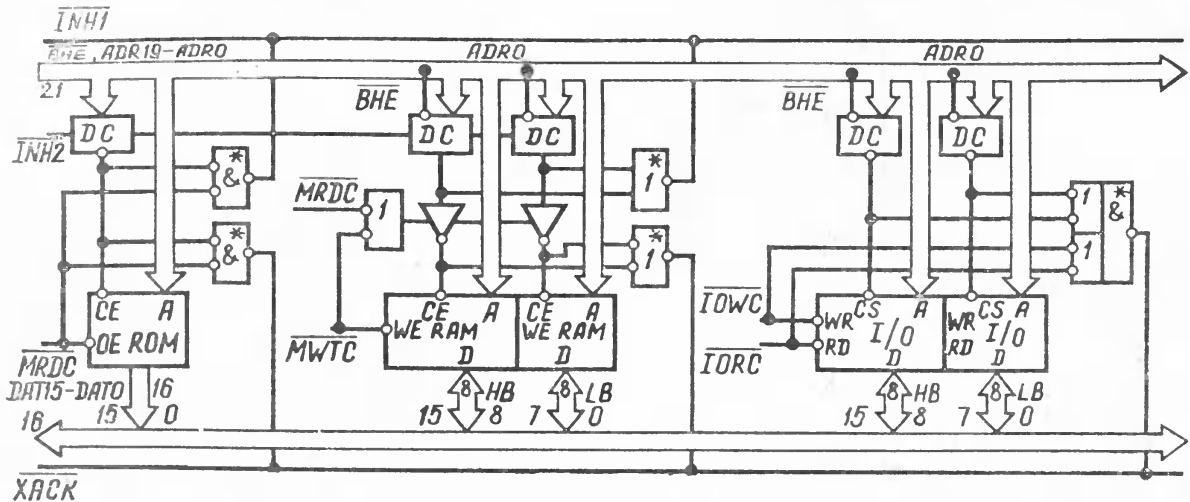


Рис. 3. Логика подключения БИС к шине И41

без модификации печатного монтажа. Более того, выходная логика контроллера шины (см. рис. 1 или рис. 2) и интерфейсная логика шины (см. рис. 3) могут быть «защиты» в одни и те же ПЗУ. Эти ПЗУ реализуют непосредственное преобразование сигналов МП в сигналы управления приборов памяти и ввода-вывода. Рассмотренная конфигурация может быть полезной в одноплатных вариантах МС, когда отсутствует необходимость в физическом наличии шины ИА1.

Практически для реализации последнего подхода требуется три микросхемы ПЗУ КР556РТ5. Схема их разводки приведена на рис. 4. При анализе схемы следует учесть, что порты БИС КР580ВВ51 (USART) и КР580ВИ53 (TIMER) были размещены в L-банке 16-разрядного физического пространства ввода-вывода. Входы и выходы ПЗУ были выбраны с учетом удобства разводки монтажа.

Логика сдвигов РАЛУ. Этот пример взят из практики проектирования МС с микропрограммами (МП) управлением на базе секционированного МПК БИС серии К1804. При построении регистрового арифметического логического устройства (РАЛУ) на основе микропроцессорных секций К1804ВС1/ВС2 возник-

ла задача реализации логики сдвигов, предназначенной для управления двунаправленными выводами устройств сдвига ОЗУ и рабочего регистра Q [1, 2]. Для этой цели может быть использована схема управления состоянием и сдвигами типа К1804ВР2. В более простых системах обычно применяют мультиплексоры связи сдвигающих устройств, реализованные на двух микросхемах К555КП12. Типовая схема такой логики сдвигов приведена на рис. 5, а. Возможности последней схемы ограничены из-за малого числа входов мультиплексора К555КП12.

На рис. 5, б представлена универсальная логика сдвигов с расширенными возможностями, реализованная на двух ППЗУ КР556РТ5. Эта же схема включает в себя мультиплексор выходного переноса С, поступающий в разряд С-флага (CF) слова состояния программы. В зависимости от класса микрооперации, выполняемой РАЛУ (арифметическая или логическая), формируется соответственно арифметический или логический перенос С. При сдвигах в качестве выходного переноса С используется содержимое одного из выдвигаемых разрядов. Реализация логики сдвигов с помощью ППЗУ замедляет работу МС примерно на 25 нс по сравнению с вариантом на мультиплексорах.

Дешифратор микроинструкций. В МР-микросистемах в качестве дешифратора микроинструкций блока микропрограммного управления (БМУ), построенного на базе БИС К1804ВУ1/ВУ2, обычно используется схема управления следующим адресом К1804ВУ3 [1, 2]. Эта схема имеет достаточно представительный набор из 16 условных и безусловных микроприказов, кодируемых 4-разрядным полем. Нетрудно заметить, что она полностью моделируется с помощью ППЗУ К155РЕ3.

Разработчик МС может по своему усмотрению сформировать систему микроприказов для К1804ВС1/ВС2 в зависимости от класса решаемых задач, например, состоящую из восьми условных команд (табл. 1). Реализация такого дешифратора на ППЗУ К155РЕ3 и схема его подключения приведены на рис. 6, а. Дешифратор обеспечивает управление тремя внешними источниками адресного кода: МАР-дешифрирующее ППЗУ начального адреса ($\overline{ME}=0$), PL — поточный регистр поля адреса ветвления ($\overline{PE}=0$) VECT — источник векторного адреса ($\overline{ME}=\overline{PE}=1$). При необходимости к дешифратору может быть подключен также счетчик микропрограммных циклов, управляемый двумя сигналами: \overline{CTL} — загрузочный счетчика и \overline{STE} — декремент счетчика. По сравнению с К1804ВУ3 такой дешифратор экономит один разряд микрокомандного слова,

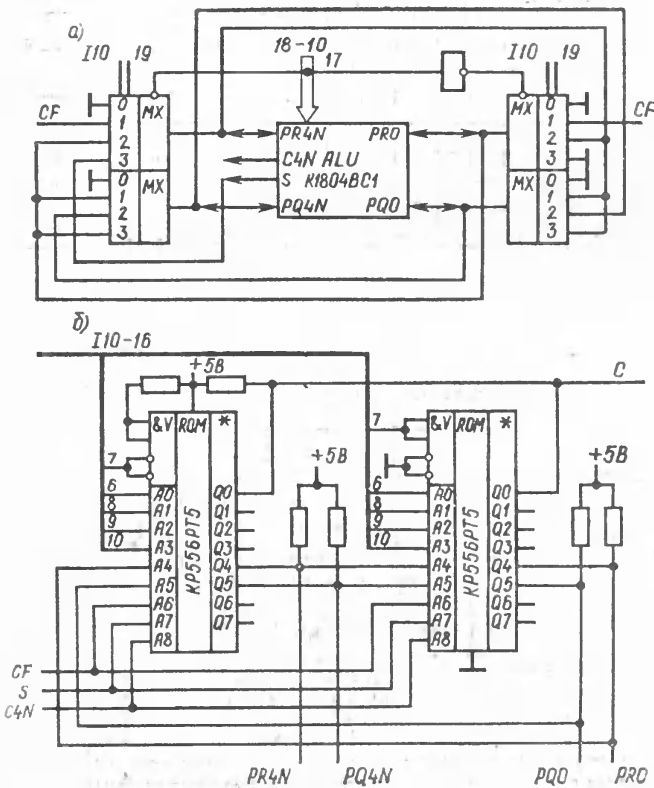


Рис. 5. Логика сдвигов на базе мультиплексоров К555КП12 (а) и на базе ППЗУ КР556РТ5 (б)

Набор условных операций управления следующим адресом

Мнемоника	Наименование операции	Вход		Выход				Источник адреса	Состояние стека	Состояние счетчика			
		T2 - I0	TST	S1	S0	PE	PUP				CTL	CTE	ME
CRTN	Условный возврат RTN/CONT	0	1 0	1 0	0 0	0 1	0 0	1 1	1 1	1 0	F0 MPC	POP	— —
CCOJS	Условный переход к подпрограмме COJS/PUSH	1	1 0	1 0	0 0	0 0	1 1	1 0	1 0	F0 MPC	PUSH MPC	PUSH	— LOAD
CLOOP	Условный переход по циклу LJMP/POP	2	1 0	1 0	0 0	1 0	0 0	1 1	1 0	F0 MPC	— MPC	— POP	DEC —
CJPP	Условный переход к выталкиванию JPP/CONT	3	1 0	1 0	1 0	0 1	0 0	1 1	1 0	PL MPC	POP MPC	POP	— —
CJSP	Условный вызов подпрограммы JSP/CONT	4	1 0	1 0	1 0	0 1	1 0	1 1	1 0	PL MPC	PUSH MPC	—	— —
CJP	Условный переход JP/CONT	5	1 0	1 0	1 0	1 0	1 0	1 1	1 0	PL MPC	— MPC	— —	— —
CJMAP	Условный переход по MAP JMAP/CONT	6	1 0	1 0	1 0	1 0	0 1	1 1	0 1	MAP MPC	— MPC	— —	— —
CJV	Условный переход по вектору JV/CONT	7	1 0	1 0	1 0	1 0	1 0	1 1	1 0	VECT MPC	— MPC	— —	— —

Примечание. CTL — загрузка счетчика циклов; CTE — декремент счетчика циклов; ME — разрешение выхода MAP; PE — разрешение выхода потокового регистра.

Таблица 2

Набор операций для схемы, приведенной на рис. 6, б

Наименование операции	NA4...NA0 HEX-код	Наименование операции	NA4...NA0 HEX-код
Условный переход по PL, если CF	0	Условный переход по PL, если не CF	10
» OF	1	» OF	11
» SF	2	» SF	12
» ZF	3	» ZF	13
» HF	4	» HF	14
» PF	5	» PF	15
» TEST	6	» TEST	16
» READY	7	» READY	17
Переход по PL, если INTR·INTE	8	Безусловный переход по MAP	18
» HOLD·LOCK	9	» VECT	19
» Z	A	» PL	1A
Резерв	B	Вызов подпрограммы	1B
»	C	Возврат	1C
»	D	Продолжение (CONT)	1D
»	E	Резерв	1E
»	F	»	1F

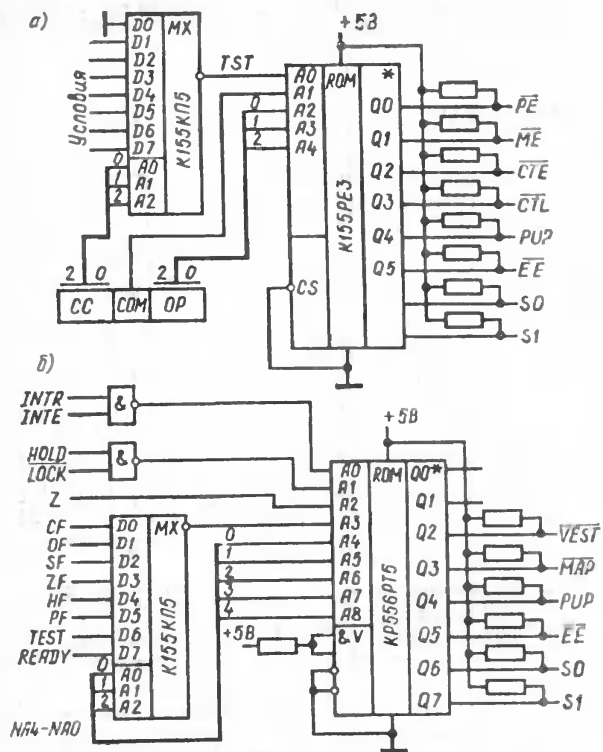


Рис. 6 Дешифратор микроинструкций управления следующим адресом с форматным (а) и бесформатным (б) кодированием

однако быстродействие МС снижается примерно на 20 нс в расчете на один такт. В ППЗУ K155PE3 этот разряд отводится для управления инверсией тестового входа TEST. Один из входов мультиплексора условий следует заземлить. Выбирая этот вход и управляя входом инверсии COM, можно реализовать дополнительно девять безусловных операций (см. табл. 1).

На рис. 6, б представлен еще один вариант дешифратора микроинструкций управления следующим адресом, отличающийся от предыдущего совмещенностью полей выбора условия CC, инверсии COM и кода операции OP, для чего используется всего пять управляющих разрядов NA4...NA0. Свободные входы ППЗУ заняты дополнительным вводом условий, которые могут потребоваться в любой из 32 микрокоманд, например, Z — признак нуля результата текущей микроинструкции РАЛУ. Этот признак важен для организации счетчиков МР-циклов на основе одного из регистров РАЛУ. Возможный набор команд для такого дешифратора представлен в табл. 2. Набор хорошо согласуется с группой команд передачи управления на командном уровне.

Приведенные примеры иллюстрируют возможность использования биполярных ППЗУ вместо логических схем произвольного типа. В связи с совершенствованием техники программирования и прожига ППЗУ [3, 4] процесс реализации таких схем не представляет труда. Рассмотренный подход к реализации схем произвольной логики дает возможность проектировать компактные МС, упростить разводку печатной платы, а также модифицировать МС за кратчайшее время. В процессе проектирования МС следует учитывать возможность появления кратковременных ложных выбросов на выходах ППЗУ в моменты переключения адресов. Именно по этой причине на схемах рис. 2 и рис. 4 строб RWS заведен не на адресный вход, а на вход выборки кристалла.

ЛИТЕРАТУРА

1. Мик Дж., Брик Дж. Проектирование микропроцессорных устройств с разрядно-модульной организацией. В 2-х книгах: Пер. с англ. — М.: Мир, 1984. — Кн. 1.
2. Проектирование цифровых систем на комплектах микропрограммируемых БИС/С. С. Булгаков, В. М. Мещеряков, В. В. Новоселов, Л. А. Шумилов; Под ред. В. Г. Колесникова. — М.: Радио и связь, 1984.
3. Дианов А. П., Щелкунов Н. Н. Модуль программирования микросхем ПЗУ. — Микропроцессорные средства и системы, 1985, № 3, с. 80.
4. Лукьянов Д. А. Схемотехника универсальных программаторов ПЗУ. — Микропроцессорные средства и системы, 1985, № 3, с. 84.

Статья поступила 21 ноября 1985 г.

В БОЛЬШОМ ЗАЛЕ ПОЛИТЕХНИЧЕСКОГО

11 марта 1986 г. журнал «Микропроцессорные средства и системы» проводит семинар «Микропроцессорная техника в медицинских приборах», а 8 апреля 1986 г. — семинар «Программирование — новая вузовская специальность» (обсуждение учебного плана по новой специальности для вузов).

Оба семинара состоятся в 12 часов в Большом зале Центрального лектория Политехнического музея.

Телефон для справок — 923-00-19 (методист Татьяна Юрьевна Ермолаева).

МИКРОСХЕМА ДИНАМИЧЕСКОГО ОЗУ КР565РУ6

Динамическое запоминающее устройство с произвольной выборкой типа КР565РУ6 выполнено на п-канальных МОП-транзисторах. Информационная емкость ОЗУ 16 Кбит. Для питания микросхемы используется один источник с напряжением $5\text{ В} \pm 10\%$.

Кристалл ОЗУ заключен в 16-выводной пластмассовый корпус (рис. 1) шириной 7,5 мм. Назначение выводов показано в табл. 1.

Кристалл ОЗУ содержит накопитель, выполненный на основе однотранзисторных ячеек памяти, систему считывающих усилителей, дешифраторы адреса строки и столбца, мультиплексированный адресный регистр, устройства ввода-вывода и схему синхронизации и управления перечисленными узлами (рис. 2).

Накопитель ОЗУ организован в виде двух блоков по 8 Кбит, которые образуют единую матрицу 128×128 бит. Для выбора одной из 16 384 ячеек памяти на 7-разрядный адресный регистр подается в мультиплексированном режиме 14-разрядный адресный код. Код адреса строки (A0...A6) фиксируется на адресном регистре строк с помощью сигнала RAS. Код адреса столбца (A7...A13) — на адресном регистре столбцов с помощью сигнала CAS.

Микросхема КР565РУ6 позволяет реализовать режимы: запись (рис. 3), считывание (рис. 4), регенерация (рис. 5), считывание — модификация — запись (рис. 6), страничные режимы записи и считывания. Временные параметры входных сигналов, обеспечивающих правильную работу ЗУ, приведены в табл. 2. Основные электрические параметры показаны в табл. 3. Данные приводятся для трех групп микросхем: КР565РУ6Б, КР565РУ6В, КР565РУ6Г.

Таблица 1

Назначение выводов

Вывод	Назначение
1	Свободный
2	Информационный вход DI
3	Вход сигнала запись WE
4	Вход сигнала выборки строк RAS
5...7	Адресные входы A0, A2, A1
8	Напряжение источника питания, U _{cc}
9	Свободный
10...13	Адресные входы A5, A4, A3, A6
14	Информационный выход DO
15	Вход сигнала выборки столбцов CAS
16	Общий вывод U _{GND}

В ОЗУ использована схема стробируемого CAS, т. е. события в цепи, определяемой сигналом CAS, запрещаются до прихода разрешающего сигнала, вырабатываемого одним из формирователей цепи RAS. Такой режим работы позволяет подавать внешний сигнал CAS, как только истекло время t_{trah} и код адреса строки на адресных входах изменился на код адреса столбца.

Рис. 1. Условное графическое изображение микросхемы КР565РУ6

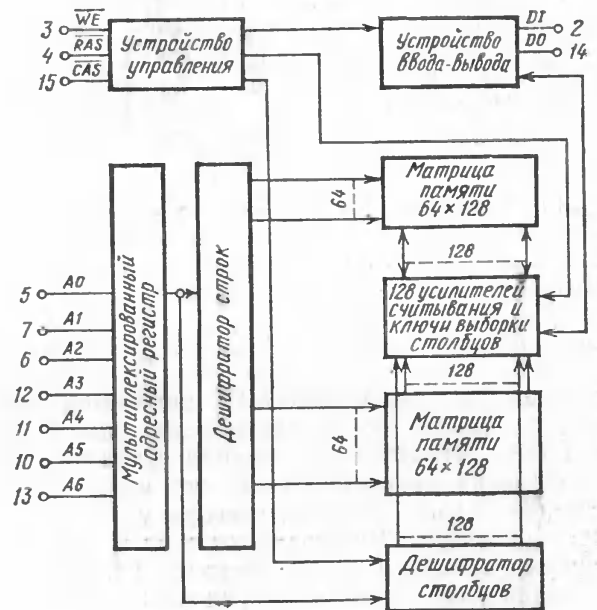
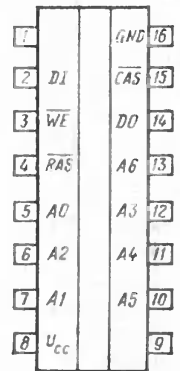


Рис. 2. Электрическая функциональная схема

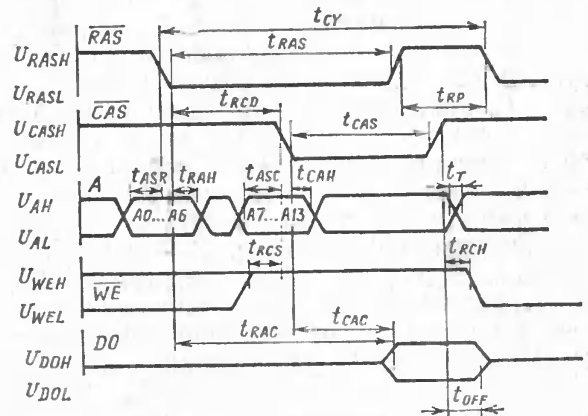


Рис. 3. Режим записи

Временные параметры входных сигналов микросхемы КР565РУ6

Наименование	Обозначение	Норма для микросхемы, нс					
		КР565РУ6Б		КР565РУ6В		КР565РУ6Г	
		мин.	макс.	мин.	макс.	мин.	макс.
Время установления сигнала адреса строк относительно \overline{RAS}	t_{ASR}	0		0		0	
Длительность сигнала \overline{RAS}	t_{RAS}	120	10 000	150	10 000	200	10 000
Время удержания сигнала адреса строк относительно \overline{RAS}	t_{RAH}	15		20		30	
Длительность интервала между сигналами \overline{RAS}	t_{RP}	100		120		150	
Длительность сигнала \overline{CAS}	t_{CAS}	70	10 000	90	10 000	120	10 000
Время установления сигнала \overline{RAS} относительно \overline{CAS}	t_{RCD}	30	50	35	60	40	80
Время удержания сигнала \overline{RAS} относительно \overline{CAS}	t_{RSH}	70		90		120	
Время установления сигнала адреса столбцов относительно \overline{CAS}	t_{ASC}	0		0		0	
Время удержания сигнала столбцов относительно \overline{CAS}	t_{CAH}	25		35		45	
Время установления сигнала считывания относительно \overline{CAS}	t_{RCS}	0		0		0	
Время сохранения сигнала считывания после \overline{CAS}	t_{RCH}	15		20		30	
Время установления сигнала DI относительно \overline{CAS}	t_{DS}						
Время удержания сигнала DI относительно \overline{CAS}	t_{DWS}	0		0		0	
Длительность сигнала \overline{WE}	t_{DCH}	30		40		50	
Время удержания сигнала \overline{WE} относительно \overline{CAS}	t_{DWH}	35		45		80	
Время установления сигнала \overline{WE} относительно \overline{CAS}	t_{WCS}	0		0		0	
Время записи по сигналу \overline{CAS}	t_{CWL}	55		70		95	
Время записи по сигналу \overline{RAS}	t_{RWL}	55		70		95	
Длительность фронта	t_{LH}, t_{HL}	3	35	3	35	3	35
Время установления сигнала \overline{CAS} относительно \overline{WE}	t_{CWD}	70		95		120	
Длительность интервала между сигналами \overline{CAS}	t_{CPN}	35		40		50	

Таблица 3

Основные электрические параметры

Наименование	Обозначение	КР565РУ6Б		КР565РУ6В		КР565РУ6Г	
		мин.	макс.	мин.	макс.	мин.	макс.
Время цикла считывания (записи, нс)	t_{CY}	230		280		360	
Время выборки относительно сигнала \overline{CAS} , нс	$t_{CAC}(1)$		70		90		120
Время выборки относительно сигнала \overline{RAS} , нс	$t_{RAC}(1)$		120		150		200
Период регенерации, мс	T_{REP}		2		2		2
Ток потребления, mA	$I_{CC}(2)$		4		4		4
Ток потребления динамический, mA	$I_{CCB}(3)$		27		25		20
Напряжение «Лог. 1» сигнала выходной информации при токе нагрузки 2 mA, В	U_{DON}	2,4		2,4		2,4	
Напряжение «Лог. 0» сигнала выходной информации при токе нагрузки 4 mA, В	U_{DON}		0,4		0,4		0,4
Время сохранения выходной информации относительно сигнала \overline{CAS} (время перехода в состояние высокого импеданса), нс	t_{OFF}	0	35	0	40	0	50

Примечание. (1) — нагрузка 100 пФ; (2) — $U_{RAS}=U_{RASH}$, $U_{CAS}=U_{CASH}$; (3) — $t_{CY} \geq 230$ нс, гр. Б; $t_{CY} \geq 280$ нс, гр. В; $t_{CY} \geq 360$ нс, гр. Г.

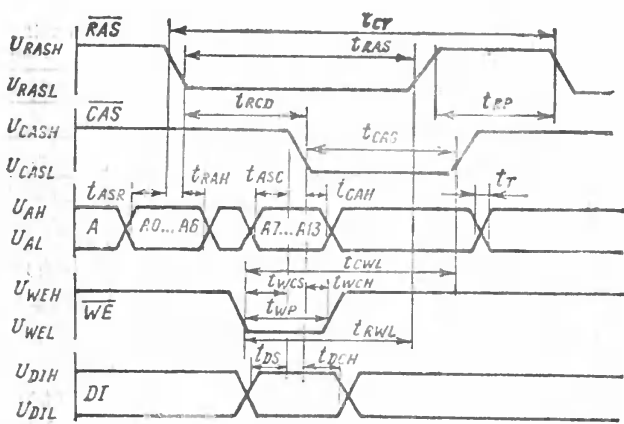


Рис. 4. Режим считывания

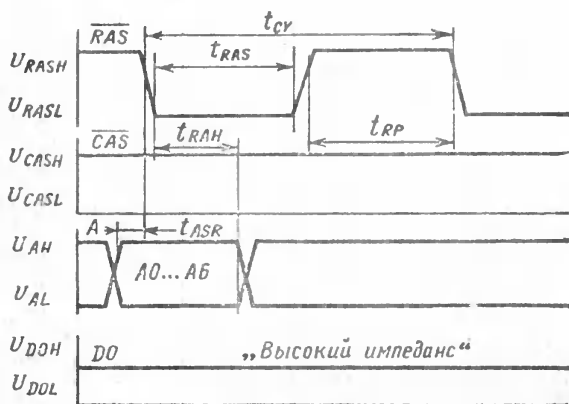


Рис. 5. Режим регенерации

При работе ОЗУ в режиме считывания информация появится на выходе через время t_{RAC} . Если задержка между сигналами \overline{RAS} и \overline{CAS} $t_{RCD} > t_{RCD\max}$, то нормальное функционирование микросхемы не нарушается, только время выборки по отношению к сигналу \overline{RAS} увеличивается на величину разности $t_{RCD} - t_{RCD\max}$.

Когда сигнал \overline{CAS} находится в состоянии «Лог. 1», информационный выход принимает состояние высокого импеданса с плавающим потенциалом. В режиме записи, когда сигнал \overline{WE} приходит раньше сигнала \overline{CAS} , информационный выход также находится в состоянии высокого импеданса в течение всего цикла записи.

При считывании информационный выход переходит из состояния с высоким импедансом в активное состояние («0» или «1») через время, равное времени выборки. В таком включенном состоянии информационный выход остается до перехода сигнала \overline{CAS} в неактивное высокое логическое состояние. При этом не играет роли, снимается сигнал \overline{RAS} или нет. В течение цикла считывания информации сигнал \overline{WE} должен поддерживаться в высоком логическом состоянии с момента подачи сигнала \overline{CAS} и до его выключения.

Реализованная схема вывода информации дает возможность осуществлять управление состоянием информационного выхода сигналом \overline{WE} при записи (варьируя момент подачи сигнала относительно \overline{CAS}) и сигналом \overline{CAS} при считывании (изменяя его ширину).

Считываемая выходная информация $D0$ имеет ту же полярность, что и входная записываемая информация. Считывание информации происходит без ее разрушения.

Так как микросхема КР565РУ6 не имеет выходных буферных регистров для фиксации выходной информации, то сигналы \overline{RAS} или \overline{CAS} могут служить сигналом «выбор микросхемы». При использовании для этой цели обоих сигналов \overline{RAS} и \overline{CAS} можно реализовать двухкоординатную (x, y) схему выбора микросхемы.

Регенерация информации осуществляется в любом цикле работы ОЗУ путем обращения к каждой из 128 строк не реже чем через каждые 2 мс. Наиболее удобно регенерацию выполнять в специальном режиме управления одним сигналом \overline{RAS} , когда сигнал \overline{CAS} находится в неактивном высоком логическом состоянии (см. рис. 5). При этом потребляемая мощность минимальна.

В источнике напряжения питания должна быть предусмотрена емкостная развязка. После подачи напряжения питания микросхема КР565РУ6 переходит в нормальный режим функционирования только спустя 2 мс или несколько (8...16) рабочих циклов.

При выходе из строя источника питания, когда напряжение выходит за допустимые пределы, дополнительной мерой против катастрофических отказов микросхем может служить принудительный перевод сигналов \overline{RAS} и \overline{CAS} в неактивное логическое состояние.

Минимальная наработка при соблюдении условий эксплуатации ($T = -10...+70^\circ\text{C}$, относительная влажность воздуха 98% при $T = 35^\circ\text{C}$) составляет не менее 15 000 ч, срок сохраняемости не менее 15 лет.

При значениях напряжения питания $U_{CC\min} = 0\text{ В}$ и $U_{CC\max} = 6\text{ В}$ и напряжения на выводах микросхемы ($A0...A6$, \overline{RAS} , \overline{CAS} , \overline{WE} , $D1$) $U_{\min} = -1\text{ В}$, $U_{\max} = +6,5\text{ В}$ (относительно U_{GND}) обеспечивается только сохранность схемы, но не ее функционирование.

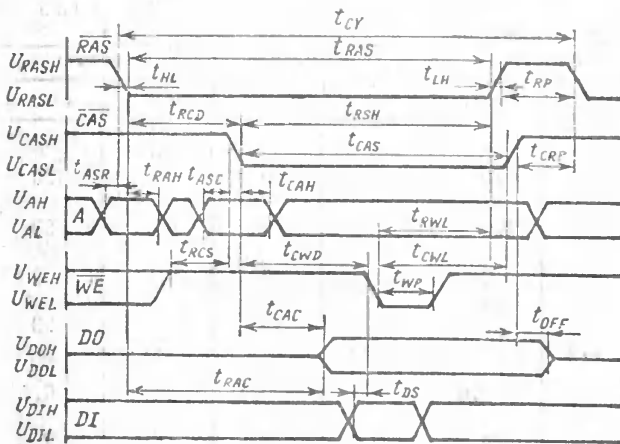


Рис. 6. Режим считывание-модификация-запись

По многочисленным просьбам читателей мы продолжаем публиковать материалы выступлений за «круглым столом» журнала, который состоялся 4 октября 1985 г. в Большом зале Центрального лектория Всесоюзного общества «Знание» (Большая аудитория Политехнического музея).

Фрагменты выступлений А. Г. Кушниренко «Обучение программированию в вузе», Н. А. Садовской «Обучение программированию в школе», В. Ю. Романова «Персональная ЭВМ «Ириша» для кабинетов информатики и вычислительной техники» были опубликованы в 3-м номере журнала за 1985 г.

Сегодня мы предлагаем вниманию читателей стенограмму доклада А. П. Ершова.

В следующем номере будут опубликованы материалы доклада Г. Р. Громова «Автоформализация профессиональных знаний».

КАК УЧИТЬ ПРОГРАММИРОВАНИЮ

А. П. Ершов

Позвольте приветствовать участников семинара и высказаться по вопросу о том, как учить программированию.

Проблема программирования является сейчас узким местом на пути к всенародному применению микроэлектронной вычислительной техники. Если в области аппаратных средств развитие в целом идет в неослабевающем темпе от победы к победе, то в программировании каждый шаг дается с трудом, а успехи не так впечатляющи.

Есть, конечно, объективные причины. Одна из них — внутренняя трудность программирования как профессиональной человеческой деятельности. Помнится, как более 25 лет назад мы в Вычислительном центре Академии наук осваивали первую советскую серийную ЭВМ «Стрела» и одновременно писали для нее транслятор.

Уже тогда наш завлаб, один из ветеранов советского программирования, Владимир Михайлович Курочкин заметил, что сконструировать хороший транслятор — труднее, чем сделать ЭВМ. А ведь с тех пор сложность и многообразие программных средств возросли многократно.

Дело, однако, не только в трудно-

сти. Есть немало профессий, требующих необычайной концентрации человеческих способностей. Космонавт — одновременно и самый ходячий, и самый яркий пример. Особенности проблемы программирования — в ее массовости. На протяжении одного поколения численность программистов возрастет на три порядка — от тысяч до миллионов.

Можно с удовлетворением отметить, что эта проблема в целом уже стала частью общественного сознания. Курс программирования читается в большей части вузов. На два или три года все факультеты повышения квалификации физико-математического профиля перекроены на двухмесячные курсы компьютерной грамотности, через которые с отрывом от работы должны пройти практически все вузовские преподаватели. Владение основами программирования и применения ЭВМ уже является минимальным квалификационным требованием ВАК СССР. Наконец, наиболее масштабное событие этого года — введение курса основ информатики и вычислительной техники во всех школах СССР. Об этом мы еще поговорим по-подробнее, но пока хотел бы сразу отметить беспрецедентный ха-

рактер этого события: наука, к существованию которой каких-нибудь 50 лет назад не было даже и предположений, становится элементом общечеловеческой культуры, воплощенной во всеобщем среднем образовании.

И все же вопросов больше, чем решений. Эти вопросы — разного уровня, но равной значимости и масштабности — ставятся, возникают, заостряются, выдвигаются. Некоторые из них были видны еще на заре программирования, другие — с неожиданностью проявляют себя в самое последнее время.

Я перечислю некоторые из них. Программирование — это фундаментальное знание или сумма трудовых навыков? Чего больше в программировании — абстрактно-математического мышления или инженерно-технической деятельности? Еще один, поистине грандиозный вопрос: нужно ли создавать среднеобразовательную профессию программиста? Очень сложный прогнозный вопрос: какой объем и какая способность к программированию должны быть возложены на так называемого оконечного пользователя? Важный методологический вопрос: в чем разница между системным аналитиком,

системным программистом и прикладным программистом?

Конечно, в широком плане эти вопросы типичны для становящейся науки и профессии. И каждая наука, и каждая профессия в ходе своего становления рано или поздно находила ответы на подобного рода вопросы, формируя — да позволю себе употребить новомодное слово — устойчивую парадигму данной науки и связанной с ней деятельности.

Особенностью программирования является уже отмеченная массовость всех связанных с ней проблем и дел.

Именно массовый характер программирования не позволяет уйти от осознанных ответов на эти вопросы путем замыкания программистов на их, так сказать, внутрисекторные проблемы, разрешаемые на основе «самороста» и накопления личного опыта. А это значит, что мировоззрение программистов, осознание ими своей роли в общественном разделении труда, основы знаний и трудовых навыков, формула непрерывного развития — все это должно быть заложено в этапы общего, ориентирующего и специального образования.

А это значит, что все перечисленные и многие другие вопросы фокусируются на один вопрос: как учить программированию?

Обсуждению этого вопроса посвящен наш семинар. Очень надеюсь, что обсуждение будет активным, горячим и многогранным. Затраченные выступления охватывают разные его стороны. Выступление Анатолия Георгиевича Кушниренко будет посвящено вузовскому курсу программирования. Выступление Григория Рафаиловича Громова связано с проблемой профессионального саморазвития специалиста, работающего с ЭВМ. Я же в своем выступлении затрону вопросы программирования и информатики в школе.

Для того чтобы сделать разговор более содержательным, я позволю себе коротко остановиться на формуле компьютерной грамотности, положенной в основу нынешнего школьного курса. Эта формула впервые прозвучала год с небольшим назад на августовском (1984 г.) совещании руководителей советского народного образования, посвященном школьной реформе.

Прошу прощения за некоторую громоздкость, вызванную желанием сказать все в одном абзаце.

Владение алгоритмической нотацией в объеме, достаточном для выражения вычислительных планов и программ для скалярных, векторных, структурных и текстовых величин, содержащих циклы, ветвления и процедуры. Понимание связи алгоритмической и общематематической нотаций. Умение составить программу решения задачи на материале

общеобразовательных дисциплин и предметов трудового обучения.

Знакомство с принципами устройства ЭВМ, владение начальными навыками обращения со школьной ЭВМ, умение применить школьную ЭВМ к решению практических задач по материалам общеобразовательных дисциплин и предметов трудового обучения. Представление о возрастающей роли ЭВМ в жизни общества, знание конкретных примеров применения ЭВМ.

Совсем недавно, в майском выпуске американского журнала «Сообщения Ассоциации по вычислительной технике» (Communications of the ACM) за прошлый год был опубликован учебный план полугодового курса основ программирования, предназначенный для всех старшеклассников и не опирающийся на какое бы то ни было предыдущее специальное знание. Поскольку образование в США полностью децентрализовано, эта программа не имеет формально нормативной силы, но тем не менее представляет интерес как авторитетное и взвешенное суждение, без сомнения, отражающее реалии американской школы.

Вот как выглядит перечень программистских «умений» для этого курса.

По окончании курса учащийся должен уметь:

- начать и завершить работу на ЭВМ;
- ввести и отредактировать программу;
- определять характеристики хорошо составленной программы;
- обнаруживать некоторые типичные ошибки программирования и составить правильную программу, содержащую:
 - ввод данных с интерактивного устройства ввода;
 - воспроизведение результатов (числовых, текстовых или графических) на интерактивном устройстве ввода;
 - присваивания простым переменным;
 - циклическую структуру, управляемую логическим условием;
 - вложенные конструкции ветвления;
 - использование цикла по параметру для нахождения накапливаемого результата.

Мне было очень интересно обнаружить практическую равнообъемность столь удаленных друг от друга учебных курсов.

Тем не менее, даже если признать либо нашу, либо заокеанскую, либо обе формулы компьютерной грамотности за бесспорные, проработка вопроса «как учить программированию?» проявляет немало до сих пор неутихающих противоречий.

Как многие из вас, возможно, знают, предварительная апробация ру-

копсии нынешнего учебника «Основы информатики и вычислительной техники» (часть I, для 9-го класса) — я буду называть его «Информатика-IX» — вызвала горячую дискуссию с диапазоном мнений от принципиального одобрения до категорического отрицания — при этом мнений, исходивших от весьма авторитетных специалистов.

В этой дискуссии были, конечно, и привходящие моменты, но если от них отвлечься и подать проблему более контрастно и объективно, то альтернативы звучали именно так, как первый из названных мною вопросов: программирование — это фундаментальное знание или сумма трудовых навыков?

Конечно, любой здравомыслящий «человек с улицы» заметит не без иронии, что спор искусственный, конечно, программирование — это и то, и это, и давайте-ка, вместо того чтобы сшибаться в теоретических дискуссиях, строить побыстрее курс, в котором теория и практика будут гармонично объединены.

Дело, однако, в том, что достичь этой гармонии можно очень нескоро, так как она полностью осуществляется лишь в сложившейся инфраструктуре школьной компьютеризации, которая должна включать и кабинеты информатики, и общедоступные школьные компьютеры, и новое поколение учителей с вузовской подготовкой по современной информатике. Пока что нам приходится действовать в становящейся и нестабильной среде, когда от идеала до реальности — дистанция огромного размера.

Есть к тому же за этим спором и одно очень глубокое объективное противоречие, которое состоит в том, что само программирование не едино. Оно имеет по крайней мере два лица — и здесь я вынужден напомнить свою же собственную работу о двух обликах программирования, опубликованную в шестом номере журнала «Кибернетика» в 1982 году.

Программирование имеет два облика, две цели; о них можно сказать так: программирование для себя (программист-хозяин) и программирование для заказчика (программист-слуга). Каждый, кто хоть сколько-нибудь повзрел в программистских коллективах, хорошо сознает как разницу этих двух программирований, так и их постоянную и каждодневную зацепленность — зацепленность и случайную, по недоразумению, и фундаментальную, глубоко диалектическую.

Нет времени вдаваться в интереснейший анализ этих двух обликов программирования, скажу только, что та или иная трактовка этих двух обликов буквально пронизывает все аспекты обучения программированию.

Приведу пример.

Недавно я был свидетелем урока информатики в девятом классе, где проходили тему «Заголовок алгоритма при его записи на алгоритмическом языке». Учитель умело пытался пробудить у учеников собственное суждение о степени полезности этого элемента оформления записи алгоритма. Интересно заметить, что наиболее явными оппонентами оказались ребята с положительным опытом программирования: «Если я все это и так знаю по необходимости, чтобы правильно запрограммировать, то зачем мне вся эта писанина?» Упоение своей способностью быть с машиной на ты, «и так», без избыточных формальностей мешало им воспринять необходимость оформления заголовка алгоритма как выражения исходного знания, предшествующего программированию. Конечно, потом они соглашались или даже сами подыскивали аргументы, но уже **другой** клеточкой своего интеллекта, не зависящей от программистской практики, а скорее из общелогических соображений.

И тут я хочу еще раз сослаться на очень интересное суждение, пришедшее к нам из компьютерной Мексики: тот же журнал «Сообщения Ассоциации по вычислительной технике» за тот же 1985 год, только выпуск не майский, а июльский.

Вот что пишет профессор Стин, президент Математической ассоциации Америки:

«Компьютерная грамотность — это одно из наиболее суетных выражений нашего времени. Мы слышим его повсюду — от школьных советов до высоких федеральных комиссий. Однако все в большей степени употребление этого выражения является привычным пустословием, нежели продуктом сколько-нибудь глубокого размышления.

Книги и курсы, претендующие на постижение компьютерной грамотности, являются бледной тенью заслуженных гуманитарных и естественных наук, стиль которых они пытаются имитировать. Вы не найдете там фундаментальных принципов или вечных истин, достойных Шекспира, Ньютона, Фолкнера или Дарвина. Вместо этого они забыты эфемерными деталями, чей срок жизни вряд ли превысит время окончания школы учеником.

Суть в том, что компьютерная грамотность преподается скорее как обучение вождению автомобиля, нежели как научение математическому исчислению».

Дальше можно не продолжать в высшей степени интересную заметку профессора Стина, потому что во многих наших дискуссиях о том, как учить программированию, метафора автовожждения была излюбленным аргументом сторонников практического подхода к информатике, при

котором теоретически мотивированный подход к курсу уравнивался со стилем школы «сухого плавания».

Статья профессора Стина не случайна. Она является озобоженной реакцией думающего человека на параферналию коммерчески обусловленного заваливания американских школ компьютерами, не сопровождаемого адекватной и перспективной методической гипотезой.

Я хотел бы всячески защитить идею перехода в информатику от математики, через понятие алгоритма, вырастающего из уже освоенной алгебраической символики, через разработку алгоритма, предшествующую выходу на машину.

Хотел бы отметить интересное смещение акцентов в восприятии теоретического материала в «Информатике-IX» (я имею в виду изучение и использование алгоритмического языка). Сначала он виделся многими как вынужденная схема изложения принципов программирования, вынужденная невозможностью сестры за реальный язык программирования, который может прийти только вместе с ЭВМ. Сейчас, при обсуждении «Информатики-X», в которой делается шаг в сторону всякой конкретики, связанной с устройством ЭВМ и использованием реальных языков программирования типа Бейсика и Рапиры, мы сталкиваемся с энергичными рекомендациями усилить «предязыковую», чисто алгоритмическую часть курса, обогатить алгоритмический язык общезыковыми конструкциями и на их уровне расширить опыт алгоритмизации опять-таки до выхода на конкретную машину, на реальный язык программирования.

Совсем недавно я столкнулся еще с одним неожиданным, но приятным подтверждением методологической ориентации на алгоритмический язык как на предмашинный носитель знания об алгоритмах. Я имею в виду только что вышедший учебник Н. П. Трифонова и В. П. Громыко по программированию на ассемблере ЕС ЭВМ, который опирается на алгоритмическую спецификацию задачи на алгоритмическом языке. Он там называется псевдокодом и использует английские ключевые слова, но это не меняет сути дела.

Эта, уже достаточно четко наблюдаемая тенденция, как и многие другие свидетельства, заражают меня еще более глубокой убежденностью в правильности срочного и всеместного введения курса информатики в школу, даже в условиях безмашинного преподавания.

Конечно, впереди спускаемого на воду еще далеко не достроенного корабля школьной информатики идет огромная, мутная волна дискредитации нового и масштабного дела через халтуру, нерасторопность, невежество, реальную неспособность.

Где-то не подобрали учителей, то не разослали вовремя по школам учебники и пособия для учителей, где-то учили на курсах чему угодно, только не предстоящему предмету.

Очень хотелось бы, чтобы эта волна неприятия и тихого саботажа не загрозила главной мысли: учебный курс информатики в школе — это единственное средство сдвинуть всю проблему с места. При любом другом подходе она неизбежно будет застревать на всевозможных трудностях, не выходя за пределы привилегированных школ, богатых шефов, факультативов, олимпиад и других форм весьма полезной, но все же внешкольной работы.

В то же время есть много свидетельств активного, творческого и продуктивного отношения к делу:

Издательство «Просвещение» сумело за два месяца напечатать 3,5 млн. экз. учебников и вовремя передать их в торговую сеть.

Преподаватели МГУ за две недели сделали великолепный Е-практикум и пропустили через него всех преподавателей, приехавших на курсы в университет.

Специалисты из Латвийского университета за три недели написали и издали в высшей степени удачное методическое пособие, которое было напечатано сериалом в «Учительской газете».

Специалисты из Куйбышева, да и многих других мест, разрабатывают методику «культурного» расписывания алгоритмов с алгоритмического языка на язык программируемого микрокалькулятора.

Библиотечка «Кванта» издала массовым тиражом книжку покойного Геннадия Анатольевича Звенигородского «Первые уроки программирования».

В Ленинграде все без исключения школы прикрепили к организациям, имеющим ЭВМ, и создали городской центр школьной информатики.

Школьники и студенты Новосибирска успешно переносят систему «Школьница» на МЭПовские компьютеры.

Кроме этих, видимых и ярких событий, ведется десятками тысяч учителей трудная, но без ошибок и сбоев, но заинтересованная и честная работа по приобщению учащихся к основным понятиям информатики и программирования.

Дело каждого из нас — достроить этот корабль, самим не отстать от него, сохранить ему плавучесть и маневренность и вывезти на нем наших детей и внуков в XXI век, полностью подготовленных к грядущим поистине эпохальным переменам, которые несет и которых требует научно-техническое обновление наших производительных сил и формируемых ими производственных отношений.

Статья поступила 2 октября 1985 г.

УДК 34:002.6

Карась И. З. Вопросы правового обеспечения информатики. — Микропроцессорные средства и системы, 1986, № 1, с. 3.

Рассмотрены проблемы, связанные с правовым регулированием общественных отношений в сфере информатики. Выявлены противоречия, связанные с выделением процессов производства машинной информации в самостоятельный вид деятельности. Дана оценка влияния вычислительных систем на социальные процессы. Сформулированы предложения по некоторым аспектам правового регулирования общественных отношений в области производства и обмена машинной информацией, а также правоотношений в социальной сфере.

УДК 681.3.06

Кобылинский А. В., Липовецкий Г. П. Однокристалльные микроЭВМ серии K1816. — Микропроцессорные средства и системы, 1986, № 1, с. 10.

Представлены технические характеристики однокристалльных 8-разрядных микроЭВМ серии K1816 — KM1816VE48, KM1816VE39, KM1816VE49. Описана структурная организация, назначение выводов, режимы работы. Приведены временные диаграммы, система команд, основные электрические параметры.

УДК 621.3.049.77:681.3.06

Бобков В. А., Бурмистров Ю. Н., Иванов В. И., Кособрюхов В. А., Уткин Ю. В., Чернуха Б. Н. Однокристалльные 4-разрядные микроЭВМ серии KR1820. — Микропроцессорные средства и системы, 1986, № 1, с. 19.

Рассмотрены основные характеристики двух однокристалльных микроЭВМ серии KR1820. Указано назначение выводов, приведены структурные схемы, система команд, описаны режимы синхронизации, обработки прерываний, структура адресации. Даны примеры применения.

УДК 681.325.5

Кобылинский А. В., Москалевский А. И., Темченко В. А. Однокристалльный высокопроизводительный 16-разрядный микропроцессор KM1810VM86. — Микропроцессорные средства и системы, 1986, № 1, с. 28.

Приведены основные характеристики однокристалльного 16-разрядного микропроцессора KM1810VM86 микропроцессорного комплекта БИС серии K1810, предназначенного для построения широкого спектра средств вычислительной техники — от высокопроизводительных одноплатных контроллеров, персональных компьютеров, до иерархических мультипроцессорных систем. Кратко рассмотрены структурная организация, режимы функционирования, возможности системы прерывания и аппаратно-программных средств организации мультипроцессорных систем, особенности системы команд и режимов адресации.

УДК 681.322.1—181.5

Велихов Е. П., Пермянцев И. Г., Рахимов А. Т., Рой Н. Н., Скурихин А. В., Шербаков О. А. Персональный компьютер в системе автоматизации физического эксперимента. — Микропроцессорные средства и системы, 1986, № 1, с. 34.

Представлена система управления физическим экспериментом на основе персональной ЭВМ ПК8001. Рассмотрены состав и основные характеристики ПЭВМ ПК8001.

UDC 34:002.6

Karas I. Z. Juridical Problems of Informatics. — Microprocessor Devices and Systems, 1986, N 1, p. 3.

The author discusses the juridical problems of social relations in the area of informatics. The contradictions connected with human activity in the data processing industry are discussed. The proposals are formulated on some aspects of legal control of social relations in the field of information production.

UDC 681.3.06

Kobylynsky A. V., Lipovetsky G. P. Monocrystal Microcomputers of the Type K1816. — Microprocessor Devices and Systems, 1986, N 1, p. 10.

The paper presents technical features of monocrystal 8-bit microcomputers of the K1816 series: KM1816VE48, KM1816VE39, KM1816VE49. The structure, pin layout and instruction sets are discussed. Timing diagrams and the main electrical parameters are presented.

UDC 621.3.049.77:681.3.06

Bobkov V. A., Burmistrov Ju. N., Ivanov V. S., Kosobryuhov V. A., Utkin Ju. V., Chernuha B. N. KR1820 Monocrystal 4-bit Microcomputers. — Microprocessor Devices and Systems, 1986, N 1, p. 19.

The paper describes the main features of two monocrystal microcomputers belonging to KR1820 IC set. Their structure, instruction set, modes of operation and addressing are presented. Examples of application are given in the paper.

UDC 681.325.5

Kobylynsky A. V., Moskalevsky A. I., Temchenko V. A. High-Efficient Monocrystal 16-Bit Microprocessor KM1810VM86. — Microprocessor Devices and Systems, 1986, N 1, p. 28.

The paper describes the basic features of monocrystal 16-bit microprocessor K1810 VM 86 intended for wide application in single-board controllers, personal computers and multi-processing systems. The structure, functioning, instruction set and other features are described.

UDC 681.322.1—181.5

Velikehov E. P., Permiantsev I. G., Rahimov A. T., Roj N. N., Skurihin A. V., Tcherbakov O. A. Personal Computer in Automation of Physical Experiments. — Microprocessor Devices and Systems, 1986, N 1, p. 34.

The paper describes the system for controlling experiments in physics which is based on PK 8001 personal computer. The structure and main features of this computer are discussed.

УДК 681.3.02

Алексенко А. Г., Гапоненко А. В., Иванов А. Д., Курилов И. Д. — Разработка и отладка микропрограммного обеспечения цифровых систем на основе секционированных микропроцессоров. — Микропроцессорные средства и системы, 1986, № 1, с. 37.

Описывается универсальный настраиваемый микроассемблер, используемый для написания микропрограмм для секционированных БИС и любых форматов микрокоманд, а система логической отладки, позволяющая по заданной принципиальной схеме автоматически получить для отладки микропрограмм эмулятор технических средств микропроцессорной системы. Микроассемблер и система отладки реализованы на мини-ЭВМ SM-4.

УДК 519.68.681.3.06

Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации. — Микропроцессорные средства и системы, 1986, № 1, с. 44.

В статье на основе аналогии «Человек в Мире объектов» и в терминах современных компьютерных игр излагается суть объектно-ориентированного взаимодействия человека с ЭВМ.

УДК 621.317:681.3.62

Мирский Г. Я. Встроенная микропроцессорная система — основа современного средства измерения. — Микропроцессорные средства и системы, 1986, № 1, с. 49.

Рассматриваются принципы построения, структурные схемы и работа программно-управляемых цифровых измерительных приборов. Раскрываются преимущества и новые возможности средств измерения, содержащих микропроцессоры. Освещаются особенности приборов, рассчитанных на применение в измерительной системе, организуемой с помощью приборного интерфейса.

УДК 681.326—181.4

А. Ю. Сасов. Микротомография и цифровая обработка изображений на микроЭВМ «Искра 226». — Микропроцессорные средства и системы, 1986, № 1, с. 55.

Рассмотрены вопросы использования микроЭВМ «Искра 226» с растровым электронным микроскопом и другими системами получения изображений. Описан комплект программ для улучшения, реставрации, анализа и классификации изображений, а также систем трехмерной реконструкции внутренней структуры микрообъектов без разрушения — «микротомографов» — на базе «Искры 226».

УДК 681.322.1

Романов В. Ю., Барышников В. Н., Воронов М. А., Паначев Ф. И. Графические возможности персональной ЭВМ «Ириша». — Микропроцессорные средства и системы, 1986, № 1, с. 61.

Дается краткая характеристика различных моделей ПЭВМ «Ириша». Приводится состав и программное обеспечение каждой модели. Подробно описывается один из основных модулей ПЭВМ — телевизионный адаптер, обеспечивающий работу с черно-белым и цветным мониторами. Рабочее поле монитора имеет размер 320×200 точек в режиме цветной графики и 640×200 точек — в режиме монохромной графики высокого разрешения.

UDC 681.3.02

Alekseenko A. G., Gaponenko A. V., Ivanov A. D., Kurilov I. D. — Microprogram Development and Debugging for Sliced Microprocessor Systems. — Microprocessor Devices and Systems, 1986, N 1, p. 37.

The paper describes universal adaptable microassembler intended for microprogram development of sliced LSI's. It allows different formats of microinstructions. Debugging system is described which provides generation of emulator for hardware components. The system is implemented on SM-4 minicomputer.

UDC 519.68.681.3.06

Lebedev G. V. Direct Editing Principle and Interactive Program Development. — Microprocessor Devices and Systems, 1986, N 1, p. 44.

The paper discusses the method of object-oriented interaction with a computer based on "A Person in the World of Objects" metaphora which is typical for computer games.

UDC 621.317:681.3.62

Mirsky R. Ja. Embedded Microprocessor System — the Basis of Advanced Measuring Equipment. — Microprocessor Devices and Systems, 1986, N 1, p. 49.

The author describes the principles, structural diagrams and functioning of the program-controlled numerical measuring instruments. The advantages of measuring devices with microprocessors are discussed. The paper describes the peculiarities of the measuring instruments utilizing the standard interface.

UDC 681.326—181.4

Sasov A. Ju. Microtomography and Image Processing on Microcomputer "ISKRA 226". — Microprocessor Devices and Systems, 1986, N 1, p. 55.

The paper discusses the problems of using "ISKRA 226" microcomputer with bit-mapped electronic microscope and other systems generating digitized images. The program package is discussed which provides improving, restoration, analysis and classification of images. The system of 3D reconstruction of microobjects internal structure ("microtomography") is described which is running on "ISKRA 226".

UDC 681.322.1

Romanov V. Ju., Baryshnikov V. N., Voronov M. A., Panachev F. I. Graphics on "Irisha" Personal Computer. — Microprocessor Devices and Systems, 1986, N 1, p. 61.

The paper describes several modifications of the personal computer "Irisha". TV-adaptor is described providing PC connection to monochrome and colour TV-sets. The high-resolution graphic formats support 320×200 pixels in colour and 640×200 pixels in monochrome mode.

Лукьянов Д. А. ПЗУ — универсальный элемент цифровой техники. — Микропроцессорные средства и системы, 1986, № 1, с. 75.

Рассматриваются применения микросхем ПЗУ, содержание которых может быть задано пользователем, в микропроцессорных и радиоэлектронных устройствах. Обсуждена адаптация ассемблера МАКРО-11 ОС РАФОС для автоматической трансляции мнемонического описания карт программирования ПЗУ. Приведены фрагменты структурных и принципиальных схем контроллера шагового двигателя, синтезатора частоты, телевизионного синхрогенератора, быстродействующего умножителя и схемы анализа и коррекции ошибок в ОЗУ.

Lukyanov D. A. ROM — Universal Digital Element. — Microprocessor Devices and Systems, 1986, N 1, p. 75.

The author considers application of ROM integrated circuits in microcomputers and radioelectronic devices. The paper discusses adaptation of microassembler MACRO — 11/OS-RAFOS for translation of ROM mnemonic codes. The structure and diagrams of controllers are given for the step motor frequency synthesizer, TV synchrogenerator, high-speed multiplier and RAM error — correction devices.

Пятиугольник «МП»

Уважаемый товарищ редактор!

В ответ на ваше письмо, в котором вы отказываетесь опубликовать мою статью «Алгоритм и программы встроенного микропроцессора для очистки настольных поверхностей», а по телефону, кроме того, сказали, что «если все публиковать, бумаги не хватит», то сообщаю следующее.

Как раз вчера у нас был один товарищ из смежной организации, довольно авторитетный специалист, и рассказал, что у них разрабатывается новая технология выпуска программной документации, которая может вас заинтересовать.

— Вся беда организаций — разработчиков программного обеспечения, говорит он, — заключается в преступном пренебрежении законами сохранения вещества и охраны природы, небрежном отношении к такому ценному носителю информации, каким является бумага.

К чему это приводит?

Из любой организации выходит гораздо больше исписанной бумаги, чем входит, образующийся дефицит приходится восполнять бумажной промышленностью. В конечном счете истребляется лес и нарушается равновесие окружающей среды.

Мы предлагаем технологию безотходного производства программного обеспечения, — говорит он, — основанную на замкнутом цикле. Это будет сверхсовременная фабрика по выпуску программного продукта, и фабриковать его мы будем без расхода бумаги.

Уже на первом этапе внедрения этой технологии предусматривается активное участие в комиссиях по приемке работ представителя Вторсырья. Вся документация рабочих проектов будет направляться в переработку немедленно, минуя заказчика, фонды алгоритмов и программ и предприятия Министерства связи. Это сразу даст колоссальный экономический эффект и поможет психо-

логической перестройке персонала предприятия, необходимой для второго этапа внедрения.

На втором этапе, чтобы уравнивать количества входящей и исходящей из нашей фабрики использованной бумаги, предлагается сделать следующий шаг: не выводить программную документацию на печатающие устройства, а все машинописные работы ограничить изготовлением сводок о выполнении основных показателей.

— Что же тогда у вас будут принимать?

— То, что и сейчас — отчет о выполнении плана выдачи машинного времени.

— Но какой же смысл такой работы?

— А какой смысл вашей?

Остается моральная сторона труда, фонды поощрения — принципиально нового здесь ничего нет: ведь программирование — не единственное занятие, где процесс важнее, чем результат. Предвидим затруднения с заказчиками: им бывает трудно доказать, что документация на неработающую программу — это чистый отход: привыкли сдавать нашу продукцию в макулатуру на абонементы, культурный уровень, видите ли, повышают. Ну, ничего, раз наши программы не используют — через арбитраж всю документацию вытребуем обратно, никуда не денутся, закон на нашей стороне! Необходимо, правда, проработать еще некоторые юридические аспекты. Только подумайте, какая польза для народного хозяйства: не говорю о бумаге, а электроэнергия, а красящая лента, а если другие организации перейдут на этот режим... Да им поэтому просто памятник поставят и напишут: сотрудникам такого-то НИИ за эффективный развал никому не нужной работы и сохранение окружающей среды. Думать на-

до, а не валить все на аэрокосмический мониторинг!

А социологический аспект — ведь определенная группа сотрудников сможет после этого спокойно проводить часть рабочего времени дома — снижается расход электроэнергии на освещение рабочих мест, нагрузка на транспорт в часы пик, нервы друг другу будут меньше трепать, а в конечном счете высвободается дополнительное время на воспитание детей, ведение хозяйства и сопутствующие процедуры.

— Интересная мысль, никогда бы не додумался. А как отнесется общественность?

— Положительно. Вот, взгляните, что пишет наша многотиражка: «Индустрия очковтирательства» — нет, это не про нас, это про соседний отдел, а, вот: «Безотходная на марше», «...дальнейшее развитие идей директора международного экспериментального ВЦ (МЭВЦ) — это про меня, значит, ну, и так далее.

— Подождите, вы меня совсем запутали, а вдруг вы делаете нужную людям программу? Ведь это иногда, как утверждают, бывает. Как ей пользоваться без документации?

— Не беспокойтесь, — это почти исключено! Во всяком случае, на моей практике такого еще не бывало. И потом существует научный закон: из всех затрат на разработки только пять процентов дают эффект, но этот эффект с лихвой окупает остальные девяносто пять. Так не разумней ли только на эти пять процентов и выпускать документацию...

Я вас хочу еще раз спросить, товарищ редактор, если действительно удастся, как он утверждает, внедрить безотходную технологию производства программной документации, сможете вы тогда напечатать в счет сэкономленной бумаги и мою статью, или нет?

С нетерпением
Л. Попель



МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКТ ДЛЯ МАКЕТИРОВАНИЯ ПРИ ОБУЧЕНИИ И РАЗРАБОТКЕ



Обобщение опыта обучения и использования микропроцессорной техники, а также анализ недостатков известных микропроцессорных учебных устройств дали возможность сформулировать основные требования к учебному лабораторному оборудованию для изучения микропроцессорной техники:

а) возможность изучения аппаратной части параллельно с программной, иными словами доступность аппаратной части микропроцессорных систем для анализа и модификации;

б) развиваемость микропроцессорной системы. Это требование связано с тем, что выпускаемые промышленностью микропроцессорные наборы и однокристалльные микроЭВМ непрерывно пополняются все новыми компонентами, которые должны быть как можно скорее использованы в соответствующем лабораторном оборудовании;

в) микропроцессорное учебное оборудование должно обязательно включать в себя средства, обеспечивающие возможность работы с современными кросс- и резидентными системами автоматизации программирования.

Для учебного оборудования, удовлетворяющего перечисленным требованиям, напрашивается альтернативное техническое решение между так называемым стендовым подходом (обучаемый полностью лишен возможности участвовать в создании микропроцессорной системы) и классическим макетированием (эффективность обучения снижается непроизводительными для обучения действиями по монтажу и отладке устройств). Активное участие обучаемого в построении микропроцессорной системы особенно важно при изучении модульных микропроцессорных наборов с наращаемой разрядностью, например, K589,

K1802, K1804, так как в этом случае изменение структуры микроконтроллера влияет на формат микрокоманды. Модификация аппаратной части позволяет применить этот подход также при макетировании вновь разрабатываемых устройств.

На кафедре вычислительной техники Винницкого политехнического института совместно с Центральным конструкторским бюро информационной техники разработан микромакетный микропроцессорный набор для обучения и разработки.

Микромакет — основной элемент предлагаемого микропроцессорного комплекта обучения — представляет собой интегральную микросхему (БИС KP580, K589, расширенный комплект K155 с дополнительными корпусами памяти и интерфейсов) либо аппаратно неизменяемый узел, конструктивно размещенный в отдельном корпусе (см. фото). Выводы микросхемы распаяны на входной и выходной линейный разъемы, расположенные на передней панели микромакета. Здесь же проставлено условное функциональное обозначение микросхемы, определяющее назначение контактов разъема. Выходы микросхем снабжены двоичной светодиодной индикацией, причем эти индикаторы совмещены с графическим обозначением выводов на передней панели.

Каждый микромакет имеет также гнезда для подключения соответствующего напряжения питания. От выхода из строя при неправильном включении питания схемы предохраняются элементами защиты. Устройства собираются из отдельных микромакетов с помощью проводников и многожильных шин. Вместе с источником питания и портативным осциллографом микромакеты размещены в портфеле типа дипломат, открытая верхняя крышка которого служит плоскостью для макетирова-

ния. Отдельные микромакеты крепятся на этой плоскости с помощью магнитных креплений.

Таким образом, пользователь в процессе обучения собирает на вертикальной плоскости структуру микроконтроллера, программирует его и контролирует выполнение программы при помощи индикации, дающей информацию о логическом состоянии всех сигналов и шин. Дополнительно в комплекте есть микромакет, обеспечивающий выход микроконтроллера на стандартный интерфейс (ИРПР, ИРПС) для подключения различных периферийных устройств либо внешней ЭВМ с целью автоматизации проектирования.

Перечисленные выше свойства предопределяют высокую эффективность использования микромакетных комплектов. Наибольший эффект будет достигнут при крупносерийном изготовлении микромакетных средств с использованием магнитной резины для крепления микромакетов на рабочей поверхности произвольной конфигурации; экономичных средств индикации состояния аналоговых и цифровых цепей; эффективных и высоконадежных разъемных соединений с учетом особенностей шинной организации структур. Предстоит совершенствовать состав и характеристики сервисных микромакетных средств с возможной разработкой специализированных микросхем.

Микромакетный микропроцессорный комплект для обучения и разработки демонстрировался на ВДНХ СССР, а также на международной выставке «Народное образование» в г. Загреб (СФРЮ) в 1984 г. В Центральном конструкторском бюро информационной техники начато серийное производство микромакетного комплекта.

А. М. Петух, А. В. Силагин,
А. П. Стахов, Ю. П. Бурштейн

ВНИМАНИЕ

Если Вы не успели подписаться на наш журнал, НАПОМИНАЕМ, что в 1986 году подписка принимается ПОКВАРТАЛЬНО. За месяц до начала любого квартала прекращается подписка на ту часть годового комплекта журналов, которая издается в следующем трехмесячном периоде.

ВНИМАНИЕ

С 1986 ГОДА ЖУРНАЛ ВЫХОДИТ 6 (ШЕСТЬ!) РАЗ В ГОД!

Если Вы подписались только на 4 номера, то имеется возможность оформить дополнительную подписку на 5 и 6 номера годового комплекта. Сделать это Вы можете в течение первого полугодия 1986 года в любом отделении «Союзпечать».

ВНИМАНИЕ

С 1986 года журнал издается 6 (шесть!) раз в год!

Подписка на журнал ежеквартальная и принимается всеми отделениями Союзпечати в течение всего года!

Цена одного номера — 1 руб. 10 коп. Индекс журнала — 70588.